Computational Semantics of Cartesian Cubical Type Theory

Carlo Angiuli

CMU-CS-19-127

September 2019

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Robert Harper, Chair
Jeremy Avigad
Karl Crary
Kuen-Bang Hou (Favonia)
Daniel R. Licata
Todd Wilson

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the United States Air Force Office of Scientific Research under grant number FA9550-15-1-0053. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government, or any other entity.

Keywords: dependent type theory, homotopy type theory, cubical type theory, two-level type theory, computational type theory, computational semantics

Abstract

Dependent type theories are a family of logical systems that serve as expressive functional programming languages and as the basis of many proof assistants. In the past decade, type theories have also attracted the attention of mathematicians due to surprising connections with homotopy theory; the study of these connections, known as *homotopy type theory*, has in turn suggested novel extensions to type theory, including higher inductive types and Voevodsky's univalence axiom. However, in their original axiomatic presentation, these extensions lack computational content, making them unusable as programming constructs and unergonomic in proof assistants.

In this dissertation, we present *Cartesian cubical type theory*, a univalent type theory that extends ordinary type theory with interval variables representing abstract hypercubes. We justify Cartesian cubical type theory by means of a *computational semantics* that generalizes Allen's semantics of Nuprl [All87] to Cartesian cubical sets. Proofs in our type theory have computational content, as evidenced by the canonicity property that all closed terms of Boolean type evaluate to true or false. It is the second univalent type theory with canonicity, after the De Morgan cubical type theory of Cohen et al. [CCHM18], and affirmatively resolves an open question of whether Cartesian interval structure constructively models univalent universes.

Acknowledgements

First and foremost, I thank Bob Harper for many years of advice, guidance, and support throughout our collaboration. One suggestion that really stuck with me was to *develop a point of view*; I hope that readers of this dissertation find mine instructive, even if they ultimately disagree. Throughout my Ph.D., my views have been informed by discussions with the Principles of Programming group in Computer Science and the Homotopy Type Theory group in Philosophy. I thank Steve Awodey for years of organizing the latter group's seminars, to which he has invited many of the best researchers in our field.

For many thoughtful conversations about mathematics, programming languages, and type theory, I particularly thank Mathieu Anel, Evan Cavallo, Favonia, Nicolas Feltman, Daniel Gratzer, Adrien Guatto, Simon Huber, Chris Kapulkin, Dan Licata, Ed Morehouse, Anders Mörtberg, Jon Sterling, Joe Tassarotti, and Todd Wilson. I continue to gain a deeper appreciation of the brilliance and vision of the late Vladimir Voevodsky, to whom many of us owe our current research directions. For discussions and feedback on this dissertation, I thank my committee, Bruno Bentzen, Mark Bickford, Evan Cavallo, Thierry Coquand, Daniel Gratzer, Alex Kavvos, Anders Mörtberg, Andrew Pitts, and Jon Sterling.

I have greatly enjoyed my time in Pittsburgh, the Paris of Appalachia. At Carnegie Mellon, I have been fortunate to take part in the annual SIGBOVIK conference in multiple capacities (`http://www.sigbovik.org`). Away from campus, I cannot thank enough my wife Cassie Orr for her support, or my friends Ben Blum, Larry Blumenthal, Jeremy Ernstoff, Gregory LeDonne, Brendan McShane, Karl Schulz, Rich Shay, and Matt Stanec for countless hours of playing board games and card games.

Contents

1 Introduction

> One might come to believe that any decision to adopt a system of axioms about sets would be *arbitrary* in that no explanation could be given why the particular system adopted had any greater claim to describe what we conceive sets and the membership relation to be like than some other system, perhaps incompatible with the one chosen. One might think that no answer could be given to the question: why adopt *this* particular system rather than that or this other one?
>
> —George Boolos, *The Iterative Conception of Set* [Boo71]

Dependent type theories are a family of logical systems employed by philosophers as foundations of constructive mathematics, and by computer scientists as expressive functional programming languages. In the past decade, type theories have also attracted the attention of mathematicians due to surprising connections with homotopy theory; the study of these connections, known as *homotopy type theory*, has in turn motivated powerful new logical principles in type theory. This dissertation extends traditional computer-scientific techniques in type theory to account for these novel principles.

1.1 The Martin-Löf ethos

To the great consternation of mathematicians, there is no *set of all type theories*. Instead, the study of type theory comprises techniques drawn from computer science, mathematics, and philosophy which, when applicable to an object, confer upon it the status of *type theory*. Many mainstream type theories, however—from the seminal work of Per Martin-Löf [ML75b; ML82; ML84] to Nuprl [Con+85] and the calculus of constructions [CH88]—share various hallmarks which we henceforth describe, with the caveat that these hallmarks are not all universal among type theories.

Unity of collections and propositions Set theory is typically presented as an axiomatic system that defines set membership as a proposition inside an ambient logic. A type theory, in contrast, is a self-contained foundation of mathematics in which types serve both as collections of objects and as logical propositions. Standard type theories comprise two core judgments—typehood (A type) and membership ($M \in A$)[1]—and the auxiliary judgments of

[1] Once standard notation [ML75b; ML82; ML84; Con+85], $M \in A$ is now more often written $M : A$.

type and member equality under hypotheses. Regarding the type A as a collection, $M \in A$ expresses that M is an element of A; regarding A instead as a proposition, $M \in A$ expresses that M is a proof of A.

Given a type A and a family of types $B(a)$ indexed by $a \in A$, the dependent function type $(a{:}A) \to B(a)$ (or $\prod_{a:A} B(a)$) classifies functions sending every $M \in A$ to an element of $B(M)$, and the dependent pair type $(a{:}A) \times B(a)$ type (or $\sum_{a:A} B(a)$) classifies pairs of $M \in A$ and an element of $B(M)$. If we interpret $B(-)$ as a predicate over A, $(a{:}A) \to B(a)$ is the proposition that $B(a)$ holds for all $a \in A$, because its proofs specify a proof of $B(M)$ for every $M \in A$, and $(a{:}A) \times B(a)$ is the proposition that $B(a)$ holds for some $a \in A$, because its proofs specify an $M \in A$ and a proof of $B(M)$.

Now, because types are both collections and propositions, it is possible for a proposition to have many distinct proofs, and for subsequent proofs or constructions to be *proof-relevant*, that is, dependent on a choice of proof. For instance, the type of natural numbers is a proposition with infinitely many distinct proofs, each of which has a distinct successor. Some authors therefore only regard as propositions (or mere propositions [UF13, Definition 3.3.1]) types with at most one element; another possibility is to reject the collection–proposition dichotomy as intrinsically non-type-theoretic.

Readers unfamiliar with standard type formers or the style of mathematics developed inside type theories may wish to consult Nordström, Petersson, and Smith [NPS90] or the *Homotopy Type Theory* book [UF13] as necessary while reading this dissertation.

Constructivity As logics, type theories are typically constructive, or *intuitionistic*, not classical. Intuitionism is a philosophical stance introduced by Brouwer, who held that mathematical objects exist only as constructions in an idealized mind, that propositions are true exactly when mental constructions of their proofs exist, and that the sole purpose of mathematical language is to convey such constructions, which are themselves extralinguistic [TD88, p. 4].

The Brouwer–Heyting–Kolmogorov (BHK) interpretation of the logical connectives therefore maps each connective to a construction—to prove $\exists n{:}\mathsf{nat}.B(n)$ is to construct a natural number n and a proof of $B(n)$; to prove a disjunction $A \vee B$ is to construct either a proof of A or a proof of B [TD88, p. 9]; *et cetera*. It follows that excluded middle ($A \vee \neg A$ for all A) is not intuitionistically valid: to prove $(\exists n{:}\mathsf{nat}.B(n)) \vee \neg(\exists n{:}\mathsf{nat}.B(n))$ for all B, one must either produce a concrete n and a proof of $B(n)$, or a proof that no such n exists.

Constructivity is popularly characterized as the failure of excluded middle, but is more accurately captured by the *existence property*: if $\exists n{:}\mathsf{nat}.B(n)$ is provable, one can construct a concrete numeral $\bar{n}$ for which $B(\bar{n})$ is provable [TD88, p. 139]. From this perspective, it is *fortunate* that most principles of classical mathematics are constructively valid. On the other hand, there are constructive logics with principles *false* in classical mathematics, including Brouwer's intuitionism and the Nuprl type theory, which adopt continuity principles for choice sequences [RB16; TD88, pp. 206–210], and Russian constructive recursive mathematics, which adopts Church's thesis [TD88, pp. 185–195].

Martin-Löf's type theories, as he explains in *Constructive mathematics and computer programs* [ML82], are rooted in a modified form of intuitionism which considers mathematical objects to be constructions not of the mind but of a *programming language*. This conception of intuitionism is incompatible with Brouwer's, as it is essentially linguistic. However, by equating construction and computation, it transforms intuitionism from merely a philosophical stance to a pragmatic one—intuitionistic mathematics is mathematics that computes, a fruitful endeavor regardless of one's beliefs.

The intuitionistic readings of Martin-Löf's type theories are provided in his *meaning explanations*, BHK interpretations which gloss each type former as a behavioral predicate over programs [ML82]. For instance, $M \in (a{:}A)\times B(a)$ when M is a program that computes a pair $\langle M_1, M_2\rangle$ with $M_1 \in A$ and $M_2 \in B(M_1)$ (that is, M_1 computes in accordance with A, and M_2 with $B(M_1)$). Hypotheses simply range over all programs with the specified behavior, so the assertion "assuming $a \in A$, $B(a)$ type" means that for every program $M \in A$, the expression $B(M)$ corresponds to a behavioral predicate. The *canonicity property*—if $M \in \mathsf{bool}$, then M computes to and is equated with a canonical Boolean, either true or false—is a type-theoretic analogue of the existence property, and an immediate consequence of the meaning explanations (because bool corresponds to precisely that predicate).

Meaning explanations are typically presented in an intuitive, premathematical style ambiguous about the programming language and rules of inference purportedly being explained. These choices are deliberate—appeals to intuition ensure the explanations take on a foundational character (that is, without appealing to an ambient logic), and ambiguity over computation and deductive systems ensures the explanations remain open-ended with respect to the choice of programs or predicates. Further discussion of the role of intuitionism in type theories is beyond the scope of this dissertation; we direct the interested reader to the writings of Martin-Löf [ML82; ML13] and Granström [Gra09].

As type theories grow in complexity, it is increasingly difficult to convey their intended meanings through intuition alone, or be confident that such intuitions are sensible. (Such is the case in this dissertation.) One can instead transform the meaning explanations into precise mathematical constructions, which we call *computational semantics*. The variants of computational semantics used in Chapters 2 and 4 of this dissertation serve as constructive proofs of consistency and canonicity, and explicit Curry–Howard or propositions-as-types[2] correspondences [How80] that construct type theories as program logics for functional programming languages.

[2] By *propositions-as-types* we mean that the logical propositions (that is, the *types*) of a type theory can equally well be viewed as the *type system* of a programming language, not merely the observation that the types of a type theory serve also as its logical propositions.

Suitability for computer implementation Of course, the study of type theories intersects computer science not only through computational semantics, but also through the development of proof assistants—software for developing computer-checked mathematical proofs—many of which are based on dependent type theories, including Agda [Agda], Coq [Coq], Lean [Mou+15], and Nuprl [Con+85].

In recent years, computer scientists have begun to coordinate large-scale uses of proof assistants in formally specifying and verifying the behavior of realistic software programs and protocols [App+17]. And while proof assistants remain on the fringe of mathematical practice, expert users have formalized increasingly sophisticated results, culminating famously in the Coq proofs of the four-color [Gon08] and odd-order theorems [Gon+13].

Proof assistants serve to bridge the gap between high-level mathematical arguments and low-level formal proofs composed of primitive inferences, thereby reducing the correctness of the former to the correctness of the latter. In practice, proof assistants require vastly more detail than is customary in paper proofs, but do perform humanly-impossible feats of logical bureaucracy and even automated combinatorial reasoning [Gon+13, p. 173].

The primitive inferences of a proof assistant, being the linchpin of its correctness, are often isolated in a trusted *kernel*; complex features are not trusted, but instead emit proofs checked by the kernel. For type theories, these primitive inferences are rules specifying when a term is a type or element of a type; in contrast to the open-ended character of meaning explanations, the kernel necessarily fixes a collection of rules, and hence, of types and elements.

The hallmarks of type theories, we argue, contribute in part to the success of type-theoretic proof assistants. First, types directly provide many basic mathematical objects (functions, products, natural numbers, *et cetera*) that are derived notions in, say, set theory. Ordinary mathematical proofs should *in theory* reduce to applications of the axioms of set theory; a proof assistant based on set theory must *in fact* perform such a reduction (or else define these objects primitively), significantly widening the gap between informal and formal proof. In contrast, Agda's interface is usable despite being quite close to its underlying type theory—users prove A by providing (most of) an M such that $M \in A$.

Secondly, the constructive nature of type theories enables proof assistants to seamlessly incorporate computation in many beneficial ways. The judgments of type theory are closed under computation, allowing users to simplify proof goals by evaluating them; canonicity ensures that elements of types constructed without hypotheses evaluate to canonical forms. Proof by reflection allows users to implement and even verify decision procedures written in the very programming language underlying the type theory itself [Gon+13]. Program extraction converts proofs into standalone, correct-by-construction functional programs, and has been used to develop a verified C compiler [Ler09]. Implementation of such features is beyond the scope of this dissertation; we discuss their relationship to computational semantics in Section 2.4.

1.2 *The algebraic perspective*

The aforementioned philosophical and computational desiderata constitute one perspective on type theories. Another perspective, particularly common amongst homotopy type theorists, is to regard as type theories *all* collections of rules bearing a superficial resemblance to type theories in the former sense.[3] A type theorist of this kind regards a type theory as an algebraic structure defined by its collection of rules, and asks: what counts as a model (resp., morphism of models) of this type theory? Does this type theory have interesting models? What properties are shared by all of its models?

One purpose of such questions is to establish metatheorems about the rules implemented in a proof assistant. Logical consistency follows from the existence of non-trivial models. Proof assistants often employ only rules whose premises are decidable, a property commonly established by normalization-by-evaluation models [ACD07]. Independence results follow from the existence of models both validating and invalidating certain principles. In a classic paper, Smith [Smi88] proved the independence of Peano's fourth axiom $(\forall n{:}\mathsf{nat}.0 \neq n + 1)$ from a type theory without universe types, by constructing a model in which the natural numbers are a one-element set. (The standard computational semantics validate Peano's axiom.) More recently, Coquand and Mannaa [CM16] proved the independence of Markov's principle from type theory by a forcing argument.

A second purpose is to establish the mathematical significance of theorems proven in a proof assistant. We expect, for example, that the Coq proof of the four-color theorem implies the truth of the four-color theorem as ordinarily construed. Such an implication rests not only on Coq's consistency but also the ability to interpret Coq's logical connectives and rules as classical set-theoretic connectives and tautologies. If classical truth is one's *only* goal, one may even choose to disrupt the constructive nature of a type theory by adding to it classical axioms for real numbers [BLM15].

Conversely, theorems in a proof assistant carry *more* content than their classical counterparts. Standard rules of type theories are valid in all locally Cartesian closed categories [Cur93; See84]—not only in sets—thus licensing type-theoretic language in presheaf categories. Orton and Pitts [OP16] have recently employed this principle to carry out cubical set constructions of Cohen et al. [CCHM18] in Agda extended with an axiomatic interval type, thereby both formalizing and generalizing the original constructions.

Homotopy type theory The *intensional* rules of type theory [ML75b], which form the basis of Agda and Coq, define an identity type $\mathsf{Id}_A(M, N)$ whose elements are proofs that $M, N \in A$ are equal. Standard computational and set-theoretic semantics model identity types as collections with one element when true and zero elements when false. Surprisingly,

[3] The precise class of such theories, which Voevodsky [Voe15] calls "dependent type theory of Martin-Löf 'genus,'" currently lacks a satisfactory definition.

Hofmann and Streicher [HS98] showed that *uniqueness of identity proofs*—the principle that any two proofs of $\mathsf{Id}_A(M, N)$ are equal—is independent of intensional type theory, by constructing a groupoid model in which some identity types have multiple elements. The Hofmann–Streicher groupoid model nevertheless validates uniqueness of identity proofs of identity proofs, that is, any two proofs of $\mathsf{Id}_{\mathsf{Id}_A(M,N)}(P, Q)$ are equal.

The field of homotopy type theory originates[4] with the observations of Awodey and Warren [War08; AW09] and Voevodsky [KL16] that n-fold iterated identity types admit complex structure closely related to that of n-dimensional paths in topological spaces, by constructing models of intensional type theory in, respectively, Quillen model categories and Kan simplicial sets. Voevodsky [Voe10a] further observed that the simplicial model validates a *univalence axiom* stating that identity of types is homotopy equivalent to homotopy equivalence of types (or informally, that isomorphic types can be regarded as equal). Finally, Lumsdaine and Shulman [LS19] and others observed that many familiar topological spaces can be defined axiomatically in type theory as *higher inductive types*.

Voevodsky subsequently championed *Univalent Foundations*—formulated as intensional type theory extended with univalence—as a new foundation for formalized, structuralist mathematics [Voe10b]. Parallel community efforts over 2012–2013 at the Institute for Advanced Study produced the *Homotopy Type Theory* book [UF13], which uses intensional type theory extended with univalence and higher inductive types as an axiom system in which to develop the homotopy theory of spaces. (In a confusing instance of synecdoche, that type theory is often called *homotopy type theory*; we instead call it *Book HoTT*.)

Are Univalent Foundations and Book HoTT—as emulated in Coq [VAG+; Bau+17], Agda [Bru+18], and Lean [DRB17]—the most perspicuous axiomatizations of univalence and higher inductive types? On what basis should we judge type theories *qua* algebraic structures? Or, as Boolos [Boo71] asks, "Why adopt *this* particular system rather than that or this other one?"

Consistency is paramount, and follows from the simplicial set semantics of Voevodsky [KL16] and Lumsdaine and Shulman [LS19].[5] Those semantics moreover establish that theorems of Book HoTT hold also in standard homotopy theory, because the homotopy theory of simplicial sets is famously the same as that of topological spaces.

Importantly, Book HoTT concerns an abstract notion of space, defined without reference to, say, topologies or real numbers; as a result, Book HoTT has already led to novel generalizations of theorems [Ane+17]. Alternatives to Book HoTT, including those proposed in this dissertation, are thus judged in part on the generality of their models.

[4] This is a woefully incomplete summary of early ideas in homotopy type theory; other early contributors to the homotopy-theoretic semantics of type theory include Gambino and Garner [GG08], Lumsdaine [Lum09], van den Berg and Garner [BG11], and Streicher [Str06].

[5] Excepting closure of type universes under parametrized higher inductive types, which was later addressed in cubical set models of Coquand, Huber, and Mörtberg [CHM18] and Cavallo and Harper [CH19a].

Moreover, homotopy type theorists conjecture that type theories with univalence are in a precise sense sound and complete for mathematical domains known as elementary ∞-toposes. However, the community is very far from proving this conjecture—the state of the art includes only dependent pairs and identity types [KS19]—and in fact, lacks consensus on the very *definition* of elementary ∞-topos.

As foundations of mathematics, Univalent Foundations and Book HoTT have several advantages over set theory: they are more amenable to computer formalization, and more elegantly express many ideas from homotopy theory and higher category theory. However, univalence is not as straightforward as the slogan *isomorphic types are equal*: a minor change in its statement renders it inconsistent [UF13, Exercise 4.6(iii)], and its standard model in simplicial sets requires powerful mathematical tools. Nor are these theories obviously constructive in any sense: Univalent Foundations and Book HoTT lack the canonicity and existence properties, and thus even the failure of excluded middle in these theories is subtle [UF13, Corollary 3.2.7]. Such subtleties, in the author's opinion, jeopardize the ontological primacy of these systems.

The failure of canonicity in Book HoTT is caused by a dearth of equations regarding univalence and higher inductive types. For instance, higher inductive types are generated by element and path constructors, and functions out of such types specify where to send those generators; in Book HoTT, a function so defined only *weakly* sends path constructors to their specified images (that is, up to a path), requiring complex bookkeeping in otherwise straightforward proofs [Soj16].

Stranger yet, because Univalent Foundations and Book HoTT co-opt type-theoretic identity to mediate path structure, they lack any internal notion of traditional mathematical equality. That is, one cannot speak of two types being exactly the same, nor define relations finer than homotopy, nor argue that a function can be silently replaced by another whose pointwise behavior is identical, despite nearly all models validating such notions. The inability to impose strict coherence conditions has in fact frustrated the definition of semisimplicial types in these systems, prompting Voevodsky to propose extending Univalent Foundations with strict equality in his Homotopy Type System [Voe13].

1.3 Contributions

This dissertation aims to understand the higher-dimensional concepts of univalence and higher inductive types through computational semantics. We describe *Cartesian cubical type theory* and its computational semantics, which reconstruct these concepts by means of a novel notion of untyped, higher-dimensional computation, supporting the thesis that *higher-dimensional types classify higher-dimensional programs extensionally according to their behaviors.* Our type theory satisfies the canonicity property, and provides a constructive, elementary (although admittedly complex) semantics of univalence.

We have three main contributions. First and foremost, ours is the second type theory admitting both univalence and canonicity, after the cubical type theory of Cohen et al. [CCHM18]. In both cases, *cubical* refers to presentations of higher-dimensional structure using interval variables, products of which represent abstract n-dimensional hypercubes. We equip our interval with *Cartesian* structure and our types with a generalized Kan filling operation, whereas Cohen et al. [CCHM18] equip their interval with a stronger De Morgan structure and their types with a weaker Kan composition operation.[6] We therefore affirmatively resolve the open question of whether Cartesian interval structure constructively models univalent universes [Coq14; LB14].

Secondly, our semantics consist of programs and thus hew closely to the syntax of type theory, illuminating which rules and even which evaluation strategies to consider. In Appendix A we present a type theory in the style of the Nuprl *computational* type theory [Con+85], which admits a strict equality type with the equality reflection principle. Ours is the first type theory with canonicity that, like Voevodsky's Homotopy Type System, has both path types and strict equality types. Such type theories are called *two-level* [ACK16], because they must stratify types into those that respect paths and those that do not (notably, certain equality types). We moreover present a finer stratification of types than previously described, allowing us to deduce that some equality types do respect paths.

Thirdly, we place a novel *validity* restriction on the shapes of Kan filling scenarios to obtain a strong canonicity result for higher inductive types: all closed 0-dimensional elements of higher inductive types compute to constructors. In contrast, Cohen et al. [CCHM18] also admit 0-dimensional Kan compositions as canonical forms. Our result simplifies many proofs involving higher inductive types, and we believe it is critical for higher-dimensional programming—for instance, all elements of a type quotient are guaranteed to compute to constructors of the underlying type.

Cartesian cubical type theory has already been implemented in two experimental proof assistants, **RedPRL** [Red16] and **redtt** [Red18], with which the author and his collaborators have considered applications to synthetic homotopy theory, constructive mathematics, and programming. Notably, we have formalized the definition of semisimplicial types in **RedPRL**.[7] Cavallo and Harper [CH19a] have extended the present work with a schema for indexed higher inductive types, and *en passant* complete the Cartesian cubical semantics of Book HoTT by defining Book HoTT–style identity types.

Outline In Chapter 2 we introduce standard computational semantics in the setting of Idealized Nuprl, a core type theory modeled after Nuprl. We discuss the rationale

[6] See Chapter 3 for further discussion of cubical semantics, including the work of Awodey [Awo18] and Bezem, Coquand, and Huber [BCH14; BCH18].

[7] Available at `https://git.io/fhpTd`.

behind different systems of inference rules implemented by proof assistants, including the *computational type theory* of Nuprl and the *intensional type theory* of Coq, Agda, and Lean.

In Chapter 3 we describe the history of cubical type theories, compare various cube categories and Kan operations, and connect the path types of Cartesian cubical type theory with the identity types of Book HoTT. We also describe *regularity*, a technical problem which arises in cubical models of type theory, and *validity*, a refinement of our Kan operations which strengthens our canonicity theorem.

Chapter 4 presents the main technical results of this dissertation, namely, the computational semantics of Cartesian cubical type theory. These semantics directly justify the two-level, computational-style rules listed in Appendix A (roughly what is implemented in **RedPRL**), and also model the intensional-style rules of Appendix B (roughly **redtt**).

In Chapter 5 we summarize our contribution, discuss recent developments in cubical type theory, and conclude with remarks on notions of equality in dependent type theory.

Publications This dissertation's main results have been published:

- *Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities* [AFH18] details the main results of this dissertation. The associated preprint *Computational Higher Type Theory III: Univalent Universes and Exact Equality* [AFH17] is superseded by Chapter 4 of this dissertation.

- *The **RedPRL** Proof Assistant* [Ang+18] discusses Cartesian cubical computational type theory and its implementation in the **RedPRL** proof assistant.

- *Meaning explanations at higher dimension* [AH18] informally describes our semantics as a generalization of Martin-Löf's meaning explanations of type theory.

- *Computational Higher-Dimensional Type Theory* [AHW17] explains an earlier version of our semantics that did not yet account for univalent universes. The associated preprint *Computational Higher Type Theory II: Dependent Cubical Realizability* [AH17] is superseded by Chapter 4.

In addition, the preprint *Syntax and Models of Cartesian Cubical Type Theory* [Ang+19] outlines an alternative presentation of Cartesian cubical type theory without exact equality, along with an Agda formalization of its denotational semantics.

Readers may also wish to consult the author's paper *Homotopical patch theory* [Ang+16] (an extended version of an earlier conference paper [Ang+14]), which predates constructive models of univalence, and describes a conjectural application of Book HoTT to dependently-typed programming: modeling version control systems and their implementations as higher inductive types and functions from such types into univalent universes.

One paper *not* coauthored by the author deserves particular mention. *Higher Inductive Types in Cubical Computational Type Theory* [CH19a], with associated preprint *Computational Higher Type Theory IV: Inductive Types* [CH18], proposes a schema for indexed higher inductive types (including Book HoTT–style identity types) in Cartesian cubical computational type theory, and extends our computational semantics accordingly.

2 *Idealized Nuprl*

> *Types* of theories:
>
> (1) Theories with *one* intended model, such as number theory and (for the Platonist) analysis and classical set theory.
>
> (2) Theories with many intended models, such as group theory. In case (1), the primary aims of axiomatization are clarity and rigor. On the contrary, in case (2), the prime goal is generality, and rigor and clarity are by-products.
>
> —Michael J. Beeson,
> *Foundations of Constructive Mathematics* [Bee85, p. 83]

This chapter introduces *Idealized Nuprl*, a (non-univalent, non-cubical) constructive type theory based on Nuprl [Con+85], with dependent functions, dependent pairs, natural numbers, booleans, and two parallel cumulative hierarchies of universes. Like Nuprl, Idealized Nuprl defines types as behavioral predicates over computer programs, using Allen-style computational semantics [All87] that mathematize Martin-Löf's meaning explanations of type theory [ML82].

We intend this chapter to serve as a primer on computational semantics and the Nuprl tradition, divorced from cubical complications that arise in Chapter 4. We therefore intentionally omit many features of modern Nuprl that we have not yet developed in the cubical setting, including untyped computational equivalence [How89]. Unlike Nuprl, we include a predicative hierarchy Prop_i of universes of subsingleton types that are subtypes of the standard universe hierarchy Type_i, foreshadowing the Kan and pretype universes in Chapter 4. Parallel universes were previously considered by Krishnaswami, Pradic, and Benton [KPB15]; as far as we know, we are the first to consider subtyping between them.

To be clear, the main results of this chapter are not novel, and were previously described by (among others) Allen [All87], Harper [Har92], and Anand and Rahli [AR14]—the last of whom provide a Coq formalization! However, the author is unsatisfied with existing presentations for various reasons, and has strived for detail and clarity in this chapter.

Why computational semantics? There are many styles of type-theoretic semantics, often based on category theory. Computational semantics are essential to Nuprl as they comprise its *only* intended semantics—unlike intensional [ML75b] and extensional [ML82] type theory which, as discussed in Section 1.2, are celebrated in part for their range of models. As a result, the rules of Nuprl permit structurally ill-typed terms, terms with many distinct types, untyped reduction, non-termination, and many other features sensible in the λ-calculus but foreign to categorical semantics of type theories. Categorically, computational semantics are a model of type theory in *modest sets*, sets whose elements are tracked by collections of elements of a partial combinatory algebra [Oos08]. However, whereas universes of modest sets model an impredicative type-theoretic universe, the universes of computational semantics do not, because the latter are inductive-recursively defined modest sets of named collections of types.

More generally, our freestyle approach has the advantage of providing self-contained, elementary, constructive semantics directly justifying program extraction: indeed, we define $M \in A$ to mean that M is a program with behavior A. Computational semantics of intensional type theory, for instance, validate *erasure* of type annotations and identity proofs during extraction, as they are unneeded at runtime. In Cartesian cubical type theory, computational semantics allow us to consider the efficiency of Kan operations, suggesting faster algorithms valid only for closed terms, or containing structurally ill-typed subterms.

Being self-contained and constructive, computational semantics are quite portable across metatheories. We have written Chapters 2 and 4 in an agnostic style compatible with both (constructive) set theory and extensional type theory. Sterling and Harper [SH18] develop the computational semantics of guarded computational type theory in the internal language of a particular presheaf topos that they simulate by Coq with axioms.

Computational semantics are an instance of the technique of *logical relations*, or Tait's method [Tai67], which is commonly used in type theory to establish completeness of normalization by evaluation [Abe13], and in programming language theory to prove properties of type systems, such as termination, parametricity, and safety. Our computational semantics constitute a logical relations proof that the rules of Idealized Nuprl enforce behavioral properties of extracts. (The Nuprl perspective, however, is typically the reverse—viewing not the rules but the semantic relations as definitive.)

2.1 *Syntax and operational semantics*

Idealized Nuprl is founded on an untyped functional programming language whose syntax is presented in Figure 2.1. This language extends the λ-calculus in a standard fashion and is an instance of both Martin-Löf's system of arities [NPS90, Chapter 3] and Harper's abstract binding trees [Har16, Chapter 1]. We include type constructors because dependency induces computation in types; types and terms cannot have distinct syntactic sorts because typehood in Idealized Nuprl is posterior to operational semantics.

We follow Harper's syntax chart notation in Figure 2.1, writing every term first as an abstract binding tree, then in a friendlier concrete notation, and finally in English. Capital

$M :=$	$\Pi(A, a.B)$	$(a{:}A) \longrightarrow B$	dependent function type
	$\lambda(a.M)$	$\lambda a.M$	lambda abstraction
	$\mathsf{app}(M, N)$	$M\ N$	function application
	$\Sigma(A, a.B)$	$(a{:}A) \times B$	dependent pair type
	$\mathsf{pair}(M, N)$	$\langle M, N \rangle$	pairing
	$\mathsf{fst}(M)$	$\mathsf{fst}(M)$	first projection
	$\mathsf{snd}(M)$	$\mathsf{snd}(M)$	second projection
	$\mathsf{Eq}(A, M, N)$	$\mathsf{Eq}_A(M, N)$	equality type
	refl	$\star$	equality proof
	nat	nat	natural number type
	z	z	zero
	$\mathsf{s}(M)$	$\mathsf{s}(M)$	successor
	$\mathsf{natrec}(M, N_1, n.a.N_2)$	$\mathsf{natrec}(M; N_1, n.a.N_2)$	natural number recursion
	bool	bool	boolean type
	true	true	true
	false	false	false
	$\mathsf{if}(M, N_1, N_2)$	$\mathsf{if}(M; N_1, N_2)$	boolean recursion
	$\mathsf{Prop}[i]$	Prop_i	ith subsingleton universe
	$\mathsf{Type}[i]$	Type_i	ith type universe

Figure 2.1: Syntax of Idealized Nuprl.

letters ($M, N, A, \dots$) represent terms, and lowercase letters (a, n) represent variables and binders. (We deliberately avoid x, y, z as Chapter 4 uses these for interval variables.) Note that in $(a{:}A) \longrightarrow B$ and $(a{:}A) \times B$, the variable a is bound in B; we write $A \longrightarrow B$ and $A \times B$ respectively when a does not occur in B.

As is customary, we equate α-equivalent terms[1] (quotienting terms by renaming of bound variables, or choosing a nameless representation of variable binders), and equip terms with a capture-avoiding substitution $M[N/a]$ that replaces free occurrences of a with N in M. Readers unfamiliar with these concepts may wish to consult Harper [Har16, Chapter 1]. We do *not* quotient terms by β- or η-equivalence at this stage. (Modern Nuprl quotients untyped terms by applicative bisimulation [How89], as we discuss in Section 2.6.)

We give operational meaning to *closed* terms (those without free variables) using a structural operational semantics [Plo81] defined in Figure 2.2. The judgment $M_0\ \mathsf{val}$ specifies when M_0 is a value, and $M \longmapsto M'$ specifies when M takes one step of weak head evaluation to M'. We will write $M \longmapsto^* M'$ when M steps to M' in zero or more steps, and $M \Downarrow M_0$, or M *evaluates* to M_0, when $M \longmapsto^* M_0$ and $M_0\ \mathsf{val}$.

[1] Surprisingly, Allen does *not* equate α-equivalent terms [All87, p. 66]!

$$\frac{}{(a{:}A) \to B\ \mathsf{val}} \qquad \frac{}{\lambda a.M\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{M\ N \longmapsto M'\ N} \qquad \frac{}{(\lambda a.M)\ N \longmapsto M[N/a]}$$

$$\frac{}{(a{:}A) \times B\ \mathsf{val}} \qquad \frac{}{\langle M, N \rangle\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{\mathsf{fst}(M) \longmapsto \mathsf{fst}(M')} \qquad \frac{M \longmapsto M'}{\mathsf{snd}(M) \longmapsto \mathsf{snd}(M')}$$

$$\frac{}{\mathsf{fst}(\langle M, N \rangle) \longmapsto M} \qquad \frac{}{\mathsf{snd}(\langle M, N \rangle) \longmapsto N}$$

$$\frac{}{\mathsf{Eq}_A(M, N)\ \mathsf{val}} \qquad \frac{}{\star\ \mathsf{val}}$$

$$\frac{}{\mathsf{nat}\ \mathsf{val}} \qquad \frac{}{\mathsf{z}\ \mathsf{val}} \qquad \frac{}{\mathsf{s}(M)\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{\mathsf{natrec}(M; Z, n.a.S) \longmapsto \mathsf{natrec}(M'; Z, n.a.S)}$$

$$\frac{}{\mathsf{natrec}(\mathsf{z}; Z, n.a.S) \longmapsto Z} \qquad \frac{}{\mathsf{natrec}(\mathsf{s}(M); Z, n.a.S) \longmapsto S[M/n][\mathsf{natrec}(M; Z, n.a.S)/a]}$$

$$\frac{}{\mathsf{bool}\ \mathsf{val}} \qquad \frac{}{\mathsf{true}\ \mathsf{val}} \qquad \frac{}{\mathsf{false}\ \mathsf{val}}$$

$$\frac{M \longmapsto M'}{\mathsf{if}(M; T, F) \longmapsto \mathsf{if}(M'; T, F)} \qquad \frac{}{\mathsf{if}(\mathsf{true}; T, F) \longmapsto T} \qquad \frac{}{\mathsf{if}(\mathsf{false}; T, F) \longmapsto F}$$

$$\frac{}{\mathsf{Prop}_i\ \mathsf{val}} \qquad \frac{}{\mathsf{Type}_i\ \mathsf{val}}$$

Figure 2.2: Operational semantics of Idealized Nuprl.

These judgments satisfy two key properties easily proven by induction on their definitions. First, if M val, then $M \not\longmapsto$. (The converse is not the case; counterexamples, such as $\mathsf{fst}(\lambda a.a)$, are called *stuck*.) Second is determinacy of evaluation: if $M \longmapsto M'$ and $M \longmapsto M''$, then $M' = M''$.

2.2 Constructing type systems

Our programming language contains building blocks of mathematics—functions, pairs, natural numbers, *et cetera*—which we must now assemble into the types and elements of Idealized Nuprl. We briefly survey the requirements of our construction before making them precise below. In short, we must define a collection $\mathbf{V}_0 \subset \mathbf{Tm}$ of closed terms that name types, and assign to each $A \in \mathbf{V}_0$ a collection $\mathbf{El}(A) \subset \mathbf{Tm}$ of its closed term elements.

We define $\mathbf{V}_0$ inductively. To define the type of Booleans, for instance, we specify that $\mathsf{bool} \in \mathbf{V}_0$. Clearly true and false denote Booleans, as do all programs evaluating to true or false, such as $(\lambda a.a)$ true; thus $\mathbf{El}(\mathsf{bool}) = \{M \mid M \Downarrow \mathsf{true} \vee M \Downarrow \mathsf{false}\}$. As for the type of equality proofs, we require at a minimum that $\mathsf{Eq}_A(M, N) \in \mathbf{V}_0$ when A is a type ($A \in \mathbf{V}_0$) with elements M and N ($M, N \in \mathbf{El}(A)$)—a strange condition, because the definition of $\mathbf{V}_0$ now refers to $\mathbf{El}$, a function *out of* $\mathbf{V}_0$! Because $\mathsf{Eq}_A(-, -)$ is a dependent type, we must also demand that it respect equality in its arguments, lest $\lambda a.\mathsf{Eq}_A(M, a)$ not form an A-indexed family of types. (In fact, without any such condition, even $\mathsf{Eq}_{\mathsf{bool}}(\mathsf{true}, \mathsf{true})$ and $\mathsf{Eq}_{\mathsf{bool}}(\mathsf{true}, (\lambda a.a)\ \mathsf{true})$ need not be equal!)

Element equality is type-sensitive (for instance, because of η principles), so we must define for each type A not only $\mathbf{El}(A)$ but also an equivalence relation on $\mathbf{El}(A)$, and check that types dependent on A send equal elements of $\mathbf{El}(A)$ to equal types. Equality of types, in turn, forms an equivalence relation on $\mathbf{V}_0$ that $\mathbf{El}$ must respect.

Finally, given $\mathbf{V}_0$ and $\mathbf{El}$, we construct a second, larger collection of types $\mathbf{V}_1$ closed under the same type formers, with an additional base type $\mathsf{Type}_0 \in \mathbf{V}_1$ whose elements are types of the previous level ($\mathbf{El}(\mathsf{Type}_0) = \mathbf{V}_0$). We obtain a cumulative hierarchy of universes in $\mathbf{V}_\omega$ by iterating this process $\mathbf{V}_0 \subset \mathbf{V}_1 \subset \cdots$ where $\mathsf{Type}_0, \mathsf{Type}_1, \ldots, \mathsf{Type}_{n-1} \in \mathbf{V}_n$.

Actualizing the vision Defining simultaneously a set $\mathbf{V}_i$, an equivalence relation on $\mathbf{V}_i$, a map $\mathbf{El} : \mathbf{V}_i \longrightarrow \mathbf{Set}$, and an equivalence relation on each $\mathbf{El}(A)$ is a process requiring careful justification. As a cautionary tale, recall that type theories with the rule $\mathsf{Type}_i \in \mathsf{Type}_i$ are inconsistent, essentially for the same reason that there is no *set of all sets* [Hur95]. Such a principle is not obviously incompatible with the semantics outlined above—in particular, there is no *size* issue with a construction in which $\mathsf{Type}_i \in \mathbf{V}_i$ and $\mathbf{El}(\mathsf{Type}_i) = \mathbf{V}_i$. Instead, we will see that such a definition is subtly ill-founded.

Induction-recursion is a generalization of type-theoretic inductive definition in which an inductive type is defined simultaneously with a recursive function out of that type [Dyb00].

Our definition of $\mathbf{V}_i$ can be justified by induction-recursion, and was in fact its motivating example. In this dissertation we opt instead for a fixed point construction described by Allen [All87]: we find Allen's construction more straightforward on paper and appreciate that it can be carried out in set theory or type theories without induction-recursion. Indeed, Allen's construction is also used by Nuprl in Coq [AR14], as Coq lacks induction-recursion.

Rather than specifying first a subset of **Tm** and then an equivalence relation on that subset, we define both simultaneously as a *partial equivalence relation* (PER), or a symmetric and transitive binary relation. A partial equivalence relation R on **Tm** corresponds to the subset $\{M \in \mathbf{Tm} \mid R(M, M)\}$ on which it is an equivalence relation; conversely, an equivalence relation R on a subset of **Tm** is a PER on the whole of **Tm**. Because typehood and membership should respect evaluation, we define them as PERs on *values* and lift to terms by evaluation. We cannot actually quotient terms by semantic equality at any point during this process because computation can distinguish semantically equal terms.

Summing up, we must define a PER of value types, and for each value type, a PER of value elements. Harper [Har92] constructs these as the least fixed point of a monotone function on the pointed directed-complete partial order $(V{:}\mathbf{PER}(\mathbf{Val})) \times (\mathbf{Val}/V \to \mathbf{PER}(\mathbf{Val}))$. Unfortunately, the existence of such a fixed point requires a classical cardinality argument, and to make Harper's construction precise, one must verify the monotone function sends PERs to PERs and respects type equality. Following Allen [All87], we instead achieve ontological reality with a least fixed point on the complete lattice of *candidate* (or *possible*) *type systems* $\mathcal{P}(\mathbf{Val} \times \mathbf{Val} \times \mathcal{P}(\mathbf{Val} \times \mathbf{Val}))$, and observe after the fact that we have constructed a proper *type system*.

Definition 2.1. A *candidate type system* is a relation $\tau(A_0, B_0, \varphi)$ over A_0 val, B_0 val, and binary relations $\varphi(M_0, N_0)$ over M_0 val and N_0 val.

Definition 2.2. A *type system* is a candidate type system τ satisfying:

1. *Unicity*: If $\tau(A_0, B_0, \varphi)$ and $\tau(A_0, B_0, \varphi')$ then $\varphi = \varphi'$.
2. *PER-valuation*: If $\tau(A_0, B_0, \varphi)$ then φ is symmetric and transitive.
3. *Symmetry*: If $\tau(A_0, B_0, \varphi)$ then $\tau(B_0, A_0, \varphi)$.
4. *Transitivity*: If $\tau(A_0, B_0, \varphi)$ and $\tau(B_0, C_0, \varphi)$ then $\tau(A_0, C_0, \varphi)$.

The relation $\tau(A_0, B_0, \varphi)$ encodes that A_0 and B_0 are equal value types with element relation φ. The conditions of Definition 2.2 ensure that types have unique element relations, and that type and element equality are PERs.

Our construction uses the Knaster–Tarski fixed point theorem, which states that any monotone (order-preserving) function $F(x)$ on a complete lattice has a least fixed point $\mu x.F(x)$ that is also its least pre-fixed point [DP02, 2.35]. We consider monotone functions on the complete lattice of candidate type systems[2] ordered by inclusion, built using the following combinators that resurface as judgments in Definitions 2.12 and 2.20:

Definition 2.3 (Candidate judgments)**.** Given a candidate type system τ:

1. $A \sim A' \downarrow \alpha \in \tau$ when $A \Downarrow A_0$, $A' \Downarrow A'_0$, and $\tau(A_0, A'_0, \alpha)$.
2. $M \sim M' \in \alpha$ when $M \Downarrow M_0$, $M' \Downarrow M'_0$, and $\alpha(M_0, M'_0)$.
3. $a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau$ when for all $M \sim M' \in \alpha$, $B[M/a] \sim B'[M'/a] \downarrow \beta_{M,M'} \in \tau$.
4. $a : \alpha \triangleright N \sim N' \in \beta$ when for all $M \sim M' \in \alpha$, $N[M/a] \sim N'[M'/a] \in \beta_{M,M'}$.

In Figure 2.3, we define each type former as a monotone function on candidate type systems. For instance, $\textsc{Pair}(\tau)$ is a candidate type system whose types are all possible dependent pairs of types and type families of τ, defined by formation and introduction rules.[3] Base types $\textsc{Nat}$ and $\textsc{Bool}$ take no input because their meaning is constant in τ. Candidate type system $\tau_0 := \mu\tau.\textsc{Types}(\varnothing, \tau)$ is the least fixed point of the union (disjoint by construction) of the type formers, and is therefore the least candidate type system closed under the non-universe type formers. As with $\mathbf{V}_0$ earlier, τ_0 will serve as universe Type_0 of the final type system.

Idealized Nuprl has a second universe hierarchy $\mathsf{Prop}_i \subset \mathsf{Type}_i$ whose elements are *subsingletons*, types all of whose elements are equal. We construct $\pi_0 := \mu\pi.\textsc{Props}(\pi, \tau_0)$ as the least fixed point of the type formers that preserve subsingletons: $\mathsf{Eq}_A(M, N)$ for any A (drawn from τ_0), $(a{:}A) \times B(a)$ for subsingletons A and $B(a)$ (both drawn from π_0), and $(a{:}A) \to B$ for A in τ_0 and $B(a)$ in π_0.

We define a third candidate type system, ν_1, containing only two types—Type_0 whose elements are types of τ_0, and Prop_0 whose elements are types of π_0—and then a fourth candidate type system, τ_1, the least fixed point of the type formers *and* the universes of ν_1. We iterate this process for all $i \in \{0, 1, \dots, \omega\}$ to obtain three sequences of candidate type systems: τ_i of all types with universes $< i$, π_i of subsingleton types with universes $< i$, and ν_i of only universes $< i$.

Suppose we had instead considered a single universe $\mathsf{Type} \in \mathsf{Type}$ containing itself:

$$\textsc{Types}(\tau) = \textsc{Fun}(\tau, \tau) \cup \cdots \cup \{(\mathsf{Type}, \mathsf{Type}, \{(A_0, B_0) \mid \tau(A_0, B_0, \varphi)\})\}$$

Such a $\textsc{Types}$ function is not monotone, since adding types to τ does not preserve the elements of Type, so we are not guaranteed a least fixed point. Readers who prefer type

[2] Type systems do not form a complete lattice, as unicity and transitivity are not preserved by joins.

[3] Standard logical relations define function and pair types not by introduction but rather *elimination*: one demands not that $M \in (a{:}A) \times B(a)$ evaluate to a pair, but instead that $\mathsf{fst}(M) \in A$ and $\mathsf{snd}(M) \in B(\mathsf{fst}(M))$. Either definition is workable in this chapter, but relying on introduction forms will be crucial in Chapter 4.

$$
\begin{aligned}
\textsc{Fun}(\tau, \pi) &\coloneqq \{((a{:}A) \to B, (a{:}A') \to B', \varphi) \mid \\
&\qquad \exists \alpha, \beta.(A \sim A' \downarrow \alpha \in \tau) \wedge (a : \alpha \triangleright B \sim B' \downarrow \beta \in \pi) \\
&\qquad \wedge\ \varphi = \{(\lambda a.N, \lambda a.N') \mid a : \alpha \triangleright N \sim N' \in \beta\}\} \\
\textsc{Pair}(\tau) &\coloneqq \{((a{:}A) \times B, (a{:}A') \times B', \varphi) \mid \\
&\qquad \exists \alpha, \beta.(A \sim A' \downarrow \alpha \in \tau) \wedge (a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau) \\
&\qquad \wedge\ \varphi = \{(\langle M, N\rangle, \langle M', N'\rangle) \mid (M \sim M' \in \alpha) \wedge (N \sim N' \in \beta_{M,M'})\}\} \\
\textsc{Eq}(\tau) &\coloneqq \{(\mathsf{Eq}_{A}(M, N), \mathsf{Eq}_{A'}(M', N'), \varphi) \mid \\
&\qquad \exists \alpha.(A \sim A' \downarrow \alpha \in \tau) \wedge (M \sim M' \in \alpha) \wedge (N \sim N' \in \alpha) \\
&\qquad \wedge\ (\varphi = \{(\star, \star) \mid M \sim N \in \alpha\})\} \\
\textsc{Nat} &\coloneqq \{(\mathsf{nat}, \mathsf{nat}, \varphi) \mid \varphi = \mu R.(\{(\mathsf{z}, \mathsf{z})\} \cup \{(\mathsf{s}(M), \mathsf{s}(M')) \mid M \sim M' \in R\})\} \\
\textsc{Bool} &\coloneqq \{(\mathsf{bool}, \mathsf{bool}, \varphi) \mid \varphi = \{(\mathsf{true}, \mathsf{true}), (\mathsf{false}, \mathsf{false})\}\} \\
\textsc{Prop}(\nu) &\coloneqq \{(\mathsf{Prop}_i, \mathsf{Prop}_i, \varphi) \mid \mathsf{Prop}_i \sim \mathsf{Prop}_i \downarrow \varphi \in \nu\} \\
\textsc{Type}(\nu) &\coloneqq \{(\mathsf{Type}_i, \mathsf{Type}_i, \varphi) \mid \mathsf{Type}_i \sim \mathsf{Type}_i \downarrow \varphi \in \nu\}
\end{aligned}
$$

$$
\begin{aligned}
\textsc{Props}(\pi, \tau) &\coloneqq \textsc{Fun}(\tau, \pi) \cup \textsc{Pair}(\pi) \cup \textsc{Eq}(\tau) \\
\textsc{Types}(\nu, \tau) &\coloneqq \textsc{Fun}(\tau, \tau) \cup \textsc{Pair}(\tau) \cup \textsc{Eq}(\tau) \cup \textsc{Nat} \cup \textsc{Bool} \cup \textsc{Prop}(\nu) \cup \textsc{Type}(\nu)
\end{aligned}
$$

$$
\begin{aligned}
\nu_n &\coloneqq \{(\mathsf{Prop}_i, \mathsf{Prop}_i, \varphi) \mid (i < n) \wedge (\varphi = \{(A_0, B_0) \mid \exists \alpha.A_0 \sim B_0 \downarrow \alpha \in \pi_i\})\} \\
&\ \cup\ \{(\mathsf{Type}_i, \mathsf{Type}_i, \varphi) \mid (i < n) \wedge (\varphi = \{(A_0, B_0) \mid \exists \alpha.A_0 \sim B_0 \downarrow \alpha \in \tau_i\})\} \\
\pi_n &\coloneqq \mu\pi.\textsc{Props}(\pi, \tau_n) \\
\tau_n &\coloneqq \mu\tau.\textsc{Types}(\nu_n, \tau)
\end{aligned}
$$

Figure 2.3: PER semantics of Idealized Nuprl.

theory as a metatheory can read Figure 2.3 quite directly as an inductive definition of $\tau_i, \pi_i : \mathbf{Val} \times \mathbf{Val} \times (\mathbf{Val} \times \mathbf{Val} \to \mathbf{Prop}) \to \mathbf{Prop}$ for an impredicative **Prop**, in which case the above Types would be insufficiently *positive*. In the absence of impredicative **Prop**, one can define n levels of Idealized Nuprl with $(n + 1)$ predicative universes [AR14].

The rest of this section is devoted to proofs that τ_i, π_i are type systems (Theorem 2.8) and that $\pi_i \subset \tau_i$ (Theorem 2.11). We hope some readers will find these proofs instructive, as the author was unable to locate any paper versions of these proofs longer than "By induction." The uninterested reader may nevertheless safely skip to Section 2.3.

Lemma 2.4. *Let* $F(R) = \{(\mathrm{z}, \mathrm{z})\} \cup \{(\mathrm{s}(M), \mathrm{s}(M')) \mid M \sim M' \in R\}$. *Then* $\mathbb{N} := \mu R.F(R)$ *is symmetric and transitive.*

Proof.

1. *Symmetry.*

   Let $\Phi = \{(M_0', M_0) \mid \mathbb{N}(M_0, M_0')\}$, and show that Φ is a pre-fixed point of F (that is, $F(\Phi) \subseteq \Phi$); $\mathbb{N}$ is the least pre-fixed point of F, so $\mathbb{N} \subseteq \Phi$, and thus $\mathbb{N}$ is symmetric.

   There are two cases. If $F(\Phi)(\mathrm{z}, \mathrm{z})$, then we must show $\Phi(\mathrm{z}, \mathrm{z})$, which is trivial. If $F(\Phi)(\mathrm{s}(M'), \mathrm{s}(M))$, then $M' \sim M \in \Phi$ and hence $M \sim M' \in \mathbb{N}$; we must show $\mathbb{N}(\mathrm{s}(M), \mathrm{s}(M'))$, which follows by $\mathbb{N} = F(\mathbb{N})$.

2. *Transitivity.*

   Let $\Phi = \{(M_0, M_0') \mid \forall M_0''.\mathbb{N}(M_0', M_0'') \implies \mathbb{N}(M_0, M_0'')\}$, and show that $F(\Phi) \subseteq \Phi$; it follows that $\mathbb{N} \subseteq \Phi$, and therefore that $\mathbb{N}$ is transitive.

   Again, there are two cases. If $F(\Phi)(\mathrm{z}, \mathrm{z})$, we must show $\Phi(\mathrm{z}, \mathrm{z})$, that is, for all M_0 such that $\mathbb{N}(\mathrm{z}, M_0)$ we have $\mathbb{N}(\mathrm{z}, M_0)$, which is trivial. If $F(\Phi)(\mathrm{s}(M), \mathrm{s}(M'))$ (that is, $M \sim M' \in \Phi$), we must show $\Phi(\mathrm{s}(M), \mathrm{s}(M'))$: for all values M_0'' such that $\mathbb{N}(\mathrm{s}(M'), M_0'')$, we must show $\mathbb{N}(\mathrm{s}(M), M_0'')$. But then $M_0'' = \mathrm{s}(M'')$ and $M' \sim M'' \in \mathbb{N}$ for some M'', so by $M \sim M' \in \Phi$, $M \sim M'' \in \mathbb{N}$. The result follows by $\mathbb{N} = F(\mathbb{N})$. □

Notice that $\mathbb{N}$ and Φ are defined only on values, but the successor case of F forces us to confront non-values as well. In (1), $M' \sim M \in \Phi$ implies $M \sim M' \in \mathbb{N}$ not immediately by definition, but by observing that $M \Downarrow M_0$, $M' \Downarrow M_0'$, and $\Phi(M_0', M_0)$, hence $\mathbb{N}(M_0, M_0')$, hence $M \sim M' \in \mathbb{N}$. The next proofs are complicated further by Fun and Pair, which lift inductive hypotheses to families of open terms; in the proof of Lemma 2.6, this lifting forces us to simultaneously prove transitivity on the left and right.

Lemma 2.5. *If* ν *is a type system, then* $\mu_{\textit{Types}}(\nu) := \mu\tau.\textit{Types}(\nu, \tau)$ *satisfies unicity.*

Proof. Let $\Phi = \{(A_0, B_0, \varphi) \mid \forall \varphi'.\mu_{\text{Types}}(\nu)(A_0, B_0, \varphi') \implies (\varphi = \varphi')\}$ and prove Φ is a pre-fixed point of Types$(\nu, -)$ (that is, Types$(\nu, \Phi) \subseteq \Phi$). Because $\mu_{\text{Types}}(\nu)$ is the least pre-fixed point of $\textsc{Types}(\nu, -)$, it will follow that $\mu_{\textsc{Types}}(\nu) \subseteq \Phi$, and therefore that $\mu_{\textsc{Types}}(\nu)$ satisfies unicity. We can consider each type former separately because they are disjoint. We will write out the case for $\textsc{Fun}$ carefully; $\textsc{Pair}$ and $\textsc{Eq}$ are similar, $\textsc{Nat}$ and $\textsc{Bool}$ are immediate, and $\textsc{Prop}$ and $\textsc{Type}$ follow from the unicity of ν.

Consider the $\textsc{Fun}$ clause of $\textsc{Types}(\nu, \Phi)$: suppose $\textsc{Fun}(\Phi, \Phi)((a{:}A) \to B, (a{:}A') \to B', \varphi)$ and show $\Phi((a{:}A) \to B, (a{:}A') \to B', \varphi)$, that is, if $\mu_{\textsc{Types}}(\nu)((a{:}A) \to B, (a{:}A') \to B', \varphi')$ then $\varphi = \varphi'$. Unrolling $\mu_{\textsc{Types}}(\nu)$, we have $A \sim A' \downarrow \alpha' \in \mu_{\textsc{Types}}(\nu)$ and $a : \alpha' \vartriangleright B \sim B' \downarrow \beta' \in \mu_{\textsc{Types}}(\nu)$ for some α', β'. Unrolling $\textsc{Fun}(\Phi, \Phi)$, we have $A \sim A' \downarrow \alpha \in \Phi$, or for any α'' such that $A \sim A' \downarrow \alpha'' \in \mu_{\textsc{Types}}(\nu)$, $\alpha = \alpha''$. Such an α'' exists (namely, α'), so it is unique. Similarly, by $a : \alpha \vartriangleright B \sim B' \downarrow \beta \in \Phi$, we know that for any M, M' with $M \sim M' \in \alpha$, there is a unique $\beta'_{M,M'}$ such that $B[M/a] \sim B'[M'/a] \downarrow \beta'_{M,M'} \in \mu_{\textsc{Types}}(\nu)$. Because φ, φ' are determined by the choice of α' and those $\beta'_{M,M'}$ we conclude that $\varphi = \varphi'$. □

Lemma 2.6. *If ν is a type system, then $\mu_{\textsc{Types}}(\nu) := \mu\tau.\textsc{Types}(\nu, \tau)$ is a type system.*

Proof. We must prove PER-valuation, symmetry, and transitivity simultaneously. Let

$$\begin{aligned}\Phi(A_0, B_0, \varphi) = \{&\mu_{\textsc{Types}}(\nu)(A_0, B_0, \varphi) \land \varphi \text{ is a PER} \land \mu_{\textsc{Types}}(\nu)(B_0, A_0, \varphi)\\ &\land (\forall C_0, \varphi'.\mu_{\textsc{Types}}(\nu)(B_0, C_0, \varphi') \implies \mu_{\textsc{Types}}(\nu)(A_0, C_0, \varphi) \land \varphi = \varphi')\\ &\land (\forall C_0, \varphi'.\mu_{\textsc{Types}}(\nu)(C_0, A_0, \varphi') \implies \mu_{\textsc{Types}}(\nu)(C_0, B_0, \varphi) \land \varphi = \varphi')\}\end{aligned}$$

We prove $\textsc{Types}(\nu, \Phi) \subseteq \Phi$; it will follow that $\mu_{\textsc{Types}}(\nu) \subseteq \Phi$, and by Lemma 2.5 that $\mu_{\textsc{Types}}(\nu)$ is a type system. (Surprisingly, unicity figures nowhere else in this proof!) Suppose $\textsc{Fun}(\Phi, \Phi)((a{:}A) \to B, (a{:}A') \to B', \varphi)$. Unfolding definitions,

1. $A \sim A' \downarrow \alpha \in \mu_{\textsc{Types}}(\nu)$,
2. α is a PER,
3. $A' \sim A \downarrow \alpha \in \mu_{\textsc{Types}}(\nu)$,
4. if $A' \sim A'' \downarrow \alpha' \in \mu_{\textsc{Types}}(\nu)$ then $A \sim A'' \downarrow \alpha \in \mu_{\textsc{Types}}(\nu)$ and $\alpha = \alpha'$,
5. if $A'' \sim A \downarrow \alpha' \in \mu_{\textsc{Types}}(\nu)$ then $A'' \sim A' \downarrow \alpha \in \mu_{\textsc{Types}}(\nu)$ and $\alpha = \alpha'$,
6. for all $M \sim M' \in \alpha$, $B[M/a] \sim B'[M'/a] \downarrow \beta_{M,M'} \in \mu_{\textsc{Types}}(\nu)$,
7. for all $M \sim M' \in \alpha$, $\beta_{M,M'}$ is a PER,
8. for all $M \sim M' \in \alpha$, $B'[M'/a] \sim B[M/a] \downarrow \beta_{M,M'} \in \mu_{\textsc{Types}}(\nu)$,
9. for all $M \sim M' \in \alpha$, if $B'[M'/a] \sim C \downarrow \beta'_{M,M'} \in \mu_{\textsc{Types}}(\nu)$ then $B[M/a] \sim C \downarrow \beta_{M,M'} \in \mu_{\textsc{Types}}(\nu)$ and $\beta_{M,M'} = \beta'_{M,M'}$,

10. for all $M \sim M' \in \alpha$, if $C \sim B[M/a] \downarrow \beta'_{M,M'} \in \mu_{\text{TYPES}}(\nu)$ then $C \sim B'[M'/a] \downarrow \beta_{M,M'} \in \mu_{\text{TYPES}}(\nu)$ and $\beta_{M,M'} = \beta'_{M,M'}$, and

11. $\varphi = \{(\lambda a.N, \lambda a.N') \mid a : \alpha \triangleright N \sim N' \in \beta\}$.

We prove each component of $\Phi((a{:}A) \to B, (a{:}A') \to B', \varphi)$ separately:

1. We have $\mu_{\text{TYPES}}(\nu)((a{:}A) \to B, (a{:}A') \to B', \varphi)$ by (1,6,11).

2. *PER-valuation.*

   To prove φ is transitive, suppose that $a : \alpha \triangleright N \sim N' \in \beta$, $a : \alpha \triangleright N' \sim N'' \in \beta$, and $M \sim M' \in \alpha$, and show $N[M/a] \sim N''[M'/a] \in \beta_{M,M'}$. Because α is a PER, $M \sim M \in \alpha$, so $N[M/a] \sim N'[M/a] \in \beta_{M,M}$ and $N'[M/a] \sim N''[M'/a] \in \beta_{M,M'}$. We know $\beta_{M,M'}$ is transitive, so it suffices to show that $\beta_{M,M} = \beta_{M,M'}$:

   $$\begin{aligned} B'[M/a] \sim B[M/a] \downarrow \beta_{M,M} &\in \mu_{\text{TYPES}}(\nu) && \text{by (8)} \\ B[M/a] \sim B'[M'/a] \downarrow \beta_{M,M'} &\in \mu_{\text{TYPES}}(\nu) && \text{by (6)} \\ \beta_{M,M} &= \beta_{M,M'} && \text{by (10)} \end{aligned}$$

   To prove φ is symmetric, suppose that $a : \alpha \triangleright N \sim N' \in \beta$ and $M \sim M' \in \alpha$, and show $N'[M/a] \sim N[M'/a] \in \beta_{M,M'}$. Because α is a PER, $M' \sim M \in \alpha$, so $N[M'/a] \sim N'[M/a] \in \beta_{M',M}$. We know $\beta_{M,M'}$ is symmetric, so it suffices to show that $\beta_{M',M} = \beta_{M,M'}$:

   $$\begin{aligned} B[M/a] \sim B'[M/a] \downarrow \beta_{M,M} &\in \mu_{\text{TYPES}}(\nu) && \text{by (6)} \\ B'[M/a] \sim B[M'/a] \downarrow \beta_{M',M} &\in \mu_{\text{TYPES}}(\nu) && \text{by (8)} \\ \beta_{M',M} = \beta_{M,M} &= \beta_{M,M'} && \text{by (9), } \beta_{M,M} = \beta_{M,M'} \end{aligned}$$

3. *Symmetry.*

   We must show $\mu_{\text{TYPES}}(\nu)((a{:}A') \to B', (a{:}A) \to B, \varphi)$. By (3) we have $A' \sim A \downarrow \alpha \in \mu_{\text{TYPES}}(\nu)$. It suffices to show $a : \alpha \triangleright B' \sim B \downarrow \beta \in \mu_{\text{TYPES}}(\nu)$. Suppose $M \sim M' \in \alpha$. By (2,8) we have $B'[M/a] \sim B[M'/a] \downarrow \beta_{M',M} \in \mu_{\text{TYPES}}(\nu)$, and by the above argument, $\beta_{M',M} = \beta_{M,M'}$.

4. *Right transitivity.*

   Suppose $\mu_{\text{TYPES}}(\nu)((a{:}A') \to B', C_0, \varphi')$, and show $\mu_{\text{TYPES}}(\nu)((a{:}A) \to B, C_0, \varphi)$ and $\varphi = \varphi'$. By inspecting the definition of TYPES, we conclude $C_0 = (a{:}A'') \to B''$, $A' \sim A'' \downarrow \alpha' \in \mu_{\text{TYPES}}(\nu)$, and $a : \alpha' \triangleright B' \sim B'' \downarrow \beta' \in \mu_{\text{TYPES}}(\nu)$. By (4), $A \sim A'' \downarrow \alpha \in$

$\mu_{\text{Types}}(\nu)$ and $\alpha = \alpha'$. It suffices to show $a : \alpha \triangleright B \sim B'' \downarrow \beta \in \mu_{\text{Types}}(\nu)$. Suppose $M \sim M' \in \alpha$.

$$\begin{aligned} B[M/a] &\sim B'[M/a] \downarrow \beta_{M,M} \in \mu_{\text{Types}}(\nu) && \text{by (6)} \\ B'[M/a] &\sim B''[M'/a] \downarrow \beta'_{M,M'} \in \mu_{\text{Types}}(\nu) && \text{by hypothesis} \\ B[M/a] &\sim B''[M'/a] \downarrow \beta_{M,M'} \in \mu_{\text{Types}}(\nu) && \text{by (9)}, \beta_{M,M} = \beta_{M,M'} \end{aligned}$$

5. *Left transitivity.*

   Suppose $\mu_{\text{Types}}(\nu)(C_0, (a{:}A) \longrightarrow B, \varphi')$, and show $\mu_{\text{Types}}(\nu)(C_0, (a{:}A') \longrightarrow B', \varphi)$ and $\varphi = \varphi'$. By the definition of Types, $C_0 = (a{:}A'') \longrightarrow B''$, $A'' \sim A \downarrow \alpha' \in \mu_{\text{Types}}(\nu)$, and $a : \alpha' \triangleright B'' \sim B \downarrow \beta' \in \mu_{\text{Types}}(\nu)$. By (5), $A'' \sim A' \downarrow \alpha \in \mu_{\text{Types}}(\nu)$ and $\alpha = \alpha'$. To show $a : \alpha \triangleright B'' \sim B' \downarrow \beta \in \mu_{\text{Types}}(\nu)$, suppose $M \sim M' \in \alpha$.

$$\begin{aligned} B''[M/a] &\sim B[M'/a] \downarrow \beta'_{M,M'} \in \mu_{\text{Types}}(\nu) && \text{by hypothesis} \\ B[M'/a] &\sim B'[M'/a] \downarrow \beta_{M',M'} \in \mu_{\text{Types}}(\nu) && \text{by (6)} \\ B''[M/a] &\sim B'[M'/a] \downarrow \beta_{M',M'} \in \mu_{\text{Types}}(\nu) && \text{by (10)} \end{aligned}$$

   It remains only to show $\beta_{M',M'} = \beta_{M,M'}$:

$$\begin{aligned} B[M/a] &\sim B'[M'/a] \downarrow \beta_{M,M'} \in \mu_{\text{Types}}(\nu) && \text{by (6)} \\ B'[M'/a] &\sim B[M'/a] \downarrow \beta_{M',M'} \in \mu_{\text{Types}}(\nu) && \text{by (8)} \\ \beta_{M',M'} &= \beta_{M,M'} && \text{by (9)} \end{aligned}$$

The cases for Pair and Eq are very similar. PER-valuation follows from Lemma 2.4 for Nat, trivially for Bool, and from PER-valuation of ν for Prop and Type. Symmetry, left transitivity, and right transitivity for Nat, Bool, Prop, and Type are immediate. □

Lemma 2.7. *If τ is a type system, then $\mu_{\text{Props}}(\tau) := \mu\pi.\text{Props}(\pi, \tau)$ is a type system.*

Proof. The proof follows those of Lemmas 2.5 and 2.6. The Pair case is identical. In the Eq case, the index type is drawn from τ and not π, so we have no inductive hypotheses and instead observe that the desired properties follow from those of τ. In the Fun case, the domain is drawn from τ and the codomain type is drawn from π, so we only have inductive hypotheses arising from $a : \alpha \triangleright B \sim B' \downarrow \beta \in \pi$ and $a : \alpha \triangleright N \sim N' \in \beta$. □

Theorem 2.8. *For all $i \in \{0, 1, \dots, \omega\}$, ν_i, τ_i, and π_i are type systems.*

Proof. By strong induction on i. In the base case, ν_0 is empty and thus a type system; $\tau_0 = \mu_{\text{Types}}(\nu_0)$ is a type system by Lemma 2.6; and $\pi_0 = \mu_{\text{Props}}(\tau_0)$ is a type system by Lemma 2.7. In the inductive case, suppose that ν_i, τ_i, π_i are type systems for all $i < n$. Then ν_n is a type system: unicity, symmetry, and transitivity are immediate, and PER-valuation follows from symmetry and transitivity of τ_i and π_i for $i < n$. Once again, τ_n and π_n are type systems by Lemmas 2.6 and 2.7. (The ω case is identical.) □

Finally, to prove our universes are cumulative ($\mathsf{Type}_i \subseteq \mathsf{Type}_j$ and $\mathsf{Prop}_i \subseteq \mathsf{Prop}_j$) we must observe that the sequences of type systems $\{\tau_i\}_i$, $\{\pi_i\}_i$ are increasing. To prove $\mathsf{Prop}_i \subseteq \mathsf{Type}_i$ we must observe $\pi_i \subseteq \tau_i$, which will require a lesser-known theorem about simultaneous fixed points [Bek84].

Lemma 2.9. *In any complete lattice, if $F(x)$ and $G(x)$ are monotone and $F(x) \subseteq G(x)$ for all x, then $\mu x.F(x) \subseteq \mu x.G(x)$.*

Proof. $\mu x.G(x)$ is a pre-fixed point of F because $F(\mu x.G(x)) \subseteq G(\mu x.G(x)) = \mu x.G(x)$. But $\mu x.F(x)$ is the least such, so $\mu x.F(x) \subseteq \mu x.G(x)$. □

Lemma 2.10. *In any complete lattice, if $F(x, y)$ and $G(x, y)$ are monotone and $F(x, y) \subseteq G(x, y)$ whenever $x \subseteq y$, then $\mu_F \subseteq \mu_G$ where $(\mu_F, \mu_G) := \mu(x, y).(F(x, y), G(x, y))$.*

Proof. Let $\mu_\cap = \mu_F \cap \mu_G$. $(\mu_\cap, \mu_G)$ is a pre-fixed point of $(x, y) \mapsto (F(x, y), G(x, y))$ because, by assumption and (μ_F, μ_G) being a fixed point, $F(\mu_\cap, \mu_G) \subseteq F(\mu_F, \mu_G) = \mu_F$ and $F(\mu_\cap, \mu_G) \subseteq G(\mu_\cap, \mu_G) \subseteq G(\mu_F, \mu_G) = \mu_G$. This implies $(\mu_F, \mu_G) \subseteq (\mu_\cap, \mu_G)$ and thus $\mu_F \subseteq \mu_G$. □

Theorem 2.11. *For $i, j \in \{0, 1, \dots, \omega\}$ where $i \leq j$, we have $\tau_i \subseteq \tau_j$, $\pi_i \subseteq \pi_j$, and $\pi_i \subseteq \tau_i$.*

Proof. The functions $\textsc{Types}$ and $\textsc{Props}$ are monotone in both arguments, so by Lemma 2.9, $\mu_{\textsc{Types}}$ and $\mu_{\textsc{Props}}$ are also monotone. When $i \leq j$ we have $\nu_i \subseteq \nu_j$ by construction, so $\tau_i = \mu_{\textsc{Types}}(\nu_i) \subseteq \mu_{\textsc{Types}}(\nu_j) = \tau_j$ and $\pi_i = \mu_{\textsc{Props}}(\tau_i) \subseteq \mu_{\textsc{Props}}(\tau_j) = \pi_j$.

By a theorem of Bekić [Bek84, p. 39] on simultaneous fixed points, for all ν,

$$\mu(\pi, \tau).(\textsc{Props}(\pi, \tau), \textsc{Types}(\nu, \tau)) = (\mu_{\textsc{Props}}(\mu_{\textsc{Types}}(\nu)), \mu_{\textsc{Types}}(\nu))$$

When $\pi \subseteq \tau$, $\textsc{Props}(\pi, \tau) \subseteq \textsc{Types}(\nu, \tau)$; thus $\pi_i \subseteq \tau_i$ follows by Lemma 2.10. □

2.3 *Closed and open judgments*

Unlike the candidate judgments of Section 2.2, the judgments of Idealized Nuprl contain only terms (not also relations α, β) and define typehood and membership under arbitrarily many hypotheses (not only zero or one). We first define the *closed judgments* of type and element equality under no hypotheses.

Definition 2.12 (Closed judgments)**.** Given a type system τ:

1. $A \doteq B$ type when $A \sim B \downarrow \alpha \in \tau$.[4]

2. $M \doteq N \in A$, presupposing $A \doteq A$ type, when $A \sim A \downarrow \alpha \in \tau$ and $M \sim N \in \alpha$.

[4] We adopt the $\doteq$ notation because homotopy type theorists commonly use $=$ for homotopy, not equality.

A *presupposition* is an assumption required to make sense of a judgment. The statement "M and N are equal elements of A" is only well-formed if A is a type (and hence equipped with a notion of equal element). Whenever we assert $M \doteq N \in A$, we implicitly assert A is a type; whenever A is a type, we will write $[\![A]\!]$ for its PER of elements (α in Definition 2.12).

The closed judgments are symmetric and transitive by the corresponding properties of type systems. As a corollary, $A \doteq A$ type whenever $A \doteq B$ type; by unicity of type systems, both judgments give rise to the same $[\![A]\!]$. We henceforth abbreviate $A \doteq A$ type by A type, and $M \doteq M \in A$ by $M \in A$.

Lemma 2.13. *If $A \doteq B$* type *and $M \doteq N \in A$ then $M \doteq N \in B$.*

Proof. By unicity of type systems, $[\![A]\!] = [\![B]\!]$; $M \sim N \in [\![A]\!]$ implies $M \sim N \in [\![B]\!]$. □

Definition 2.12 is parametrized by a choice of type system serving as an *ambient universe* of types. The type system τ_ω constructed in Section 2.2 is the ambient universe of Idealized Nuprl, but we must also consider τ_i, π_i to prove properties of its internal universes Type_i, Prop_i. We write $\tau \models \mathcal{J}$ for any judgment $\mathcal{J}$ to make the choice of τ explicit; when unspecified, judgments are relative to τ_ω.

Lemma 2.14. *If $A \doteq B \in \mathsf{Prop}_i$ or $A \doteq B \in \mathsf{Type}_i$ then $A \doteq B$* type.

Proof. Suppose $\tau_\omega \models (A \doteq B \in \mathsf{Prop}_i)$; then $A \sim B \downarrow \alpha \in \pi_i$. By Theorem 2.11, $\pi_i \subseteq \tau_\omega$, so $A \sim B \downarrow \alpha \in \tau_\omega$, and thus $A \doteq B$ type. Similarly, when $\tau_\omega \models (A \doteq B \in \mathsf{Type}_i)$, we have $A \sim B \downarrow \alpha \in \tau_i$ and $\tau_i \subseteq \tau_\omega$. □

Types whose elements are types (as in Lemma 2.14) are known as *universes à la Russell*, in contrast to *universes à la Tarski*, whose elements are the names of types ($\widehat{\mathsf{nat}} \in \mathsf{Type}_i$ and $\mathsf{El}(\widehat{\mathsf{nat}}) \doteq \mathsf{nat}$ type) [ML84]. The distinction is blurred in computational semantics, because elements of universes *are* the names of types—$[\![\mathsf{Type}_i]\!]$ contains nat, not $[\![\mathsf{nat}]\!]$—but the typehood judgment also concerns the names of types!

The signature feature of Nuprl is that its judgments range over untyped terms, allowing users to establish typehood and membership judgments by untyped evaluation—for instance, A type by $A \Downarrow A_0$ and A_0 type. In contrast, the judgments of most type theories range over structurally well-typed terms, but are closed under evaluation—if A type (hence $A \doteq A$ type) and $A \Downarrow A_0$, then $A \doteq A_0$ type.

Lemma 2.15 (Head expansion)**.**

1. *If A'* type *and $A \longmapsto^* A'$, then $A \doteq A'$* type.
2. *If $M' \in A$ and $M \longmapsto^* M'$, then $M \doteq M' \in A$.*

Proof. For part (1), A' type implies $A' \Downarrow A_0$ and $\tau_\omega(A_0, A_0, \alpha)$. By determinacy of evaluation, $A \Downarrow A_0$ as well, and therefore $A \doteq A'$ type by definition. Part (2) follows similarly. □

Lemma 2.16 (Evaluation)**.** *If $M \in A$ then $M \doteq M_0 \in A$ where $M \Downarrow M_0$.*

Proof. By definition, $M \Downarrow M_0$ and $[\![A]\!](M_0, M_0)$, and therefore $M \doteq M_0 \in A$. □

Open judgments The open judgments of type and element equality under hypotheses express that an equality holds as a closed judgment for all instantiations by elements (proofs) of the hypotheses. We begin by defining lists of hypotheses and their instantiations.

Definition 2.17. A *telescope* is either

1. nil, written $\cdot$, or

2. cons$(A, a.\Gamma)$, written $(a : A, \Gamma)$, for a term A and telescope Γ.

Definition 2.18. For γ, γ' lists and Γ a telescope, $\gamma \sim \gamma' \in \Gamma$ when

1. $\cdot \sim \cdot \in \cdot$, or

2. $(M, \gamma) \sim (M', \gamma') \in (a : A, \Gamma)$ when $M \doteq M' \in A$ and $\gamma \sim \gamma' \in \Gamma[M/a]$.

A telescope is a cons list whose tail has a bound variable, allowing later hypotheses to be parametrized over earlier ones. Telescopes are abstract binding trees and inherit notions of α-equivalence and substitution $\Gamma[M/a]$. The relation $\gamma \sim \gamma' \in \Gamma$, which Crary [Cra98, p. 56] calls *assignment similarity*, expresses that γ, γ' are pointwise equal in the types of Γ; however, because $\Gamma[M/a]$ and $\Gamma[M'/a]$ need not have equal elements when $M \doteq M' \in A$, assignment similarity is neither symmetric nor transitive! Following Martin-Löf [ML82], we restrict attention to *contexts* $(a : A, \Gamma)$ in which the telescope $\Gamma(a)$ respects equality in A. (Allen [All87] has a rather different definition of context, as we discuss in Section 2.6.)

Definition 2.19 (Contexts)**.** For Γ, Γ' telescopes, $\Gamma \doteq \Gamma'$ ctx when

1. $\cdot \doteq \cdot$ ctx, or

2. $(a : A, \Gamma) \doteq (a : A', \Gamma')$ ctx when $A \doteq A'$ type and for all $M \doteq M' \in A$, $\Gamma[M/a] \doteq \Gamma'[M'/a]$ ctx.

Thus, as in the categorical semantics of type theory, a context Γ is simply a nested dependent pair type written as a telescope, an assignment $\gamma \sim \gamma \in \Gamma$ is an element of that type written as a list, and an open judgment $\Gamma \gg \mathcal{J}$ is a dependent function $(\gamma{:}\Gamma) \to \mathcal{J}(\gamma)$.

Definition 2.20 (Open judgments)**.**

1. $\Gamma \gg A \doteq A'$ type, presupposing $\Gamma \doteq \Gamma$ ctx, when for all $\gamma \sim \gamma' \in \Gamma$, $A\gamma \doteq A'\gamma'$ type.

2. $\Gamma \gg M \doteq M' \in A$, presupposing $\Gamma \gg A \doteq A$ type, when for all $\gamma \sim \gamma' \in \Gamma$, $M\gamma \doteq M'\gamma' \in A\gamma$.

In Definition 2.20, A, A', M, and M' extend the telescope Γ on the right (that is, they are under all the binders of Γ), and the operation $M\gamma$ simultaneously substitutes the terms of γ for the corresponding variables of Γ in M. These judgments are symmetric and transitive despite omitting the crossed instantiations $M\gamma'$ and $M'\gamma$, because assignment similarity is symmetric and transitive for contexts.

As with the closed judgments, we will henceforth abbreviate $\Gamma \doteq \Gamma$ ctx by Γ ctx, $\Gamma \gg A \doteq A$ type by $\Gamma \gg A$ type, and $\Gamma \gg M \doteq M \in A$ by $\Gamma \gg M \in A$. We adopt the $\Gamma \gg \mathcal{J}$ notation used in the Nuprl book [Con+85] over the more standard $\Gamma \vdash \mathcal{J}$ to emphasize that the open judgments of (Idealized) Nuprl express *semantic consequence*, not syntactic consequence as in most type theories. That is, in Idealized Nuprl, if a type in Γ has no elements, then *any* judgment whatsoever holds because the antecedent is vacuous (even, say, $\Gamma \gg \mathsf{z} \in \mathsf{z}$), whereas in Coq one can only derive elements of *structurally well-typed* types from a contradiction.

Most authors define contexts as snoc lists to match the rules of type theory, which add hypotheses on the right. In such presentations, the open judgments must be defined mutually—for instance, $(\Gamma, a : A)$ ctx when $\Gamma \gg A$ type, which holds when Γ ctx *et cetera*. In exchange for avoiding this difficulty, we must define context extension $(\Gamma, a : A)$ as a derived operation (which concatenates the telescopes Γ and $\mathsf{cons}(A, a.\mathsf{nil})$ by recursion on Γ), and must prove its usual property:

Lemma 2.21 (Context extension). *If* $\Gamma \doteq \Gamma'$ ctx *and* $\Gamma \gg A \doteq A'$ type *then* $(\Gamma, a : A) \doteq (\Gamma', a : A')$ ctx.

Proof. By induction on $\Gamma \doteq \Gamma'$ ctx. If $\Gamma = \Gamma' = \cdot$, we must show that if $A \doteq A'$ type then $(a : A) \doteq (a : A')$ ctx, which is immediate. Now suppose $(b : B, \Gamma) \doteq (b : B', \Gamma')$ ctx and $b : B, \Gamma \gg A \doteq A'$ type, and show $((b : B, \Gamma), a : A) \doteq ((b : B', \Gamma'), a : A')$ ctx. By definition, $((b : B, \Gamma), a : A) = (b : B, (\Gamma, a : A))$. We know $B \doteq B'$ type, so it suffices to show for any $N \doteq N' \in B$ that $(\Gamma[N/b], a : A[N/b]) \doteq (\Gamma'[N'/b], a : A'[N'/b])$ ctx. The result follows by the inductive hypothesis, $\Gamma[N/b] \doteq \Gamma'[N'/b]$ ctx, and $\Gamma[N/b] \gg A[N/b] \doteq A'[N'/b]$ type (by unrolling the definition of $b : B, \Gamma \gg A \doteq A'$ type). □

Judgments satisfy the *structural rules* of hypothesis, weakening, and substitution.

Lemma 2.22 (Hypothesis). *If* $(\Gamma, a : A, \Delta)$ ctx *then* $\Gamma, a : A, \Delta \gg a \in A$.

Proof. We establish the presupposition $\Gamma, a : A, \Delta \gg A$ type by induction on Γ, unrolling $(\Gamma, a : A, \Delta)$ ctx to obtain $A\gamma \doteq A\gamma'$ type for all $\gamma \sim \gamma' \in \Gamma$, and observing that a and the variables of Δ cannot occur in A. Now suppose $(\gamma, M, \delta) \sim (\gamma', M', \delta') \in (\Gamma, a : A, \Delta)$ (hence $\gamma \sim \gamma' \in \Gamma$ and $M \doteq M' \in A\gamma$) and show $M \doteq M' \in A\gamma$, which is immediate. □

Lemma 2.23 (Weakening)**.** *Supposing* $\Gamma \gg A$ type*:*

1. *If* $\Gamma, \Delta \gg B \doteq B'$ type *then* $\Gamma, a : A, \Delta \gg B \doteq B'$ type*.*
2. *If* $\Gamma, \Delta \gg N \doteq N' \in B$ *then* $\Gamma, a : A, \Delta \gg N \doteq N' \in B$*.*

Proof. Suppose $\gamma \sim \gamma' \in \Gamma$. We establish the presupposition $(\Gamma, a : A, \Delta)$ ctx by induction on Γ, obtaining $A\gamma \doteq A\gamma'$ type from $\Gamma \gg A$ type and $\Delta\gamma \doteq \Delta\gamma'$ ctx from (Γ, Δ) ctx. (The latter suffices because a cannot occur in Δ.) For part (1), suppose $(\gamma, M, \delta) \sim (\gamma', M', \delta') \in (\Gamma, a : A, \Delta)$ and show $B\gamma[M/a]\delta \doteq B'\gamma'[M'/a]\delta'$ type. Because a cannot occur in B, this is exactly $B\gamma\delta \doteq B'\gamma'\delta'$ type, which holds by assumption. Part (2) follows similarly. □

Lemma 2.24 (Substitution)**.** *Supposing* $\Gamma \gg M \doteq M' \in A$*:*

1. *If* $\Gamma, a : A, \Delta \gg B \doteq B'$ type *then* $\Gamma, \Delta[M/a] \gg B[M/a] \doteq B'[M'/a]$ type*.*
2. *If* $\Gamma, a : A, \Delta \gg N \doteq N' \in B$ *then* $\Gamma, \Delta[M/a] \gg N[M/a] \doteq N'[M'/a] \in B[M/a]$*.*

Proof. As in the previous lemmas, the presupposition $(\Gamma, \Delta[M/a])$ ctx is established by induction on Γ, noting that for any $\gamma \sim \gamma' \in \Gamma$, we have $M\gamma \in A\gamma$ and $\Delta[M/a]\gamma = \Delta\gamma[M\gamma/a]$. For part (1), suppose $(\gamma, \delta) \sim (\gamma', \delta') \in (\Gamma, \Delta[M/a])$, where $\gamma \sim \gamma' \in \Gamma$ and $\delta \sim \delta' \in \Delta[M/a]\gamma$. Then $(\gamma, M\gamma, \delta) \sim (\gamma', M'\gamma', \delta') \in (\Gamma, a : A, \Delta)$ by $M\gamma \doteq M'\gamma' \in A\gamma$ and $\Delta[M/a]\gamma = \Delta\gamma[M\gamma/a]$, so by hypothesis we have $B\gamma[M\gamma/a]\delta \doteq B'\gamma'[M'\gamma'/a]\delta'$ type and hence $B[M/a]\gamma\delta \doteq B'[M'/a]\gamma'\delta'$ type as required. Part (2) follows similarly. □

Surprisingly, Lemma 2.24 does *not* hold in ordinary Nuprl—in Allen's nonstandard semantics of open judgments, the cut principle holds only if a does not occur in Δ—but it does hold in Idealized Nuprl and Chapter 4 of this dissertation.

It remains to show that τ_ω admits the formation, introduction, elimination, and computation principles of dependent function types, universe types, *et cetera.* We will prove only the closed, binary forms of these principles, as the open forms follow directly by commuting substitutions past constructors. By way of illustration, consider the open form of Lemma 2.13:

Lemma 2.25. *If* $\Gamma \gg A \doteq B$ type *and* $\Gamma \gg M \doteq N \in A$ *then* $\Gamma \gg M \doteq N \in B$*.*

Proof. Suppose $\gamma \sim \gamma' \in \Gamma$ and show $M\gamma \doteq N\gamma' \in B\gamma$. By our second assumption, $M\gamma \doteq N\gamma' \in A\gamma$. By our first assumption and $\gamma \sim \gamma \in \Gamma$ (by Γ ctx), $A\gamma \doteq B\gamma$ type. The result follows by Lemma 2.13. □

2.4 Rules of inference

Sections 2.2 and 2.3 define Idealized Nuprl as a mathematical object, but these definitions alone are neither useful in practice (how does one establish a judgment?) nor obviously correct (is $(a{:}A) \to B$ a dependent function type in the usual sense?). We address both concerns by proving that Idealized Nuprl admits standard type-theoretic rules of inference, listed in Figure 2.4. Each rule is annotated with the lemma containing its proof: structural rules were proven in Section 2.3, and rules governing type formers are proven in Section 2.5. We suppress ambient hypotheses in these rules for clarity; every judgment implicitly contains additional hypotheses Γ.

These natural deduction–style rules characterize type formers by rules of *formation*, establishing when an instance is a type; *introduction*, establishing how to construct an element of that type; and *elimination*, establishing how to use an element of that type [Pra65]. Some type formers also have a *uniqueness*, or η, rule establishing that all elements are equal to introduction forms.[5] In standard natural deduction, type formers have *computation*, or β, rules establishing how elimination acts on introduction:

$$\frac{a : A \gg M \in B \qquad N \in A}{(\lambda a.M)\ N \doteq M[N/a] \in B[N/a]}$$

Computation rules are true but superfluous in Idealized Nuprl, because they are instances of head expansion (Lemma 2.15).

Choosing rules As the computation rules have already illustrated, there are many possible collections of inference rules for any type theory. Their relative merits are a topic of frequent debate amongst type theorists, and while we do not enter the fray in this dissertation, we will nevertheless sketch various considerations. First, the very meaning of a type theory *qua* algebraic structure (Section 1.2), by definition, depends on its collection of rules; the biggest open problem in homotopy type theory is to characterize all models of Book HoTT and various cubical type theories.

Secondly, proof assistants invariably implement type theories as collections of primitive inference steps (Section 1.1). These steps must be machine-checkable and, unlike Figure 2.4, cannot rely on notational conventions or semantic presuppositions. For instance, if a reader claimed to find an inconsistency in Figure 2.4 by shadowing variable names, we would admonish them to recall standard notational conventions; a proof assistant, on the other hand, must actually implement variable binding.

Unlike variable freshness, which is fully automated, our judgments' presuppositions can induce subgoals requiring user input. For instance, to maintain the invariant that

[5] Many type theories have few or no uniqueness rules. In Idealized Nuprl, every type has a uniqueness principle; those of nat and bool can be derived from their elimination rules.

$$\frac{A \in \mathsf{Type}_i}{a : A \gg a \in A}\ (2.22) \qquad \frac{M \doteq M' \in B \qquad A \in \mathsf{Type}_i}{a : A \gg M \doteq M' \in B}\ (2.23)$$

$$\frac{M \doteq M' \in A \qquad A \doteq A' \in \mathsf{Type}_i}{M \doteq M' \in A'}\ (2.13) \qquad \frac{a : A \gg M \doteq M' \in B \qquad N \doteq N' \in A}{M[N/a] \doteq M'[N'/a] \in B[N/a]}\ (2.24)$$

$$\frac{M' \in A \qquad M \longmapsto^{*} M'}{M \doteq M' \in A}\ (2.15) \qquad \frac{M \in A \qquad M \Downarrow M_0}{M \doteq M_0 \in A}\ (2.16)$$

$$\frac{A \in \mathsf{Type}_i \qquad a : A \gg B \in \mathsf{Type}_i}{(a{:}A) \longrightarrow B \in \mathsf{Type}_i}\ (2.26) \qquad \frac{A \in \mathsf{Type}_i \qquad a : A \gg B \in \mathsf{Prop}_i}{(a{:}A) \longrightarrow B \in \mathsf{Prop}_i}\ (2.26)$$

$$\frac{a : A \gg M \in B}{\lambda a.M \in (a{:}A) \longrightarrow B}\ (2.27) \qquad \frac{M \in (a{:}A) \longrightarrow B \qquad N \in A}{M\ N \in B[N/a]}\ (2.28)$$

$$\frac{M \in (a{:}A) \longrightarrow B}{M \doteq \lambda a.M\ a \in (a{:}A) \longrightarrow B}\ (2.30)$$

$$\frac{A \in \mathsf{Type}_i \qquad a : A \gg B \in \mathsf{Type}_i}{(a{:}A) \times B \in \mathsf{Type}_i}\ (2.31) \qquad \frac{A \in \mathsf{Prop}_i \qquad a : A \gg B \in \mathsf{Prop}_i}{(a{:}A) \times B \in \mathsf{Prop}_i}\ (2.31)$$

$$\frac{a : A \gg B \in \mathsf{Type}_i \qquad M \in A \qquad N \in B[M/a]}{\langle M, N \rangle \in (a{:}A) \times B}\ (2.32) \qquad \frac{P \in (a{:}A) \times B}{\mathsf{fst}(P) \in A}\ (2.33)$$

$$\frac{P \in (a{:}A) \times B}{\mathsf{snd}(P) \in B[\mathsf{fst}(P)/a]}\ (2.33) \qquad \frac{P \in (a{:}A) \times B}{P \doteq \langle \mathsf{fst}(P), \mathsf{snd}(P) \rangle \in (a{:}A) \times B}\ (2.34)$$

Figure 2.4: Idealized Nuprl: structural rules, functions, pairs.

$$\frac{A \in \mathsf{Type}_i \qquad M \in A \qquad N \in A}{\mathsf{Eq}_A(M, N) \in \mathsf{Prop}_i}\ (2.35) \qquad\qquad \frac{M \doteq N \in A}{\star \in \mathsf{Eq}_A(M, N)}\ (2.36)$$

$$\frac{E \in \mathsf{Eq}_A(M, N)}{M \doteq N \in A}\ (2.37) \qquad\qquad \frac{E \in \mathsf{Eq}_A(M, N)}{E \doteq \star \in \mathsf{Eq}_A(M, N)}\ (2.38)$$

$$\frac{}{\mathsf{nat} \in \mathsf{Type}_i}\ (2.39) \qquad\qquad \frac{}{\mathsf{z} \in \mathsf{nat}}\ (2.40) \qquad\qquad \frac{M \in \mathsf{nat}}{\mathsf{s}(M) \in \mathsf{nat}}\ (2.40)$$

$$\frac{\begin{array}{c} n : \mathsf{nat} \gg A \in \mathsf{Type}_i \\ M \in \mathsf{nat} \qquad Z \in A[\mathsf{z}/n] \qquad n : \mathsf{nat}, a : A \gg S \in A[\mathsf{s}(n)/n] \end{array}}{\mathsf{natrec}(M; Z, n.a.S) \in A[M/n]}\ (2.41)$$

$$\frac{}{\mathsf{bool} \in \mathsf{Type}_i}\ (2.43) \qquad\qquad \frac{}{\mathsf{true} \in \mathsf{bool}}\ (2.44) \qquad\qquad \frac{}{\mathsf{false} \in \mathsf{bool}}\ (2.44)$$

$$\frac{b : \mathsf{bool} \gg A \in \mathsf{Type}_i \qquad M \in \mathsf{bool} \qquad T \in A[\mathsf{true}/b] \qquad F \in A[\mathsf{false}/b]}{\mathsf{if}(M; T, F) \in A[M/b]}\ (2.45)$$

$$\frac{}{\mathsf{Type}_i \in \mathsf{Type}_{i+1}}\ (2.48) \qquad\qquad \frac{}{\mathsf{Prop}_i \in \mathsf{Type}_{i+1}}\ (2.48) \qquad\qquad \frac{A \in \mathsf{Prop}_i}{A \in \mathsf{Type}_i}\ (2.49)$$

$$\frac{A \in \mathsf{Type}_i}{A \in \mathsf{Type}_{i+1}}\ (2.50) \qquad \frac{A \in \mathsf{Prop}_i}{A \in \mathsf{Prop}_{i+1}}\ (2.50) \qquad \frac{M \in A \qquad N \in A \qquad A \in \mathsf{Prop}_i}{M \doteq N \in A}\ (2.51)$$

Figure 2.4: Idealized Nuprl: equality, natural numbers, Booleans, universes.

Γ ctx holds whenever $\Gamma \gg B$ type is provable in our proof assistant, we might demand that users prove new hypotheses are types before extending contexts. Typehood cannot be established automatically, as it is undecidable (if $\mathsf{Eq}_A(M, M)$ type then M halts) and commonly intractable, often requiring proofs that two functions are extensionally equal.

Computational type theories—implemented in the Nuprl [Con+85] and **RedPRL** [Red16] proof assistants—alleviate the burden by allowing users to employ computation in their proofs. Computational type theories are defined by two characteristics: having a computational semantics, and using inference rules that irrevocably commit to that semantics. The latter is evident in the head expansion and evaluation rules of Figure 2.4, which expose the operational semantics, but crucially, allow users to prove theorems by first computing them. On top of that, Nuprl's developers frequently extend Nuprl with useful rules subject only to validation by their computational semantics.

Figure 2.4 differs from the rules used by Nuprl and **RedPRL** in a few important ways. Both proof assistants adopt sequent calculi, not natural deduction rules, in which elimination rules are replaced by *left rules* explaining how to use hypotheses of each type. In practice, one needs head expansion for open terms, despite our operational semantics being defined only for closed terms. Fortunately, $\longmapsto$ is stable under substitution when lifted to open terms—that is, if a rule of $\longmapsto$ applies to open terms M and M', then $M\gamma \longmapsto M'\gamma$ for all substitutions γ—allowing us to reduce subgoals like $\Gamma \gg \mathsf{fst}(\langle a, b\rangle) \in A$ to $\Gamma \gg a \in A$. (We will not be as fortunate in Chapter 4.)

Intensional type theories—implemented in the Agda [Agda] and Coq [Coq] proof assistants—axiomatize and automate a decidable underapproximation of the judgments, reducing users' proof burden while restricting what is provable. These type theories notably lack an *equality reflection* rule converting proofs of $\mathsf{Eq}_A(M, N)$ to $M \doteq N \in A$; instead, silent equations are restricted to those provable by computation, whereas any appeals to non-trivial equations (those requiring inductive proofs) must be annotated.

Those silent equations, known as *definitional equalities* [ML75a], do not constitute a mathematically natural notion of equality, unlike the equality judgment considered in this dissertation—many types lack definitional uniqueness principles, and therefore lack strict universal properties in the categorical sense.[6] Definitional equality is also brittle, as extensions require significant effort both devising algorithms and proving them correct. However, the benefit of being fully automated cannot be understated.

Computational type theory is often considered odd for having undecidable judgments. On the contrary, it is perfectly natural for a judgment to be sensible without being *obviously* sensible; in ordinary mathematical practice, a theorem's statement may require explanation even before its proof. Definitional equality is decidable not because decidability is most natural, but because it allows proof assistants to fully automate all appeals to computation.

[6] Nor, in the author's opinion, does definitional equality constitute a philosophically natural notion of *intension*, as the name of intensional type theory might suggest.

2.5 Semantics of types

We now prove the formation, introduction, elimination, and uniqueness rules that govern each type former. We will also establish that Idealized Nuprl is consistent (Theorem 2.47), and that it enjoys the canonicity (Theorem 2.46) and existence properties (Theorem 2.42) characteristic of constructive type theories.

Recall that the judgments of Idealized Nuprl are defined relative to the type system τ_ω constructed in Section 2.2. Formation and introduction rules for each type hold essentially by construction, after expanding judgments into the candidate judgments of Figure 2.3; elimination rules hold by observing that elimination forms compute on introduction forms; and uniqueness rules hold by induction (all elements are equal to introduction forms) coupled with introduction rules (to see that two introduction forms are equal).

2.5.1 Dependent functions

Recall from Figure 2.3 that for $i \in \{0, 1, \dots, \omega\}$ and $R \in \{\tau_i, \pi_i\}$,

$$R((a{:}A) \to B, (a{:}A') \to B', \{(\lambda a.N, \lambda a.N') \mid a : \alpha \triangleright N \sim N' \in \beta\})$$

if and only if $A \sim A' \downarrow \alpha \in \tau_i$ and $a : \alpha \triangleright B \sim B' \downarrow \beta \in R$.

Rule 2.26 (Formation).

1. *If* $A \doteq A' \in \mathsf{Type}_i$ *and* $a : A \gg B \doteq B' \in \mathsf{Type}_i$ *then* $(a{:}A) \to B \doteq (a{:}A') \to B' \in \mathsf{Type}_i$.
2. *If* $A \doteq A' \in \mathsf{Type}_i$ *and* $a : A \gg B \doteq B' \in \mathsf{Prop}_i$ *then* $(a{:}A) \to B \doteq (a{:}A') \to B' \in \mathsf{Prop}_i$.
3. *If* $A \doteq A'$ type *and* $a : A \gg B \doteq B'$ type *then* $(a{:}A) \to B \doteq (a{:}A') \to B'$ type.

Proof. For part (1), $A \doteq A' \in \mathsf{Type}_i$ implies $A \sim A' \downarrow \alpha \in \tau_i$ (by the definition of $[\![\mathsf{Type}_i]\!]$), $A \doteq A'$ type (by Lemma 2.14), and $\alpha = [\![A]\!]$ (by Theorem 2.11). Furthermore, $a : A \gg B \doteq B' \in \mathsf{Type}_i$ states that for any $M \doteq M' \in A$ (equivalently, $M \sim M' \in \alpha$), $B[M/a] \doteq B'[M'/a] \in \mathsf{Type}_i$, and thus $a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau_i$. We conclude that $(a{:}A) \to B \sim (a{:}A') \to B' \downarrow [\![(a{:}A) \to B]\!] \in \tau_i$, and therefore $(a{:}A) \to B \doteq (a{:}A') \to B' \in \mathsf{Type}_i$, where $[\![(a{:}A) \to B]\!](\lambda a.N, \lambda a.N')$ holds if and only if $a : \alpha \triangleright N \sim N' \in \beta$. Parts (2) and (3) are similar, appealing to the definitions of π_i and τ_ω, respectively, instead of τ_i. □

Rule 2.27 (Introduction). *If* $a : A \gg M \doteq M' \in B$ *then* $\lambda a.M \doteq \lambda a.M' \in (a{:}A) \to B$.

Proof. The presupposition $(a{:}A) \to B$ type holds by the formation rule applied to the premise's presupposition $a : A \gg B$ type. We know for any $N \doteq N' \in A$ that $M[N/a] \doteq M'[N'/a] \in B[N/a]$; thus $a : [\![A]\!] \triangleright M \sim M' \in \beta$ where $a : [\![A]\!] \triangleright B \sim B \downarrow \beta \in \tau_\omega$ (because unicity implies $\beta_{N,N'} = [\![B[N/a]]\!]$), and hence $[\![(a{:}A) \to B]\!](\lambda a.M, \lambda a.M')$. □

Observe that $a{:}A \gg B \doteq B'$ type and $a{:}A \gg M \doteq M' \in B$ are equivalent to $a{:}[\![A]\!] \triangleright B \sim B' \downarrow \beta \in \tau_\omega$ and $a{:}[\![A]\!] \triangleright M \sim M' \in \beta$ respectively. We henceforth freely use this equivalence when unfolding definitions of type formers.

Rule 2.28 (Elimination). *If $M \doteq M' \in (a{:}A) \to B$ and $N \doteq N' \in A$ then $M\ N \doteq M'\ N' \in B[N/a]$.*

Proof. By the definition of $[\![(a{:}A) \to B]\!]$, we know $M \Downarrow \lambda a.O$, $M \Downarrow \lambda a.O'$, and $a : A \gg O \doteq O' \in B$. By the latter, $O[N/a] \doteq O'[N'/a] \in B[N/a]$. By Lemma 2.15 and $M\ N \longmapsto^* (\lambda a.O)\ N \longmapsto O[N/a]$, we know $M\ N \doteq O[N/a] \in B[N/a]$, and analogously, $M'\ N' \doteq O'[N'/a] \in B[N/a]$. The result follows by symmetry and transitivity of equality. □

As discussed in Section 2.4, we omit computation rules from Figure 2.4. We will prove Rule 2.29 for illustrative purposes, and leave further computation rules as an exercise.

Rule 2.29 (Computation). *If $a{:}A \gg M \in B$ and $N \in A$ then $(\lambda a.M)\ N \doteq M[N/a] \in B[N/a]$.*

Proof. Our hypotheses imply $M[N/a] \in B[N/a]$. The result follows by Lemma 2.15 and $(\lambda a.M)\ N \longmapsto M[N/a]$. □

Rule 2.30 (Uniqueness). *If $M \in (a{:}A) \to B$ then $M \doteq \lambda a.M\ a \in (a{:}A) \to B$.*

Proof. By $M \in (a{:}A) \to B$, we know $M \Downarrow \lambda a.O$ and $a : A \gg O \in B$. By the definition of $[\![(a{:}A) \to B]\!]$, it suffices to show $a : A \gg O \doteq M\ a \in B$. Suppose $N \doteq N' \in A$. Then by $a : A \gg O \in B$ we have $O[N/a] \doteq O[N'/a] \in B[N/a]$; by Lemma 2.15 and $M\ N' \longmapsto^* (\lambda a.O)\ N' \longmapsto O[N'/a]$ we have $O[N'/a] \doteq M\ N' \in B[N/a]$; and thus $O[N/a] \doteq M\ N' \in B[N/a]$ as required. □

2.5.2 *Dependent pairs*

For $i \in \{0, 1, \dots, \omega\}$ and $R \in \{\tau_i, \pi_i\}$,

$$R((a{:}A) \times B, (a{:}A') \times B', \{(\langle M, N\rangle, \langle M', N'\rangle) \mid (M \sim M' \in \alpha) \wedge (N \sim N' \in \beta_{M,M'})\})$$

if and only if $A \sim A' \downarrow \alpha \in R$ and $a : \alpha \triangleright B \sim B' \downarrow \beta \in R$.

Rule 2.31 (Formation).

1. *If $A \doteq A' \in \mathsf{Type}_i$ and $a : A \gg B \doteq B' \in \mathsf{Type}_i$ then $(a{:}A) \times B \doteq (a{:}A') \times B' \in \mathsf{Type}_i$.*
2. *If $A \doteq A' \in \mathsf{Prop}_i$ and $a : A \gg B \doteq B' \in \mathsf{Prop}_i$ then $(a{:}A) \times B \doteq (a{:}A') \times B' \in \mathsf{Prop}_i$.*
3. *If $A \doteq A'$ type and $a : A \gg B \doteq B'$ type then $(a{:}A) \times B \doteq (a{:}A') \times B'$ type.*

Proof. The proof proceeds identically to that of Rule 2.26. Note that the Pair clause of π_i requires $A \sim A' \downarrow \alpha \in \pi_i$ and not $A \sim A' \downarrow \alpha \in \tau_i$ as in the Fun clause. □

Rule 2.32 (Introduction). *If* $a : A \gg B \in \mathsf{Type}_i$, $M \doteq M' \in A$, *and* $N \doteq N' \in B[M/a]$, *then* $\langle M, N\rangle \doteq \langle M', N'\rangle \in (a{:}A) \times B$.

Proof. The first premise and the formation rule ensure the presupposition $(a{:}A) \times B$ type. It remains to show $[\![(a{:}A) \times B]\!](\langle M, N\rangle, \langle M', N'\rangle)$, which follows from $M \sim M' \in [\![A]\!]$ and $N \sim N' \in \beta_{M,M'}$ where $a : [\![A]\!] \triangleright B \sim B \downarrow \beta \in \tau_\omega$ (by $\beta_{M,M'} = [\![B[M/a]]\!]$). □

Rule 2.33 (Elimination). *If* $P \doteq P' \in (a{:}A) \times B$ *then*

1. $\mathsf{fst}(P) \doteq \mathsf{fst}(P') \in A$ *and*
2. $\mathsf{snd}(P) \doteq \mathsf{snd}(P') \in B[\mathsf{fst}(P)/a]$.

Proof. By the definition of $[\![(a{:}A) \times B]\!]$, we know $P \Downarrow \langle M, N\rangle$, $P' \Downarrow \langle M', N'\rangle$, $M \doteq M' \in A$, and $N \doteq N' \in B[M/a]$. For part (1), by Lemma 2.15 and $\mathsf{fst}(P) \longmapsto^* \mathsf{fst}(\langle M, N\rangle) \longmapsto M$, we know $\mathsf{fst}(P) \doteq M \in A$ and analogously $M' \doteq \mathsf{fst}(P') \in A$; the result follows by transitivity.

For part (2), by Lemma 2.15 and $\mathsf{snd}(P) \longmapsto^* \mathsf{snd}(\langle M, N\rangle) \longmapsto N$, we know $\mathsf{snd}(P) \doteq N \in B[M/a]$ and analogously $N' \doteq \mathsf{snd}(P') \in B[M/a]$, hence $\mathsf{snd}(P) \doteq \mathsf{snd}(P') \in B[M/a]$. We have $a : A \gg B$ type by $(a{:}A) \times B$ type and inversion on the definition of τ_ω; thus $B[M/a] \doteq B[\mathsf{fst}(P)/a]$ type, and the result follows by Lemma 2.13. □

Rule 2.34 (Uniqueness). *If* $P \in (a{:}A) \times B$ *then* $P \doteq \langle \mathsf{fst}(P), \mathsf{snd}(P)\rangle \in (a{:}A) \times B$.

Proof. By $P \in (a{:}A) \times B$, we know $P \Downarrow \langle M, N\rangle$, $M \in A$, and $N \in B[M/a]$. By the definition of $[\![(a{:}A) \times B]\!]$, it suffices to show $M \doteq \mathsf{fst}(P) \in A$ and $N \doteq \mathsf{snd}(P) \in B[M/a]$; both are immediate by Lemma 2.15. □

2.5.3 Equalities

For $i \in \{0, 1, \dots, \omega\}$ and $R \in \{\tau_i, \pi_i\}$,

$$R(\mathsf{Eq}_A(M, N), \mathsf{Eq}_{A'}(M', N'), \{(\star, \star) \mid M \sim N \in \alpha\})$$

if and only if $A \sim A' \downarrow \alpha \in \tau_i$, $M \sim M' \in \alpha$, and $N \sim N' \in \alpha$.

Rule 2.35 (Formation).

1. *If* $A \doteq A' \in \mathsf{Type}_i$, $M \doteq M' \in A$, *and* $N \doteq N' \in A$, *then* $\mathsf{Eq}_A(M, N) \doteq \mathsf{Eq}_{A'}(M', N') \in \mathsf{Type}_i$.
2. *If* $A \doteq A' \in \mathsf{Type}_i$, $M \doteq M' \in A$, *and* $N \doteq N' \in A$, *then* $\mathsf{Eq}_A(M, N) \doteq \mathsf{Eq}_{A'}(M', N') \in \mathsf{Prop}_i$.
3. *If* $A \doteq A'$ type, $M \doteq M' \in A$, *and* $N \doteq N' \in A$, *then* $\mathsf{Eq}_A(M, N) \doteq \mathsf{Eq}_{A'}(M', N')$ type.

Proof. For part (1), $A \doteq A' \in \mathsf{Type}_i$ implies $A \sim A' \downarrow [\![A]\!] \in \tau_i$. From $M \sim M' \in [\![A]\!]$ and $N \sim N' \in [\![A]\!]$ we conclude $\mathsf{Eq}_A(M,N) \sim \mathsf{Eq}_{A'}(M',N') \downarrow [\![\mathsf{Eq}_A(M,N)]\!] \in \tau_i$ and therefore $\mathsf{Eq}_A(M,N) \doteq \mathsf{Eq}_{A'}(M',N') \in \mathsf{Type}_i$, where $[\![\mathsf{Eq}_A(M,N)]\!](\star,\star)$ holds if and only if $M \sim N \in [\![A]\!]$. Parts (2) and (3) are similar, appealing to the definitions of π_i and τ_ω, respectively, instead of τ_i. □

Rule 2.36 (Introduction)**.** *If* $M \doteq N \in A$ *then* $\star \in \mathsf{Eq}_A(M,N)$.

Proof. The premise implies A type, $M \in A$, and $N \in A$, so by the formation rule, the presupposition $\mathsf{Eq}_A(M,N)$ type holds. It remains to show $[\![\mathsf{Eq}_A(M,N)]\!](\star,\star)$, which is immediate by $M \sim N \in [\![A]\!]$. □

Rule 2.37 (Elimination)**.** *If* $E \in \mathsf{Eq}_A(M,N)$ *then* $M \doteq N \in A$.

Proof. By the definition of $[\![\mathsf{Eq}_A(M,N)]\!]$, $E \Downarrow \star$ and $M \doteq N \in A$. □

Rule 2.37 is commonly known as equality reflection. As discussed in Section 2.4, intensional type theories omit such a rule despite it holding in their computational semantics (in the non-univalent case).

Rule 2.38 (Uniqueness)**.** *If* $E \in \mathsf{Eq}_A(M,N)$ *then* $E \doteq \star \in \mathsf{Eq}_A(M,N)$.

Proof. By $E \in \mathsf{Eq}_A(M,N)$ we know $E \Downarrow \star$ and $M \doteq N \in A$, which by the definition of $[\![\mathsf{Eq}_A(M,N)]\!]$ is exactly what we must show. □

2.5.4 Natural numbers

For $i \in \{0,1,\dots,\omega\}$, $\tau_i(\mathsf{nat},\mathsf{nat},\mu R.F(R))$ holds, where

$$F(R) = \{(\mathsf{z},\mathsf{z})\} \cup \{(\mathsf{s}(M),\mathsf{s}(M')) \mid M \sim M' \in R\}.$$

Rule 2.39 (Formation)**.** $\mathsf{nat} \in \mathsf{Type}_i$.

Proof. Immediate by the definition of $[\![\mathsf{Type}_i]\!]$. □

Rule 2.40 (Introduction)**.**

1. $\mathsf{z} \in \mathsf{nat}$.
2. *If* $M \doteq M' \in \mathsf{nat}$ *then* $\mathsf{s}(M) \doteq \mathsf{s}(M') \in \mathsf{nat}$.

Proof. The presupposition nat type holds by the formation rule and Lemma 2.14. Part (1) is immediate by $F([\![\mathsf{nat}]\!])(\mathsf{z},\mathsf{z})$, because $[\![\mathsf{nat}]\!] = F([\![\mathsf{nat}]\!])$. Part (2) is similarly immediate by $F([\![\mathsf{nat}]\!])(\mathsf{s}(M),\mathsf{s}(M'))$ whenever $M \sim M' \in [\![\mathsf{nat}]\!]$. □

Rule 2.41 (Elimination)**.** *If* $n : \mathsf{nat} \gg A \in \mathsf{Type}_i$*,* $M \doteq M' \in \mathsf{nat}$*,* $Z \doteq Z' \in A[\mathsf{z}/n]$*, and* $n{:}\mathsf{nat}, a{:}A \gg S \doteq S' \in A[\mathsf{s}(n)/n]$*, then* $\mathsf{natrec}(M; Z, n.a.S) \doteq \mathsf{natrec}(M'; Z', n.a.S') \in A[M/n]$*.*

Proof. Let $\Phi = \{(N_0, N_0') \mid [\![\mathsf{nat}]\!](N_0, N_0') \wedge \mathsf{natrec}(N_0; Z, n.a.S) \doteq \mathsf{natrec}(N_0'; Z', n.a.S') \in A[N_0/n]\}$. First, show $N \sim N' \in \Phi$ implies $\mathsf{natrec}(N; Z, n.a.S) \doteq \mathsf{natrec}(N'; Z', n.a.S') \in A[N/n]$. Since $\Phi \subseteq [\![\mathsf{nat}]\!]$, we have $N \in \mathsf{nat}$ and thus the presupposition $A[N/n]$ type. By definition, $N \Downarrow N_0$, $N' \Downarrow N_0'$, and $\Phi(N_0, N_0')$. Therefore, by Lemma 2.15, it suffices to show $\mathsf{natrec}(N_0; Z, n.a.S) \doteq \mathsf{natrec}(N_0'; Z', n.a.S') \in A[N/n]$. These terms are equal in the type $A[N_0/n]$ by $\Phi(N_0, N_0')$; the result follows by $A[N_0/n] \doteq A[N/n]$ type (by $N \doteq N_0 \in \mathsf{nat}$) and Lemma 2.13.

Now let us show $F(\Phi) \subseteq \Phi$; because $[\![\mathsf{nat}]\!]$ is the least such relation, it will follow that $[\![\mathsf{nat}]\!] \subseteq \Phi$. Suppose $F(\Phi)(N_0, N_0')$ and show $\Phi(N_0, N_0')$. There are two possibilities. If $N_0 = N_0' = \mathsf{z}$, then $[\![\mathsf{nat}]\!](\mathsf{z}, \mathsf{z})$. By Lemma 2.15 and $\mathsf{natrec}(\mathsf{z}; Z, n.a.S) \longmapsto Z$, it suffices to show $Z \doteq Z' \in A[\mathsf{z}/n]$, which we have assumed. Otherwise, $N_0 = \mathsf{s}(N)$, $N_0' = \mathsf{s}(N')$, and $N \sim N' \in \Phi$. Since $\Phi \subseteq [\![\mathsf{nat}]\!]$ we have $[\![\mathsf{nat}]\!](\mathsf{s}(N), \mathsf{s}(N'))$. By Lemma 2.15 and $\mathsf{natrec}(\mathsf{s}(N); Z, n.a.S) \longmapsto S[N/n][\mathsf{natrec}(M; Z, n.a.S)/a]$, it suffices to show

$$S[N/n][\mathsf{natrec}(N; Z, n.a.S)/a] \doteq S'[N'/n][\mathsf{natrec}(N'; Z', n.a.S')/a] \in A[\mathsf{s}(N)/n]$$

which follows from $N \doteq N' \in \mathsf{nat}$, $\mathsf{natrec}(N; Z, n.a.S) \doteq \mathsf{natrec}(N'; Z, n.a.S) \in A[N/n]$ (by the observation in the previous paragraph), and our assumption on S and S'.

By $M \doteq M' \in \mathsf{nat}$ and $[\![\mathsf{nat}]\!] \subseteq \Phi$, $M \sim M' \in \Phi$, from which the result is immediate. □

Idealized Nuprl, as formulated in Figure 2.4, is constructive in the sense of satisfying the existence property [TD88, p. 139]. We demand not only that our computational semantics have existential witnesses but in fact that our *chosen rules* determine these witnesses—in the absence of the latter, users of Idealized Nuprl would not reap the benefits of the former.

Theorem 2.42 (Existence)**.** *If* $n : \mathsf{nat} \gg B \in \mathsf{Type}_i$ *and* $M \in (n{:}\mathsf{nat}) \times B$*, there exist terms* $\bar{n} = \mathsf{s} \cdots \mathsf{s}(\mathsf{z})$ *and* P *such that* $P \in B[\bar{n}/n]$*. Furthermore, the rules of* Figure 2.4 *suffice to derive* $\bar{n}$ *and* P*.*

Proof. By Lemma 2.16 (and the definition of $[\![(n{:}\mathsf{nat}) \times B]\!]$), $M \Downarrow \langle N, P\rangle$ and $M \doteq \langle N, P\rangle \in (n{:}\mathsf{nat}) \times B$. By Rule 2.33 and Lemma 2.16, $\mathsf{fst}(M) \doteq N \in \mathsf{nat}$ and $\mathsf{snd}(M) \doteq P \in B[\mathsf{fst}(M)/n]$. To determine $\bar{n} = \mathsf{s} \cdots \mathsf{s}(\mathsf{z})$ such that $N \doteq \bar{n} \in \mathsf{nat}$, apply Lemma 2.16 recursively to N—either $N \Downarrow \mathsf{z}$ or $N \Downarrow \mathsf{s}(N')$ such that $N' \doteq \bar{n}' \in \mathsf{nat}$, and in the latter case, $\mathsf{s}(N') \doteq \mathsf{s}(\bar{n}') \in \mathsf{nat}$ by Rule 2.40. (To project N' from $\mathsf{s}(N')$, subtract $\mathsf{s}(\mathsf{z})$ and evaluate.) Then $B[\mathsf{fst}(M)/n] \doteq B[\bar{n}/n] \in \mathsf{Type}_i$ by Lemma 2.24, and $P \in B[\bar{n}/n]$ by Lemma 2.13.

To see why the recursive procedure succeeds, let $\Phi = \{(N_0, N_0') \mid \exists \bar{n}. N_0 \doteq \bar{n} \in \mathsf{nat}\}$. We show $F(\Phi) \subseteq \Phi$, and therefore $[\![\mathsf{nat}]\!] \subseteq \Phi$; it follows that whenever $N \in \mathsf{nat}$, $N \Downarrow N_0$ and $N_0 \doteq \bar{n} \in \mathsf{nat}$. Suppose $F(\Phi)(N_0, N_0')$. If $F(\Phi)(\mathsf{z}, \mathsf{z})$ then $\bar{n} = \mathsf{z}$ and $\mathsf{z} \doteq \mathsf{z} \in \mathsf{nat}$. Otherwise $F(\Phi)(\mathsf{s}(M), \mathsf{s}(M'))$ and $\Phi(M, M')$, and therefore $M \Downarrow M_0$ and $M_0 \doteq \bar{n}' \in \mathsf{nat}$; then $\bar{n} = \mathsf{s}(\bar{n}')$ and $\mathsf{s}(M) \doteq \bar{n} \in \mathsf{nat}$ by Lemma 2.16 and Rule 2.40. □

2.5.5 *Booleans*

For $i \in \{0, 1, \dots, \omega\}$, $\tau_i(\mathsf{bool}, \mathsf{bool}, \{(\mathsf{true}, \mathsf{true}), (\mathsf{false}, \mathsf{false})\})$.

Rule 2.43 (Formation)**.** $\mathsf{bool} \in \mathsf{Type}_i$.

Proof. Immediate by the definition of $[\![\mathsf{Type}_i]\!]$. □

Rule 2.44 (Introduction)**.**

1. $\mathsf{true} \in \mathsf{bool}$.

2. $\mathsf{false} \in \mathsf{bool}$.

Proof. The presupposition bool type holds by the formation rule and Lemma 2.14. Both parts are immediate by the definition of $[\![\mathsf{bool}]\!]$. □

Rule 2.45 (Elimination)**.** *If* $b : \mathsf{bool} \gg A \in \mathsf{Type}_i$*,* $M \doteq M' \in \mathsf{bool}$*,* $T \doteq T' \in A[\mathsf{true}/b]$*, and* $F \doteq F' \in A[\mathsf{false}/b]$*, then* $\mathsf{if}(M; T, F) \doteq \mathsf{if}(M'; T', F') \in A[M/b]$.

Proof. There are two possibilities for $M \doteq M' \in \mathsf{bool}$. If $M \Downarrow \mathsf{true}$ and $M' \Downarrow \mathsf{true}$, then by Lemma 2.15 and $\mathsf{if}(M; T, F) \longmapsto^* \mathsf{if}(\mathsf{true}; T, F) \longmapsto T$, it suffices to show $T \doteq T' \in A[M/b]$. The result follows by $T \doteq T' \in A[\mathsf{true}/b]$, $A[\mathsf{true}/b] \doteq A[M/b]$ type (by $\mathsf{true} \doteq M \in \mathsf{bool}$), and Lemma 2.13. The false case is analogous. □

By the definition of $[\![\mathsf{bool}]\!]$, every closed Boolean equals either true or false. Canonicity, like the existence property, states that the rules of Idealized Nuprl derive such equations.

Theorem 2.46 (Canonicity)**.** *If* $M \in \mathsf{bool}$ *then either* $M \Downarrow \mathsf{true}$ *and* $M \doteq \mathsf{true} \in \mathsf{bool}$ *or* $M \Downarrow \mathsf{false}$ *and* $M \doteq \mathsf{false} \in \mathsf{bool}$*; moreover, the rules of Figure 2.4 derive these equations.*

Proof. By the definition of $[\![\mathsf{bool}]\!]$, either $M \Downarrow \mathsf{true}$ or $M \Downarrow \mathsf{false}$; the result follows by Lemma 2.16, which is a rule in Figure 2.4. □

Of course, Idealized Nuprl is also consistent, as it has empty types.

Theorem 2.47 (Consistency)**.** *There is no* E *such that* $E \in \mathsf{Eq}_{\mathsf{bool}}(\mathsf{true}, \mathsf{false})$.

Proof. Suppose such an E existed. Then $\mathsf{true} \doteq \mathsf{false} \in \mathsf{bool}$ by Rule 2.37, which contradicts the definition of $[\![\mathsf{bool}]\!]$. □

2.5.6 Universes

For $i, j \in \{0, 1, \dots, \omega\}$ with $i < j$:

$$\tau_j(\mathsf{Prop}_i, \mathsf{Prop}_i, \{(A_0, B_0) \mid \exists\alpha.\pi_i(A_0, B_0, \alpha)\})$$
$$\tau_j(\mathsf{Type}_i, \mathsf{Type}_i, \{(A_0, B_0) \mid \exists\alpha.\tau_i(A_0, B_0, \alpha)\})$$

Rule 2.48 (Formation)**.**

1. $\mathsf{Type}_i \in \mathsf{Type}_{i+1}$.
2. $\mathsf{Prop}_i \in \mathsf{Type}_{i+1}$.

Proof. The presupposition Type_{i+1} type is immediate by the definition of τ_ω; parts (1) and (2) are immediate by the definition of $[\![\mathsf{Type}_{i+1}]\!]$. □

We have already stated and proven the introduction rules of Prop_i and Type_i as formation rules of other types. The elimination rules of Prop_i and Type_i are simply that their elements are types, which we established in Lemma 2.14.

Rule 2.49 (Subtyping)**.** *If* $A \doteq A' \in \mathsf{Prop}_i$ *then* $A \doteq A' \in \mathsf{Type}_i$.

Proof. By the definition of $[\![\mathsf{Prop}_i]\!]$, $A \sim A' \downarrow \alpha \in \pi_i$. By Theorem 2.11 and monotonicity of candidate judgments, $A \sim A' \downarrow \alpha \in \tau_i$; the result follows by the definition of $[\![\mathsf{Type}_i]\!]$. □

Rule 2.50 (Cumulativity)**.**

1. *If* $A \doteq A' \in \mathsf{Type}_i$ *then* $A \doteq A' \in \mathsf{Type}_{i+1}$.
2. *If* $A \doteq A' \in \mathsf{Prop}_i$ *then* $A \doteq A' \in \mathsf{Prop}_{i+1}$.

Proof. For part (1), $A \sim A' \downarrow \alpha \in \tau_i$ implies $A \sim A' \downarrow \alpha \in \tau_{i+1}$ by Theorem 2.11, and the result follows by the definition of $[\![\mathsf{Type}_{i+1}]\!]$. Part (2) is analogous. □

Rule 2.51 (Subsingleton)**.** *If* $M \in A$, $M' \in A$, *and* $A \in \mathsf{Prop}_i$, *then* $M \doteq M' \in A$.

Proof. Let $\Phi = \{(A_0, A_0', \varphi) \mid \pi_i(A_0, A_0', \varphi) \wedge \forall M, M'.(M, M' \in A_0 \implies M \doteq M' \in A_0)\}$. (Note that $\pi_i(A_0, A_0', \varphi)$ implies A_0 type because $\pi_i \subseteq \tau_\omega$ by Theorem 2.11.) We begin by showing $\textsc{Props}(\Phi, \tau_i) \subseteq \Phi$; consulting Figure 2.3, there are three cases.

1. $\textsc{Fun}(\tau_i, \Phi)((a{:}A) \to B, (a{:}A') \to B', \varphi)$.

   By construction, $\Phi \subseteq \pi_i$, and therefore $\pi_i((a{:}A) \to B, (a{:}A') \to B', \varphi)$ by $\textsc{Fun}(\tau_i, \Phi) \subseteq \textsc{Fun}(\tau_i, \pi_i) \subseteq \pi_i$. Suppose $M, M' \in (a{:}A) \to B$. To show M and M' are equal, it suffices by Rules 2.27 and 2.30 to show $a : A \gg M\ a \doteq M'\ a \in B$, and hence that $M\ N \doteq M'\ N' \in B[N/a]$ for any $N \doteq N' \in A$. Unrolling the definition of $\textsc{Fun}(\tau_i, \Phi)$,

$a : [\![A]\!] \triangleright B \sim B' \downarrow \beta \in \Phi$ and therefore $B[N/a] \Downarrow B_0$ where $B_0 \in \mathsf{Prop}_i$ and all elements of B_0 are equal. But then $B[N/a] \doteq B_0$ type, so $M\ N \doteq M'\ N' \in B[N/a]$ by Lemma 2.13.

2. $\textsc{Pair}(\Phi)((a{:}A) \times B, (a{:}A') \times B', \varphi)$.

   Again, $\pi_i((a{:}A) \times B, (a{:}A') \times B', \varphi)$ by $\textsc{Pair}(\Phi) \subseteq \textsc{Pair}(\pi_i) \subseteq \pi_i$. Suppose $M, M' \in (a{:}A) \times B$. By Rules 2.32 and 2.34, it suffices to show $\mathsf{fst}(M) \doteq \mathsf{fst}(M') \in A$ and $\mathsf{snd}(M) \doteq \mathsf{snd}(M') \in B[\mathsf{fst}(M)/a]$. Unrolling the definition of $\textsc{Pair}(\Phi)$, $A \sim A' \downarrow [\![A]\!] \in \Phi$ and $a : [\![A]\!] \triangleright B \sim B' \downarrow \beta \in \Phi$. By the former, $A \Downarrow A_0$, $A_0 \in \mathsf{Prop}_i$, and all elements of A_0 are equal; thus by Lemma 2.13, $\mathsf{fst}(M) \doteq \mathsf{fst}(M') \in A$. By the latter, $B[\mathsf{fst}(M)/a] \Downarrow B_0$, $B_0 \in \mathsf{Prop}_i$, and all elements of B_0 are equal; again, by Lemma 2.13, $\mathsf{snd}(M) \doteq \mathsf{snd}(M') \in B[\mathsf{fst}(M)/a]$.

3. $\textsc{Eq}(\tau_i)(\mathsf{Eq}_A(M, N), \mathsf{Eq}_{A'}(M', N'), \varphi)$.

   We have $\pi_i(\mathsf{Eq}_A(M, N), \mathsf{Eq}_{A'}(M', N'), \varphi)$ by $\textsc{Eq}(\tau_i) \subseteq \pi_i$. Suppose $E, E' \in \mathsf{Eq}_A(M, N)$. By Rule 2.38, $E \doteq \star \in \mathsf{Eq}_A(M, N)$ and $E' \doteq \star \in \mathsf{Eq}_A(M, N)$, and therefore $E \doteq E' \in \mathsf{Eq}_A(M, N)$.

Because π_i is the least pre-fixed point of $\textsc{Props}(-, \tau_i)$, $\pi_i \subseteq \Phi$. Suppose $A \in \mathsf{Prop}_i$. Then $A \Downarrow A_0$, $A \doteq A_0 \in \mathsf{Prop}_i$, and by $[\![\mathsf{Prop}_i]\!] = \pi_i \subseteq \Phi$, $\Phi(A_0, A_0)$; therefore all elements of A_0 (and A) are equal, completing the proof. □

On open-endedness Computational semantics are *open-ended* in the sense of being modular and, like the BHK interpretation, easily extended with additional type formers, thereby reflecting the intuitionistic perspective that mathematics evolves over time [Dum77]. However, Idealized Nuprl's universes are not open-ended; on the contrary, they are fixed, inductively-defined collections of types! In this regard, computational semantics deviate significantly from categorical semantics, in which universes typically classify all collections below some size cutoff.

The inductive nature of our universes has some surprising consequences. First, not all subsingleton types are elements of Prop_i; for instance, $(n{:}\mathsf{nat}) \times \mathsf{Eq}_{\mathsf{nat}}(n, \mathsf{z}) \notin \mathsf{Prop}_i$ because nat is not subsingleton.[7] Secondly, functions out of Type_i can be defined by cases on type formers if the underlying programming language has such a facility, often called typecase. We sketch the operational semantics of typecase and the resulting Type_i elimination rule in Figure 2.5.

There are several ways to limit the power of typecase. Allen suggests a variant of computational semantics parametric in extensions of the $\textsc{Types}$ operator (Figure 2.3) that

[7] In this chapter, of course, one can simply define $\mathsf{Prop}_i := (A{:}\mathsf{Type}_i) \times ((a{:}A) \to (a'{:}A) \to \mathsf{Eq}_A(a, a'))$, but in Chapter 4 we cannot internalize the statement that a type is Kan.

$$\frac{}{\mathsf{typecase}(\mathsf{bool}; C_{\mathsf{bool}}, c.c'.C_{\mathsf{fun}}, \dots) \longmapsto C_{\mathsf{bool}}}$$

$$\frac{}{\begin{array}{c}\mathsf{typecase}((a{:}A) \to B; C_{\mathsf{bool}}, c.c'.C_{\mathsf{fun}}, \dots) \longmapsto \\ C_{\mathsf{fun}}[A/c][\lambda a.\mathsf{typecase}(B; C_{\mathsf{bool}}, c.c'.C_{\mathsf{fun}}, \dots)/c']\end{array}} \qquad \dots$$

$$\frac{C_{\mathsf{bool}} \in C[\mathsf{bool}/a] \qquad \begin{array}{c} a : \mathsf{Type}_i \gg C \in \mathsf{Type}_i \qquad A \in \mathsf{Type}_i \\ c : \mathsf{Type}_i, c' : c \to \mathsf{Type}_i \gg C_{\mathsf{fun}} \in C[(a{:}c) \to (c'\ a)/a] \end{array} \qquad \dots}{\mathsf{typecase}(A; C_{\mathsf{bool}}, c.c'.C_{\mathsf{fun}}, \dots) \in C[A/a]}$$

Figure 2.5: Fragment of the syntax and semantics of typecase.

generates the universe hierarchy [All87, Chapter 6]. In Chapter 4 of this dissertation, universes are inductively defined but include a univalence operator requiring maps out of $\mathcal{U}_i^{\mathsf{Kan}}$ to respect type isomorphism. Cavallo and Harper [CH19b] extend cubical type theory with a *relativity* principle that refutes excluded middle by ensuring maps out of the universe are in some sense parametric. Conversely, inductively-defined universes are in some cases desirable; Dagand and McBride have proposed closed universes of datatypes as an implementation strategy for datatype-generic dependent programming [Dag13].

Although its universes are not open-ended, Idealized Nuprl is nevertheless open-ended in two significant respects. First, because its judgments are monotone in the choice of type system, the ambient universe τ_ω is easily extended to a larger type system τ'_ω. For instance, to add an empty type to Idealized Nuprl, define

$$\tau'_\omega := \mu\tau.(\textsc{Types}(\nu_\omega, \tau) \cup \{(\mathsf{void}, \mathsf{void}, \{\})\}).$$

The containment $\tau_\omega \subseteq \tau'_\omega$ follows from Lemma 2.9, τ'_ω is a type system by a trivial modification to Lemma 2.6, and thus, by monotonicity of judgments, the rules of Figure 2.4 hold relative to τ'_ω (as does void type, but not $\mathsf{void} \in \mathsf{Type}_i$).

Secondly, Idealized Nuprl is stable under extensions of the syntax and operational semantics of its underlying programming language, subject to certain weak restrictions (notably, determinacy of evaluation) [How91]. Howe calls this property *computational open-endedness*, and, with Stoller, has defined a variant of the Nuprl type theory that includes all classical set-theoretic functions [HS94].

2.6 *Modern Nuprl*

The PRL group at Cornell has used the Nuprl theorem prover [Con+85] continuously since the mid-1980s, when Allen [All87] first described its computational semantics. They have extended the system and its semantics many times in the intervening decades, both increasing its expressivity and refining their methodology of computational type theory (described in Section 2.4). We close our discussion of Idealized Nuprl by summarizing some of modern Nuprl's most significant innovations.

Continuity principles In Nuprl, any $F \in (\mathsf{nat} \to \mathsf{nat}) \to \mathsf{nat}$ provably depends on only finitely many values of its input functions [RB16]; intuitively, a terminating program cannot check infinitely many values. Brouwer used this and other *continuity principles* to prove that, intuitionistically, all functions on the real numbers are continuous [TD88, pp. 206–210]. (Both of these statements are classically false, of course, so modern Nuprl is not compatible with classical logic.) Nuprl validates this principle by means of exceptions [Har16, Chapter 29] in its programming language that allow users to intercept F's calls to its input and compute an explicit bound.

Non-termination Nuprl's programming language extends the untyped λ-calculus, which famously contains divergent terms [Har16, Chapter 21]. Building on earlier work of Smith [Smi89], Crary [Cra98] extends Nuprl with *partial types* $\bar{A}$ whose elements are either divergent or elements of A. Unlike Idealized Nuprl, in which all elements terminate, Crary defines types as evaluation-respecting PERs on **Tm** (if $[\![A]\!](M,M)$ and $M \longmapsto M'$ then $[\![A]\!](M,M')$). Finally, Crary establishes a broad class of *admissible types* A whose partial types contain all fixed points of functions $\bar{A} \to \bar{A}$:

$$\frac{M \in \bar{A} \to \bar{A} \qquad A\ \mathsf{admiss}}{\mathsf{fix}(M) \in \bar{A}}$$

Computational equivalence As discussed in Section 2.4, Idealized Nuprl respects open computation at top level in both terms and types. In practice, one wishes also to compute *inside* larger terms to prove, for instance, that $\Gamma \gg M[\mathsf{fst}(\langle a,b\rangle)/c] \doteq M[a/c] \in A$. Such a principle holds definitionally in intensional type theory, but is defeasible in computational semantics—in theory, M could detect the head constructor of an unevaluated subterm.

Howe [How89] proves that Nuprl's type system respects *applicative bisimulation* (essentially, untyped equivalence of evaluation behavior), and moreover, that applicative bisimulation is a congruence if every operator depends only on the evaluation behavior of its arguments, unlike our hypothetical M—a condition satisfied by Nuprl's programming language. Nuprl's users can therefore replace arbitrary subterms of proof goals by equivalent terms, including but not limited to β-reductions; Rahli, Bickford, and Anand [RBA13] have applied non-trivial optimizations to extracted Paxos code. Modern Nuprl defines types as applicative bisimulation–respecting PERs, and includes a base type whose elements are all closed terms modulo applicative bisimulation [AR14].

Membership and PER types The base type is an essential component of modern Nuprl's expressive power. In Idealized Nuprl, the equality type on A ranges over elements of A, so the proposition $\mathsf{Eq}_A(M, M)$ carries no information—whenever it is a type, it is also true. In Nuprl, equality types range instead over base, allowing them to internalize not only equality but also membership in the form of $\mathsf{Eq}_A(M, M)$ [AR14].

Anand et al. [Ana+14] recently extended Nuprl with a $\mathsf{pertype}(R)$ type former that, for any symmetric and transitive $R \in \mathsf{base} \to \mathsf{base} \to \mathsf{Type}_i$, has R as its underlying PER of elements. Many standard type formers are instances of PER types and can therefore be defined internally in Nuprl, including dependent function types, inductive types, partial types, and quotient types.

Pointwise functionality Idealized Nuprl's open judgments range over contexts, telescopes of types that respect the equality of, or *are functional in*, all earlier types. Nuprl is based instead on *pointwise functionality*, a notion introduced by Allen [All87, Chapter 8] and defined by Anand and Rahli [AR14] as follows:

Definition 2.52 (Pointwise functionality)**.** For γ a list and Γ a telescope, $\Gamma\ \mathsf{ctx} \mathbin{@} \gamma$ when $\Gamma @ \gamma \sim \Gamma @ \gamma'\ \overrightarrow{\mathsf{type}}$ for all γ' such that $\gamma \sim \gamma' \in \Gamma$, where

1. $\cdot @ \cdot \sim \cdot @ \cdot\ \overrightarrow{\mathsf{type}}$, and
2. $(a : A, \Gamma)@(M, \gamma) \sim (a : A', \Gamma')@(M', \gamma')\ \overrightarrow{\mathsf{type}}$ when $A \doteq A'\ \mathsf{type}$ and $\Gamma[M/a]@\gamma \sim \Gamma'[M'/a]@\gamma'\ \overrightarrow{\mathsf{type}}$.

Definition 2.53 (Pointwise open judgments)**.**

1. $\Gamma \gg A \doteq A'\ \mathsf{type}$ when for all $\gamma \sim \gamma' \in \Gamma$, if $\Gamma\ \mathsf{ctx} \mathbin{@} \gamma$ then $A\gamma \doteq A'\gamma'\ \mathsf{type}$.
2. $\Gamma \gg M \doteq M' \in A$ when for all $\gamma \sim \gamma' \in \Gamma$, if $\Gamma\ \mathsf{ctx} \mathbin{@} \gamma$ then $A\gamma \doteq A\gamma'\ \mathsf{type}$ and $M\gamma \doteq M'\gamma' \in A\gamma$.

Unlike our Definition 2.19, which requires each type of Γ to respect all equalities in its indices, pointwise functionality requires only that each type respect equality at *total* assignments γ of Γ. As a consequence, Nuprl admits several nonstandard rules, notably:

$$\frac{\begin{array}{c}\Gamma, a : \mathsf{base}, a' : \mathsf{base}, e : \mathsf{Eq}_A(a, a'), \Delta \gg B \doteq B[a'/a] \in \mathsf{Type}_i \\ \Gamma, a : \mathsf{base}, e : \mathsf{Eq}_A(a, a), \Delta \gg M \in B \qquad a, e \# M\end{array}}{\Gamma, a : A, \Delta \gg M \in B}$$

$$\frac{M \in \mathsf{nat} \qquad Z \in A[\mathsf{z}/n] \qquad n : \mathsf{nat}, a : A \gg S \in A[\mathsf{s}(n)/n]}{\mathsf{natrec}(M; Z, n.a.S) \in A[M/n]}$$

The first, called *pointwise functionality*, allows users to rewrite a hypothesis $a : A$ as a hypothesis $a : \mathsf{base}$ such that $a \in A$, so long as the conclusion is functional in equality in A. The second is a very strong elimination rule for nat that omits the usual premise $n : \mathsf{nat} \gg A \in \mathsf{Type}_i$. In Idealized Nuprl, that premise is required for the third premise to be well-formed, because $(n : \mathsf{nat}, a : A)$ must be a context; in Nuprl, the pointwise functionality of A in nat is established inductively by earlier instances of S. This rule nearly halves the number of inductions on nat one performs, because establishing $n : \mathsf{nat} \gg A \in \mathsf{Type}_i$ itself often requires induction.

On the other hand, Nuprl's open judgments do not validate the dependent cut principle[8] (Lemma 2.24), but instead the weaker rule

$$\frac{\Gamma, a : A, \Delta \gg M \in B \qquad \Gamma \gg N \in A}{\Gamma, \Delta \gg M[N/a] \in B[N/a]}$$

where Δ does not depend on a. Why does dependent cut fail? In both Nuprl and Idealized Nuprl, $\Delta[M/a]$ may respect equality in Γ while Δ does not respect equality in $(\Gamma, a : A)$. In Nuprl, this causes $(\Gamma, \Delta[M/a])\ \mathsf{ctx} \mathbin{@} (\gamma, \delta)$ to hold while $(\Gamma, a : A, \Delta)\ \mathsf{ctx} \mathbin{@} (\gamma, M, \delta)$ fails, and because these occur in antecedents of implications in Definition 2.53, the conclusion of dependent cut fails while its premises hold.

[8] Dependent cut is only admissible in intensional type theory, allowing one in principle to consider models not closed under it; however, usual notions of categorical model of type theory [Dyb96] are closely related to explicit substitution calculi [Gra09, Chapter V] in which dependent cut is derivable.

3 Cubical methods

In this note it will be indicated how a homotopy theory may be developed for all abstract cubical complexes which satisfy only a certain *extension axiom*; homotopy groups will be introduced for all such complexes.

—Daniel M. Kan, *Abstract Homotopy. I* [Kan55, p. 1092]

This dissertation extends the computational semantics of type theory to account for the homotopy-type-theoretic features of *univalence* and *higher inductive types.* We begin this chapter by describing these features' original formulations in Book HoTT [UF13]. Next, we motivate and compare various *cubical* approaches to homotopy type theory, from the symmetric monoidal cubical model of Bezem, Coquand, and Huber [BCH14; BCH18] to our Cartesian cubical type theory and the De Morgan cubical type theory of Cohen et al. [CCHM18]. Finally, we discuss possible variations on existing cubical techniques.

Book HoTT The equality type of Idealized Nuprl, like those of Nuprl [Con+85] and extensional type theory [ML82], satisfies straightforward rules that capture a mathematical notion of equality—proofs of equality are unique, equality is extensional at every type, and any element can be silently replaced by an equal element. In contrast, as discussed in Section 2.4, intensional type theory [ML75b] restricts silent *definitional equality* to α-, β-, and certain η-equivalences, mediating all other equations through an *identity type* whose rules are listed in Figure 3.1 [ML75b; UF13, Section A.2.10]. (We write $\Gamma \vdash M = N : A$ for the judgments of intensional type theories, to emphasize that they have different intended properties than judgments of computational semantics $\Gamma \gg M \doteq N \in A$.)

To understand the effects of definitional equality, consider addition of natural numbers, defined by recursion on the left argument (that is, $\mathsf{z} + m = m$ and $\mathsf{s}(n) + m = \mathsf{s}(n + m)$):

$$n + m := \mathsf{natrec}(n; m, \_.a.\mathsf{s}(a))$$

Here the equation $\mathsf{z} + m = m$ holds definitionally, whereas $\mathsf{Id}_{\mathsf{nat}}(n + \mathsf{z}, n)$ only holds *propositionally*, or up to the identity type, because the latter equation must be established by induction on n. Given a nat-indexed family of types $C(n)$, an element of $C(\mathsf{z} + m)$ is also an element of $C(m)$, because these types are definitionally equal, but an element of $C(n + \mathsf{z})$ must be *transported* over a proof of $\mathsf{Id}_{\mathsf{nat}}(n + \mathsf{z}, n)$ to obtain an element of $C(n)$.

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : A}{\Gamma \vdash \mathsf{Id}_A(M, N) : \mathsf{Type}_i} \qquad\qquad \frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{refl}(M) : \mathsf{Id}_A(M, M)}$$

$$\frac{\begin{array}{c}\Gamma, a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash C : \mathsf{Type}_i \\ \Gamma, z : A \vdash Q : C[z, z, \mathsf{refl}(z)/a, b, p] \qquad \Gamma \vdash P : \mathsf{Id}_A(M, N)\end{array}}{\Gamma \vdash \mathsf{J}_{a.b.p.C}(P; z.Q) : C[M, N, P/a, b, p]}$$

$$\frac{\begin{array}{c}\Gamma, a : A, b : A, p : \mathsf{Id}_A(x, y) \vdash C : \mathsf{Type}_i \\ \Gamma, z : A \vdash Q : C[z, z, \mathsf{refl}(z)/a, b, p] \qquad \Gamma \vdash M : A\end{array}}{\Gamma \vdash \mathsf{J}_{a.b.p.C}(\mathsf{refl}(M); z.Q) = Q[M/a] : C[M, M, \mathsf{refl}(M)/a, b, p]}$$

Figure 3.1: Rules of intensional identity types.

Transport is a crucial but straightforward consequence of identity elimination:

$$\begin{gathered}\mathsf{transport} : (C{:}B \to \mathsf{Type}_i) \to (p{:}\mathsf{Id}_B(b, b')) \to C\ b \to C\ b' \\ \mathsf{transport}\ C\ p := \mathsf{J}_{a.a'.\_.C(a)\to C(a')}(p; \_.\lambda c.c)\end{gathered}$$

The requirement that proofs explicitly annotate uses of non-trivial equations allows implementations of intensional type theory to decide equality of elements and thence typing judgments, at the cost of added bureaucracy. Computational semantics nevertheless validate erasing all uses of transport—because all closed identity proofs are reflexive—allowing extracted programs to avoid any overhead induced by these annotations.

Many semantically-natural equations hold only propositionally in intensional type theory, including uniqueness principles for many types [UF13, Corollary 2.7.3]. Many others fail completely, including *function extensionality* [Str93, p. 106], the principle that $(a{:}A) \to \mathsf{Id}_B(f(a), g(a))$ implies $\mathsf{Id}_{A\to B}(f, g)$, and *uniqueness of identity proofs* [HS98], the principle that $\mathsf{Id}_{\mathsf{Id}_A(a,a')}(p, p')$ for all $p, p' : \mathsf{Id}_A(a, a')$. However, identity proofs are unique in the following weaker sense, often called *singleton contractibility*:

$$\begin{gathered}\widehat{\mathsf{Id}}_A(a) := (b{:}A) \times \mathsf{Id}_A(a, b) \\ \mathsf{uniqueness} : (a{:}A) \to (\hat{p}{:}\widehat{\mathsf{Id}}_A(a)) \to \mathsf{Id}_{\widehat{\mathsf{Id}}_A(a)}(\langle a, \mathsf{refl}(a)\rangle, \hat{p}) \\ \mathsf{uniqueness}\ a\ \langle b, p\rangle := \mathsf{J}_{a.b.p.\mathsf{Id}_{\widehat{\mathsf{Id}}_A(a)}(\langle a,\mathsf{refl}(a)\rangle,\langle b,p\rangle)}(p; z.\mathsf{refl}(\langle z, \mathsf{refl}(z)\rangle))\end{gathered}$$

As discussed in Section 1.2, the intensional identity type is compatible with non-trivial elements of not only $\mathsf{Id}_A(a, b)$ but also $\mathsf{Id}_{\mathsf{Id}_A(a,b)}(p, q)$, $\mathsf{Id}_{\mathsf{Id}_{\mathsf{Id}_A(a,b)}(p,q)}(\alpha, \beta)$, *et cetera*. One such element is postulated by Voevodsky's univalence axiom [Voe10a], originally formulated:

$$\begin{gathered}
\mathsf{IsEquiv} : (A \to B) \to \mathsf{Type}_i \\
\mathsf{idequiv} : (A{:}\mathsf{Type}_i) \to \mathsf{IsEquiv}(\lambda a.a) \\
\mathsf{idtoequiv} : \mathsf{Id}_{\mathsf{Type}_i}(A, B) \to ((f{:}A \to B) \times \mathsf{IsEquiv}(f)) \\
\mathsf{idtoequiv}\ p := \mathsf{J}_{a.b.\_.(f:a\to b)\times\mathsf{IsEquiv}(f)}(p; z.\langle \lambda a.a, \mathsf{idequiv}(z)\rangle) \\
\mathsf{univalence} : \mathsf{IsEquiv}(\mathsf{idtoequiv})
\end{gathered}$$

Here, $\mathsf{IsEquiv}(f)$ is a propositionally-subsingleton notion of isomorphism up to the identity type, and $\mathsf{idequiv}(A)$ is a proof that the identity function $A \to A$ is such an equivalence. (Consult the *Homotopy Type Theory* book for many possible definitions of equivalence [UF13, Chapter 4].) For the purposes of this dissertation, we consider an equivalent but simpler presentation of univalence suggested by Licata [Lic16]:

$$\begin{gathered}
\mathsf{ua} : (f{:}A \to B) \to \mathsf{IsEquiv}(f) \to \mathsf{Id}_{\mathsf{Type}_i}(A, B) \\
\mathsf{ua}\beta : (f{:}A \to B) \to (e{:}\mathsf{IsEquiv}(f)) \to (a{:}A) \to \mathsf{Id}_B(\mathsf{transport}\ (\lambda A.A)\ (\mathsf{ua}\ f\ e)\ a, f\ a)
\end{gathered}$$

To see that univalence refutes uniqueness of identity proofs, consider the two proofs of $\mathsf{Id}_{\mathsf{Type}_i}(\mathsf{bool}, \mathsf{bool})$ obtained by applying ua to the equivalences $\lambda b.b$ and $\lambda b.\mathsf{if}(b; \mathsf{false}, \mathsf{true})$. If these were equal, $(\mathsf{transport}\ (\lambda A.A)\ -\ \mathsf{true})$ across each would be equal, but $\mathsf{ua}\beta$ propositionally equates these transports to true and false respectively. Moreover, a hierarchy of n univalent universes refutes uniqueness of n-fold iterated identity proofs [KS15].

Models of univalent type theory must therefore account for arbitrarily-iterated identity structure; Voevodsky's model [KL16] interprets types as *simplicial sets*—a combinatorial representation of topological spaces using triangles and tetrahedra of arbitrary dimension—and n-fold iterated identity types as (approximately) the collection of n-dimensional simplices of a type.

Computational content Univalence essentially adds new introduction rules (ua and $\mathsf{ua}\beta$) to Book HoTT's identity type without adding computation rules describing how $\mathsf{J}_{a.b.p.C}(-; z.Q)$ acts on these identity proofs. Book HoTT therefore lacks the canonicity and existence properties characteristic of type theories; for instance, transporting true across univalence yields a closed Boolean definitionally equal to neither true nor false:

$$\mathsf{transport}\ (\lambda A.A)\ (\mathsf{ua}\ (\lambda b.b)\ (\mathsf{idequiv}\ \mathsf{bool}))\ \mathsf{true} : \mathsf{bool}$$

As a result, proof assistants based on Book HoTT cannot fully simplify proof goals by open computation. Users of Book HoTT instead "compute propositionally" by manually invoking lemmas to simplify transports according to type family or identity proof.[1] The $\mathsf{ua}\beta$

[1] Readers can consult Angiuli et al. [Ang+16] for many examples of "propositional computation."

axiom is one such lemma, simplifying the above transport to true; another propositionally equates the transport of $\langle M, N\rangle$ in a family of product types to the pair of transports of M and N [UF13, Theorem 2.6.4].

In fact, Sattler and Kapulkin have recently shown[2] that "propositional computation" enjoys *homotopy canonicity* [Voe10b]—every closed Boolean can be propositionally simplified to true or false. Such simplifications are nevertheless challenging in practice, as famously demonstrated by *Brunerie's number*, a topological invariant of type nat defined in Book HoTT. In ordinary type theories, any such definition must compute to a concrete numeral. In Book HoTT, however, Brunerie could not prove it propositionally equal to any numeral; he later gave a different construction in which the invariant is 2 [Bru16].

These and other difficulties led researchers to seek univalent type theories with definitional canonicity, hoping that such type theories would be more usable (as their proof goals would simplify further), would enable the use of univalence in dependently-typed programming (as explored, for instance, by Angiuli et al. [Ang+16]), and would produce a constructive model of univalence simpler than Voevodsky's model, which is essentially classical [BC15].

To this end, Licata and Harper [LH12] established canonicity for a "truncated" type theory with non-trivial proofs of $\mathsf{Id}_A(M, N)$, but definitional uniqueness of $\mathsf{Id}_{\mathsf{Id}_A(M,N)}(P, Q)$ proofs. Their type theory introduces an auxiliary judgment $\Gamma \vdash P : M \simeq_A N$ expressing that P is a "term equivalence" between M and N in A; any such term equivalence gives rise to an element of $\mathsf{Id}_A(M, N)$, and *vice versa*. Using their auxiliary judgment, Licata and Harper explicitly axiomatize all operations on term equivalences (transport, symmetry and transitivity of identity, *et cetera*), providing enough computation rules (transport in a family of product types, transport across univalence, *et cetera*) to achieve canonicity.

Licata and Harper's term equivalence judgment reconciles non-trivial identity proofs with the type-theoretic aphorism[3] that *type formers internalize judgmental structure.* Indeed, dependent function types internalize membership-under-hypotheses ($\lambda a.M : (a{:}A) \to B$ whenever $a{:}A \vdash M : B$), identity types originally internalized definitional equality ($\mathsf{refl}(M) : \mathsf{Id}_A(M, N)$ whenever $M = N : A$), and the identity types of Licata and Harper [LH12] internalize term equivalence. The failure of this aphorism for Book HoTT's identity type—which is generated by $\mathsf{refl}(M)$, univalence, and even identity elimination—syntactically obscures the principle that $\mathsf{Id}_A(M, N)$ mirrors the structure of A (for instance, that transport in a family of product types is a pair of transports, or that propositional equality in a product type is a pair of propositional equalities).

However, Licata and Harper's approach does not scale to the infinitely-iterated identity structure found in Book HoTT, which would require not only infinitely many term equivalence judgments, but also infinitely many operations and computation rules.

[2] See the TYPES '19 abstract at `http://www.ii.uib.no/~bezem/abstracts/TYPES_2019_paper_110`.

[3] Also known as *Martin-Löf's judo move.*

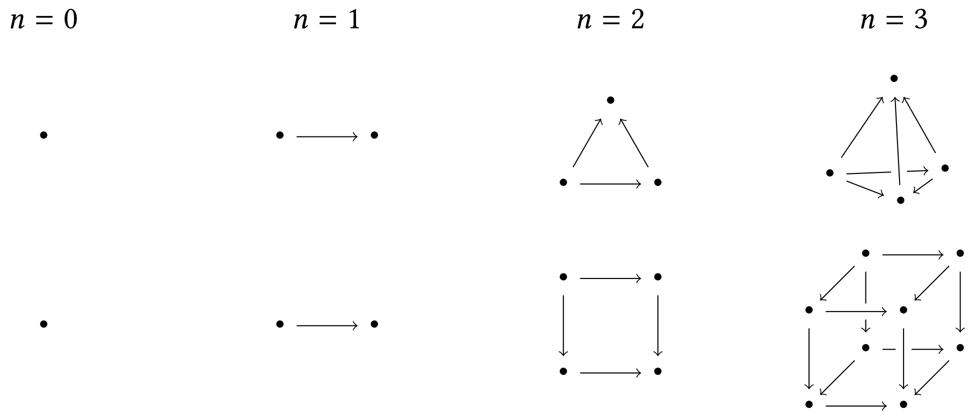


Figure 3.2: n-simplices (top) and n-cubes (bottom).

3.1 Symmetric monoidal cubes

Bezem, Coquand, and Huber [BCH14; BCH18] sparked a cubical revolution by presenting a constructive model of univalent type theory in *symmetric monoidal cubical sets*, a combinatorial presentation of infinite-dimensional structure analogous to simplicial sets. Cubical sets and simplicial sets generalize directed graphs ($n = 1$) in two different ways, as illustrated in Figure 3.2—an n-dimensional simplex has $n + 1$ faces of dimension $n - 1$, but an n-dimensional cube has $2n$ faces of dimension $n - 1$. Symmetric monoidal cubical sets are the first of three variations on cubical sets discussed in this dissertation, each with more operations than the last.

Definition 3.1. The *symmetric monoidal cube category* $\square_\otimes$ has as objects finite sets, and as morphisms $I \longrightarrow J$ set-theoretic functions $f : J \longrightarrow I + \{0, 1\}$ for which $f|_{f^{-1}(I)}$ is injective (equivalently, for which $f(x) = f(y) \in I$ implies $x = y$) [BCH14; Pit15].[4] Composition of morphisms is the evident Kleisli composition, namely, $(g \circ f)(x) = f(x)$ when $f(x) \in \{0, 1\}$ and $g(f(x))$ otherwise.

Notable morphisms of $\square_\otimes$ include *face maps*

$$(x_i := 0), (x_i := 1) : (\{x_1, \ldots, x_n\} \setminus \{x_i\}) \longrightarrow \{x_1, \ldots, x_n\}$$

that are identity functions except for sending x_i to 0 or 1, and *degeneracy maps*

$$\hat{y} : \{x_1, \ldots, x_n, y\} \longrightarrow \{x_1, \ldots, x_n\}$$

[4] Bernardy, Coquand, and Moulin [BCM15] consider a very similar category whose morphisms $I \longrightarrow J$ are functions $f : J \longrightarrow I + \{0\}$ satisfying the same injectivity condition.

Faces
($\square_\otimes$, $\square$, $\square_{\mathrm{DM}}$)

$$\begin{tikzcd}
\bullet \arrow[r, "p"] \arrow[d, "r"'] & \bullet \arrow[d, "q"] \\
\bullet \arrow[r, "s"'] & \bullet
\end{tikzcd}
\mapsto
\begin{tikzcd}
\bullet \arrow[r, "p"] & \bullet & \bullet \arrow[r, "q"] & \bullet & \bullet \arrow[r, "r"] & \bullet & \bullet \arrow[r, "s"] & \bullet
\end{tikzcd}$$

Degeneracies
($\square_\otimes$, $\square$, $\square_{\mathrm{DM}}$)

$$\begin{tikzcd}
a \arrow[r, "p"] & b
\end{tikzcd}
\mapsto
\begin{tikzcd}
a \arrow[r, "p"] \arrow[d, equal] & b \arrow[d, equal] \\
a \arrow[r, "p"'] & b
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, equal] \arrow[d, "p"'] & a \arrow[d, "p"] \\
b \arrow[r, equal] & b
\end{tikzcd}$$

Diagonals
($\square$, $\square_{\mathrm{DM}}$)

$$\begin{tikzcd}
a \arrow[r] \arrow[d] & \bullet \arrow[d] \\
\bullet \arrow[r] & b
\end{tikzcd}
\mapsto
\begin{tikzcd}
a \arrow[r] & b
\end{tikzcd}$$

Connections
($\square_{\mathrm{DM}}$)

$$\begin{tikzcd}
a \arrow[r, "p"] & b
\end{tikzcd}
\mapsto
\begin{tikzcd}
a \arrow[r, "p"] \arrow[d, "p"'] & b \arrow[d, equal] \\
b \arrow[r, equal] & b
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, equal] \arrow[d, equal] & a \arrow[d, "p"] \\
a \arrow[r, "p"'] & b
\end{tikzcd}$$

Reversals
($\square_{\mathrm{DM}}$)

$$\begin{tikzcd}
a \arrow[r] & b
\end{tikzcd}
\mapsto
\begin{tikzcd}
b \arrow[r] & a
\end{tikzcd}$$

Figure 3.3: Operations in cubical sets.

that miss y and are otherwise identity functions. (Because of degeneracy maps, $\square_\otimes$ is in fact a symmetric *semicartesian* monoidal category.)

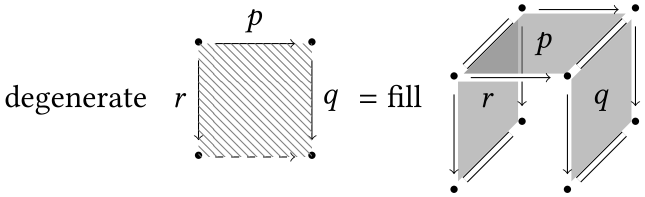
A symmetric monoidal cubical set F is a presheaf on $\square_\otimes$, that is, a functor $\square_\otimes^{\mathbf{op}} \to \mathbf{Set}$, or a family of sets $F(I)$ for each $I \in \square_\otimes$ equipped with functions $F(f) : F(I) \to F(J)$ for each $f : J \to I$ in $\square_\otimes$. Geometrically, one can imagine F as a space and $F(\{x_1, \ldots, x_n\})$ as the set of continuous functions $[0,1]^n \to F$, or *n-cubes of F*; then face maps $F(x_i := \varepsilon)$ restrict n-cubes to $(n-1)$-cubes by setting the x_i coordinate to ε, and degeneracy maps $F(\hat{y})$ regard n-cubes as $(n+1)$-cubes constant in the new y coordinate. (See Figure 3.3.) Faces and degeneracies satisfy the expected geometric identities—the "top" face of the "left" face of a square is also the "left" face of its "top" face (because $(x := 0) \circ (y := 0) = (y := 0) \circ (x := 0)$), and the "top" and "bottom" faces of a vertically-degenerate square are equal (because $\hat{y} \circ (y := 0) = \hat{y} \circ (y := 1) = \mathbf{id}$).

Cubical sets are a convenient representation of infinitely-iterated data, but lack enough structure needed to model spaces or identity types. In a space X, every path $p : [0,1] \to X$ has an inverse path p^{-1} with $p^{-1}(0) = p(1)$ and $p^{-1}(1) = p(0)$; similarly, in intensional type theory, identity is symmetric:

$$\lambda p.\mathsf{transport}\ (\lambda c.\mathsf{Id}_A(c,a))\ p\ \mathsf{refl}(a) : \mathsf{Id}_A(a,b) \longrightarrow \mathsf{Id}_A(b,a)$$

In symmetric monoidal (or Cartesian) cubical sets, however, n-cubes need not be invertible. Mathematicians describe spacelike cubical sets as those satisfying the *Kan condition* [Kan55], namely, for any configuration of $(n-1)$-cubes forming all but one face of an n-cube, there exists an n-cube with that boundary. For $n = 2$, the Kan condition asserts that for any three lines forming the left, top, and right sides of a square (evocatively called an *open box*), there exists a square (a *filler*) whose left, top, and right faces are those lines.

In their model of type theory, Bezem, Coquand, and Huber [BCH14] introduce *uniform Kan operations*, which refine the Kan condition in three critical ways.[5] First, rather than stipulating the mere existence of fillers, the uniform Kan operations determine functions sending each open box to a chosen filler. Secondly, open boxes are allowed to omit more than one face from a cube, and in fact can specify as few as one face. Thirdly, the uniform Kan operations must commute with face and degeneracy maps—degenerating the filler of an open box must coincide with the filler of the degenerated open box:



Bezem, Coquand, and Huber [BCH14; BCH18] therefore model types as symmetric monoidal cubical sets equipped with a uniform Kan operation; their model includes dependent function types, dependent pair types, identity types, and univalent universes.

Towards cubical type theory The model of Bezem, Coquand, and Huber [BCH14] brought us closer to a univalent type theory with canonicity—it allows one to compute the values of well-typed terms,[6] but not to reflect those computations as definitional (or even propositional) equalities of the theory. However, various researchers immediately set out to develop a type theory based on uniform Kan cubical sets, using the insight that cubical sets have a natural syntactic representation.

[5] In Section 3.2 we will define uniform Kan operations more precisely in the Cartesian setting. Readers specifically interested in the setting of Bezem, Coquand, and Huber [BCH14] may wish to consult the detailed exposition of Harper and Hou (Favonia) [HF15].

[6] In December 2013, Cohen, Coquand, Huber, and Mörtberg released `cubical` (`https://github.com/simhu/cubical`), a prototype evaluator with precisely this purpose.

By the Yoneda lemma, the n-cubes of a cubical set F are in bijection with the natural transformations $\mathbf{hom}_{\square_\otimes}(-, \{x_1, \dots, x_n\}) \to F$. The Yoneda embedding sends the monoidal product $\otimes$ of $\square_\otimes$ (for which $\{x_1, \dots, x_n\} = \{x_1\} \otimes \cdots \otimes \{x_n\}$) to a Day convolution product of presheaves. Therefore, writing $\mathbb{I} := \mathbf{hom}_{\square_\otimes}(-, \{\star\})$ for the *representable* 1*-cube*, we obtain a bijection between the n-cubes of F and natural transformations $\mathbb{I}^{\otimes n} \to F$.

These calculations justify our earlier intuition that n-cubes of F are continuous functions $[0, 1]^n \to F$. (Unlike $[0, 1]$, $\mathbb{I}$ is an *abstract* interval with only two points, 0 and 1.) Critically, F's structure is determined by morphisms out of a product of cubical sets, just as the judgments of type theory describe morphisms out of products of types. We therefore consider *cubical judgments* expressing that M is an n-cube of A:

$$x_1 : \mathbb{I}, \dots, x_n : \mathbb{I} \vdash M : A$$

Cubical judgments involve a new form of *interval variable* x_i, subject to *interval substitutions* $\langle 0/x_i \rangle$ and $\langle 1/x_i \rangle$ which implement the action of face maps $(x_i := 0)$ and $(x_i := 1)$, and weakenings, which implement the action of degeneracy maps. *Cubical type theory* therefore scales Licata and Harper's type theory [LH12] to arbitrary dimension: interval variables uniformly define n-dimensional judgments, and uniform Kan operations, which commute with interval substitution and can therefore be represented syntactically, compactly axiomatize all operations on n-cubes.

A frequent question from mathematicians is: why cubical type theory and not simplicial type theory? The representable n-simplex is not a product of representable 1-simplices, making simplices less amenable than cubes to syntactic encoding. Mathematicians prefer simplicial sets because geometric realization commutes with products of simplicial sets up to homotopy, whereas the same is not true of symmetric monoidal cubical sets [BM17]. These concerns appear irrelevant to type theory, however—they affect the spacelike structure induced abstractly on presheaves through equivalence of categories [Cis06], whereas we induce spacelike structure explicitly on types through uniform Kan operations.

Symmetric monoidal cubes nevertheless pose two difficulties to type theorists. First, their interval variables are substructural—by the injectivity condition of Definition 3.1, there are no *contraction* substitutions joining two variables—and many subtleties arise when unifying dependency and substructurality [KPB15]. More importantly, the elimination principles of higher inductive types appear to require contraction. Suppose we have a higher inductive type $\mathbb{T}$ with 1-cube constructor $x : \mathbb{I} \vdash \mathsf{line}(x)$. Functions $(t{:}\mathbb{T}) \to A$ are therefore determined by 1-cubes $x : \mathbb{I} \vdash L : A[\mathsf{line}(x)/t]$, and send constructors $\mathsf{line}(y)$ to $L\langle y/x \rangle$, which contracts x and y in L:

$$\frac{\begin{array}{c}\Gamma, t : \mathbb{T} \vdash A : \mathsf{Type}_i \\ \Gamma \vdash M : \mathbb{T} \\ \Gamma, x : \mathbb{I} \vdash L : A[\mathsf{line}(x)/t]\end{array}}{\Gamma \vdash \mathsf{elim}_{t.A}(M; x.L) : A[M/t]} \qquad \frac{\Gamma, t : \mathbb{T} \vdash A : \mathsf{Type}_i \qquad \Gamma, x : \mathbb{I} \vdash L : A[\mathsf{line}(x)/t]}{\Gamma \vdash \mathsf{elim}_{t.A}(\mathsf{line}(y); x.L) = L\langle y/x \rangle : A[\mathsf{line}(y)/t]}$$

$$\frac{\begin{array}{ll} & \Gamma \vdash M : A \\ (\forall i) & \Gamma, \xi_i, y : \mathbb{I} \vdash N_i : A \\ (\forall i, j) & \Gamma, \xi_i, \xi_j, y : \mathbb{I} \vdash N_i = N_j : A \\ (\forall i) & \Gamma, \xi_i \vdash N_i\langle r/y \rangle = M : A \end{array}}{\begin{array}{l} \Gamma, y : \mathbb{I} \vdash \mathsf{hcom}_A^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) : A \\ \qquad = \begin{cases} M & \text{under } \langle r/y \rangle \\ N_i & \text{under } \xi_i \end{cases} \end{array}} \qquad \frac{\Gamma, x : \mathbb{I} \vdash A : \mathsf{Type}_i \qquad \Gamma \vdash M : A\langle r/x \rangle}{\begin{array}{l} \Gamma \vdash \mathsf{coe}_{x.A}^{r \rightsquigarrow r'}(M) : A\langle r'/x \rangle \\ \quad = M \quad \text{under } r = r' \end{array}}$$

Figure 3.4: Cartesian cubical Kan operations.

3.2 *Cartesian cubes*

Cartesian cubical sets augment symmetric monoidal cubical sets with contraction substitutions $\langle y/x \rangle$, whose actions compute the *diagonals* of $(n+2)$-cubes. (See Figure 3.3.) Cartesian interval variables are structural (satisfy exchange, weakening, and contraction) and, unlike De Morgan interval variables, are subject to no equations.

Definition 3.2. The *Cartesian cube category* $\square$ has as objects finite sets, and as morphisms $I \longrightarrow J$ set-theoretic functions $J \longrightarrow I + \{0, 1\}$, with the evident Kleisli composition [AFH18; Ang+19; Awo18; Par15].

Work on Cartesian cubes dates to around 2014, when Coquand[7] and Licata and Brunerie [LB14] presented fragments of Cartesian cubical type theories. In 2016, Awodey [Awo18] presented a model of identity types in Cartesian cubical sets. In 2017, the author and collaborators succeeded in defining a univalent Cartesian cubical type theory with canonicity, presented both as a computational type theory (Appendix A) with a computational semantics (Chapter 4) [AFH18], and as an intensional type theory (Appendix B) with a denotational semantics in Cartesian cubical sets [Ang+19].

In Chapter 4 we will decompose the uniform Kan operation into *homogeneous composition*, which fills boxes in cubically-constant types, and *coercion*, a form of transport for cubically-dependent types. This decomposition is essential when defining higher inductive types [CH19a]; De Morgan cubical type theory adopts a similar decomposition but *only* at higher inductive types [CHM18]. Figure 3.4 contains rules for our Kan operations, in which r and r' are *interval terms*, or elements of $\mathbb{I}$ (concretely, either 0, 1, or interval variables), and ξ_i are *interval equations* (concretely, $r = r'$) that restrict judgments to particular subcubes.

[7] See the note at `http://www.cse.chalmers.se/~coquand/diag.pdf`, dated April 9, 2014.

$$\frac{\begin{array}{c}\Gamma, x : \mathbb{I} \vdash A : \mathsf{Type}_i \\ \Gamma \vdash P_0 : A\langle 0/x\rangle \\ \Gamma \vdash P_1 : A\langle 1/x\rangle\end{array}}{\Gamma \vdash \mathsf{Path}_{x.A}(P_0, P_1) : \mathsf{Type}_i} \qquad \frac{\Gamma, x : \mathbb{I} \vdash M : A}{\Gamma \vdash \langle x\rangle M : \mathsf{Path}_{x.A}(M\langle 0/x\rangle, M\langle 1/x\rangle)}$$

$$\frac{\Gamma \vdash M : \mathsf{Path}_{x.A}(P_0, P_1)}{\Gamma \vdash M@r : A\langle r/x\rangle} \qquad \frac{\Gamma \vdash M : \mathsf{Path}_{x.A}(P_0, P_1)}{\Gamma \vdash M@0 = P_0 : A\langle 0/x\rangle} \qquad \frac{\Gamma \vdash M : \mathsf{Path}_{x.A}(P_0, P_1)}{\Gamma \vdash M@1 = P_1 : A\langle 1/x\rangle}$$

$$\frac{\Gamma, x : \mathbb{I} \vdash M : A}{\Gamma \vdash (\langle x\rangle M)@r = M\langle r/x\rangle : A\langle r/x\rangle} \qquad \frac{\Gamma \vdash M : \mathsf{Path}_{x.A}(P_0, P_1)}{\Gamma \vdash M = \langle x\rangle M@x : \mathsf{Path}_{x.A}(P_0, P_1)}$$

Figure 3.5: Rules of path types in cubical type theory.

Identity elimination To familiarize ourselves with these Kan operations, we will consider *path types* (Figure 3.5), a cubical analogue of identity types[8] that propositionally—but *not* definitionally—model the computation rule in Figure 3.1. Elements of $\mathsf{Path}_{x.A}(P_0, P_1)$ are dependent functions $(x{:}\mathbb{I}) \to A$ that send 0 and 1 to P_0 and P_1 respectively; like ordinary functions, paths are introduced by abstraction ($\langle x\rangle M$) and eliminated by application ($M@r$). The reflexive path on $M : A$ is a constant function $\langle\_\rangle M$, which has type $\mathsf{Path}_{\_.A}(M, M)$.

We define the "identity eliminator" of path types using Kan operations and the folkloric observation that identity elimination is interderivable with transport and uniqueness. Transport for path types is a straightforward instance of coercion:

$$\begin{aligned} \mathsf{transport}_{\square} &: (C{:}B \to \mathsf{Type}_i) \to (p{:}\mathsf{Path}_{\_.B}(b, b')) \to C\ b \to C\ b' \\ \mathsf{transport}_{\square}\ C\ p\ c &:= \mathsf{coe}^{0 \rightsquigarrow 1}_{x.C\ (p@x)}(c) \end{aligned}$$

observing that $(C\ (p@x))\langle 0/x\rangle = C\ b$ and $(C\ (p@x))\langle 1/x\rangle = C\ b'$.

Singleton contractibility follows from homogeneous composition (hcom) in a manner best explained graphically:

$$\begin{aligned} \widehat{\mathsf{Path}}_A(a) &:= (b{:}A) \times \mathsf{Path}_{\_.A}(a, b) \\ \mathsf{uniqueness}_{\square} &: (a{:}A) \to (\hat{p}{:}\widehat{\mathsf{Path}}_A(a)) \to \mathsf{Path}_{\_.\widehat{\mathsf{Path}}_A(a)}(\langle a, \langle\_\rangle a\rangle, \hat{p}) \end{aligned}$$

$$\mathsf{uniqueness}_{\square}\ a\ \langle b, p\rangle := \langle x\rangle \left\langle \begin{tikzcd} {} \arrow[r, "a"] \arrow[d, "a"'] & {} \arrow[d, "p"] \\ {} \arrow[r, dashed] & {} \end{tikzcd}, \langle y\rangle \begin{tikzcd} {} \arrow[r, "a"] \arrow[d, "a"'] & {} \arrow[d, "p"] \\ {} \arrow[r, dashed] & {} \end{tikzcd} \right\rangle$$

[8] In fact, they correspond to Book HoTT's *dependent paths* [UF13, Section 2.3], because the type A can vary in $x : \mathbb{I}$.

To construct an element of type $\mathsf{Path}_{\_.\widehat{\mathsf{Path}}_A(a)}(\langle a, \langle\_\rangle a\rangle, \hat{p})$, we abstract $x : \mathbb{I}$ and construct an element of type $(b{:}A) \times \mathsf{Path}_{\_.A}(a, b)$ equal to $\langle a, \langle\_\rangle a\rangle$ when $x = 0$ (on the left) and $\langle b, p\rangle$ when $x = 1$ (on the right). Such an element is itself a pair; for its first component we could choose $p@x$, as p is a path from a to b, but let us momentarily postpone our choice.

By the ordinary rules of dependent pair types, the second component must have type $\mathsf{Path}_{\_.A}(a, q)$, where q is the first component. We exhibit such a path by abstracting $y : \mathbb{I}$ and constructing an element of A equal to a when $y = 0$ (on the top) and q when $y = 1$ (on the bottom). However, these are not our only constraints—as we noted when abstracting $x : \mathbb{I}$, the path must also equal $\langle\_\rangle a$ when $x = 0$ and p when $x = 1$.

In sum, we require a square equal to a when $x = 0$ or $y = 0$, to p when $x = 1$, and to q when $y = 1$. We select the following filler, depicted in our definition of $\mathsf{uniqueness}_{\square}$ as a hatched square:

$$F := \mathsf{hcom}_A^{0 \rightsquigarrow y}(a; x = 0 \hookrightarrow \_.a, x = 1 \hookrightarrow y.p@y)$$

The premises of the hcom rule require (1) A is a type, (2) $a : A$, (3) $a : A$ when $x = 0$ and $p@y : A$ when $x = 1$, (4) $a = p@y : A$ whenever both $x = 0$ and $x = 1$, which is vacuous, and (5) $a\langle 0/y\rangle = a : A$ when $x = 0$ and $(p@y)\langle 0/y\rangle = a : A$ when $x = 1$. The conclusions assert (1) $F : A$, (2) $F\langle 0/y\rangle = a : A$, and (3) $F = a : A$ when $x = 0$ and $F = p@y : A$ when $x = 1$. These conclusions discharge all of our obligations except $F\langle 1/y\rangle = q : A$, which we discharge by defining $q := F\langle 1/y\rangle$, the missing bottom face of the filler.

The hcom rule in Figure 3.4 is quite technical because it systematically accounts for all possible filling scenarios; particular instances, such as F, are often much simpler. For example, it is essential that ξ_i not depend on $y : \mathbb{I}$; otherwise, one could fill *closed boxes* and obtain uniqueness of identity proofs. We defer most technical details to Section 4.3.2, whose definition of hcom differs from Figure 3.4 in primarily cosmetic ways: the computational semantics divide interval variables, interval equations, and ordinary variables into three separate contexts, and explicitly close hcom under interval substitution by admitting composition to any r', not only $y : \mathbb{I}$. (A more significant difference is the *validity condition*, discussed in Section 3.5.)

We are finally prepared to derive an uncurried variant of identity elimination:

$$\mathsf{J}_{\square} : (a{:}A) \to (C{:}\widehat{\mathsf{Path}}_A(a) \to \mathsf{Type}_i) \to C\ \langle a, \langle\_\rangle a\rangle \to (\hat{p}{:}\widehat{\mathsf{Path}}_A(a)) \to C\ \hat{p}$$
$$\mathsf{J}_{\square}\ a\ C\ c\ \hat{p} := \mathsf{transport}_{\square}\ C\ (\mathsf{uniqueness}_{\square}\ a\ \hat{p})\ c$$

We do not obtain $\mathsf{J}_{\square}\ a\ C\ c\ \langle a, \langle\_\rangle a\rangle = c : C\ \langle a, \langle\_\rangle a\rangle$, which seems to require both $\mathsf{uniqueness}_{\square}\ a\ \langle a, \langle\_\rangle a\rangle$ to be a constant path, and $\mathsf{transport}_{\square}\ C\ p\ c$ to be the identity function when p is constant. These conditions, jointly named *regularity*, are *a priori* sensible but have proven quite elusive; we discuss regularity further in Section 3.4.

Instead, we exhibit a path between $\mathsf{J}_{\square}\ a\ C\ c\ \langle a, \langle\_\rangle a\rangle$ and c, or an element of:

$$(a{:}A) \to (C{:}\widehat{\mathsf{Path}}_A(a) \to \mathsf{Type}_i) \to (c{:}C\ \langle a, \langle\_\rangle a\rangle) \to \mathsf{Path}_{\_.C\ \langle a, \langle\_\rangle a\rangle}(\mathsf{J}_{\square}\ a\ C\ c\ \langle a, \langle\_\rangle a\rangle, c)$$

Because $\mathsf{J}_{\square}$ is an instance of $\mathsf{transport}_{\square}$, which is in turn an instance of coercion, we obtain a dependent path between $\mathsf{J}_{\square}\ a\ C\ c\ \langle a, \langle\_\rangle a\rangle$ and c by coercing to an interval variable:

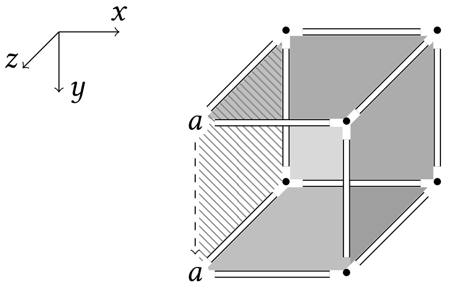
$$\lambda C.\lambda p.\lambda c.\langle x\rangle \mathsf{coe}^{0\rightsquigarrow x}_{x.C\ (p@x)}(c) :$$
$$(C{:}B \to \mathsf{Type}_i) \to (p{:}\mathsf{Path}_{\_.B}(b, b')) \to (c{:}C\ b) \to \mathsf{Path}_{x.C\ (p@x)}(c, \mathsf{transport}_{\square}\ C\ p\ c)$$

observing that $(\mathsf{coe}^{0\rightsquigarrow x}_{x.C\ (p@x)}(c))\langle 0/x\rangle = c : C\ b$ and $(\mathsf{coe}^{0\rightsquigarrow x}_{x.C\ (p@x)}(c))\langle 1/x\rangle$ is by definition $\mathsf{transport}_{\square}\ C\ p\ c$.

When specialized to $\mathsf{J}_{\square}$, we obtain an element of

$$\mathsf{Path}_{x.C\ ((\mathsf{uniqueness}_{\square}\ a\ \langle a, \langle\_\rangle a\rangle)@x)}(c, \mathsf{J}_{\square}\ a\ C\ c\ \langle a, \langle\_\rangle a\rangle)$$

a path with the desired endpoints but in the wrong type. To correct its type, we exhibit a path between $\mathsf{uniqueness}_{\square}\ a\ \langle a, \langle\_\rangle a\rangle$ and $\langle\_\rangle\langle a, \langle\_\rangle a\rangle$, which amounts to a path between $\langle\_\rangle a$ and the composition of a, a, and a, defined as the front face of a filler whose back, right, top, and bottom faces are a, and whose left face is the filler of a, a, and a:



We complete the proof by transporting our dependent path over the propositional equality between $\mathsf{uniqueness}_{\square}\ a\ \langle a, \langle\_\rangle a\rangle$ and $\langle\_\rangle\langle a, \langle\_\rangle a\rangle$, then inverting the result.

Path types are incredibly useful in cubical type theory, notwithstanding their failure to strictly model the computation rule of identity types. However, there are also constructions that *do* strictly model the rules of identity types (but not the rules of path types). The first, due to Swan [Swa18a], was originally conceived in the symmetric monoidal and De Morgan settings but has later been ported to the Cartesian setting by Angiuli et al. [Ang+19]. The second, due to Cavallo and Harper [CH19a], obtains identity types as an instance of indexed inductive types.

Closure under type formers To achieve canonicity, all hcom and coe terms must compute. In Chapter 4, we include computation steps defining the Kan operations at compound types in terms of the Kan operations at constituent types. The same principle applies in denotational semantics, in which one must check that, for example, the product of

Kan cubical sets is again Kan. Closure of Kan operations under the type formers, and particularly univalent universes, is one of the most technical aspects of cubical type theory.

Our construction of $\mathsf{J}_{\square}$ has already required coercions $0 \rightsquigarrow r'$ and homogeneous compositions $0 \rightsquigarrow r'$ with equations $(x = 0), (x = 1)$. We now argue that closure under dependent function and path types requires coercion and composition *from* arbitrary interval terms r as well.

Suppose $x : \mathbb{I} \vdash (a{:}A) \to B : \mathsf{Type}_i$ and $M : ((a{:}A) \to B)\langle 0/x\rangle$, and define a function $((a{:}A) \to B)\langle r'/x\rangle$ by abstracting $a : A\langle r'/x\rangle$. We cannot apply M to a directly, but we can apply M to $\mathsf{coe}^{r'\rightsquigarrow 0}_{x.A}(a)$, obtaining an element of $B\langle 0/x\rangle[\mathsf{coe}^{r'\rightsquigarrow 0}_{x.A}(a)/a]$. If B were not dependent, we would have an element of $B\langle 0/x\rangle$ and could simply coerce $0 \rightsquigarrow r'$ in B. Instead, we must coerce in $B[\mathsf{coe}^{r'\rightsquigarrow x}_{x.A}(a)/a]$, a type whose $\langle 0/x\rangle$ face is $B\langle 0/x\rangle[\mathsf{coe}^{r'\rightsquigarrow 0}_{x.A}(a)/a]$, and whose $\langle r'/x\rangle$ aspect is $B\langle r'/x\rangle$ (by $\mathsf{coe}^{r'\rightsquigarrow r'}_{x.A}(a) = a : A\langle r'/x\rangle$). In sum:[9]

$$\mathsf{coe}^{0\rightsquigarrow r'}_{x.(a:A)\to B}(M) := \lambda a.\mathsf{coe}^{0\rightsquigarrow r'}_{x.B[\mathsf{coe}^{r'\rightsquigarrow x}_{x.A}(a)/a]}(M\ \mathsf{coe}^{r'\rightsquigarrow 0}_{x.A}(a))$$

Because we need coercion $0 \rightsquigarrow r'$ in all types to define $\mathsf{J}_{\square}$, and we need coercion $r' \rightsquigarrow x$ in all types to obtain coercion $0 \rightsquigarrow r'$ in all dependent function types, we conclude that all coercions $r \rightsquigarrow r'$ are required.

Now suppose $y : \mathbb{I} \vdash \mathsf{Path}_{x.A}(P_0, P_1) : \mathsf{Type}_i$ and $M : (\mathsf{Path}_{x.A}(P_0, P_1))\langle r/y\rangle$ and define a path $(\mathsf{Path}_{x.A}(P_0, P_1))\langle r'/y\rangle$ by abstracting $x : \mathbb{I}$. The coercion $\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M@x) : A\langle r'/y\rangle$ has the correct type, but its $\langle \varepsilon/x\rangle$ faces are $\mathsf{coe}^{r\rightsquigarrow r'}_{y.A\langle \varepsilon/x\rangle}(P_\varepsilon\langle r/y\rangle)$ and not $P_\varepsilon\langle r'/y\rangle$ as required. However, under $x = \varepsilon$, $\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(P_\varepsilon) : A\langle r'/y\rangle$ is a y-cube whose $\langle r/y\rangle$ aspect agrees with our coercion of $M@x$, and whose $\langle r'/y\rangle$ aspect agrees with $P_\varepsilon\langle r'/y\rangle$. We obtain the result by composing the three aforementioned elements of $A\langle r'/y\rangle$:

$$\mathsf{coe}^{r\rightsquigarrow r'}_{y.\mathsf{Path}_{x.A}(P_0,P_1)}(M) := \langle x\rangle \mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle r'/y\rangle}(\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M@x); \overrightarrow{x = \varepsilon \hookrightarrow y.\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(P_\varepsilon)})$$

To define coercion $r \rightsquigarrow r'$ in path types, we thus require composition $r \rightsquigarrow r'$ in all types.

Finally, consider homogeneous composition in $\mathsf{Path}_{x.A}(P_0, P_1)$:

$$\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{Path}_{x.A}(P_0,P_1)}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) := \langle x\rangle \mathsf{hcom}^{r\rightsquigarrow r'}_{A}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow \_.P_\varepsilon}, \overrightarrow{\xi_i \hookrightarrow y.N_i@x})$$

If we omitted the equations $(x = 0), (x = 1)$ in the definiens, its $\langle \varepsilon/x\rangle$ faces would be $\mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle \varepsilon/x\rangle}(P_\varepsilon; \overrightarrow{\xi_i \hookrightarrow \_.P_\varepsilon})$ and not P_ε as required. Therefore, to define composition of shape $\xi_1, \dots, \xi_n$ in path types, we require composition of shape $\xi_1, \dots, \xi_n, (x = 0), (x = 1)$.

Coercion $r \rightsquigarrow r'$ and composition $r \rightsquigarrow r'$ of shape $(x_1 = 0), (x_1 = 1), \dots, (x_n = 0), (x_n = 1)$ suffice to define dependent function, dependent pair, and path types [AHW17].

[9] One must also verify $\mathsf{coe}^{0\rightsquigarrow 0}_{x.(a:A)\to B}(M) = M$. Simplifying our definition, $\mathsf{coe}^{0\rightsquigarrow 0}_{x.(a:A)\to B}(M) = \lambda a.M\ a$, so the result follows by the (definitional) uniqueness rule for dependent function types.

As we will see in Chapter 4, univalent universes require also (1) equations corresponding to diagonals $x = z$, introduced by Angiuli, Hou (Favonia), and Harper [AFH17], and (2) closure of composition shapes under the deletion of equations that depend on $x : \mathbb{I}$, an operation named "$\forall x$" by Cohen et al. [CCHM18].

At the time of writing, two prototype proof assistants implement Cartesian cubical type theory. The first, **RedPRL** [Red16], implements a computational type theory—in the sense of Section 2.4—with both equality types (as in Chapter 2) and path types. The second, **redtt** [Red18], implements an intensional type theory—in the sense of Section 2.4—and currently lacks equality types. (With Anders Mörtberg, the author has also implemented `yacctt` (`https://github.com/mortberg/yacctt`), a defunct experimental type-checker for intensional Cartesian cubical type theory.)

3.3 De Morgan cubes

In 2016, while investigations into Cartesian cubical type theory were ongoing, Cohen et al. [CCHM18] discovered that one can substantially simplify the Cartesian uniform Kan operation at the cost of adding further operations to the cube category, leading them to define a univalent De Morgan cubical type theory with canonicity [Hub18] and a denotational semantics in De Morgan cubical sets [CCHM18; OP16; Lic+18].

Definition 3.3. The *De Morgan cube category* $\square_{\mathbf{DM}}$ has as objects finite sets, and as morphisms $I \longrightarrow J$ set-theoretic functions $J \longrightarrow \mathbf{DM}(I)$, where $\mathbf{DM}(I)$ is the free De Morgan algebra on I, with the evident Kleisli composition [CCHM18]. (A De Morgan algebra has binary operations $\wedge, \vee$ and constants $0, 1$ forming a bounded distributive lattice, and an involution $\neg$ satisfying $\neg(r \vee r') = \neg r \wedge \neg r'$ and $\neg(r \wedge r') = \neg r \vee \neg r'$.)

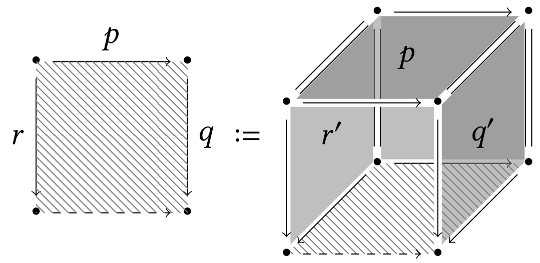
De Morgan cubical type theory augments Cartesian cubical type theory with three interval term formers: *connections* $r \wedge r'$ and $r \vee r'$ which compute the minimum and maximum of r, r' respectively, and *reversals* $\neg r$ which compute $1 - r$. Geometrically, connections provide an alternative form of degeneracy that is constant on two *non-opposing* faces, and reversals exchange the endpoints of one interval. (See Figure 3.3.)

These connections and reversals allow Cohen et al. [CCHM18] to consider a uniform Kan operation limited to $0 \rightsquigarrow 1$ and shapes generated by conjunctions and disjunctions of $(x_i = 0)$ and $(x_i = 1)$. Figure 3.6 describes their *heterogeneous* composition operation for cubically-dependent types, which encompasses both homogeneous composition (when A is constant in $y : \mathbb{I}$) and coercion (when there are no ξ_i). It is often easier to construct instances of comp than of the Cartesian hcom and coe—unlike hcom and coe, the source and target $r \rightsquigarrow r'$ of comp can never become equal, so comp is subject only to one equation ("under ξ_i") and not two ("under $\langle r/y \rangle$").

$$\frac{\begin{array}{ll} & \Gamma, y:\mathbb{I} \vdash A : \mathsf{Type}_i \\ & \Gamma \vdash M : A\langle 0/y\rangle \\ (\forall i) & \Gamma,\xi_i, y:\mathbb{I}\vdash N_i : A \\ (\forall i,j) & \Gamma,\xi_i,\xi_j,y:\mathbb{I}\vdash N_i = N_j : A \\ (\forall i) & \Gamma,\xi_i\vdash N_i\langle 0/y\rangle = M : A\langle 0/y\rangle\end{array}}{\begin{array}{l}\Gamma\vdash \mathsf{comp}_{y.A}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}) : A\langle 1/y\rangle \\ \qquad = N_i\langle 1/y\rangle \quad \text{under } \xi_i\end{array}}$$

Figure 3.6: De Morgan cubical Kan operation.

Surprisingly, many of the constructions described in Section 3.2 can be recovered in the De Morgan setting. For instance, forward coercion $0 \rightsquigarrow 1$ in $x:\mathbb{I}\vdash (a{:}A)\to B : \mathsf{Type}_i$ previously required backward coercion $1 \rightsquigarrow 0$ in $x:\mathbb{I}\vdash A : \mathsf{Type}_i$, but here we can simply coerce (forward) in $x:\mathbb{I}\vdash A\langle\neg x/x\rangle : \mathsf{Type}_i$.[10] As before, $\mathsf{J}_{\square}$ (with propositional computation rule) follows from $\mathsf{transport}_{\square}$ and $\mathsf{uniqueness}_{\square}$, but the latter is now a direct consequence of connections: $\mathsf{uniqueness}_{\square}\ a\ \langle b,p\rangle := \langle x\rangle\langle p@x, \langle y\rangle p@(x\wedge y)\rangle$. Finally, although composition $0 \rightsquigarrow 1$ directly provides only the "missing face" of a filler and not its interior, one can recover filling from composition and connections in an ingenious fashion:



That is, we can define the filler of p, q, and r as the missing bottom face of an open box whose top and back faces are p degenerated, and whose left and right faces are connections of r and q, that is, squares whose top and back faces are degenerate and whose front and bottom faces are r and q respectively.

Conversely, in the Cartesian setting, one can define *weak* connections and reversals that have the correct faces and diagonals but do not satisfy the equations of a De Morgan algebra; for example, reversing a path twice is only propositionally an involution.

De Morgan cubical type theory is the basis of a recent cubical extension to the Agda proof assistant [VMA19]; earlier, Cohen, Coquand, Huber, and Mörtberg implemented it in an experimental type-checker `cubicaltt` (`https://github.com/mortberg/cubicaltt`).

[10] In fact, Cohen et al. [CCHM18] need reversals only for backward composition, and implicitly also define a *distributive lattice* cubical type theory that replaces reversals by $\mathsf{comp}^{0\rightsquigarrow 1}_{y.A}$ and $\mathsf{comp}^{1\rightsquigarrow 0}_{y.A}$.

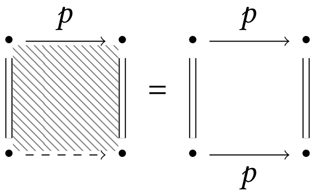
3.4 *Regularity*

In cubical type theories, paths in A correspond to open elements of A. Path types are therefore quite well-behaved: they are automatically dependent (in the sense of Book HoTT [UF13, Section 2.3]), and commute appropriately with other type formers, including dependent function types:

$$\mathsf{funext} : ((a{:}A) \to \mathsf{Path}_{\_.B}(f\ a, g\ a)) \to \mathsf{Path}_{\_.(a:A)\to B}(f, g)$$
$$\mathsf{funext}\ h := \langle x\rangle \lambda a.(h\ a)@x$$

One of their few drawbacks is that path types do not strictly satisfy the rules of identity types, which would allow cubical type theory to import proofs from Book HoTT in a natural way, sending identity types to path types.

Recall our cubical definition of $\mathsf{J}_{\square}$ in Section 3.2. To validate the definitional computation rule of identity types, we seem to require $\mathsf{transport}_{\square}\ C\ (\langle\_\rangle b)\ c$ to be the identity function on $C\ b$, and $\mathsf{uniqueness}_{\square}\ a\ \langle a, \langle\_\rangle a\rangle$ to be a degenerate path at $\langle a, \langle\_\rangle a\rangle$. In Cartesian cubical type theory, the former holds if coercion in degenerate types ($x : \mathbb{I} \vdash C : \mathsf{Type}_i$ such that $C = C\langle 0/x\rangle : \mathsf{Type}_i$) is constant, and the latter holds if any open box whose specified ξ_i sides are degenerate has a degenerate filler:



(In De Morgan cubical type theory, $\mathsf{uniqueness}_{\square}\ a\ \langle a, \langle\_\rangle a\rangle$ is reflexive when defined with connections, but $\mathsf{transport}_{\square}\ C\ (\langle\_\rangle b)\ c$ is not the identity.)

These conditions on coercion and composition, jointly known as *regularity*, were discovered by several researchers, including Coquand[11] and Awodey [Awo18]. Early investigations into cubical type theory focused on defining regular uniform Kan operations closed under the type formers of Book HoTT. In April 2015, Cohen, Coquand, Huber, and Mörtberg circulated an early version of De Morgan cubical type theory apparently satisfying regularity. In May 2015, Licata, Harper, Morehouse, and the author discovered an error in their treatment of universes, which Cohen et al. corrected in July 2015 at the cost of abandoning regularity.

Regularity for universes All open boxes in a universe must have fillers. Because those fillers are elements of a universe, they must *themselves* be types. Consider the bottom

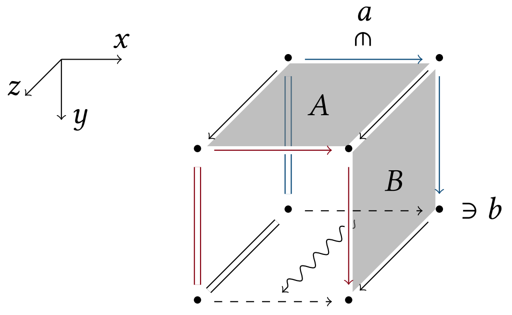
[11] See the note at `http://www.cse.chalmers.se/~coquand/comp.pdf`, dated November 8, 2014.

face of a filler whose other faces are types A, B, and C (below, right). Coercion in the filler's y-coordinate requires this type to be equivalent to A. However, this type is not *equal* to A, because its $\langle 0/x\rangle$ and $\langle 1/x\rangle$ faces are $C\langle 1/y\rangle$ and $B\langle 1/y\rangle$. Instead, we regard it as a new type whose elements are triples of $a : A$, $b : B\langle 1/y\rangle$, and $c : C\langle 1/y\rangle$ such that $a\langle 0/x\rangle = \mathrm{coe}^{1\rightsquigarrow 0}_{y.C}(c) : A\langle 0/x\rangle$ and $a\langle 1/x\rangle = \mathrm{coe}^{1\rightsquigarrow 0}_{y.B}(b) : A\langle 1/x\rangle$:

$$\begin{tikzcd}
{} \arrow[r, "x"] \arrow[d, "y"'] & {} & \mathrm{coe}^{1\rightsquigarrow 0}_{y.C}(c) \arrow[r, "a"] & \mathrm{coe}^{1\rightsquigarrow 0}_{y.B}(b) & & \bullet \arrow[r, "A"] \arrow[d, "C"'] & \bullet \arrow[d, "B"] \\
{} & & c \arrow[r, dashed] & b & : & \bullet \arrow[r, dashed] & \bullet
\end{tikzcd}$$

This type must moreover admit a Kan operation, which failed to be regular in the April 2015 draft of De Morgan cubical type theory. It remains unknown how to equip this type with a regular Kan operation, and a semantic analysis by Swan [Swa18b] proves regularity is in fact impossible to achieve in certain classes of models. A representative case of the difficulty with regularity follows.

Suppose A, B, and C vary also in $z : \mathbb{I}$; for simplicity, let C be degenerate in $y : \mathbb{I}$. Coercion $0 \rightsquigarrow 1$ in the composite type's z-coordinate takes as input $x : \mathbb{I} \vdash a : A\langle 0/z\rangle$ and $b : B\langle 0/z\rangle\langle 1/y\rangle$ such that $a\langle 1/x\rangle = \mathrm{coe}^{1\rightsquigarrow 0}_{y.B\langle 0/z\rangle}(b) : B\langle 0/z\rangle\langle 0/y\rangle$, and must produce $x : \mathbb{I} \vdash a' : A\langle 1/z\rangle$ and $b' : B\langle 1/z\rangle\langle 1/y\rangle$ such that $a'\langle 1/x\rangle = \mathrm{coe}^{1\rightsquigarrow 0}_{y.B\langle 1/z\rangle}(b') : B\langle 1/z\rangle\langle 0/y\rangle$:



By definition, b' is the $\langle 1/x\rangle$ face of our coercion. On the other hand, the $\langle 1/x\rangle$ face of the type in which we coerce equals $B\langle 1/y\rangle$, so the $\langle 1/x\rangle$ face of our coercion must equal $\mathrm{coe}^{0\rightsquigarrow 1}_{z.B\langle 1/y\rangle}(b)$ by commuting coercion and substitution. We therefore define $b' := \mathrm{coe}^{0\rightsquigarrow 1}_{z.B\langle 1/y\rangle}(b)$.

It is tempting to define $x : \mathbb{I} \vdash a' : A\langle 1/z\rangle$ as $\mathsf{coe}^{0\rightsquigarrow 1}_{z.A}(a)$, but this definition does not necessarily satisfy $a'\langle 1/x\rangle = \mathsf{coe}^{1\rightsquigarrow 0}_{y.B\langle 1/z\rangle}(b') : B\langle 1/z\rangle\langle 0/y\rangle$ as required:

$$\begin{aligned} a'\langle 1/x\rangle &= \mathsf{coe}^{0\rightsquigarrow 1}_{z.A\langle 1/x\rangle}(a\langle 1/x\rangle) = \mathsf{coe}^{0\rightsquigarrow 1}_{z.A\langle 1/x\rangle}(\mathsf{coe}^{1\rightsquigarrow 0}_{y.B\langle 0/z\rangle}(b)) \\ &\neq \mathsf{coe}^{1\rightsquigarrow 0}_{y.B\langle 1/z\rangle}(b') = \mathsf{coe}^{1\rightsquigarrow 0}_{y.B\langle 1/z\rangle}(\mathsf{coe}^{0\rightsquigarrow 1}_{z.B\langle 1/y\rangle}(b)) \end{aligned}$$

That is, coercing b up then forward may not be the same as coercing it forward then up. There is, however, a path between those coercions, which we obtain by coercing b up to $B\langle 0/z\rangle$, then coercing forward along all of B:

$$\begin{tikzcd} \mathsf{coe}^{0\rightsquigarrow 1}_{z.B\langle 0/y\rangle}(\mathsf{coe}^{1\rightsquigarrow 0}_{y.B\langle 0/z\rangle}(b)) \arrow[r, "\mathsf{coe}^{y\rightsquigarrow 0}_{y.B\langle 1/z\rangle}(\mathsf{coe}^{0\rightsquigarrow 1}_{z.B}(\mathsf{coe}^{1\rightsquigarrow y}_{y.B\langle 0/z\rangle}(b)))"] & \mathsf{coe}^{1\rightsquigarrow 0}_{y.B\langle 1/z\rangle}(\mathsf{coe}^{0\rightsquigarrow 1}_{z.B\langle 1/y\rangle}(b)) \end{tikzcd}$$

Define a' as the composite of $\mathsf{coe}^{0\rightsquigarrow 1}_{z.A}(a)$ and the above adjustment path.

We have just defined coercion in the composite of A and B, but our definition does not satisfy regularity. Suppose the composite type (and hence also A and B) is degenerate in $z : \mathbb{I}$; then regularity requires our coercion to be the identity function (that is, $a' = a$ and $b' = b$). Assuming regularity holds for coercion in A and B, we have $b' = b$ and $\mathsf{coe}^{0\rightsquigarrow 1}_{z.A}(a) = a$, but the adjustment path above requires coercion in $y : \mathbb{I}$ and thus only simplifies to $\mathsf{coe}^{y\rightsquigarrow 0}_{y.B}(\mathsf{coe}^{1\rightsquigarrow y}_{y.B}(b))$. Therefore $a' \neq a$, and regularity fails for our definition.

Regularity for other types On the other hand, regular uniform Kan operations *are* straightforwardly closed under dependent function, dependent pair, and path types, and in fact, regularity simplifies the definition of composition in $\mathsf{Path}_{x.A}(P_0, P_1)$ that we presented in Section 3.2. Unlike in ordinary cubical type theory, we can define:

$$\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{Path}_{x.A}(P_0,P_1)}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) := \langle x\rangle \mathsf{hcom}^{r\rightsquigarrow r'}_{A}(M@x; \overrightarrow{\xi_i \hookrightarrow y.N_i@x})$$

The $\langle \varepsilon/x\rangle$ faces of the definiens are $\mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle \varepsilon/x\rangle}(P_\varepsilon; \overrightarrow{\xi_i \hookrightarrow \_.P_\varepsilon})$, which by regularity equal P_ε (because P_0, P_1 are constant in $y : \mathbb{I}$).

To obtain a cubical type theory closed under dependent function, dependent pair, and path types that strictly model identity types, it suffices to equip each type with regular coercion $r \rightsquigarrow r'$ and regular composition $r \rightsquigarrow r'$ of shape $(x = 0), (x = 1)$ *only.*[12] (We still need compositions of such shape to define $\mathsf{J}_{\square}$ and coercion in path types, but composition in path types no longer alters composition shapes.) Awodey [Awo18] models identity types in Cartesian cubical sets by a similar observation.

[12] Sterling, Angiuli, and Gratzer [SAG19] use exactly these Kan operations in a cubical type theory with non-univalent universes and path types satisfying uniqueness of identity proofs.

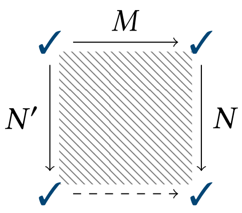
3.5 *Validity and other considerations*

Regularity is one of many subtleties in defining a uniform Kan operation. Another is: do we permit 0-dimensional filling problems? Cohen et al. [CCHM18] answer in the affirmative, allowing compositions of empty shape $\cdot \vdash \mathsf{comp}_{y.A}(M;\cdot) : A\langle 1/y\rangle$ for $\cdot \vdash M : A\langle 0/y\rangle$. Such compositions simplify only according to type-specific principles analogous to those in Section 3.2, because the sole type-generic equation governing comp (Figure 3.6) does not apply to empty shapes. (If comp were regular, then $\mathsf{comp}_{\_.A}(M;\cdot) = M$ would hold generically.) For a higher inductive type $\mathbb{T}$, $\mathsf{comp}_{\_.\mathbb{T}}(M;\cdot)$ *never* simplifies, because $\mathbb{T}$'s elimination principle stipulates that $\mathbb{T}$ is the least uniform Kan type with given constructors.

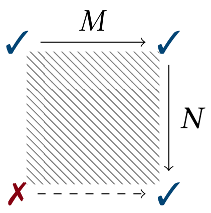
Perhaps surprisingly, empty compositions in higher inductive types pose major difficulties in `cubicaltt`, where closed terms of higher inductive type often compute to values with *hundreds* of nested empty compositions $\mathsf{comp}_{\_.\mathbb{T}}(\mathsf{comp}_{\_.\mathbb{T}}(\cdots;\cdot);\cdot)$. Such empty compositions partially negate the benefits of computation. Imagine, for instance, computing a closed element of a set quotient A/R [UF13, Section 6.10]. In `cubicaltt`, such an element's value may contain arbitrarily many empty compositions; if we could disallow empty compositions, its value would always be the injection $\mathsf{in}(M)$ of an element of A.

We contribute a method for disallowing empty compositions, allowing us to obtain a strong canonicity theorem for 0-cubes of higher inductive type (Theorem 4.77); recently, Vezzosi, Mörtberg, and Abel [VMA19] have developed such a method in the De Morgan setting in cubical Agda. To understand our approach, recall that uniform Kan operations must commute with interval substitutions. One must therefore prohibit not only empty compositions but also all compositions with empty substitution instances. For example, although $\mathsf{hcom}_A^{0\rightsquigarrow 1}(M; x = 1 \hookrightarrow y.N)$ is not empty, its $\langle 0/x\rangle$ face has shape $(0 = 1)$, which is either empty or equally problematic, because it is unsatisfiable.

We thus restrict hcom to shapes $\xi_1, \dots, \xi_n$ all of whose closed instantiations contain at least one true clause—that is, for which $(\forall x_1, \dots, x_m \in \{0,1\}.\xi_1 \vee \cdots \vee \xi_n)$ holds. Geometrically, such open boxes cover all 0-cells of their fillers (below, left but not right):



$\mathsf{hcom}_A^{0\rightsquigarrow y}(M; x = 1 \hookrightarrow y.N, x = 0 \hookrightarrow y.N')$



$\mathsf{hcom}_A^{0\rightsquigarrow y}(M; x = 1 \hookrightarrow y.N)$

In Chapter 4 we underapproximate the above condition, requiring simply that *valid* composition shapes contain either a true clause or a pair of clauses $(x = 0), (x = 1)$ for some x (Definition 4.28).

Following the observations of Section 3.2, valid homogeneous composition $r \rightsquigarrow r'$ and coercion $r \rightsquigarrow r'$ suffice to define dependent function, dependent pair, and path types. Univalent universes are more challenging, as the aforementioned $\forall x$ operation [CCHM18] produces invalid shapes from valid ones, for example, when deleting all x clauses from the shape $(x = 0), (x = 1), (z = 1)$. In such cases we use a derived operation called *generalized* homogeneous composition, written $\mathsf{ghcom}_A^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})$, which fills boxes of possibly-invalid shape by iterated valid fillers (Theorem 4.34).[13] Our generalized compositions compute not according to their type but rather the length of $\xi_1, \dots, \xi_n$; moreover, although generalized compositions satisfy the rules of Figure 3.4, they do not equal the corresponding hcoms if applied to valid shapes.

Equality of compositions Our discussion of validity raises another question: when are two compositions equal? Many syntactically distinct filling problems have equal extensions as open boxes, including:

$$\begin{aligned}
\mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \xi \hookrightarrow y.N, 0 = 1 \hookrightarrow y.N') &= \mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \xi \hookrightarrow y.N) \\
\mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \xi \hookrightarrow y.N, \xi' \hookrightarrow y.N') &= \mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \xi' \hookrightarrow y.N', \xi \hookrightarrow y.N) \\
\mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \xi \hookrightarrow y.N, \xi \hookrightarrow y.N) &= \mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \xi \hookrightarrow y.N) \\
\mathsf{hcom}_A^{r \rightsquigarrow r'}(M; s = s' \hookrightarrow y.N) &= \mathsf{hcom}_A^{r \rightsquigarrow r'}(M; s' = s \hookrightarrow y.N)
\end{aligned}$$

The first three are definitional equalities of De Morgan cubical type theory, where they are used in the definition of identity types [CCHM18, Section 9.1]. (The fourth is moot because Cohen et al. [CCHM18] consider only easily-oriented equations $(x = 0), (x = 1)$.) We adopt none of the above equations in this dissertation; Cavallo and Harper [CH19a] provide an alternative definition of identity types not reliant on these equations. Aside from identity types, such equations seem expendable in practice, and complicate both implementations and our definition of generalized composition.

Readers often ask whether coercions satisfy the equation:

$$\mathsf{coe}_{x.A}^{r' \rightsquigarrow r''}(\mathsf{coe}_{x.A}^{r \rightsquigarrow r'}(M)) = \mathsf{coe}_{x.A}^{r \rightsquigarrow r''}(M)$$

In fact, this equation implies that every path in Type_i induces a strict isomorphism between its endpoints (by $\lambda a.\mathsf{coe}_{x.p@x}^{0 \rightsquigarrow 1}(a)$ and $\lambda a.\mathsf{coe}_{x.p@x}^{1 \rightsquigarrow 0}(a)$) rather than an equivalence, ruling out many instances of univalence. These coercions are nevertheless related by $\mathsf{coe}_{x.A}^{x \rightsquigarrow r''}(\mathsf{coe}_{x.A}^{r \rightsquigarrow x}(M))$, which equals the left one under $\langle r'/x \rangle$ and the right under $\langle r/x \rangle$.

[13] Our construction of univalent universes is only simplified by omitting validity (and thus, generalized composition); see Appendix B or, for an alternate construction, Angiuli et al. [Ang+19].

Compositions of types As suggested in Section 3.4, we regard homogeneous composition of types as a type former in its own right, and must therefore specify its introduction, elimination, computation, and uniqueness principles. For example, an element of $\mathsf{hcom}^{0\rightsquigarrow 1}_{\mathsf{Type}_i}(A; x = 0 \hookrightarrow y.C, x = 1 \hookrightarrow y.B)$, written $\mathsf{box}^{0\rightsquigarrow 1}(a; x = 0 \hookrightarrow c, x = 1 \hookrightarrow b)$, consists of an element $a : A$ and two partial elements $(x = 0) \vdash c : C\langle 1/y\rangle$ and $(x = 1) \vdash b : B\langle 1/y\rangle$ satisfying $(x = 0) \vdash a = \mathsf{coe}^{1\rightsquigarrow 0}_{y.C}(c) : A$ and $(x = 1) \vdash a = \mathsf{coe}^{1\rightsquigarrow 0}_{y.B}(b) : A$.

$$\begin{tikzcd}
{} \arrow[r, "x"] \arrow[d, "y"'] & {} & \mathsf{coe}^{1\rightsquigarrow 0}_{y.C}(c) \arrow[r, "a"] & \mathsf{coe}^{1\rightsquigarrow 0}_{y.B}(b) & & \bullet \arrow[r, "A"] \arrow[d, "C"'] & \bullet \arrow[d, "B"] \\
{} & & c \arrow[r, dashed] & b & : & \bullet \arrow[r, dashed] & \bullet
\end{tikzcd}$$

This introduction principle is far from arbitrary. If $x = 1$, the composite type must equal $B\langle 1/y\rangle$; accordingly, if $x = 1$, the above data specifies precisely an element of $B\langle 1/y\rangle$—the partial element c is vacuous because $x = 0$ is impossible,[14] and a must be exactly $\mathsf{coe}^{1\rightsquigarrow 0}_{y.B}(b)$ because $x = 1$ is tautologous. Coercion in $y.\mathsf{hcom}^{0\rightsquigarrow y}_{\mathsf{Type}_i}(A; x = 0 \hookrightarrow y.C, x = 1 \hookrightarrow y.B)$ implies the composite type is equivalent to A, which is why boxes include $a : A$ (as well as its equations with b and c, lest the type be equivalent instead to $A \times C\langle 1/y\rangle \times B\langle 1/y\rangle$). Conversely, if composite types do not admit coercion, they need not be equivalent to A; in Section 4.4.10 we define compositions of non-Kan types to be empty.

Finally, to compose and coerce in composite types, we must access the components of boxes. We easily obtain b and c as the $\langle 1/x\rangle$ and $\langle 0/x\rangle$ faces of the box above, and we compute a using cap, the elimination form of composite types. In general, eliminating an element of $\mathsf{hcom}^{0\rightsquigarrow 1}_{\mathsf{Type}_i}(A; x = 0 \hookrightarrow y.C, x = 1 \hookrightarrow y.B)$ produces an element of A. The cap of a box is simply its $a : A$ component; if $x = 1$, the input to cap is not a box but an element of $B\langle 1/y\rangle$, and cap instead applies $\mathsf{coe}^{1\rightsquigarrow 0}_{y.B}(-)$. (These computations agree by $(x = 1) \vdash a = \mathsf{coe}^{1\rightsquigarrow 0}_{y.B}(b) : A$.) See Section 4.4.11 for details.

V-types If Type_i is univalent, we have a function:

$$\mathsf{ua}_{🗇} : (A\ B{:}\mathsf{Type}_i) \to ((f{:}A \to B) \times \mathsf{IsEquiv}(f)) \to \mathsf{Path}_{\_.\mathsf{Type}_i}(A, B)$$

Applying elimination rules, $(\mathsf{ua}_{🗇}\ A\ B\ \langle f, e\rangle)@x$ must therefore be a type with extent in $x : \mathbb{I}$. In this dissertation, we regard this type as a new type former written $\mathsf{V}_x(A, B, \langle f, e\rangle)$, equal to A when $x = 0$ and B when $x = 1$.

What is the formation principle of $\mathsf{V}_x(A, B, \langle f, e\rangle)$? In particular, what do occurrences of x mean in A, B, f, e? (No such occurrences exist in our motivating example above.) We

[14] We discuss judgments under interval equations in Section 4.3.2.

cannot insist that x does not occur, because formation would not be closed under diagonals—typehood of $\mathsf{V}_x(A, B, \langle f, e\rangle)$ would not imply typehood of $(\mathsf{V}_x(A, B, \langle f, e\rangle))\langle y/x\rangle$ when y occurs in A, B, f, e.[15] Nor can we ask that A and B be disjoint partial types $(x = 0) \vdash A : \mathsf{Type}_i$ and $(x = 1) \vdash B : \mathsf{Type}_i$, because we would not be able to form $A \to B$.

Instead, we require types $(x = 0) \vdash A : \mathsf{Type}_i$ and $B : \mathsf{Type}_i$ and an equivalence $(x = 0) \vdash \langle f, e\rangle : (f{:}A \to B) \times \mathsf{IsEquiv}(f)$. That is, given a type B and a partial type A equivalent to $B\langle 0/x\rangle$ under $x = 0$, we obtain a line $\mathsf{V}_x(A, B, \langle f, e\rangle)$ between A and $B\langle 1/x\rangle$, effectively composing the equivalence f and the type B:

$$\begin{tikzcd}
\xrightarrow{x} & f\ a \arrow[r, "b"] & \bullet & & \bullet \arrow[r, "B"] & \bullet \\
& a \arrow[ur, dashed] & & : & A \arrow[u, dotted, "f"] \arrow[ur, dashed] &
\end{tikzcd}$$

By analogy with composite types, an element of $\mathsf{V}_x(A, B, \langle f, e\rangle)$, written $\mathsf{Vin}_x(a, b)$, consists of $b : B$ and a partial element $(x = 0) \vdash a : A$ satisfying $(x = 0) \vdash f\ a = b : B$; $\mathsf{Vin}_x(a, b)$ equals a when $x = 0$ and $b\langle 1/x\rangle$ when $x = 1$. The elimination form for $\mathsf{V}_x(A, B, \langle f, e\rangle)$, written $\mathsf{Vproj}_x(-, f)$, produces an element of B. Applied to $\mathsf{Vin}_x(a, b)$, Vproj returns b; if $x = 0$, the argument of Vproj is an element of A instead, and Vproj applies f. See Section 4.4.9 for details.

V-types and composite types arise as special cases of the *Glue types* of Cohen et al. [CCHM18], in which a single type is composed with any configuration of partial types equivalent to it. (One obtains composite types by using coercion to transform each $y.B_i$ into an equivalence.) Angiuli et al. [Ang+19] develop Glue types in the Cartesian setting; we do not discuss them further in this dissertation.

Extension types We consider several variations on path types in **RedPRL** and **redtt**. Elements of *line types* are dependent functions from $\mathbb{I}$ with arbitrary endpoints:

$$\frac{\Gamma, x : \mathbb{I} \vdash A : \mathsf{Type}_i}{\Gamma \vdash (x{:}\mathbb{I}) \to A : \mathsf{Type}_i} \qquad \frac{\Gamma, x : \mathbb{I} \vdash M : A}{\Gamma \vdash \langle x\rangle M : (x{:}\mathbb{I}) \to A} \qquad \frac{\Gamma \vdash M : (x{:}\mathbb{I}) \to A}{\Gamma \vdash M@r : A\langle r/x\rangle}$$

Elements of a *restriction type* $A\,[\overrightarrow{\xi_i \hookrightarrow N_i}]$ are elements of A that equal N_i when ξ_i:

$$\frac{\begin{array}{ll} & \Gamma \vdash A : \mathsf{Type}_k \\ (\forall i) & \Gamma, \xi_i \vdash N_i : A \\ (\forall i, j) & \Gamma, \xi_i, \xi_j \vdash N_i = N_j : A \end{array}}{\Gamma \vdash A\,[\overrightarrow{\xi_i \hookrightarrow N_i}] : \mathsf{Type}_k} \qquad \frac{\begin{array}{ll} & \Gamma \vdash M : A \\ (\forall i) & \Gamma, \xi_i \vdash M = N_i : A \end{array}}{\Gamma \vdash \mathsf{in}(M) : A\,[\overrightarrow{\xi_i \hookrightarrow N_i}]} \qquad \frac{\Gamma \vdash M : A\,[\overrightarrow{\xi_i \hookrightarrow N_i}]}{\begin{array}{c}\Gamma \vdash \mathsf{out}(M) : A \\ = N_i \ \text{ under } \xi_i\end{array}}$$

[15] This definition, known as the G-type, is possible in the symmetric monoidal setting [BCH18].

(One can avoid the bureaucracy of explicit in/out coercions by considering $A\,[\overrightarrow{\xi_i \hookrightarrow N_i}]$ a subtype of A; similarly, one can consider $\mathsf{Path}_{x.A}(M,N)$ a subtype of $(x{:}\mathbb{I}) \to A$.)

We can recover path types as a combination of line and restriction types:

$$\mathsf{Path}_{x.A}(a,a') := (x{:}\mathbb{I}) \to A\,[x=0 \hookrightarrow a, x=1 \hookrightarrow a']$$

However, the flexibility of line and restriction types makes them more ergonomic than path types. Consider binary path composition, which takes three points $a,b,c : A$ and two paths $\mathsf{Path}_{\_.A}(a,b)$ and $\mathsf{Path}_{\_.A}(b,c)$. Using line types, we can eliminate the first three arguments a,b,c, instead stating that composition takes a line $p : \mathbb{I} \to A$ and a line $q : (x{:}\mathbb{I}) \to A\,[x=0 \hookrightarrow p@1]$ whose left endpoint equals the right endpoint of p:

$$\begin{aligned} \mathsf{composition} &: (a\;b\;c{:}A) \to \mathsf{Path}_{\_.A}(a,b) \to \mathsf{Path}_{\_.A}(b,c) \to \mathsf{Path}_{\_.A}(a,c) \\ \mathsf{composition}' &: (p{:}\mathbb{I} \to A) \to (q{:}(x{:}\mathbb{I}) \to A\,[x=0 \hookrightarrow p@1]) \to \mathsf{Path}_{\_.A}(p@0, q@1) \end{aligned}$$

Another example is the type of squares bounded by four paths $p : \mathsf{Path}_{\_.A}(a,b)$, $q : \mathsf{Path}_{\_.A}(a,c)$, $r : \mathsf{Path}_{\_.A}(b,d)$, and $s : \mathsf{Path}_{\_.A}(c,d)$, typically defined as the iterated path type $\mathsf{Path}_{x.\mathsf{Path}_{\_.A}(p@x,s@x)}(q,r)$. We can express the geometry of these constraints more directly using line and restriction types:

$$(x\;y{:}\mathbb{I}) \to A\,[x=0 \hookrightarrow q@y, x=1 \hookrightarrow r@y, y=0 \hookrightarrow p@x, y=1 \hookrightarrow s@x]$$

If $x : \mathbb{I} \vdash A : \mathsf{Type}_i$ is Kan then $(x{:}\mathbb{I}) \to A$ is Kan. The same is not true for restriction types, as that would imply that all paths exist:

$$\langle x\rangle \mathsf{out}(\mathsf{coe}^{0 \rightsquigarrow x}_{x.A\,[x=0\hookrightarrow\mathsf{true},x=1\hookrightarrow\mathsf{false}]}(\mathsf{in}(\mathsf{true}))) : \mathsf{Path}_{\_.\mathsf{bool}}(\mathsf{true},\mathsf{false})$$

Fortunately, $(\overrightarrow{x_i}{:}\mathbb{I}) \to A\,[\overrightarrow{\xi_j \hookrightarrow N_j}]$ is Kan whenever $\overrightarrow{\xi_j}$ is empty or valid, and moreover mentions only interval variables in $\overrightarrow{x_i}$—that is, whenever line types bind all the variables involved in a restriction type. The latter condition is satisfied by path types, composition′, square types, and many other examples.

For that reason, we consider $(\overrightarrow{x_i}{:}\mathbb{I}) \to A\,[\overrightarrow{\xi_j \hookrightarrow N_j}]$ to be a primitive type former known as *extension types*, following Riehl and Shulman [RS17], who first introduced them. The Kan operations for extension types are very similar to those of path types (Section 3.2), adjusting homogeneous compositions by $\overrightarrow{\xi_j \hookrightarrow \_.N_j}$ and coercions by $\overrightarrow{\xi_j \hookrightarrow y.\mathsf{coe}^{y \rightsquigarrow r'}_{y.A}(N_j)}$.

The **redtt** proof assistant currently supports extension types, and uses them to reify boundary constraints on subgoals. For example, if a user writes the partial term:

$$\mathsf{hcom}^{0 \rightsquigarrow 1}_{A}(M; x=0 \hookrightarrow y.N, x=1 \hookrightarrow ?)$$

redtt displays $(y{:}\mathbb{I}) \to A\,[y=0 \hookrightarrow M, 1=0 \hookrightarrow \_]$ for the type of the hole ?. Cubical Agda currently supports line and restriction types (the latter via “partial cubical types”) but not Kan extension types, nor does it formulate subgoals using extension types [VMA19, Section 3.3]. We anticipate these features will be implemented in the future.

4

Cartesian cubical type theory

Now, it is the contention of the intuitionists (or constructivists, I shall use these terms synonymously) that the basic mathematical notions, above all the notion of function, ought to be interpreted in such a way that the cleavage between mathematics (classical mathematics, that is) and programming that we are witnessing at present disappears.

—Per Martin-Löf,
Constructive mathematics and computer programming [ML82]

This chapter describes the computational semantics of Cartesian cubical type theory, which extends ordinary constructive type theory with path, circle, and univalent universe types. Like De Morgan cubical type theory [CCHM18], Cartesian cubical type theory uses interval variables $x : \mathbb{I}$ and uniform Kan operations to mediate the higher-dimensional structure induced by univalence and higher inductive types.

Our computational semantics directly justify *Cartesian cubical computational type theory* (Appendix A), a Nuprl-style type theory implemented in the **RedPRL** proof assistant [Red16]. Cartesian cubical computational type theory is a *two-level type theory*, like Voevodsky's Homotopy Type System [Voe13], containing both *Kan types* equipped with homogeneous composition and coercion operations (as described in Section 3.2), and *pretypes* possibly lacking such operations. As in Chapter 2, we establish consistency (Theorem 4.58) and canonicity (Theorems 4.63 and 4.77) for our computational type theory.

Cartesian cubical computational type theory contains two cumulative hierarchies of universes: $\mathcal{U}_i^{\mathsf{Kan}}$ whose elements are Kan types, and $\mathcal{U}_i^{\mathsf{pre}} \supset \mathcal{U}_i^{\mathsf{Kan}}$ whose elements are pretypes. Both $\mathcal{U}_i^{\mathsf{Kan}}$ and $\mathcal{U}_i^{\mathsf{pre}}$ are Kan types, and each $\mathcal{U}_i^{\mathsf{Kan}}$ is univalent. Our pretypes include strict equality $\mathsf{Eq}_A(M, N)$ with equality reflection; strict equality is not generally Kan, as that would imply that all paths are trivial:

$$\lambda p.\mathsf{transport}_{\square}\,(\lambda b.\mathsf{Eq}_A(a, b))\; p\; \star : \mathsf{Path}_{\_.A}(a, a') \to \mathsf{Eq}_A(a, a')$$

This chapter derives heavily from the author's *Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities* [AFH18] and its associated preprint [AFH17], but incorporates additional exposition and various technical improvements, notably: simpler definitions of coercion in V-types and homogeneous composition in $\mathsf{hcom}_{\mathcal{U}_i^{\mathsf{Kan}}}$ types suggested by Anders Mörtberg (Figure 4.2); a construction of $\mathsf{hcom}_{\mathcal{U}_i^{\mathsf{pre}}}$

types suggested by Andrew Pitts enabling $\mathcal{U}_i^{\mathsf{pre}}$ to be Kan (Section 4.4.10); and new notation for candidate judgments after Chapter 2 (Definition 4.5).

Cavallo and Harper [CH19a] extend our computational semantics with a schema for indexed higher inductive types accounting for many types considered in Book HoTT, including spheres, homotopy pushouts, truncations, and the identity type (with strict computation rule). In a separate line of work, Cavallo and Harper [CH19b] extend our computational semantics with *bridge variables* expressing internal parametricity primitives in the style of Bernardy, Coquand, and Moulin [BCM15], which refute the law of excluded middle for homotopy subsingleton types [UF13, Section 3.4]. We do not present these extensions in this dissertation.

Denotational and intensional formulations Although we analyze Cartesian cubical type theory from the computational perspective, our uniform Kan operation can be formulated for arbitrary Cartesian cubical sets. Angiuli et al. [Ang+19] show that Cartesian cubical sets with such an operation are closed under dependent functions, dependent pairs, univalent universes, *et cetera*, accompanied by an Agda formalization in the axiomatic style developed by Orton and Pitts [OP16] and Licata et al. [Lic+18].

Moreover, just as the computational semantics of Idealized Nuprl also model intensional type theory (in the sense of Section 2.4), we can define a *Cartesian cubical intensional type theory* and regard this chapter as a proof of its consistency.[1] Angiuli et al. [Ang+19] propose one such type theory; we propose another in Appendix B closer to what is implemented in the **redtt** [Red18] proof assistant and described by our computational semantics (namely, with V and $\mathsf{hcom}_{\mathcal{U}_i^{\mathsf{Kan}}}$ type formers instead of Glue types [Ang+19, Section 2.11]).

Neither Cartesian cubical intensional type theory contains strict equality types—although equality reflection is semantically justified and natural in computational type theory, it is undesirable for definitional equality as reflection disrupts its decidability. Altenkirch, Capriotti, and Kraus [ACK16] and Boulier and Tabareau [BT17] have proposed two-level intensional type theories (without canonicity) whose strict equality types are identity types augmented by axioms for function extensionality and uniqueness of identity proofs. More recently, Sterling, Angiuli, and Gratzer [SAG19] proposed XTT, a cubical intensional type theory with canonicity, whose path type satisfies function extensionality and uniqueness of identity proofs. We hope to integrate XTT into **redtt** to obtain strict equality without reflection in an intensional, two-level cubical type theory with canonicity.

Outline Before diving into technical details, we briefly outline the difficulties that arise in this chapter. Cartesian cubical type theory is based on a cubical programming language whose programs contain interval variables that can be instantiated with 0 and 1. As in

[1] Canonicity does *not* follow immediately, as intensional type theories contain only typed equations, but we expect that this chapter and Huber's canonicity proof [Hub18] contain all the necessary ingredients.

Chapter 2, our operational semantics range only over programs with no free term variables, and open judgments express the truth of their conclusion for all closed elements of their hypotheses. However, interval variables cannot stand only for their endpoints 0 and 1, because such an interpretation would equate all lines with equal boundaries and thus imply uniqueness of identity proofs.

We therefore define our cubical operational semantics for programs with free interval variables (but no free term variables), and define types by their PERs of closed value elements *at every dimension*, that is, for every set of interval variables. We ensure that the PERs of elements of each closed type determine a Cartesian cubical set whose restriction maps are given by interval substitution followed by evaluation. (Evaluation is needed because the endpoints of values are not necessarily values.)

We equip every Kan type with *coercion* and *homogeneous Kan composition* operations sufficient for defining transport and singleton contractibility (Section 3.2). To define homogeneous Kan composition (Definition 4.29), we must specify when a collection of n-cubes forms an open box of a given shape. We implement the Kan operations of each type former using operational semantics rules which rely on the Kan operations of constituent types. As outlined in Section 3.4, homogeneous compositions of Kan types are themselves types whose elements are formal boxes of elements of the constituent types. To ensure our universes are univalent, given any types A, B and equivalences E between them, we define a type $\mathsf{V}_x(A, B, E)$ whose endpoints are A and B. These "V-types" are a special case of the "Glue types" of Cohen et al. [CCHM18], and closely related to the "G-types" of Bezem, Coquand, and Huber [BCH18].

Finally, as in Section 2.5, we prove the formation, introduction, elimination, and uniqueness rules for each type former. In the cubical setting, we must also verify that each type is Kan, that is, that its operational semantics rules properly implement its Kan operations. In Section 2.5, we proved most rules using head expansion (Lemma 2.15), by which it suffices to consider a term's evaluation behavior; in Section 4.4, we must often rely instead on *coherent expansion* (Lemma 4.18), which requires us to consider the evaluation behavior of all interval substitution instances of a term.

4.1 Syntax and operational semantics

Cartesian cubical type theory is based on a untyped functional programming language with two sorts: interval terms and ordinary terms. The only interval terms are 0, 1, and *interval variables* $x, y, \dots$; ordinary terms include standard type and term constructors previously seen in Idealized Nuprl, as well as new constructs (for Kan operations, higher inductive types, *et cetera*) with interval term arguments.

We define the syntax of ordinary terms in Figure 4.1. As before, capital letters $M, N, A, \dots$ represent ordinary terms and $a, b, \dots$ represent ordinary term variables and

$M :=$	$\Pi(A, a.B)$	$(a{:}A) \to B$	dependent function type
	$\lambda(a.M)$	$\lambda a.M$	lambda abstraction
	$\mathsf{app}(M, N)$	$M\ N$	function application
	$\Sigma(A, a.B)$	$(a{:}A) \times B$	dependent pair type
	$\mathsf{pair}(M, N)$	$\langle M, N \rangle$	pairing
	$\mathsf{fst}(M)$	$\mathsf{fst}(M)$	first projection
	$\mathsf{snd}(M)$	$\mathsf{snd}(M)$	second projection
	$\mathsf{Eq}(A, M, N)$	$\mathsf{Eq}_A(M, N)$	equality pretype
	refl	$\star$	equality proof
	nat	nat	natural number type
	z	z	zero
	$\mathsf{s}(M)$	$\mathsf{s}(M)$	successor
	$\mathsf{natrec}(M, N_1, n.a.N_2)$	$\mathsf{natrec}(M; N_1, n.a.N_2)$	natural number recursion
	bool	bool	boolean type
	true	true	true
	false	false	false
	$\mathsf{if}(M, N_1, N_2)$	$\mathsf{if}(M; N_1, N_2)$	boolean recursion
	void	void	empty type
	$\mathsf{Path}(x.A, M, N)$	$\mathsf{Path}_{x.A}(M, N)$	path type
	$\mathsf{ilam}(x.M)$	$\langle x \rangle M$	interval abstraction
	$\mathsf{iapp}(M, r)$	$M@r$	interval application
	$\mathsf{V}(r, A, B, E)$	$\mathsf{V}_r(A, B, E)$	weak univalence axiom
	$\mathsf{Vin}(r, M, N)$	$\mathsf{Vin}_r(M, N)$	V-type injection
	$\mathsf{Vproj}(r, M, F)$	$\mathsf{Vproj}_r(M, F)$	V-type projection
	circle	$\mathbb{S}^1$	circle type
	base	base	base point
	$\mathsf{loop}(r)$	loop_r	loop
	$\mathsf{celim}(c.A, M, N_1, x.N_2)$	$\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; N_1, x.N_2)$	circle induction
	$\mathsf{Pre}[i]$	$\mathcal{U}_i^{\mathsf{pre}}$	ith pretype universe
	$\mathsf{Kan}[i]$	$\mathcal{U}_i^{\mathsf{Kan}}$	ith Kan type universe

Figure 4.1: Syntax of Cartesian cubical type theory: basic constructs.

binders. We additionally write $r, s, \ldots$ for interval terms, $x, y, \ldots$ for interval variables and binders, ε for interval constants 0 and 1, and $\mathsf{FI}(M)$ for the set of interval variables free in M. One can substitute ordinary terms into ordinary terms $M[N/a]$, and interval terms into either sort: $M\langle r/x \rangle$ and $s\langle r/x \rangle$.

$\mathsf{coe}(x.A, r, r', M)$	$\mathsf{coe}^{r \rightsquigarrow r'}_{x.A}(M)$	coercion
$\mathsf{hcom}[n](A, r, r', M, \overrightarrow{r_i, r'_i, y.N_i})$	$\mathsf{hcom}^{r \rightsquigarrow r'}_{A}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i})$	homog. comp.
$\mathsf{com}[n](y.A, r, r', M, \overrightarrow{r_i, r'_i, y.N_i})$	$\mathsf{com}^{r \rightsquigarrow r'}_{y.A}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i})$	heterog. comp.
$\mathsf{ghcom}[n](A, r, r', M, \overrightarrow{r_i, r'_i, y.N_i})$	$\mathsf{ghcom}^{r \rightsquigarrow r'}_{A}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i})$	generalized hcom
$\mathsf{gcom}[n](y.A, r, r', M, \overrightarrow{r_i, r'_i, y.N_i})$	$\mathsf{gcom}^{r \rightsquigarrow r'}_{y.A}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i})$	generalized com
$\mathsf{box}[n](r, r', M, \overrightarrow{r_i, r'_i, N_i})$	$\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{r_i = r'_i \hookrightarrow N_i})$	box formation
$\mathsf{cap}[n](r, r', M, \overrightarrow{r_i, r'_i, y.B_i})$	$\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.B_i})$	cap projection

Figure 4.1: Syntax of Cartesian cubical type theory: Kan operations.

The Kan composition operators are indexed by external natural numbers. For instance, $\mathsf{hcom}[n]$ represents homogeneous composition along n equations, and has $3n+4$ arguments, the last $3n$ of which are grouped in triples $r_i, r'_i, y.N_i$ specifying N_i as the $r_i = r'_i$ component of an open box. We write ξ_i for equations $r_i = r'_i$ (formally, pairs of interval terms), and pun ξ_i and $\neg\xi_i$ for the propositions that such an equation holds or fails to hold.

In Figure 4.2, we define a structural operational semantics [Plo81] over terms with free interval variables but no free term variables. The judgment $M_0\ \mathsf{val}$ specifies when M_0 is a value, and $M \longmapsto M'$ specifies when M takes one step of evaluation to M'. We write $M \longmapsto^* M'$ when M steps to M' in zero or more steps, and $M \Downarrow M_0$ when $M \longmapsto^* M_0$ and $M_0\ \mathsf{val}$. Curiously, coercion is the only place where evaluation descends under interval binders, as coe must evaluate its type argument $x.A$ and dispatch on A's head constructor.

These judgments satisfy three key properties easily proven by induction on their definitions. First, if $M\ \mathsf{val}$, then $M \not\longmapsto$. (As before, the converse is not the case.) Secondly, evaluation is deterministic: if $M \longmapsto M'$ and $M \longmapsto M''$, then $M' = M''$. Finally, evaluation never introduces additional free interval variables: if $M \longmapsto M'$, then $\mathsf{FI}(M') \subseteq \mathsf{FI}(M)$.

Finite sets of interval variables and total interval substitutions present the Cartesian cube category described in Definition 3.2.

Definition 4.1. Given finite sets Ψ, Ψ' of interval variables, a *total interval substitution* $\psi : \Psi' \longrightarrow \Psi$ assigns to each element of Ψ either 0, 1, or an element of Ψ'.

Our computational semantics associate to each closed type a Cartesian cubical set whose Ψ-cubes are values M_0 with $\mathsf{FI}(M_0) \subseteq \Psi$, and whose restriction maps are given by interval substitution followed by evaluation. Evaluation is necessary because, crucially, our operational semantics are not *cubically stable*, or stable under interval substitution. (Recall that, in contrast, evaluation in Idealized Nuprl is stable under substitution when lifted to open terms.) They do, however, respect permutations of interval variables, which form a “nominal” [Pit13; Pit15] wide subcategory of the Cartesian cube category.

To see why the operational semantics do not respect interval substitution, consider the circle $\mathbb{S}^1$, a higher inductive type generated by a point base and a line loop_x. The constructors base and loop_x are values; we ensure the faces of loop_x are base by the operational step $(\mathsf{loop}_x)\langle 0/x\rangle = \mathsf{loop}_0 \longmapsto \mathsf{base}$. (Thus $M\ \mathsf{val}$ does not imply $M\psi\ \mathsf{val}$.) Maps out of the circle are determined by a point P (the image of base) and an abstracted line $x.L$ (the image of loop_x). The operational step $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_x; P, x.L) \longmapsto L$ is not stable under interval substitution, because:

$$\begin{aligned}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_x; P, x.L))\langle 0/x\rangle &= \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\langle 0/x\rangle}(\mathsf{loop}_0; P\langle 0/x\rangle, x.L)\\ &\longmapsto \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\langle 0/x\rangle}(\mathsf{base}; P\langle 0/x\rangle, x.L)\\ &\longmapsto P\langle 0/x\rangle\end{aligned}$$

where $P\langle 0/x\rangle$ need not equal $L\langle 0/x\rangle$. (Thus $M \longmapsto M'$ does not imply $M\psi \longmapsto^* M'\psi$.)

Fortunately, many operational semantics rules are in fact cubically stable, including all the rules shared with Idealized Nuprl. In Figure 4.2, many rules are annotated with ⧉, defining an additional pair of judgments $M\ \mathsf{val}_{⧉}$ and $M \longmapsto_{⧉} M'$ by replacing every occurrence of val (resp., $\longmapsto$) in those rules with $\mathsf{val}_{⧉}$ (resp., $\longmapsto_{⧉}$). The judgments $M\ \mathsf{val}_{⧉}$ and $M \longmapsto_{⧉} M'$ are cubically-stable subrelations of the operational semantics.

Lemma 4.2 (Cubically-stable evaluation). *For all total interval substitutions ψ:*

1. *If $M\ \mathsf{val}_{⧉}$ then $M\psi\ \mathsf{val}$.*

2. *If $M \longmapsto_{⧉} M'$ then $M\psi \longmapsto M'\psi$.*

Cubically-stable evaluation figures prominently in this chapter's proofs, as head expansion (Lemma 4.32) holds only for cubically-stable steps. In **RedPRL**, we define a more sophisticated $\longmapsto_{⧉}$ relation which accounts also for steps that are stable by virtue of occurring under interval binders [Ang+18]. For instance:

$$\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_x;P,x.L)}(M) \longmapsto_{⧉} \mathsf{coe}^{r\rightsquigarrow r'}_{x.L}(M)$$

is stable because the x of loop_x is bound and thus unaffected by interval substitutions.

Note that we fix a strategy for simplifying hcom—checking first the type, then (at base types) $r = r'$, followed by each ξ_i in order. Other strategies, such as inspecting r, r', ξ_i *before* the type, are semantically equivalent, but exhibit different performance characteristics.

4.2 Constructing cubical type systems

We now assemble the cubical programs of Section 4.1 into the types and elements of our Cartesian cubical type theory. As in Section 2.2, we use Allen's fixed point construction

$$\frac{A \longmapsto A'}{\mathsf{hcom}_{A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \mathsf{hcom}_{A'}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})}$$

$$\frac{A \longmapsto A'}{\mathsf{coe}_{x.A}^{r \rightsquigarrow r'}(M) \longmapsto \mathsf{coe}_{x.A'}^{r \rightsquigarrow r'}(M)}$$

$$\frac{}{\mathsf{com}_{y.A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \mathsf{hcom}_{A\langle r'/y\rangle}^{r \rightsquigarrow r'}(\mathsf{coe}_{y.A}^{r \rightsquigarrow r'}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}_{y.A}^{y \rightsquigarrow r'}(N_i)})}$$

$$\frac{r = r' \qquad A \in \{\mathbb{S}^1, \mathcal{U}_j^{\mathsf{Kan}}, \mathcal{U}_j^{\mathsf{pre}}\}}{\mathsf{hcom}_{A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto M}$$

$$\frac{r \neq r' \qquad \neg\xi_i \ (\forall i < j) \qquad \xi_j \qquad A \in \{\mathbb{S}^1, \mathcal{U}_j^{\mathsf{Kan}}, \mathcal{U}_j^{\mathsf{pre}}\}}{\mathsf{hcom}_{A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto N_j\langle r'/y\rangle}$$

$$\frac{r \neq r' \qquad \neg\xi_i \ (\forall i) \qquad A \in \{\mathbb{S}^1, \mathcal{U}_j^{\mathsf{Kan}}, \mathcal{U}_j^{\mathsf{pre}}\}}{\mathsf{hcom}_{A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})\ \mathsf{val}}$$

$$\frac{}{\mathsf{ghcom}_{A}^{r \rightsquigarrow r'}(M; \cdot) \longmapsto M}$$

$$\frac{T_{\varepsilon,\bar{\varepsilon}} := \mathsf{hcom}_{A}^{r \rightsquigarrow z}(M; s' = \varepsilon \hookrightarrow y.N, s' = \bar{\varepsilon} \hookrightarrow y.\mathsf{ghcom}_{A}^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}), \overrightarrow{\xi_i \hookrightarrow y.N_i})}{\begin{array}{c}\mathsf{ghcom}_{A}^{r \rightsquigarrow r'}(M; s = s' \hookrightarrow y.N, \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \\ \mathsf{hcom}_{A}^{r \rightsquigarrow r'}(M; s = 0 \hookrightarrow z.T_{0,1}, s = 1 \hookrightarrow z.T_{1,0}, s = s' \hookrightarrow y.N, \overrightarrow{\xi_i \hookrightarrow y.N_i})\end{array}}$$

$$\frac{}{\mathsf{gcom}_{y.A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \mathsf{ghcom}_{A\langle r'/y\rangle}^{r \rightsquigarrow r'}(\mathsf{coe}_{y.A}^{r \rightsquigarrow r'}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}_{y.A}^{y \rightsquigarrow r'}(N_i)})}$$

Figure 4.2: Operational semantics: generic Kan operations.

$$\frac{}{(a{:}A) \to B\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{M\ N \longmapsto M'\ N} \qquad \frac{}{(\lambda a.M)\ N \longmapsto M[N/a]}$$

$$\frac{}{\lambda a.M\ \mathsf{val}} \qquad \frac{}{\mathsf{hcom}^{r \rightsquigarrow r'}_{(a:A)\to B}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \lambda a.\mathsf{hcom}^{r \rightsquigarrow r'}_{B}(M\ a; \overrightarrow{\xi_i \hookrightarrow y.N_i\ a})}$$

$$\frac{}{\mathsf{coe}^{r \rightsquigarrow r'}_{x.(a:A)\to B}(M) \longmapsto \lambda a.\mathsf{coe}^{r \rightsquigarrow r'}_{x.B[\mathsf{coe}^{r' \rightsquigarrow x}_{x.A}(a)/a]}(M\ \mathsf{coe}^{r' \rightsquigarrow r}_{x.A}(a))}$$

$$\frac{}{(a{:}A) \times B\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{\mathsf{fst}(M) \longmapsto \mathsf{fst}(M')} \qquad \frac{M \longmapsto M'}{\mathsf{snd}(M) \longmapsto \mathsf{snd}(M')}$$

$$\frac{}{\langle M, N\rangle\ \mathsf{val}} \qquad \frac{}{\mathsf{fst}(\langle M, N\rangle) \longmapsto M} \qquad \frac{}{\mathsf{snd}(\langle M, N\rangle) \longmapsto N}$$

$$\frac{F := \mathsf{hcom}^{r \rightsquigarrow z}_{A}(\mathsf{fst}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{fst}(N_i)})}{\begin{array}{c}\mathsf{hcom}^{r \rightsquigarrow r'}_{(a:A)\times B}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \\ \langle \mathsf{hcom}^{r \rightsquigarrow r'}_{A}(\mathsf{fst}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{fst}(N_i)}), \mathsf{com}^{r \rightsquigarrow r'}_{z.B[F/a]}(\mathsf{snd}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{snd}(N_i)})\rangle\end{array}}$$

$$\frac{}{\mathsf{coe}^{r \rightsquigarrow r'}_{x.(a:A)\times B}(M) \longmapsto \langle \mathsf{coe}^{r \rightsquigarrow r'}_{x.A}(\mathsf{fst}(M)), \mathsf{coe}^{r \rightsquigarrow r'}_{x.B[\mathsf{coe}^{r \rightsquigarrow x}_{x.A}(\mathsf{fst}(M))/a]}(\mathsf{snd}(M))\rangle}$$

$$\frac{}{\mathsf{Path}_{x.A}(M, N)\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{M@r \longmapsto M'@r} \qquad \frac{}{(\langle x\rangle M)@r \longmapsto M\langle r/x\rangle}$$

$$\frac{}{\langle x\rangle M\ \mathsf{val}} \qquad \frac{}{\begin{array}{c}\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{Path}_{x.A}(P_0,P_1)}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \\ \langle x\rangle \mathsf{hcom}^{r \rightsquigarrow r'}_{A}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow \_.P_\varepsilon}, \overrightarrow{\xi_i \hookrightarrow y.N_i@x})\end{array}}$$

$$\frac{}{\mathsf{coe}^{r \rightsquigarrow r'}_{y.\mathsf{Path}_{x.A}(P_0,P_1)}(M) \longmapsto \langle x\rangle \mathsf{com}^{r \rightsquigarrow r'}_{y.A}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow y.P_\varepsilon})}$$

Figure 4.2: Operational semantics: functions, pairs, paths.

$$\frac{}{\mathsf{Eq}_A(M,N)\ \mathsf{val}} \qquad \frac{}{\star\ \mathsf{val}} \qquad \frac{}{\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{Eq}_A(E_0,E_1)}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \star}$$

$$\frac{}{\mathsf{void}\ \mathsf{val}}$$

$$\frac{}{\mathsf{nat}\ \mathsf{val}} \qquad \frac{}{\mathsf{z}\ \mathsf{val}} \qquad \frac{}{\mathsf{s}(M)\ \mathsf{val}}$$

$$\frac{M \longmapsto M'}{\mathsf{natrec}(M;Z,n.a.S) \longmapsto \mathsf{natrec}(M';Z,n.a.S)} \qquad \frac{}{\mathsf{natrec}(\mathsf{z};Z,n.a.S) \longmapsto Z}$$

$$\frac{}{\mathsf{natrec}(\mathsf{s}(M);Z,n.a.S) \longmapsto S[M/n][\mathsf{natrec}(M;Z,n.a.S)/a]}$$

$$\frac{}{\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{nat}}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto M} \qquad \frac{}{\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{nat}}(M) \longmapsto M}$$

$$\frac{}{\mathsf{bool}\ \mathsf{val}} \qquad \frac{}{\mathsf{true}\ \mathsf{val}} \qquad \frac{}{\mathsf{false}\ \mathsf{val}} \qquad \frac{M \longmapsto M'}{\mathsf{if}(M;T,F) \longmapsto \mathsf{if}(M';T,F)}$$

$$\frac{}{\mathsf{if}(\mathsf{true};T,F) \longmapsto T} \qquad \frac{}{\mathsf{if}(\mathsf{false};T,F) \longmapsto F}$$

$$\frac{}{\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto M} \qquad \frac{}{\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{bool}}(M) \longmapsto M}$$

Figure 4.2: Operational semantics: equality, void, natural numbers, Booleans.

$$\frac{}{\mathbb{S}^1\ \mathsf{val}}\ \text{⧉} \qquad \frac{}{\mathsf{loop}_\varepsilon \longmapsto \mathsf{base}}\ \text{⧉} \qquad \frac{}{\mathsf{base}\ \mathsf{val}}\ \text{⧉} \qquad \frac{}{\mathsf{loop}_x\ \mathsf{val}}$$

$$\frac{M \longmapsto M'}{\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L) \longmapsto \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M'; P, x.L)}\ \text{⧉} \qquad \frac{}{\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{base}; P, x.L) \longmapsto P}\ \text{⧉}$$

$$\frac{}{\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_y; P, x.L) \longmapsto L\langle y/x\rangle}$$

$$\frac{r \neq r' \qquad \neg\xi_i\ (\forall i) \qquad F := \mathsf{hcom}^{r\rightsquigarrow z}_{\mathbb{S}^1}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})}{\begin{array}{c}\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}); P, x.L) \longmapsto \\ \mathsf{com}^{r\rightsquigarrow r'}_{z.A[F/c]}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L); \overrightarrow{\xi_i \hookrightarrow y.\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(N_i; P, x.L)})\end{array}}$$

$$\frac{}{\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathbb{S}^1}(M) \longmapsto M}\ \text{⧉}$$

$$\frac{\begin{array}{c} s \neq s' \qquad \neg\xi_i\ (\forall i) \qquad N_i := \mathsf{coe}^{s'\rightsquigarrow z}_{z.B_i}(\mathsf{coe}^{r\rightsquigarrow x}_{x.B_i\langle s'/z\rangle}(M)) \\ O := \mathsf{hcom}^{s'\langle r/x\rangle\rightsquigarrow z}_{A\langle r/x\rangle}(\mathsf{cap}^{s\langle r/x\rangle\leftsquigarrow s'\langle r/x\rangle}(M; \overrightarrow{\xi_i\langle r/x\rangle \hookrightarrow z.B_i\langle r/x\rangle}); \overrightarrow{T}) \\ \overrightarrow{T} := \overrightarrow{\xi_i\langle r/x\rangle \hookrightarrow z.\mathsf{coe}^{z\rightsquigarrow s\langle r/x\rangle}_{z.B_i\langle r/x\rangle}(\mathsf{coe}^{s'\langle r/x\rangle\rightsquigarrow z}_{z.B_i\langle r/x\rangle}(M))} \\ P := \mathsf{gcom}^{r\rightsquigarrow r'}_{x.A}(O\langle s\langle r/x\rangle/z\rangle; \overrightarrow{\xi_i \hookrightarrow x.N_i\langle s/z\rangle}|_{(x\#\xi_i)}, s = s' \hookrightarrow x.\mathsf{coe}^{r\rightsquigarrow x}_{x.A}(M)|_{(x\#s,s')}) \\ Q_k := \mathsf{gcom}^{s\langle r'/x\rangle\rightsquigarrow z}_{z.B_k\langle r'/x\rangle}(P; \overrightarrow{\xi_i \hookrightarrow z.N_i\langle r'/x\rangle}|_{(x\#\xi_i)}, r = r' \hookrightarrow z.\mathsf{coe}^{s'\langle r'/x\rangle\rightsquigarrow z}_{z.B_k\langle r'/x\rangle}(M)) \\ H := \mathsf{hcom}^{s\langle r'/x\rangle\rightsquigarrow s'\langle r'/x\rangle}_{A\langle r'/x\rangle}(P; \overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow z.\mathsf{coe}^{z\rightsquigarrow s\langle r'/x\rangle}_{z.B_i\langle r'/x\rangle}(Q_i)}, r = r' \hookrightarrow z.O) \end{array}}{\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_j}(A;\overrightarrow{\xi_i\hookrightarrow z.B_i})}(M) \longmapsto \mathsf{box}^{s\langle r'/x\rangle\rightsquigarrow s'\langle r'/x\rangle}(H; \overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow Q_i\langle s'\langle r'/x\rangle/z\rangle})}$$

Figure 4.2: Operational semantics: circle, universes.

$$\frac{}{\mathcal{U}^{\mathsf{pre}}_j\ \mathsf{val}}\ \text{⧉} \qquad \frac{}{\mathcal{U}^{\mathsf{Kan}}_j\ \mathsf{val}}\ \text{⧉} \qquad \frac{}{\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathcal{U}^{\kappa}_j}(M) \longmapsto M}\ \text{⧉}$$

$$\frac{r = r'}{\mathsf{box}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \longmapsto M}\ \text{⧉} \qquad \frac{r \neq r' \qquad \neg\xi_i\ (\forall i < j) \qquad \xi_j}{\mathsf{box}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \longmapsto N_j}$$

$$\frac{r \neq r' \qquad \neg\xi_i\ (\forall i)}{\mathsf{box}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i})\ \mathsf{val}} \qquad \frac{r = r'}{\mathsf{cap}^{r\leftrightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \longmapsto M}\ \text{⧉}$$

$$\frac{r \neq r' \qquad \neg\xi_i\ (\forall i < j) \qquad \xi_j}{\mathsf{cap}^{r\leftrightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \longmapsto \mathsf{coe}^{r'\rightsquigarrow r}_{y.B_j}(M)}$$

$$\frac{r \neq r' \qquad \neg\xi_i\ (\forall i) \qquad M \longmapsto M'}{\mathsf{cap}^{r\leftrightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \longmapsto \mathsf{cap}^{r\leftrightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.B_i})}$$

$$\frac{r \neq r' \qquad r_i \neq r_i'\ (\forall i)}{\mathsf{cap}^{r\leftrightsquigarrow r'}(\mathsf{box}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}); \overrightarrow{r_i = r_i' \hookrightarrow y.B_i}) \longmapsto M}$$

$$\frac{\begin{array}{c} s \neq s' \qquad s_j \neq s_j'\ (\forall j) \qquad O_i := \mathsf{cap}^{s\leftrightsquigarrow s'}(N_i; \overrightarrow{s_j = s_j' \hookrightarrow z.B_j}) \\ P_j := \mathsf{hcom}^{r\rightsquigarrow y}_{B_j\langle s'/z\rangle}(M; \overrightarrow{r_i = r_i' \hookrightarrow y.N_i}) \qquad Q := \mathsf{hcom}^{r\rightsquigarrow r'}_{A}(\mathsf{cap}^{s\leftrightsquigarrow s'}(M; \overrightarrow{s_j = s_j' \hookrightarrow z.B_j}); \overrightarrow{T}) \\ \overrightarrow{T} := \overrightarrow{r_i = r_i' \hookrightarrow y.O_i}, \overrightarrow{s_j = s_j' \hookrightarrow y.\mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(P_j)}, s = s' \hookrightarrow y.\mathsf{hcom}^{r\rightsquigarrow y}_{A}(M; \overrightarrow{r_i = r_i' \hookrightarrow y.N_i}) \end{array}}{\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{s_j = s_j' \hookrightarrow z.B_j})}(M; \overrightarrow{r_i = r_i' \hookrightarrow y.N_i}) \longmapsto \mathsf{box}^{s\rightsquigarrow s'}(Q; \overrightarrow{s_j = s_j' \hookrightarrow P_j\langle r'/y\rangle})}$$

Figure 4.2: Operational semantics: universes (continued).

$$\frac{}{\mathsf{V}_x(A,B,E)\ \mathsf{val}} \qquad \frac{}{\mathsf{V}_0(A,B,E) \longmapsto A}\ \text{⧉} \qquad \frac{}{\mathsf{V}_1(A,B,E) \longmapsto B}\ \text{⧉} \qquad \frac{}{\mathsf{Vin}_x(M,N)\ \mathsf{val}}$$

$$\frac{}{\mathsf{Vin}_0(M,N) \longmapsto M}\ \text{⧉} \qquad \frac{}{\mathsf{Vin}_1(M,N) \longmapsto N}\ \text{⧉} \qquad \frac{}{\mathsf{Vproj}_0(M,F) \longmapsto F\ M}\ \text{⧉}$$

$$\frac{}{\mathsf{Vproj}_1(M,F) \longmapsto M}\ \text{⧉} \qquad \frac{M \longmapsto M'}{\mathsf{Vproj}_x(M,F) \longmapsto \mathsf{Vproj}_x(M',F)}$$

$$\frac{}{\mathsf{Vproj}_x(\mathsf{Vin}_x(M,N),F) \longmapsto N}$$

$$\frac{\begin{array}{c} O := \mathsf{hcom}_{A}^{r\rightsquigarrow y}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \\ \overrightarrow{T} := x=0 \hookrightarrow y.\mathsf{fst}(E)\ O, x=1 \hookrightarrow y.\mathsf{hcom}_{B}^{r\rightsquigarrow y}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \end{array}}{\begin{array}{c} \mathsf{hcom}_{\mathsf{V}_x(A,B,E)}^{r\rightsquigarrow r'}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto \\ \mathsf{Vin}_x(O\langle r'/y\rangle, \mathsf{hcom}_{B}^{r\rightsquigarrow r'}(\mathsf{Vproj}_x(M,\mathsf{fst}(E));\overrightarrow{\xi_i \hookrightarrow y.\mathsf{Vproj}_x(N_i,\mathsf{fst}(E))},\overrightarrow{T})) \end{array}}$$

$$\frac{\begin{array}{c} N := \mathsf{coe}_{x.B}^{r\rightsquigarrow r'}(\mathsf{Vproj}_r(M,\mathsf{fst}(E\langle r/x\rangle))) \\ F := (a{:}A\langle r'/x\rangle) \times \mathsf{Path}_{\_.B\langle r'/x\rangle}(\mathsf{fst}(E\langle r'/x\rangle)\ a, N) \qquad C := \mathsf{snd}(E\langle r'/x\rangle)\ N \\ O := \mathsf{hcom}_{F}^{1\rightsquigarrow 0}(\mathsf{fst}(C); r=0 \hookrightarrow z.(\mathsf{snd}(C)\ \langle M, \langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M)\rangle)@z, r=1 \hookrightarrow \_.\mathsf{fst}(C)) \\ P := \mathsf{hcom}_{B\langle r'/x\rangle}^{1\rightsquigarrow 0}(N; r'=0 \hookrightarrow z.\mathsf{snd}(O)@z, r'=1 \hookrightarrow \_.N, r=r' \hookrightarrow \_.\mathsf{Vproj}_r(M,\mathsf{fst}(E\langle r/x\rangle))) \end{array}}{\mathsf{coe}_{x.\mathsf{V}_x(A,B,E)}^{r\rightsquigarrow r'}(M) \longmapsto \mathsf{Vin}_{r'}(\mathsf{fst}(O),P)}\ \text{⧉}$$

$$\frac{x \neq y \qquad \overrightarrow{T} := x=0 \hookrightarrow y.\mathsf{fst}(E)\ \mathsf{coe}_{y.A}^{r\rightsquigarrow y}(M), x=1 \hookrightarrow y.\mathsf{coe}_{y.B}^{r\rightsquigarrow y}(M)}{\mathsf{coe}_{y.\mathsf{V}_x(A,B,E)}^{r\rightsquigarrow r'}(M) \longmapsto \mathsf{Vin}_x(\mathsf{coe}_{y.A}^{r\rightsquigarrow r'}(M), \mathsf{com}_{y.B}^{r\rightsquigarrow r'}(\mathsf{Vproj}_x(M,\mathsf{fst}(E\langle r/y\rangle));\overrightarrow{T}))}$$

Figure 4.2: Operational semantics: univalence.

[All87] to define a PER of value types, and for each value type, a PER of value elements. In the cubical setting, however, we must define such PERs at every dimension (that is, for each finite set of interval variables).

Definition 4.3. A *candidate cubical type system* is a relation $\tau(\Psi, A_0, B_0, \varphi)$ over a finite set Ψ of interval variables, A_0 val, B_0 val, and binary relations $\varphi(M_0, N_0)$ over M_0 val and N_0 val, where $\mathsf{FI}(A_0, B_0, M_0, N_0) \subseteq \Psi$.

The relation $\tau(\{x_1, \dots, x_n\}, A_0, B_0, \varphi)$ encodes that A_0 and B_0 are equal n-dimensional value types ($x_1 : \mathbb{I}, \dots, x_n : \mathbb{I} \vdash A_0 = B_0 : \mathsf{Type}_i$ in Chapter 3's notation) whose n-dimensional value elements are specified by φ.

In Idealized Nuprl, we define the closed judgments (Definition 2.12) as the evaluation lifting of τ, thereby ensuring they are closed under evaluation. In cubical type theory, judgments must be closed under both evaluation and interval substitution—if $x_1 : \mathbb{I}, \dots, x_n : \mathbb{I} \vdash M : A$ and $\psi : \{x_1, \dots, x_m\} \to \{x_1, \dots, x_n\}$ then $x_1 : \mathbb{I}, \dots, x_m : \mathbb{I} \vdash M\psi : A\psi$. The latter condition is complicated by the fact that types contain interval variables, as types depend on terms, and terms contain interval variables. Therefore, not only must a type itself evaluate to a value type, but also all of its interval substitution instances must as well. We track this data by assigning to each type over Ψ a Ψ-relation of elements.

Definition 4.4. A *Ψ-relation* is a family of binary relations $\alpha_\psi(M, N)$ indexed by interval substitutions $\psi : \Psi' \to \Psi$, over terms $\mathsf{FI}(M, N) \subseteq \Psi'$. We write $\alpha(M, N)$ for $\alpha_{\mathrm{id}_\Psi}(M, N)$. A Ψ-relation that depends only on Ψ' (and not ψ) is *context-indexed*, and we write $\alpha_{\Psi'}(M, N)$.

A Ψ-PER captures the data of a functor $(\square/\Psi)^{\mathbf{op}} \to \mathbf{Set}$. Intuitively, such functors are *dependent Cartesian cubical sets*, because a dependent type over the Yoneda embedding of Ψ is an object of $[\square^{\mathbf{op}}, \mathbf{Set}]/\mathbf{hom}_\square(-, \Psi) \simeq [(\square/\Psi)^{\mathbf{op}}, \mathbf{Set}]$. Like such a functor, we can precompose a Ψ-relation α by an interval substitution $\psi : \Psi' \to \Psi$, obtaining a Ψ'-relation $(\alpha\psi)_{\psi'}(M, N) := \alpha_{\psi\psi'}(M, N)$. A context-indexed PER captures the data of an ordinary Cartesian cubical set, or a functor $\square^{\mathbf{op}} \to \mathbf{Set}$.

As in Section 2.2, we define *candidate cubical judgments* for use in our fixed point construction. Every type A over Ψ must give rise to a Ψ-PER $[\![A]\!]$ of value elements, whose $\psi_1 : \Psi_1 \to \Psi$ component we extract from τ as the relation φ for which $\tau(\Psi_1, A_1, A_1, \varphi)$ where $A\psi_1 \Downarrow A_1$. Moreover, meanings of types must be preserved by both evaluation ($[\![A]\!] = [\![A_0]\!]$ when $A \Downarrow A_0$) and interval substitution ($[\![A]\!]\psi_1 = [\![A\psi_1]\!]$ when $\psi_1 : \Psi_1 \to \Psi$). Thus $[\![A]\!]\psi_1 = [\![A_1]\!]$ when $A\psi_1 \Downarrow A_1$; unrolling definitions, we must require that for all $\psi_2 : \Psi_2 \to \Psi_1$, τ assigns the same relation to the values of $A\psi_1\psi_2$ and $A_1\psi_2$ at Ψ_2.

Definition 4.5 (Candidate cubical judgments)**.** Given a candidate cubical type system τ, and writing $\tau^{\Downarrow}(\Psi, A, B, \varphi)$ for $A \Downarrow A_0$, $B \Downarrow B_0$, and $\tau(\Psi, A_0, B_0, \varphi)$:

1. $A \sim A' \downarrow \alpha \in \tau\ [\Psi]$ when for all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$, we have $A\psi_1 \Downarrow A_1$ and $A'\psi_1 \Downarrow A_1'$; there exists a φ such that $\tau^{\Downarrow}(\Psi_2, -, -, \varphi)$ relates the pairs $(A_1\psi_2, A\psi_1\psi_2)$, $(A\psi_1\psi_2, A_1\psi_2)$, $(A_1'\psi_2, A'\psi_1\psi_2)$, $(A'\psi_1\psi_2, A_1'\psi_2)$, and $(A_1\psi_2, A_1'\psi_2)$; and α is a Ψ-relation on values satisfying $\tau^{\Downarrow}(\Psi', A\psi, A'\psi, \alpha_\psi)$ for all $\psi : \Psi' \to \Psi$.

2. $M \sim M' \in \alpha\ [\Psi]$ when for all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$, we have $M\psi_1 \Downarrow M_1$, $M'\psi_1 \Downarrow M_1'$, and $(\alpha_{\psi_1\psi_2})^{\Downarrow}$ relates the pairs $(M_1\psi_2, M\psi_1\psi_2)$, $(M\psi_1\psi_2, M_1\psi_2)$, $(M_1'\psi_2, M'\psi_1\psi_2)$, $(M'\psi_1\psi_2, M_1'\psi_2)$, and $(M_1\psi_2, M_1'\psi_2)$.

3. $a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau\ [\Psi]$ when for all $\psi : \Psi' \to \Psi$ and $M \sim M' \in \alpha\psi\ [\Psi']$, $B\psi[M/a] \sim B'\psi[M'/a] \downarrow \beta^{\psi,M,M'} \in \tau\ [\Psi']$.

4. $a : \alpha \triangleright N \sim N' \in \beta\ [\Psi]$ when for all $\psi : \Psi' \to \Psi$ and $M \sim M' \in \alpha\psi\ [\Psi']$, $N\psi[M/a] \sim N'\psi[M'/a] \in \beta^{\psi,M,M'}\ [\Psi']$.

One can easily check that the candidate judgments are monotone in τ and closed under interval substitution. Monotonicity is essential, as we define the computational semantics of Cartesian cubical type theory as a *cubical type system* given by the least fixed point of a monotone function on the complete lattice of candidate cubical type systems.

Definition 4.6. A *cubical type system* is a candidate cubical type system τ satisfying:

1. *Unicity*: If $\tau(\Psi, A_0, B_0, \varphi)$ and $\tau(\Psi, A_0, B_0, \varphi')$ then $\varphi = \varphi'$.

2. *PER-valuation*: If $\tau(\Psi, A_0, B_0, \varphi)$ then φ is symmetric and transitive.

3. *Symmetry*: If $\tau(\Psi, A_0, B_0, \varphi)$ then $\tau(\Psi, B_0, A_0, \varphi)$.

4. *Transitivity*: If $\tau(\Psi, A_0, B_0, \varphi)$ and $\tau(\Psi, B_0, C_0, \varphi)$ then $\tau(\Psi, A_0, C_0, \varphi)$.

5. *Value-coherence*: If $\tau(\Psi, A_0, B_0, \varphi)$ then $A_0 \sim B_0 \downarrow \alpha \in \tau\ [\Psi]$ for some α.

If τ is a cubical type system, then $A \sim A' \downarrow \alpha \in \tau\ [\Psi]$ enjoys unicity, symmetry, transitivity, and is Ψ-PER-valued. If α is a Ψ-PER, then $- \sim - \in \alpha\ [\Psi]$ is a PER. *Value-coherence* ensures that the value types of τ are types in the sense of the candidate judgments, which was automatic in Section 2.2; it also ensures that value types form a Cartesian cubical set whose restriction maps are given by interval substitution followed by evaluation. (Without it, one might have $\tau(\Psi, A_0, A_0, \varphi)$ but not $\tau^{\Downarrow}(\Psi', A_0\psi, A_0\psi, \varphi')$.) In Definition 4.11, we will impose an analogous condition on the meaning of each type.

Definition 4.7. A Ψ-relation on values α is *value-coherent*, written $\mathsf{Coh}(\alpha)$, when for all $\psi : \Psi' \to \Psi$, if $\alpha_\psi(M_0, N_0)$ then $M_0 \sim N_0 \in \alpha\psi\ [\Psi']$. Given $a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau\ [\Psi]$, we write $\mathsf{CohFam}(\beta)$ when $\mathsf{Coh}(\beta^{\psi,M,M'})$ for all $\psi : \Psi' \to \Psi$ and $M \sim M' \in \alpha\psi\ [\Psi']$.

Despite their complexity, the above definitions in fact coincide with Definitions 2.2 and 2.3 if our cubical type theory has no cubical structure (that is, if programs cannot contain interval variables and τ is constant in Ψ). Thinking in terms of categories, one can regard a set-theoretic model as a cubical set model restricted to constant presheaves.

In Figure 4.3, we define each type former by its formation and introduction rules, as a monotone function on candidate cubical type systems, using the abbreviations:

$$\begin{aligned}
\mathsf{valid}(\overrightarrow{r_i = r_i'}) &:= \exists j, k.(r_j = r_k) \wedge (r_j' = 0) \wedge (r_k' = 1)\\
\mathsf{isContr}(C) &:= (c{:}C) \times ((c'{:}C) \to \mathsf{Path}_{\_.C}(c', c))\\
\mathsf{Equiv}(A, B) &:= (f{:}A \to B) \times ((b{:}B) \to \mathsf{isContr}((a{:}A) \times \mathsf{Path}_{\_.B}(f\ a, b)))
\end{aligned}$$

In the cubical setting, it is essential that we define types by their introduction (rather than elimination) rules, because our proofs of value-coherence in Section 4.4 rely on knowing the head constructors of value elements in order to characterize their evaluation behavior under interval substitutions.

In Figure 4.3, the candidate cubical type system $K(\nu, \sigma)$ generates Kan types, drawing universes from ν and constituent Kan types from σ; $P(\nu, \sigma, \tau)$ generates pretypes (arbitrary types), drawing universes from ν, constituent Kan types from σ, and constituent pretypes from τ. Pretypes depend on Kan types because of homogeneous compositions of Kan types (HKan); the only non-Kan type formers are Eq and HPre. The value elements of certain base types (Nat and Circ) are mutually defined for all Ψ and are therefore defined as fixed points ($\mathbb{N}$ and $\mathbb{C}$).

As in Idealized Nuprl, we have two universe hierarchies $\mathcal{U}_i^{\mathsf{Kan}}$ and $\mathcal{U}_i^{\mathsf{pre}}$ whose elements are Kan types and pretypes respectively. For all $i \in \{0, 1, \dots, \omega\}$, we define ν_i as the candidate cubical type system consisting of universes $< i$, $\tau_i^{\mathsf{Kan}} := \mu\sigma.K(\nu_i, \sigma)$ as the ith Kan type universe, and $\tau_i^{\mathsf{pre}} := \mu\tau.P(\nu_i, \tau_i^{\mathsf{Kan}}, \tau)$ as the ith pretype universe. The rest of this section is devoted to rather technical proofs that τ_i^{pre} and τ_i^{Kan} are cubical type systems (Theorem 4.9), and that $\tau_i^{\mathsf{Kan}} \subset \tau_i^{\mathsf{pre}}$ (Theorem 4.10).

Lemma 4.8. *If ν and σ are cubical type systems, then $\mu^{\mathsf{Kan}}(\nu) := \mu\sigma.K(\nu, \sigma)$ and $\mu^{\mathsf{pre}}(\nu, \sigma) := \mu\tau.P(\nu, \sigma, \tau)$ are cubical type systems.*

Proof. We can consider each type former Fun, Pair, . . . separately because they are disjoint. We describe the proof for $\mu^{\mathsf{pre}}(\nu, \sigma)$; the proof for $\mu^{\mathsf{Kan}}(\nu)$ is analogous. Unlike in Lemma 2.6, we can prove each property separately, as the candidate cubical judgments inadvertently contain "cross cases" needed for symmetry and transitivity of type families.

1. *Unicity.*

   Let $\Phi = \{(\Psi, A_0, B_0, \varphi) \mid \forall\varphi'.\mu^{\mathsf{pre}}(\nu, \sigma)(\Psi, A_0, B_0, \varphi') \implies (\varphi = \varphi')\}$, and show that Φ is a pre-fixed point of $P(\nu, \sigma, -)$ (that is, $P(\nu, \sigma, \Phi) \subseteq \Phi$). Because $\mu^{\mathsf{pre}}(\nu, \sigma)$ is the least pre-fixed point, it will follow that $\mu^{\mathsf{pre}}(\nu, \sigma) \subseteq \Phi$, and that $\mu^{\mathsf{pre}}(\nu, \sigma)$ has unicity.

$$\begin{aligned}
\textsc{Fun}(\tau) := \{&(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi) \mid \\
&\exists \alpha, \beta^{(-,-,-)}.(A \sim A' \downarrow \alpha \in \tau\ [\Psi]) \wedge \mathsf{Coh}(\alpha) \\
&\wedge (a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau\ [\Psi]) \wedge \mathsf{CohFam}(\beta) \\
&\wedge (\varphi = \{(\lambda a.N, \lambda a.N') \mid a : \alpha \triangleright N \sim N' \in \beta\ [\Psi]\})\} \\
\textsc{Pair}(\tau) := \{&(\Psi, (a{:}A) \times B, (a{:}A') \times B', \varphi) \mid \\
&\exists \alpha, \beta^{(-,-,-)}.(A \sim A' \downarrow \alpha \in \tau\ [\Psi]) \wedge \mathsf{Coh}(\alpha) \\
&\wedge (a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau\ [\Psi]) \wedge \mathsf{CohFam}(\beta) \\
&\wedge (\varphi = \{(\langle M, N\rangle, \langle M', N'\rangle) \mid (M \sim M' \in \alpha\ [\Psi]) \wedge (N \sim N' \in \beta^{\mathsf{id}_\Psi, M, M'}\ [\Psi])\})\} \\
\textsc{Path}(\tau) := \{&(\Psi, \mathsf{Path}_{x.A}(P_0, P_1), \mathsf{Path}_{x.A'}(P'_0, P'_1), \varphi) \mid \\
&\exists \alpha.(A \sim A' \downarrow \alpha \in \tau\ [\Psi, x]) \wedge \mathsf{Coh}(\alpha) \wedge (\forall \varepsilon.P_\varepsilon \sim P'_\varepsilon \in \alpha\langle \varepsilon/x\rangle\ [\Psi]) \\
&\wedge (\varphi = \{(\langle x\rangle M, \langle x\rangle M') \mid \\
&\qquad (M \sim M' \in \alpha\ [\Psi, x]) \wedge (\forall \varepsilon.M\langle \varepsilon/x\rangle \sim P_\varepsilon \in \alpha\langle \varepsilon/x\rangle\ [\Psi])\})\} \\
\textsc{Eq}(\tau) := \{&(\Psi, \mathsf{Eq}_A(M, N), \mathsf{Eq}_{A'}(M', N'), \varphi) \mid \\
&\exists \alpha.(A \sim A' \downarrow \alpha \in \tau\ [\Psi]) \wedge \mathsf{Coh}(\alpha) \wedge (M \sim M' \in \alpha\ [\Psi]) \wedge (N \sim N' \in \alpha\ [\Psi]) \\
&\wedge (\varphi = \{(\star, \star) \mid M \sim N \in \alpha\ [\Psi]\})\} \\
\mathrm{V}(\tau) := \{&((\Psi, x), \mathsf{V}_x(A, B, E), \mathsf{V}_x(A', B', E'), \varphi) \mid \\
&\exists \beta, \alpha^{(-)}, \eta^{(-)}.(B \sim B' \downarrow \beta \in \tau\ [\Psi, x]) \wedge \mathsf{Coh}(\beta) \\
&\wedge (\forall \psi.(x\psi = 0) \implies (A\psi \sim A'\psi \downarrow \alpha^{\psi} \in \tau\ [\Psi']) \wedge \mathsf{Coh}(\alpha^{\psi}) \\
&\qquad \wedge (\mathsf{Equiv}(A\psi, B\psi) \sim \mathsf{Equiv}(A\psi, B\psi) \downarrow \eta^{\psi} \in \tau\ [\Psi']) \wedge (E\psi \sim E'\psi \in \eta^{\psi}\ [\Psi'])) \\
&\wedge (\varphi = \{(\mathsf{Vin}_x(M, N), \mathsf{Vin}_x(M', N')) \mid (N \sim N' \in \beta\ [\Psi, x]) \wedge (\forall \psi.(x\psi = 0) \implies \\
&\qquad (M\psi \sim M'\psi \in \alpha^{\psi}\ [\Psi']) \wedge (\mathsf{fst}(E\psi)\ M\psi \sim N\psi \in \beta\psi\ [\Psi']))\})\} \\
\textsc{HPre}(\tau) := \{&(\Psi, \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}), \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A'; \overrightarrow{\xi_i \hookrightarrow y.B'_i}), \{\}) \mid \\
&\exists \alpha, \beta^{(-,-,-)}.(r \neq r') \wedge (\forall i.\neg \xi_i) \wedge \mathsf{valid}(\overrightarrow{\xi_i}) \wedge (A \sim A' \downarrow \alpha \in \tau\ [\Psi]) \wedge \mathsf{Coh}(\alpha) \\
&\wedge (\forall i, j, \psi.(\xi_i\psi \wedge \xi_j\psi) \implies (B_i\psi \sim B'_j\psi \downarrow \beta^{\psi, i, j} \in \tau\ [\Psi']) \wedge \mathsf{Coh}(\beta^{\psi, i, j})) \\
&\wedge (\forall i, \psi.\xi_i\psi \implies B_i\langle r/y\rangle\psi \sim A\psi \downarrow \_ \in \tau\ [\Psi'])\}
\end{aligned}$$

Figure 4.3: PER semantics.

$$
\begin{aligned}
\textsc{Void} &:= \{(\Psi, \mathsf{void}, \mathsf{void}, \{\})\} \\
\textsc{Nat} &:= \{(\Psi, \mathsf{nat}, \mathsf{nat}, \varphi) \mid \varphi = \{(M_0, M_0') \mid \mathbb{N}(\Psi, M_0, M_0')\}\} \\
\textsc{Bool} &:= \{(\Psi, \mathsf{bool}, \mathsf{bool}, \{(\mathsf{true}, \mathsf{true}), (\mathsf{false}, \mathsf{false})\})\} \\
\textsc{Circ} &:= \{(\Psi, \mathbb{S}^1, \mathbb{S}^1, \varphi) \mid \varphi = \{(M_0, M_0') \mid \mathbb{C}(\Psi, M_0, M_0')\}\} \\
\textsc{UPre}(\nu) &:= \{(\Psi, \mathcal{U}_j^{\mathsf{pre}}, \mathcal{U}_j^{\mathsf{pre}}, \varphi) \mid \nu(\Psi, \mathcal{U}_j^{\mathsf{pre}}, \mathcal{U}_j^{\mathsf{pre}}, \varphi)\} \\
\textsc{UKan}(\nu) &:= \{(\Psi, \mathcal{U}_j^{\mathsf{Kan}}, \mathcal{U}_j^{\mathsf{Kan}}, \varphi) \mid \nu(\Psi, \mathcal{U}_j^{\mathsf{Kan}}, \mathcal{U}_j^{\mathsf{Kan}}, \varphi)\} \\
\textsc{HKan}(\tau) &:= \{(\Psi, \mathsf{hcom}_{\mathcal{U}_k^{\mathsf{Kan}}}^{r \rightsquigarrow r'}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}), \mathsf{hcom}_{\mathcal{U}_k^{\mathsf{Kan}}}^{r \rightsquigarrow r'}(A'; \overrightarrow{\xi_i \hookrightarrow y.B_i'}), \varphi) \mid \\
&\qquad \exists \alpha, \beta^{(-,-,-)}.(r \neq r') \wedge (\forall i. \neg \xi_i) \wedge \mathsf{valid}(\overrightarrow{\xi_i}) \wedge (A \sim A' \downarrow \alpha \in \tau\ [\Psi]) \wedge \mathsf{Coh}(\alpha) \\
&\qquad \wedge (\forall i, j, \psi : \Psi' \to (\Psi, y).(\xi_i \psi \wedge \xi_j \psi) \implies \\
&\qquad\qquad (B_i \psi \sim B_j' \psi \downarrow \beta^{\psi,i,j} \in \tau\ [\Psi']) \wedge \mathsf{Coh}(\beta^{\psi,i,j})) \\
&\qquad \wedge (\forall i, \psi. \xi_i \psi \implies B_i \langle r/y \rangle \psi \sim A\psi \downarrow \_ \in \tau\ [\Psi']) \\
&\qquad \wedge (\varphi = \{(\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}), \mathsf{box}^{r \rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow N_i'})) \mid (M \sim M' \in \alpha\ [\Psi]) \\
&\qquad\quad \wedge (\forall i, j, \psi. (\xi_i \psi \wedge \xi_j \psi) \implies N_i \psi \sim N_j' \psi \in \beta^{\psi,i,j} \langle r'\psi / y \rangle\ [\Psi']) \\
&\qquad\quad \wedge (\forall i, \psi. \xi_i \psi \implies M\psi \sim \mathsf{coe}_{y.B_i\psi}^{r'\psi \rightsquigarrow r\psi}(N_i \psi) \in \alpha\psi\ [\Psi'])\})\}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{N} &:= \mu R.(\{(\Psi, \mathsf{z}, \mathsf{z})\} \cup \{(\Psi, \mathsf{s}(M), \mathsf{s}(M')) \mid M \sim M' \in R\ [\Psi]\}) \\
\mathbb{C} &:= \mu R.(\{(\Psi, \mathsf{base}, \mathsf{base}), ((\Psi, x), \mathsf{loop}_x, \mathsf{loop}_x)\} \cup \\
&\quad \{(\Psi, \mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}), \mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'})) \mid \\
&\quad (r \neq r') \wedge (\forall i. \neg \xi_i) \wedge \mathsf{valid}(\overrightarrow{\xi_i}) \wedge (M \sim M' \in R\ [\Psi]) \\
&\quad \wedge (\forall i, j, \psi : \Psi' \to (\Psi, y).(\xi_i \psi \wedge \xi_j \psi) \implies N_i \psi \sim N_j' \psi \in R\ [\Psi']) \\
&\quad \wedge (\forall i, \psi : \Psi' \to \Psi. \xi_i \psi \implies N_i \langle r/y \rangle \psi \sim M\psi \in R\ [\Psi'])\})
\end{aligned}
$$

Figure 4.3: PER semantics.

$$\begin{aligned}
P(\nu,\sigma,\tau) &:= \textsc{Fun}(\tau) \cup \textsc{Pair}(\tau) \cup \textsc{Path}(\tau) \cup \textsc{Eq}(\tau) \cup \mathrm{V}(\tau) \cup \mathrm{H}\textsc{Kan}(\sigma) \cup \mathrm{H}\textsc{Pre}(\tau)\\
&\quad \cup\ \textsc{Void} \cup \textsc{Nat} \cup \textsc{Bool} \cup \textsc{Circ} \cup \mathrm{U}\textsc{Pre}(\nu) \cup \mathrm{U}\textsc{Kan}(\nu)\\
K(\nu,\sigma) &:= \textsc{Fun}(\sigma) \cup \textsc{Pair}(\sigma) \cup \textsc{Path}(\sigma) \cup \mathrm{V}(\sigma) \cup \mathrm{H}\textsc{Kan}(\sigma)\\
&\quad \cup\ \textsc{Void} \cup \textsc{Nat} \cup \textsc{Bool} \cup \textsc{Circ} \cup \mathrm{U}\textsc{Pre}(\nu) \cup \mathrm{U}\textsc{Kan}(\nu)
\end{aligned}$$

$$\begin{aligned}
\nu_n &:= \{(\Psi, \mathcal{U}_i^\kappa, \mathcal{U}_i^\kappa, \varphi) \mid (i < n) \land (\varphi = \{(A_0, B_0) \mid \tau_i^\kappa(\Psi, A_0, B_0, \_)\})\}\\
\tau_n^{\mathsf{Kan}} &:= \mu\sigma.K(\nu_n, \sigma)\\
\tau_n^{\mathsf{pre}} &:= \mu\tau.P(\nu_n, \tau_n^{\mathsf{Kan}}, \tau)
\end{aligned}$$

Figure 4.3: PER semantics.

Assume that $\textsc{Fun}(\Phi)(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. Thus $A \sim A' \downarrow \alpha \in \Phi\ [\Psi]$, and in particular, for all $\psi : \Psi' \to \Psi$, $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A\psi, A'\psi, \varphi')$ implies $\alpha_\psi = \varphi'$, so α is unique in $\mu^{\mathsf{pre}}(\nu,\sigma)$ when it exists. Similarly, each $\beta^{(-,-,-)}$ is unique in $\mu^{\mathsf{pre}}(\nu,\sigma)$ when it exists. The relation φ is determined uniquely by α and $\beta^{(-,-,-)}$. Now let us show $\Phi(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$, that is, assume $\mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi')$ and show $\varphi = \varphi'$. It follows that $A \sim A' \downarrow \alpha' \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$ for some α', and similarly for some family β', but $\alpha = \alpha'$ and $\beta^{(-,-,-)} = \beta'^{(-,-,-)}$. Because φ' is defined using the same α and $\beta^{(-,-,-)}$ as φ, we conclude $\varphi = \varphi'$. Other cases are similar; for $\mathrm{H}\textsc{Kan}$, $\mathrm{U}\textsc{Pre}$, $\mathrm{U}\textsc{Kan}$ we use that ν, σ have unicity.

2. *PER-valuation.*

   Let $\Phi = \{(\Psi, A_0, B_0, \varphi) \mid \varphi \text{ is a PER}\}$, and show that Φ is a pre-fixed point of $P(\nu,\sigma,-)$. It follows that $\mu^{\mathsf{pre}}(\nu,\sigma)$ is PER-valued, by $\mu^{\mathsf{pre}}(\nu,\sigma) \subseteq \Phi$.

   Assume that $\textsc{Fun}(\Phi)(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. Then $A \sim A' \downarrow \alpha \in \Phi\ [\Psi]$, and in particular, for all $\psi : \Psi' \to \Psi$, $\Phi^{\Downarrow}(\Psi', A\psi, A'\psi, \alpha_\psi)$, so each α_ψ is a PER. Similarly, each $\beta_{\psi'}^{\psi,M,M'}$ is a PER. Now we must show $\Phi(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. The relation φ is a PER because α and β are PER-valued so their candidate equality judgments are PERs. Most cases proceed in this fashion. For $\textsc{Nat}$ and $\textsc{Circ}$ we show that $\mathbb{N}$ and $\mathbb{C}$ are symmetric and transitive at all Ψ (employing the same strategy as in parts (3–4)); for $\mathrm{H}\textsc{Kan}$, $\mathrm{U}\textsc{Pre}$, $\mathrm{U}\textsc{Kan}$ we use that σ, ν are PER-valued.

3. *Symmetry.*

   Let $\Phi = \{(\Psi, A_0, B_0, \varphi) \mid \mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, B_0, A_0, \varphi)\}$, and show that Φ is a pre-fixed point of $P(\nu,\sigma,-)$. It will follow that $\mu^{\mathsf{pre}}(\nu,\sigma)$ is symmetric, by $\mu^{\mathsf{pre}}(\nu,\sigma) \subseteq \Phi$.

Assume that $\textsc{Fun}(\Phi)(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. Then $A \sim A' \downarrow \alpha \in \Phi\ [\Psi]$ and $\mathsf{Coh}(\alpha)$, and therefore $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A'\psi, A\psi, \alpha_\psi)$, $A\psi_1 \Downarrow A_1$, $A'\psi_1 \Downarrow A'_1$, and $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi_2, -, -, \varphi)$ relates the pairs $(A\psi_1\psi_2, A_1\psi_2)$, $(A_1\psi_2, A\psi_1\psi_2)$, $(A'\psi_1\psi_2, A'_1\psi_2)$, $(A'_1\psi_2, A'\psi_1\psi_2)$, and $(A'_1\psi_2, A_1\psi_2)$. Similar facts hold for instances of B, B', β. We must show $\Phi(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$, that is, $\mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, (a{:}A') \to B', (a{:}A) \to B, \varphi)$. This requires $A' \sim A \downarrow \alpha \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$ and $\mathsf{Coh}(\alpha)$, which follows from the above facts; and also $B'\psi[M/a] \sim B\psi[M'/a] \downarrow \beta^{\psi,M,M'} \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$ and $\mathsf{Coh}(\beta^{\psi,M,M'})$ whenever $M \sim M' \in \alpha\psi\ [\Psi']$, which follows from the symmetry of $- \sim - \in \alpha\psi\ [\Psi']$ (because each α_ψ is a PER, by (2)), and the above facts. Other cases are similar; for HKan we use that σ is symmetric.

4. *Transitivity.*

   Let $\Phi = \{(\Psi, A_0, B_0, \varphi) \mid \forall C_0.\mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, B_0, C_0, \varphi) \implies \mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, A_0, C_0, \varphi)\}$, and show that Φ is a pre-fixed point of $P(\nu,\sigma,-)$. It will follow that $\mu^{\mathsf{pre}}(\nu,\sigma)$ is transitive, by $\mu^{\mathsf{pre}}(\nu,\sigma) \subseteq \Phi$.

   Assume that $\textsc{Fun}(\Phi)(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. Then $A \sim A' \downarrow \alpha \in \Phi\ [\Psi]$, and thus if $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A'\psi, C_0, \alpha_\psi)$ then $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A\psi, C_0, \alpha_\psi)$. Furthermore, $A\psi_1 \Downarrow A_1$, $A'\psi_1 \Downarrow A'_1$, and for any C_0, $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi_2, -, -, \varphi)$ relates $(A\psi_1\psi_2, C_0)$ if and only if $(A_1\psi_2, C_0)$; $(A'\psi_1\psi_2, C_0)$ if and only if $(A'_1\psi_2, C_0)$; and if $(A'_1\psi_2, C_0)$ then $(A_1\psi_2, C_0)$. Similar facts hold for B, B', β by $a : \alpha \triangleright B \sim B' \downarrow \beta \in \Phi\ [\Psi]$.

   Now we must show $\Phi(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$, that is, if $\mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, (a{:}A') \to B', C_0, \varphi)$ then $\mu^{\mathsf{pre}}(\nu,\sigma)(\Psi, (a{:}A) \to B, C_0, \varphi)$. Inspecting P, $C_0 = (a{:}A'') \to B''$; thus $A' \sim A'' \downarrow \alpha' \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$ and $\mathsf{Coh}(\alpha')$, so $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A'\psi, A''\psi, \alpha'_\psi)$, and by hypothesis, $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A\psi, A''\psi, \alpha_\psi)$ and $\mathsf{Coh}(\alpha)$. We already know $A\psi_1 \Downarrow A_1$, $A''\psi_1 \Downarrow A''_1$, and that $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi_2, -, -, \varphi)$ relates $(A''\psi_1\psi_2, A''_1\psi_2)$ and vice versa. By $(A'_1\psi_2, A''_1\psi_2)$ and the above, we have $(A_1\psi_2, A''_1\psi_2)$. Finally, by $(A'\psi_1\psi_2, A'_1\psi_2)$ and transitivity we have $(A'_1\psi_2, A'_1\psi_2)$, hence by transitivity and symmetry $(A'_1\psi_2, A_1\psi_2)$, and again by transitivity $(A_1\psi_2, A_1\psi_2)$; as needed, $(A_1\psi_2, A_0\psi_2)$ and vice versa follow by transitivity. Similarly, using transitivity of each α_ψ (by (2)), whenever $M \sim M' \in \alpha\psi\ [\Psi']$, $B\psi[M/a] \sim B''\psi[M'/a] \downarrow \beta^{\psi,M,M'} \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi']$ and $\mathsf{Coh}(\beta^{\psi,M,M'})$. Other cases are similar; for HKan we use that σ is transitive.

5. *Value-coherence.*

   Let $\Phi = \{(\Psi, A_0, B_0, \varphi) \mid A_0 \sim B_0 \downarrow \alpha \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]\}$, and show that Φ is a pre-fixed point of $P(\nu,\sigma,-)$. It will follow that $\mu^{\mathsf{pre}}(\nu,\sigma)$ is value-coherent, by $\mu^{\mathsf{pre}}(\nu,\sigma) \subseteq \Phi$. We will check Fun (Pair, Path, and Eq are similar) and V (HKan and HPre are similar); the property $P(\nu,\sigma,\Phi) \subseteq \Phi$ is trivial for base types (Void, Nat, …) and universes (UPre, UKan).

Assume $\textsc{Fun}(\Phi)(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. Then by $A \sim A' \downarrow \alpha \in \Phi\ [\Psi]$ and $\mathsf{Coh}(\alpha)$, we have $\Phi^{\Downarrow}(\Psi', A\psi, A'\psi, \alpha_\psi)$, $A\psi_1 \Downarrow A_1$, $A'\psi_1 \Downarrow A'_1$, $\Phi^{\Downarrow}(\Psi_2, A_1\psi_2, A\psi_1\psi_2, \varphi')$, and so forth. Note that on values, $A_0 \sim B_0 \downarrow \alpha \in \tau\ [\Psi]$ implies $\tau(\Psi, A_0, B_0, \alpha_{\mathsf{id}_\Psi})$. Therefore $\mu^{\mathsf{pre}}(\nu,\sigma)^{\Downarrow}(\Psi', A\psi, A'\psi, \alpha_\psi)$, and so forth. Similar facts hold for B, B', β by $a : \alpha \triangleright B \sim B' \downarrow \beta \in \Phi\ [\Psi]$ and $\mathsf{CohFam}(\beta)$. We must show $\Phi(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi')$, that is, $(a{:}A) \to B \sim (a{:}A') \to B' \downarrow \gamma \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$. We know $(a{:}A) \to B\ \mathsf{val}_{\square}$, and by the above, $A \sim A' \downarrow \alpha \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$, $\mathsf{Coh}(\alpha)$, and for all $M \sim M' \in \alpha\psi\ [\Psi']$, $B\psi[M/a] \sim B'\psi[M'/a] \downarrow \beta^{\psi,M,M'} \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi']$ and $\mathsf{Coh}(\beta^{\psi,M,M'})$. The result holds because value-coherence and the candidate cubical judgments are closed under interval substitution.

The V case is more complicated because not all instances of $\mathsf{V}_x(A,B,E)$ have the same head constructor. As before, by $\mathsf{V}(\Phi)(\Psi, \mathsf{V}_x(A,B,E), \mathsf{V}_x(A',B',E'), \varphi)$ we have $B \sim B' \downarrow \beta \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$ and for all ψ with $x\psi = 0$, $A\psi \sim A'\psi \downarrow \alpha^{\psi} \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi']$. However, to see that $\mathsf{V}_x(A,B,E) \sim \mathsf{V}_x(A',B',E') \downarrow \gamma \in \mu^{\mathsf{pre}}(\nu,\sigma)\ [\Psi]$, we must consider the cases in which $\mathsf{V}_{x\psi}(A\psi, B\psi, E\psi)$ steps to $A\psi$ or $B\psi$, and show that for every ψ_1, ψ_2, the appropriate relations hold in $\mu^{\mathsf{pre}}(\nu,\sigma)$. See Rule 4.78 for full details, and Lemma 4.88 and Rule 4.87 for corresponding results about HKan and HPre. □

Theorem 4.9. *For all $i \in \{0, 1, \dots, \omega\}$, ν_i, τ_i^{pre}, and τ_i^{Kan} are cubical type systems.*

Proof. By strong induction on i. In the base case, ν_0 is empty and thus a cubical type system; therefore, by Lemma 4.8, so are $\tau_0^{\mathsf{Kan}} = \mu^{\mathsf{Kan}}(\nu_0)$ and $\tau_0^{\mathsf{pre}} = \mu^{\mathsf{pre}}(\nu_0, \mu^{\mathsf{Kan}}(\nu_0))$. In the inductive case, suppose τ_i^{κ} are cubical type systems for $i < n$. Then ν_n is a cubical type system: unicity, symmetry, transitivity, and value-coherence are immediate; PER-valuation follows from symmetry and transitivity of τ_i^{κ} for $i < n$. By Lemma 4.8, τ_n^{Kan} and thence τ_n^{pre} are cubical type systems. (The ω case is identical.) □

Theorem 4.10. *For $i, j \in \{0, 1, \dots, \omega\}$ where $i \le j$, we have $\tau_i^{\kappa} \subseteq \tau_j^{\kappa}$ and $\tau_i^{\mathsf{Kan}} \subseteq \tau_i^{\mathsf{pre}}$.*

Proof. The functions P and K are monotone in all arguments, so by Lemma 2.9, μ^{pre} and μ^{Kan} are as well. When $i \le j$, $\nu_i \subseteq \nu_j$ by construction, so $\tau_i^{\mathsf{Kan}} = \mu^{\mathsf{Kan}}(\nu_i) \subseteq \mu^{\mathsf{Kan}}(\nu_j) = \tau_j^{\mathsf{Kan}}$, and similarly, $\tau_i^{\mathsf{pre}} = \mu^{\mathsf{pre}}(\nu_i, \tau_i^{\mathsf{Kan}}) \subseteq \mu^{\mathsf{pre}}(\nu_j, \tau_j^{\mathsf{Kan}}) = \tau_j^{\mathsf{pre}}$.

By a theorem of Bekić [Bek84] on simultaneous fixed points, for all ν,

$$(\mu^{\mathsf{Kan}}(\nu), \mu^{\mathsf{pre}}(\nu, \mu^{\mathsf{Kan}}(\nu))) = \mu(\sigma,\tau).(K(\nu,\sigma), P(\nu,\sigma,\tau))$$

When $\sigma \subseteq \tau$, $K(\nu,\sigma) \subseteq P(\nu,\sigma,\tau)$; thus $\tau_i^{\mathsf{Kan}} \subseteq \tau_i^{\mathsf{pre}}$ follows by Lemma 2.10. □

4.3 Cubical judgments and Kan operations

The typehood and membership judgments of Cartesian cubical type theory mirror those of Idealized Nuprl, parametrized by finite sets of interval variables Ψ. This section nevertheless differs from Section 2.3 in two key respects. First, the candidate cubical judgments (Definition 4.5) are more complex than Idealized Nuprl's candidate judgments (Definition 2.3), and must be established by more sophisticated means—one must consider the evaluation behavior of all interval substitution instances of a term, not only the term itself. Secondly, we must define the *uniform Kan operations* (Definition 4.29), introduced in Chapter 3, which serve an analogous role to Book HoTT's identity elimination.

Definition 4.11 (Closed judgments)**.** Given a cubical type system τ:

1. $A \doteq B$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ when $A \sim B \downarrow \alpha \in \tau$ $[\Psi]$ and $\mathsf{Coh}(\alpha)$.
2. $M \doteq N \in A$ $[\Psi]$, presupposing $A \doteq A$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$, when $A \sim A \downarrow \alpha \in \tau$ $[\Psi]$ and $M \sim N \in \alpha$ $[\Psi]$.

The closed judgments are symmetric and transitive, and therefore $A \doteq A$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ whenever $A \doteq B$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$. As in Section 2.3, we abbreviate $A \doteq A$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ by A $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ and $M \doteq M \in A$ $[\Psi]$ by $M \in A$ $[\Psi]$ (and similarly for all subsequent judgments). Every judgment $\mathcal{J}$ $[\Psi]$ presupposes implicitly that $\mathsf{FI}(\mathcal{J}) \subseteq \Psi$. All judgments are relative to a choice of cubical type system τ, which we can notate explicitly by $\tau \models \mathcal{J}$ $[\Psi]$; when unspecified, judgments are relative to $\tau_\omega^{\mathsf{pre}}$. When $A \sim B \downarrow \alpha \in \tau$ $[\Psi]$ and τ is a cubical type system, the Ψ-PER α is uniquely determined and independent of B, so we write $[\![A]\!]$ for α.

As before, equal pretypes have equal elements; moreover, the closed judgments are preserved by interval substitutions.

Lemma 4.12. *If* $A \doteq B$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ *and* $M \doteq N \in A$ $[\Psi]$ *then* $M \doteq N \in B$ $[\Psi]$.

Proof. By the unicity and symmetry properties of τ, $[\![A]\!] = [\![B]\!]$; thus $M \sim N \in [\![A]\!]$ $[\Psi]$ implies $M \sim N \in [\![B]\!]$ $[\Psi]$. □

Lemma 4.13. *Supposing* $\psi : \Psi' \to \Psi$*:*

1. *If* $A \doteq B$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ *then* $A\psi \doteq B\psi$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi']$.
2. *If* $M \doteq N \in A$ $[\Psi]$ *then* $M\psi \doteq N\psi \in A\psi$ $[\Psi']$.

Proof. For part (1), $A \sim B \downarrow [\![A]\!] \in \tau$ $[\Psi]$ implies $A\psi \sim B\psi \downarrow [\![A]\!]\psi \in \tau$ $[\Psi']$ by precomposing ψ_1 with ψ in Definition 4.5, and by the unicity property of τ. Similarly, $\mathsf{Coh}([\![A]\!]\psi)$ follows from $\mathsf{Coh}([\![A]\!])$ by precomposition with ψ in Definition 4.7. For part (2), the presupposition $A\psi$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi']$ is resolved by part (1); the result follows by $[\![A\psi]\!] = [\![A]\!]\psi$ and precomposing ψ_1 with ψ in Definition 4.5. □

The judgments of Idealized Nuprl are closed under both forward (Lemma 2.16) and backward (Lemma 2.15) evaluation, essentially by definition. Cubical typehood and membership are also closed under forward evaluation to a value, but the arguments are more subtle and require value-coherence (of the cubical type system and of pretypes, respectively).

Lemma 4.14 (Evaluation I)**.** *If* A $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$ *then* $A \Downarrow A_0$ *where* $A \doteq A_0$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$.

Proof. We obtain $A \Downarrow A_0$ by instantiating $A \sim A \downarrow [\![A]\!] \in \tau\ [\Psi]$ at substitutions id_Ψ and id_Ψ. It remains only to show $A \sim A_0 \downarrow [\![A]\!] \in \tau\ [\Psi]$; $\mathsf{Coh}([\![A]\!])$ follows from A $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$. Suppose $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$. We must show that $\tau^{\Downarrow}(\Psi_2, A\psi_1\psi_2, A_1\psi_2, \varphi)$, $\tau^{\Downarrow}(\Psi_2, A\psi_1\psi_2, A_0\psi_1\psi_2, \varphi')$, and $\tau^{\Downarrow}(\Psi_2, A_1\psi_2, A_1'\psi_2, \varphi'')$ where $A\psi_1 \Downarrow A_1$ and $A_0\psi_1 \Downarrow A_1'$; the remaining conditions of Definition 4.5 follow from unicity, symmetry, and transitivity of τ.

The first holds by $A \sim A \downarrow [\![A]\!] \in \tau\ [\Psi]$ at the substitutions ψ_1 and ψ_2. The second holds by $A \sim A \downarrow [\![A]\!] \in \tau\ [\Psi]$ at id_Ψ and $\psi_1\psi_2$. For the third, $A \sim A \downarrow [\![A]\!] \in \tau\ [\Psi]$ at $\psi_1, \mathsf{id}_{\Psi_1}$ implies $\tau^{\Downarrow}(\Psi_1, A_1, A\psi_1, \_)$ and at id_Ψ, ψ_1 implies $\tau^{\Downarrow}(\Psi_1, A_0\psi_1, A\psi_1, \_)$. By transitivity, $\tau(\Psi_1, A_1, A_1', \_)$; by value-coherence of τ, $A_1 \sim A_1' \downarrow [\![A_1]\!] \in \tau\ [\Psi_1]$. Therefore $\tau^{\Downarrow}(\Psi_2, A_1\psi_2, A_1'\psi_2, \varphi'')$ as required. □

Lemma 4.15. *If* $M \in A\ [\Psi]$, $N \in A\ [\Psi]$, *and* $[\![A]\!]^{\Downarrow}(M, N)$, *then* $M \doteq N \in A\ [\Psi]$.

Proof. For all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$, we must show $[\![A]\!]^{\Downarrow}_{\psi_1\psi_2}(M\psi_1\psi_2, N\psi_1\psi_2)$; the remaining conditions of Definition 4.5 follow from $M \in A\ [\Psi]$ and $N \in A\ [\Psi]$. By presupposition we have $\mathsf{Coh}([\![A]\!])$, so $[\![A]\!]^{\Downarrow}(M, N)$ implies $M_0 \sim N_0 \in [\![A]\!]\ [\Psi]$ where $M \Downarrow M_0$ and $N \Downarrow N_0$. We conclude that $[\![A]\!]^{\Downarrow}_{\psi_1\psi_2}(M_0\psi_1\psi_2, N_0\psi_1\psi_2)$. From $M \in A\ [\Psi]$ we know $[\![A]\!]^{\Downarrow}_{\psi_1\psi_2}(M_0\psi_1\psi_2, M\psi_1\psi_2)$ and similarly for N; the result follows by transitivity. □

Lemma 4.16 (Evaluation II). *If* $M \in A\ [\Psi]$ *then* $M \doteq M_0 \in A\ [\Psi]$ *where* $M \Downarrow M_0$.

Proof. By $M \in A\ [\Psi]$ at $\mathsf{id}_\Psi, \mathsf{id}_\Psi$ we have $M \Downarrow M_0$ and $[\![A]\!](M_0, M_0)$; therefore, by $\mathsf{Coh}([\![A]\!])$, $M_0 \in A\ [\Psi]$. The result follows by Lemma 4.15. □

Lemma 4.15 is much weaker than its counterpart in Idealized Nuprl, in which $M \doteq N \in A$ follows directly from $[\![A]\!]^{\Downarrow}(M, N)$! That statement is easy to refute in cubical type theory, as the term $\mathbb{S}^1\text{-}\mathsf{elim}_{\_.\mathsf{bool}}(\mathsf{loop}_x; \mathsf{z}, \_.\mathsf{true})$ evaluates to true but cannot be a Boolean because its $\langle 0/x \rangle$ face evaluates to z. Nor does it suffice to check that $[\![A]\!]^{\Downarrow}_{\psi}(M\psi, N\psi)$ for all ψ—although all instances of $\mathbb{S}^1\text{-}\mathsf{elim}_{\_.\mathsf{bool}}(\mathsf{loop}_x; \mathsf{false}, \_.\mathsf{true})$ evaluate to Booleans, evaluation under id_Ψ then $\langle 0/x \rangle$ yields true, whereas evaluation under $\langle 0/x \rangle$ then id_Ψ yields false.

In Idealized Nuprl, we commonly use head expansion (or closure under backward evaluation) to establish judgments. Its analogue in this chapter is *coherent expansion*,[2] which states that a term is a pretype (resp., element) if all of its interval substitution instances eventually step to pretypes (resp., elements) in a suitably coherent way. More precisely, to prove A $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$, it suffices to exhibit a family of reducts $\{A^{\Psi'}_{\psi} \mid A\psi \longmapsto^* A^{\Psi'}_{\psi}\}_{\psi:\Psi'\to\Psi}$ for which $A^{\Psi'}_{\psi}$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi']$ and $A^{\Psi''}_{\psi\psi'} \doteq (A^{\Psi'}_{\psi})\psi'$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi'']$ (or equivalently, $A^{\Psi'}_{\psi} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi']$).

[2] A similar lemma appears independently in Huber's canonicity proof [Hub18, Lemma 9].

Lemma 4.17 (Coherent expansion I)**.** *Suppose A is a term and $\{A^{\Psi'}_{\psi}\}_{\psi:\Psi'\to\Psi}$ is a family of terms such that $A^{\Psi'}_{\psi} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$ and $A\psi \longmapsto^* A^{\Psi'}_{\psi}$ for all $\psi : \Psi' \to \Psi$. Then $A \doteq A^{\Psi}_{\mathsf{id}_\Psi}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.*

Proof. It suffices to check for all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$ that $A\psi_1 \Downarrow A_1$, $(A^{\Psi}_{\mathsf{id}_\Psi})\psi_1 \Downarrow A_1'$, and $\tau^{\Downarrow}(\Psi_2, -, -, \_)$ relates $A_1\psi_2$, $A\psi_1\psi_2$, $(A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2$, and $A_1'\psi_2$.

1. $A\psi_1 \Downarrow A_1$ and $\tau^{\Downarrow}(\Psi_2, A_1\psi_2, A\psi_1\psi_2, \varphi)$.

   By $A^{\Psi_1}_{\psi_1}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_1]$ at $\mathsf{id}_{\Psi_1}, \psi_2$ and $A\psi_1 \longmapsto^* A^{\Psi_1}_{\psi_1}$, we know $\tau^{\Downarrow}(\Psi_2, A_1\psi_2, (A^{\Psi_1}_{\psi_1})\psi_2, \varphi)$ where $A\psi_1 \Downarrow A_1$. By $A^{\Psi_1}_{\psi_1} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_1]$ and $(A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2 \doteq A^{\Psi_2}_{\psi_1\psi_2}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_2]$, we have $(A^{\Psi_1}_{\psi_1})\psi_2 \doteq A^{\Psi_2}_{\psi_1\psi_2}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_2]$ and thus $\tau^{\Downarrow}(\Psi_2, (A^{\Psi_1}_{\psi_1})\psi_2, A^{\Psi_2}_{\psi_1\psi_2}, \varphi)$. The result follows by transitivity of τ and $A\psi_1\psi_2 \longmapsto^* A^{\Psi_2}_{\psi_1\psi_2}$.

2. $\tau^{\Downarrow}(\Psi_2, A\psi_1\psi_2, (A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2, \varphi')$.

   By $A^{\Psi_2}_{\psi_1\psi_2} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_2]$ we have $\tau^{\Downarrow}(\Psi_2, A^{\Psi_2}_{\psi_1\psi_2}, (A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2, \varphi')$; the result follows by $A\psi_1\psi_2 \longmapsto^* A^{\Psi_2}_{\psi_1\psi_2}$.

3. $(A^{\Psi}_{\mathsf{id}_\Psi})\psi_1 \Downarrow A_1'$ and $\tau^{\Downarrow}(\Psi_2, (A^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2, A_1'\psi_2, \varphi'')$.

   By $A^{\Psi}_{\mathsf{id}_\Psi}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$. □

Lemma 4.18 (Coherent expansion II). *Suppose $A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, M is a term, and $\{M^{\Psi'}_{\psi}\}_{\psi:\Psi'\to\Psi}$ is a family of terms such that $M^{\Psi'}_{\psi} \doteq (M^{\Psi}_{\mathsf{id}_\Psi})\psi \in A\psi\ [\Psi']$ and $M\psi \longmapsto^* M^{\Psi'}_{\psi}$ for all $\psi : \Psi' \to \Psi$. Then $M \doteq M^{\Psi}_{\mathsf{id}_\Psi} \in A\ [\Psi]$.*

Proof. It suffices to check for all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$ that $M\psi_1 \Downarrow M_1$, $(M^{\Psi}_{\mathsf{id}_\Psi})\psi_1 \Downarrow M_1'$, and $\llbracket A\rrbracket^{\Downarrow}_{\psi_1\psi_2}$ relates $M_1\psi_2$, $M\psi_1\psi_2$, $(M^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2$, and $M_1'\psi_2$.

1. $M\psi_1 \Downarrow M_1$ and $\llbracket A\rrbracket^{\Downarrow}_{\psi_1\psi_2}(M_1\psi_2, M\psi_1\psi_2)$.

   By $M^{\Psi_1}_{\psi_1} \in A\psi_1\ [\Psi_1]$ at $\mathsf{id}_{\Psi_1}, \psi_2$ and $M\psi_1 \longmapsto^* M^{\Psi_1}_{\psi_1}$, we know $\llbracket A\rrbracket^{\Downarrow}_{\psi_1\psi_2}(M_1\psi_2, (M^{\Psi_1}_{\psi_1})\psi_2)$ where $M\psi_1 \Downarrow M_1$. By $M^{\Psi_1}_{\psi_1} \doteq (M^{\Psi}_{\mathsf{id}_\Psi})\psi_1 \in A\psi_1\ [\Psi_1]$ and $(M^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2 \doteq M^{\Psi_2}_{\psi_1\psi_2} \in A\psi_1\psi_2\ [\Psi_2]$, we have $(M^{\Psi_1}_{\psi_1})\psi_2 \doteq M^{\Psi_2}_{\psi_1\psi_2} \in A\psi_1\psi_2\ [\Psi_2]$ and thus $\llbracket A\rrbracket^{\Downarrow}_{\psi_1\psi_2}((M^{\Psi_1}_{\psi_1})\psi_2, M^{\Psi_2}_{\psi_1\psi_2})$. The result follows by transitivity of τ and $M\psi_1\psi_2 \longmapsto^* M^{\Psi_2}_{\psi_1\psi_2}$.

2. $\llbracket A\rrbracket^{\Downarrow}_{\psi_1\psi_2}(M\psi_1\psi_2, (M^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2)$.

   By $M^{\Psi_2}_{\psi_1\psi_2} \doteq (M^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2 \in A\psi_1\psi_2\ [\Psi_2]$ we have $\llbracket A\rrbracket^{\Downarrow}_{\psi_1\psi_2}(M^{\Psi_2}_{\psi_1\psi_2}, (M^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2)$; the result follows by $M\psi_1\psi_2 \longmapsto^* M^{\Psi_2}_{\psi_1\psi_2}$.

3. $(M^{\Psi}_{\mathsf{id}_\Psi})\psi_1 \Downarrow M'_1$ and $[\![A]\!]^{\Downarrow}_{\psi_1\psi_2}((M^{\Psi}_{\mathsf{id}_\Psi})\psi_1\psi_2, M'_1\psi_2)$.

   By $M^{\Psi}_{\mathsf{id}_\Psi} \in A\ [\Psi]$. □

In the special case that $M \longmapsto^*_{\square} M'$ and $M' \in A\ [\Psi]$, we can invoke Lemma 4.18 with M and the family $\{M'\psi\}_{\psi:\Psi'\to\Psi}$, because $M'\psi \in A\psi\ [\Psi']$ (where $M'\psi = (M'\mathsf{id}_\Psi)\psi$ on the nose) and $M\psi \longmapsto^* M'\psi$ for all $\psi : \Psi' \to \Psi$. We therefore obtain head expansion with respect to *cubically-stable* evaluation, as we observe in Lemma 4.32.

4.3.1 *Open judgments*

As in Idealized Nuprl, open judgments express that an equality holds as a closed judgment for all instantiations by elements of the hypotheses Γ. Our notation $\Gamma \gg \mathcal{J}\ [\Psi]$ distinguishes the contexts Γ and Ψ in accordance with their differing semantics. One can instead interleave term $(a : A)$ and interval $(x : \mathbb{I})$ variables in a single context; in fact, all current cubical proof assistants do so, including **RedPRL**, **redtt**, and cubical Agda.

Definition 4.19. For γ, γ' lists and Γ a telescope (Definition 2.17), $\gamma \sim \gamma' \in \Gamma\ [\Psi]$ when

1. $\cdot \sim \cdot \in \cdot\ [\Psi]$, or

2. $(M, \gamma) \sim (M', \gamma') \in (a : A, \Gamma)\ [\Psi]$ when $M \doteq M' \in A\ [\Psi]$ and $\gamma \sim \gamma' \in \Gamma[M/a]\ [\Psi]$.

Definition 4.20 (Contexts)**.** For Γ, Γ' telescopes, $\Gamma \doteq \Gamma'\ \mathsf{ctx}\ [\Psi]$ when

1. $\cdot \doteq \cdot\ \mathsf{ctx}\ [\Psi]$, or

2. $(a : A, \Gamma) \doteq (a : A', \Gamma')\ \mathsf{ctx}\ [\Psi]$ when $A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$ and for all $\psi : \Psi' \to \Psi$ and $M \doteq M' \in A\psi\ [\Psi']$, $\Gamma\psi[M/a] \doteq \Gamma'\psi[M'/a]\ \mathsf{ctx}\ [\Psi']$.

The relation $\gamma \sim \gamma' \in \Gamma\ [\Psi]$ is symmetric and transitive when $\Gamma\ \mathsf{ctx}\ [\Psi]$; the judgment $\Gamma \doteq \Gamma'\ \mathsf{ctx}\ [\Psi]$ is always symmetric and transitive. Both are closed under interval substitution as a straightforward consequence of Lemma 4.13.

Definition 4.21 (Open judgments)**.**

1. $\Gamma \gg A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$, presupposing $\Gamma\ \mathsf{ctx}\ [\Psi]$, when for all $\psi : \Psi' \to \Psi$ and $\gamma \sim \gamma' \in \Gamma\psi\ [\Psi']$, $A\psi\gamma \doteq A'\psi\gamma'\ \mathsf{type_{pre}}\ [\Psi']$.

2. $\Gamma \gg M \doteq M' \in A\ [\Psi]$, presupposing $\Gamma \gg A\ \mathsf{type_{pre}}\ [\Psi]$, when for all $\psi : \Psi' \to \Psi$ and $\gamma \sim \gamma' \in \Gamma\psi\ [\Psi']$, $M\psi\gamma \doteq M'\psi\gamma' \in A\psi\gamma\ [\Psi']$.

The open judgments are symmetric, transitive, and closed under interval substitution. It remains to prove that contexts can be extended on the right by open types, and that the open judgments satisfy the *structural rules* of hypothesis, weakening, and substitution.

Lemma 4.22 (Context extension)**.** *If* $\Gamma \doteq \Gamma'\ \mathsf{ctx}\ [\Psi]$ *and* $\Gamma \gg A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$ *then* $(\Gamma, a:A) \doteq (\Gamma', a:A')\ \mathsf{ctx}\ [\Psi]$.

Proof. The telescope $(\Gamma, a:A)$ is defined by recursion on Γ, so we prove the lemma by induction on $\Gamma \doteq \Gamma'\ \mathsf{ctx}\ [\Psi]$. If $\Gamma = \Gamma' = \cdot$, we must show that if $A\psi \doteq A'\psi\ \mathsf{type_{pre}}\ [\Psi']$ for all $\psi : \Psi' \to \Psi$ then $A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$, which is immediate.

Now suppose $(b:B,\Gamma) \doteq (b:B',\Gamma')\ \mathsf{ctx}\ [\Psi]$ and $b:B,\Gamma \gg A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$, and show $((b:B,\Gamma), a:A) \doteq ((b:B',\Gamma'), a:A')\ \mathsf{ctx}\ [\Psi]$. By definition, $((b:B,\Gamma),a:A) = (b:B,(\Gamma,a:A))$. We know $B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$, and must show for any $\psi : \Psi' \to \Psi$ and $N \doteq N' \in B\psi\ [\Psi']$ that $(\Gamma\psi[N/b], a:A\psi[N/b]) \doteq (\Gamma'\psi[N'/b], a:A'\psi[N'/b])\ \mathsf{ctx}\ [\Psi']$.

It suffices to establish $\Gamma\psi[N/b] \gg A\psi[N/b] \doteq A'\psi[N'/b]\ \mathsf{type_{pre}}\ [\Psi']$; the result then follows from the inductive hypothesis and $\Gamma\psi[N/b] \doteq \Gamma'\psi[N'/b]\ \mathsf{ctx}\ [\Psi']$. We must show for all $\psi' : \Psi'' \to \Psi'$ and $\gamma \sim \gamma' \in \Gamma\psi[N/b]\psi'\ [\Psi'']$ that $A\psi[N/b]\psi'\gamma \doteq A'\psi[N'/b]\psi'\gamma'\ \mathsf{type_{pre}}\ [\Psi'']$. This follows from $b{:}B,\Gamma \gg A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$ instantiated at $\psi\psi'$ and $(N\psi',\gamma) \sim (N'\psi',\gamma') \in (b:B,\Gamma)\psi\psi'\ [\Psi'']$. □

Lemma 4.23 (Hypothesis)**.** *If* $(\Gamma, a:A, \Delta)\ \mathsf{ctx}\ [\Psi]$ *then* $\Gamma, a:A, \Delta \gg a \in A\ [\Psi]$.

Proof. First, we establish the presupposition $\Gamma, a:A,\Delta \gg A\ \mathsf{type_{pre}}\ [\Psi]$ by induction on Γ, unrolling the definition of $(\Gamma, a:A,\Delta)\ \mathsf{ctx}\ [\Psi]$, gathering iterated interval substitutions (as in the previous proof), and observing that a and the variables of Δ cannot occur in A. To establish the judgment itself, suppose $\psi : \Psi' \to \Psi$ and $(\gamma, M, \delta) \sim (\gamma', M', \delta') \in (\Gamma, a:A,\Delta)\psi\ [\Psi']$ (hence $\gamma \sim \gamma' \in \Gamma\psi\ [\Psi']$ and $M \doteq M' \in A\psi\gamma\ [\Psi']$); we must show $M \doteq M' \in A\psi\gamma\ [\Psi']$, which is immediate. □

Lemma 4.24 (Weakening)**.** *Supposing* $\Gamma \gg A\ \mathsf{type_{pre}}\ [\Psi]$*:*

1. *If* $\Gamma, \Delta \gg B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$ *then* $\Gamma, a:A, \Delta \gg B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$.

2. *If* $\Gamma, \Delta \gg M \doteq M' \in B\ [\Psi]$ *then* $\Gamma, a:A,\Delta \gg M \doteq M' \in B\ [\Psi]$.

Proof. For part (1), as previously, we establish the presupposition $(\Gamma, a:A,\Delta)\ \mathsf{ctx}\ [\Psi]$ by induction on Γ, observing for all $\psi : \Psi' \to \Psi$ and $\gamma \sim \gamma' \in \Gamma\psi\ [\Psi']$ that $A\psi\gamma \doteq A\psi\gamma'\ \mathsf{type_{pre}}\ [\Psi']$ and $\Delta\psi\gamma \doteq \Delta\psi\gamma'\ \mathsf{ctx}\ [\Psi']$. To establish the judgment, suppose $(\gamma, M, \delta) \sim (\gamma', M', \delta') \in (\Gamma, a:A,\Delta)\psi\ [\Psi']$ and show $B\psi\gamma[M/a]\delta \doteq B'\psi\gamma'[M'/a]\delta'\ \mathsf{type_{pre}}\ [\Psi']$. Because a cannot occur in B, this is exactly $B\psi\gamma\delta \doteq B'\psi\gamma'\delta'\ \mathsf{type_{pre}}\ [\Psi']$, which is immediate. Part (2) follows similarly. □

Lemma 4.25 (Substitution)**.** *Supposing* $\Gamma \gg M \doteq M' \in A\ [\Psi]$*:*

1. *If* $\Gamma, a:A, \Delta \gg B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$ *then* $\Gamma, \Delta[M/a] \gg B[M/a] \doteq B'[M'/a]\ \mathsf{type_{pre}}\ [\Psi]$.

2. *If* $\Gamma, a{:}A, \Delta \gg N \doteq N' \in B\ [\Psi]$ *then* $\Gamma, \Delta[M/a] \gg N[M/a] \doteq N'[M'/a] \in B[M/a]\ [\Psi]$.

Proof. For part (1), once again we establish the presupposition $(\Gamma, \Delta[M/a])$ ctx $[\Psi]$ by induction on Γ, observing for all $\psi : \Psi' \to \Psi$ and $\gamma \sim \gamma' \in \Gamma\psi\ [\Psi']$ that $(\gamma, M\psi\gamma) \sim (\gamma', M\psi\gamma') \in (\Gamma, a : A)\psi\ [\Psi']$ and thus $(\Delta[M/a])\psi\gamma \doteq (\Delta[M/a])\psi\gamma'$ ctx $[\Psi']$ (because $(\Delta[M/a])\psi\gamma = \Delta\psi\gamma[M\psi\gamma/a]$). To establish the judgment, show if $(\gamma, \delta) \sim (\gamma', \delta') \in (\Gamma, \Delta[M/a])\psi\ [\Psi']$ then $B[M/a]\psi\gamma\delta \doteq B'[M'/a]\psi\gamma'\delta'$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi']$, which follows by $B[M/a]\psi\gamma\delta = B\psi\gamma[M\psi\gamma/a]\delta$ and $(\gamma, M\psi\gamma, \delta) \sim (\gamma', M'\psi\gamma', \delta') \in (\Gamma, a : A, \Delta)\psi\ [\Psi']$. Part (2) follows similarly. □

4.3.2 *Kan operations*

As we explained in Chapter 3, cubical type theories are motivated by two key insights. First, interval variables allow us to generalize the typehood and membership judgments to arbitrary dimension, and thereby represent the arbitrary-dimensional data induced by univalence and higher inductive types. Secondly, uniform Kan operations equip these types with structure that in Book HoTT is induced by identity elimination: composition and inversion of paths, transport, *et cetera*.

We therefore define a new judgment expressing that a type is *Kan*, or equipped with the uniform Kan operations of coercion and homogeneous composition described in Section 3.2. This judgment has similar force to the typehood judgment of Cohen et al. [CCHM18], in whose type theory all types are Kan. In contrast, we explicitly consider pretypes that are not Kan, including many equality types.

Like Cohen et al. [CCHM18], we express open boxes using *restricted judgments*, which limit the force of a cubical judgment to a subcube of Ψ [Ang+19].

Definition 4.26. An interval substitution $\psi : \Psi' \to \Psi$ *satisfies* a set of unordered equational constraints $\Xi = \{r_1 = r_1', \dots, r_n = r_n'\}$ for which $\mathsf{FI}(\Xi) \subseteq \Psi$ when for all i, $r_i\psi = r_i'\psi$.

Definition 4.27 (Restricted judgments)**.** For every judgment form $\mathcal{J}$, we say $\mathcal{J}\ [\Psi \mid \Xi]$, presupposing $\mathsf{FI}(\Xi) \subseteq \Psi$, when $\mathcal{J}\psi\ [\Psi']$ for all $\psi : \Psi' \to \Psi$ satisfying Ξ.

Restricted judgments maintain the presuppositions of their unrestricted forms; for instance, $M \in A\ [\Psi \mid \Xi]$ presupposes A $\mathsf{type}_{\mathsf{pre}}$ $[\Psi \mid \Xi]$. (This presupposition is needed to ensure the presupposition of $M\psi \in A\psi\ [\Psi']$ for all $\psi : \Psi' \to \Psi$ satisfying Ξ.) One can translate any $\mathcal{J}\ [\Psi \mid \Xi]$ into an unrestricted judgment by case analysis on Ξ:

1. $\mathcal{J}\ [\Psi \mid \cdot]$ if and only if $\mathcal{J}\ [\Psi]$.

   Both are equivalent to $\mathcal{J}\psi\ [\Psi']$ for all ψ, because all interval substitutions satisfy $\cdot$, and all judgment forms are closed under interval substitution.

2. $\mathcal{J}\ [\Psi \mid \Xi, r = r]$ if and only if $\mathcal{J}\ [\Psi \mid \Xi]$.

   An interval substitution satisfies $\{\Xi, r = r\}$ if and only if it satisfies Ξ.

3. $\mathcal{J}\ [\Psi \mid \Xi, 0 = 1]$ always.

   No interval substitutions satisfy $\{\Xi, 0 = 1\}$.

4. $\mathcal{J}\ [\Psi, x \mid \Xi, x = r]$ if and only if $\mathcal{J}\langle r/x\rangle\ [\Psi \mid \Xi\langle r/x\rangle]$.

   An interval substitution $\psi : \Psi' \to (\Psi, x)$ satisfies $\{\Xi, x = r\}$ if and only if $\psi = \langle r/x\rangle\psi'$ and $\psi' : \Psi' \to \Psi$ satisfies $\Xi\langle r/x\rangle$.

Restrictions are nevertheless critical because they specify subcubes in a manner that is both closed under interval substitution (because ψ' satisfies $\Xi\psi$ if and only if $\psi\psi'$ satisfies Ξ) and amenable to intersection. To illustrate the latter point, consider three lines in the $x = 0$, $x = 1$, and $x = y$ instances of A $\mathsf{type_{pre}}\ [x, y]$ that agree on their intersections. Using restricted judgments, we can express such a scenario with three singly-restricted elements and three doubly-restricted equations (below, left). Using interval substitutions, it is easy to express the three lines themselves, but their intersections are not computed uniformly—the intersection of $x = 0$ and $x = 1$ is empty, whereas the intersection of $x = \varepsilon$ and $x = y$ corresponds to the substitution $\langle\varepsilon/x\rangle\langle\varepsilon/y\rangle$ (below, right):

$$
\begin{array}{c|c}
\begin{array}{c}
M \in A\ [x, y \mid x = 0] \\
N \in A\ [x, y \mid x = 1] \\
O \in A\ [x, y \mid x = y] \\
\hline
M \doteq N \in A\ [x, y \mid x = 0, x = 1] \\
M \doteq O \in A\ [x, y \mid x = 0, x = y] \\
N \doteq O \in A\ [x, y \mid x = 1, x = y]
\end{array}
&
\begin{array}{c}
M \in A\langle 0/x\rangle\ [y] \\
N \in A\langle 1/x\rangle\ [y] \\
O \in A\langle y/x\rangle\ [y] \\
\hline
- \\
M\langle 0/y\rangle \doteq O\langle 0/y\rangle \in A\langle 0/x\rangle\langle 0/y\rangle\ [\cdot] \\
N\langle 1/y\rangle \doteq O\langle 1/y\rangle \in A\langle 1/x\rangle\langle 1/y\rangle\ [\cdot]
\end{array}
\end{array}
$$

We write A $\mathsf{type_{Kan}}\ [\Psi]$ when A is a pretype whose interval substitution instances admit the Cartesian cubical Kan operations (Figure 3.4), restricted to *valid* open box shapes.[3] As discussed in Section 3.5, the validity condition is defeasible (Angiuli et al. [Ang+19] omit it) and engenders additional complications (Theorem 4.34), but allows us to obtain a sharper canonicity result in which 0-dimensional elements of higher inductive types evaluate to constructors (Theorem 4.77).

Definition 4.28. A set of equational constraints $\overrightarrow{r_i = r_i'}$ is *valid* when either $r_i = r_i'$ for some i, or $r_i = r_j$, $r_i' = 0$, and $r_j' = 1$ for some i, j.

Definition 4.29 (Kan types). $A \doteq B$ $\mathsf{type_{Kan}}\ [\Psi]$, presupposing $A \doteq B$ $\mathsf{type_{pre}}\ [\Psi]$, when for any $\psi : \Psi' \to \Psi$:

[3] More precisely, we require not only that $A\psi$ admit Kan operations but moreover that the particular terms $\mathsf{hcom}_{A\psi}$ and $\mathsf{coe}_{x.A\psi}$ define such operations.

1. If

   a) $\overrightarrow{\xi_i}$ is valid,

   b) $M \doteq M' \in A\psi\ [\Psi']$,

   c) $N_i \doteq N'_j \in A\psi\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and

   d) $N_i\langle r/y\rangle \doteq M \in A\psi\ [\Psi' \mid \xi_i]$ for any i,

   then

   a) $\mathsf{hcom}^{r\rightsquigarrow r'}_{A\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{B\psi}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in A\psi\ [\Psi']$;

   b) if $r = r'$ then $\mathsf{hcom}^{r\rightsquigarrow r'}_{A\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in A\psi\ [\Psi']$; and

   c) if ξ_i then $\mathsf{hcom}^{r\rightsquigarrow r'}_{A\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in A\psi\ [\Psi']$.

2. If $\Psi' = (\Psi'', x)$ and $M \doteq M' \in A\psi\langle r/x\rangle\ [\Psi'']$, then

   a) $\mathsf{coe}^{r\rightsquigarrow r'}_{x.A\psi}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{x.B\psi}(M') \in A\psi\langle r'/x\rangle\ [\Psi'']$; and

   b) if $r = r'$ then $\mathsf{coe}^{r\rightsquigarrow r'}_{x.A\psi}(M) \doteq M \in A\psi\langle r/x\rangle\ [\Psi'']$.

$\Gamma \gg A \doteq A'\ \mathsf{type_{Kan}}\ [\Psi]$, presupposing $\Gamma \gg A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$, when for all $\psi : \Psi' \to \Psi$ and $\gamma \sim \gamma' \in \Gamma\psi\ [\Psi']$, $A\psi\gamma \doteq A'\psi\gamma'\ \mathsf{type_{Kan}}\ [\Psi']$.

The closed and open Kan judgments are symmetric, transitive, and closed under interval substitution. We can establish further properties of these judgments by appealing to earlier evaluation and coherent expansion lemmas.

Lemma 4.30. *Suppose* $A\ \mathsf{type_{Kan}}\ [\Psi]$, $B\ \mathsf{type_{Kan}}\ [\Psi]$, *and for all* $\psi : \Psi' \to \Psi$ *we have* $A_\psi \doteq B_\psi\ \mathsf{type_{Kan}}\ [\Psi']$ *where* $A\psi \Downarrow A_\psi$ *and* $B\psi \Downarrow B_\psi$. *Then* $A \doteq B\ \mathsf{type_{Kan}}\ [\Psi]$.

Proof. Let $\psi : \Psi' \to \Psi$. By Lemma 4.14, $A\psi \doteq A_\psi\ \mathsf{type_{pre}}\ [\Psi']$ and $B\psi \doteq B_\psi\ \mathsf{type_{pre}}\ [\Psi']$; therefore $A\psi \doteq B\psi\ \mathsf{type_{pre}}\ [\Psi']$, and in particular, the presupposition $A \doteq B\ \mathsf{type_{pre}}\ [\Psi]$ holds. Now, suppose

1. $\overrightarrow{\xi_i}$ is valid,

2. $M \doteq M' \in A\psi\ [\Psi']$,

3. $N_i \doteq N'_j \in A\psi\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and

4. $N_i\langle r/y\rangle \doteq M \in A\psi\ [\Psi' \mid \xi_i]$ for any i,

and show $\mathsf{hcom}^{r \rightsquigarrow r'}_{A\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r \rightsquigarrow r'}_{B\psi}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}) \in A\psi\ [\Psi']$. Both sides of this equation are elements of $A\psi$ (by Definition 4.29 and $A\psi \doteq B\psi\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$), so by Lemma 4.15 it suffices to check that these terms are related by $\llbracket A\psi \rrbracket^{\Downarrow}$ or equivalently $\llbracket A_\psi \rrbracket^{\Downarrow}$. This holds by $\mathsf{hcom}_{A_\psi} \doteq \mathsf{hcom}_{B_\psi} \in A_\psi\ [\Psi']$ (using $A_\psi \doteq B_\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$ and, for the premises, $A\psi \doteq A_\psi\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$), $\mathsf{hcom}_{A\psi} \longmapsto^* \mathsf{hcom}_{A_\psi}$, and $\mathsf{hcom}_{B\psi} \longmapsto^* \mathsf{hcom}_{B_\psi}$. The remaining hcom equations of Definition 4.29 follow by transitivity and the equations of $A_\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$; the coe equations follow similarly. □

Lemma 4.31 (Coherent expansion III). *Suppose A is a term and $\{A^{\Psi'}_{\psi}\}_{\psi:\Psi'\to\Psi}$ is a family of terms such that $A^{\Psi'}_{\psi} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$ and $A\psi \longmapsto^* A^{\Psi'}_{\psi}$ for all $\psi : \Psi' \to \Psi$. Then $A \doteq A^{\Psi}_{\mathsf{id}_\Psi}\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.*

Proof. The presupposition $A \doteq A^{\Psi}_{\mathsf{id}_\Psi}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ is immediate by Lemma 4.17. Now, suppose $\psi : \Psi' \to \Psi$,

1. $\overrightarrow{\xi_i}$ is valid,
2. $M \doteq M' \in A\psi\ [\Psi']$,
3. $N_i \doteq N_j' \in A\psi\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and
4. $N_i\langle r/y\rangle \doteq M \in A\psi\ [\Psi' \mid \xi_i]$ for any i,

and show $\mathsf{hcom}^{r \rightsquigarrow r'}_{A\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r \rightsquigarrow r'}_{(A^{\Psi}_{\mathsf{id}_\Psi})\psi}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}) \in A\psi\ [\Psi']$. We apply Lemma 4.18 at $A\psi\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$ to the term $\mathsf{hcom}^{r \rightsquigarrow r'}_{A\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})$ and family of terms $\{\mathsf{hcom}^{r\psi' \rightsquigarrow r'\psi'}_{A^{\Psi''}_{\psi\psi'}}(M\psi'; \overrightarrow{\xi_i\psi' \hookrightarrow y.N_i\psi'})\}^{\Psi''}_{\psi'}$.

We know $\mathsf{hcom}_{A\psi\psi'} \longmapsto^* \mathsf{hcom}_{A^{\Psi''}_{\psi\psi'}}$ by $A\psi\psi' \longmapsto^* A^{\Psi''}_{\psi\psi'}$, and $\mathsf{hcom}_{A^{\Psi''}_{\psi\psi'}} \doteq \mathsf{hcom}_{(A^{\Psi'}_{\psi})\psi'} \in A\psi\psi'\ [\Psi'']$ by $A^{\Psi''}_{\psi\psi'} \doteq (A^{\Psi'}_{\psi})\psi'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'']$ and $A^{\Psi''}_{\psi\psi'} \doteq A\psi\psi'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi'']$ (both by transitivity through $(A^{\Psi}_{\mathsf{id}_\Psi})\psi\psi'$). We conclude that $\mathsf{hcom}_{A\psi} \doteq \mathsf{hcom}_{A^{\Psi'}_{\psi}} \in A\psi\ [\Psi']$, and the desired result follows by $A^{\Psi'}_{\psi} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$. The remaining hcom equations of Definition 4.29 follow by transitivity and $A^{\Psi}_{\mathsf{id}_\Psi}\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Now, suppose $\psi : (\Psi', x) \to \Psi$ and $M \doteq M' \in A\psi\langle r/x\rangle\ [\Psi']$ and show $\mathsf{coe}^{r \rightsquigarrow r'}_{x.A\psi}(M) \doteq \mathsf{coe}^{r \rightsquigarrow r'}_{x.(A^{\Psi}_{\mathsf{id}_\Psi})\psi}(M') \in A\psi\langle r'/x\rangle\ [\Psi']$. We apply Lemma 4.18 at $A\psi\langle r'/x\rangle\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$ to the term $\mathsf{coe}^{r \rightsquigarrow r'}_{x.A\psi}(M)$ and family of terms $\{\mathsf{coe}^{r\psi' \rightsquigarrow r'\psi'}_{x.A^{\Psi''}_{\psi\psi'}}(M\psi')\}^{\Psi''}_{\psi'}$. As above, we obtain $\mathsf{coe}_{x.A\psi} \doteq \mathsf{coe}_{x.A^{\Psi'}_{\psi}} \in A\psi\langle r'/x\rangle\ [\Psi']$, and the desired result follows by $A^{\Psi'}_{\psi} \doteq (A^{\Psi}_{\mathsf{id}_\Psi})\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', x]$. The remaining coe equation of Definition 4.29 follows similarly. □

Lemma 4.32 (Head expansion)**.**

1. *If* A' $\mathsf{type_{pre}}$ $[\Psi]$ *and* $A \longmapsto^*_{\square} A'$ *then* $A \doteq A'$ $\mathsf{type_{pre}}$ $[\Psi]$.
2. *If* $M' \in A\ [\Psi]$ *and* $M \longmapsto^*_{\square} M'$ *then* $M \doteq M' \in A\ [\Psi]$.
3. *If* A' $\mathsf{type_{Kan}}$ $[\Psi]$ *and* $A \longmapsto^*_{\square} A'$ *then* $A \doteq A'$ $\mathsf{type_{Kan}}$ $[\Psi]$.

Proof. Part (1) follows from Lemma 4.17 applied to the term A and family $\{A'\psi\}^{\Psi'}_{\psi}$, observing that $A'\psi$ $\mathsf{type_{pre}}$ $[\Psi']$ and $A\psi \longmapsto^* A'\psi$ for all ψ. Parts (2–3) follow similarly, appealing instead to Lemma 4.18 and Lemma 4.31 respectively. □

Using homogeneous composition, which fills open boxes in constant types, and coercion, which varies the type of a single element, we can define a *heterogeneous* composition that fills open boxes in a varying type. In fact, we have already seen an instance of heterogeneous composition in our definition of coercion in path types (Section 3.2).

Theorem 4.33 (Heterogeneous composition)**.** *If* $A \doteq B$ $\mathsf{type_{Kan}}$ $[\Psi, y]$,

1. $\overrightarrow{\xi_i}$ *is valid,*
2. $M \doteq M' \in A\langle r/y\rangle\ [\Psi]$,
3. $N_i \doteq N'_j \in A\ [\Psi, y \mid \xi_i, \xi_j]$ *for any* i, j, *and*
4. $N_i\langle r/y\rangle \doteq M \in A\langle r/y\rangle\ [\Psi \mid \xi_i]$ *for any* i,

then

1. $\mathsf{com}^{r\rightsquigarrow r'}_{y.A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{com}^{r\rightsquigarrow r'}_{y.B}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in A\langle r'/y\rangle\ [\Psi]$;
2. *if* $r = r'$ *then* $\mathsf{com}^{r\rightsquigarrow r'}_{y.A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in A\langle r/y\rangle\ [\Psi]$; *and*
3. *if* ξ_i *then* $\mathsf{com}^{r\rightsquigarrow r'}_{y.A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in A\langle r'/y\rangle\ [\Psi]$.

Proof. By hypothesis (3), $N_i\psi \doteq N'_j\psi \in A\psi\ [\Psi']$ for all $\psi : \Psi' \to (\Psi, y)$ satisfying ξ_i and ξ_j. By $A \doteq B$ $\mathsf{type_{Kan}}$ $[\Psi, y]$ we have $(\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i))\psi \doteq (\mathsf{coe}^{y\rightsquigarrow r'}_{y.B}(N'_j))\psi \in A\langle r'/y\rangle\psi\ [\Psi']$, and therefore $\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i) \doteq \mathsf{coe}^{y\rightsquigarrow r'}_{y.B}(N'_j) \in A\ [\Psi, y \mid \xi_i, \xi_j]$. Similarly, by hypothesis (4) we conclude $(\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i))\langle r/y\rangle \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M) \in A\langle r'/y\rangle\ [\Psi \mid \xi_i]$, and by hypothesis (2), $\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{y.B}(M') \in A\langle r'/y\rangle\ [\Psi]$. Therefore, by hypothesis (1) and

$A \doteq B\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, y]$:

$$\begin{aligned}&\mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle r'/y\rangle}(\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i)})\\ \doteq\ &\mathsf{hcom}^{r\rightsquigarrow r'}_{B\langle r'/y\rangle}(\mathsf{coe}^{r\rightsquigarrow r'}_{y.B}(M'); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}^{y\rightsquigarrow r'}_{y.B}(N_i')}) \in A\langle r'/y\rangle\ [\Psi]\end{aligned}$$

Part (1) follows by Lemma 4.32 on each side. Moreover, when $r = r'$,

$$\begin{aligned}\mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle r'/y\rangle}(\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i)}) &\doteq \mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M)\\ &\doteq M \in A\langle r'/y\rangle\ [\Psi]\end{aligned}$$

and when ξ_i,

$$\begin{aligned}\mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle r'/y\rangle}(\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i)}) &\doteq (\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i))\langle r'/y\rangle\\ &\doteq N_i\langle r'/y\rangle \in A\langle r'/y\rangle\ [\Psi].\end{aligned}$$

Parts (2–3) follow from the above facts and Lemma 4.32. □

Our definition of coercion in $x.\mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_j}(A; \overrightarrow{\xi_i \hookrightarrow z.B_i})$ requires filling open boxes of possibly-invalid shape—namely, the subset of $\overrightarrow{\xi_i}$ not containing x (Section 4.4.11).[4] Such box fillers are not instances of homogeneous composition in the sense of Definition 4.29; however, as discussed in Section 3.5, they are definable by recursion on the list of equations.

Theorem 4.34 (Generalized homogeneous composition). *If $A \doteq B\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$,*

1. *$M \doteq M' \in A\ [\Psi]$,*
2. *$N_i \doteq N_j' \in A\ [\Psi, y \mid \xi_i, \xi_j]$ for any i, j, and*
3. *$N_i\langle r/y\rangle \doteq M \in A\ [\Psi \mid \xi_i]$ for any i,*

then

1. *$\mathsf{ghcom}^{r\rightsquigarrow r'}_A(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{ghcom}^{r\rightsquigarrow r'}_B(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}) \in A\ [\Psi]$;*
2. *if $r = r'$ then $\mathsf{ghcom}^{r\rightsquigarrow r'}_A(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in A\ [\Psi]$; and*
3. *if ξ_i then $\mathsf{ghcom}^{r\rightsquigarrow r'}_A(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in A\ [\Psi]$.*

[4] Following Cohen et al. [CCHM18], Angiuli et al. [Ang+19] notate this filtering operation $\forall x$.

Proof. By induction on the length of $\overrightarrow{\xi_i}$. If $\overrightarrow{\xi_i} = \cdot$, part (1) requires $\mathsf{ghcom}_A^{r\rightsquigarrow r'}(M;\cdot) \doteq \mathsf{ghcom}_B^{r\rightsquigarrow r'}(M';\cdot) \in A\ [\Psi]$, which is immediate by Lemma 4.32 on both sides; part (2) is immediate by Lemma 4.32 on the left; and part (3) is impossible.

Now consider $\mathsf{ghcom}_A^{r\rightsquigarrow r'}(M; s = s' \hookrightarrow y.N, \overrightarrow{\xi_i \hookrightarrow y.N_i})$, armed with the inductive hypothesis that ghcoms with one fewer attached face satisfy properties (1–3). By Lemma 4.32 we must show (the binary form of)

$$\mathsf{hcom}_A^{r\rightsquigarrow r'}(M; \overrightarrow{s = \varepsilon \hookrightarrow z.T_\varepsilon}, s = s' \hookrightarrow y.N, \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\ [\Psi] \text{ where}$$
$$T_\varepsilon = \mathsf{hcom}_A^{r\rightsquigarrow z}(M; s' = \varepsilon \hookrightarrow y.N, s' = \overline{\varepsilon} \hookrightarrow y.\mathsf{ghcom}_A^{r\rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}), \overrightarrow{\xi_i \hookrightarrow y.N_i})$$

writing $\overline{\varepsilon}$ for 1 when $\varepsilon = 0$ and *vice versa*. Let us first show $T_\varepsilon \in A\ [\Psi, z \mid s = \varepsilon]$ by Definition 4.29, noting that the composition is valid by $s' = \varepsilon, s' = \overline{\varepsilon}$ and that

1. $M \in A\ [\Psi \mid s = \varepsilon]$ by $M \in A\ [\Psi]$,
2. $N \in A\ [\Psi, y \mid s = \varepsilon, s' = \varepsilon]$ (by $N \in A\ [\Psi, y \mid s = s']$, because $s = s'$ whenever $s = \varepsilon, s' = \varepsilon$), $N \doteq N_i \in A\ [\Psi, y \mid s = \varepsilon, s' = \varepsilon, \xi_i]$ (by $N \doteq N_i \in A\ [\Psi, y \mid s = s', \xi_i]$), and $N\langle r/y\rangle \doteq M \in A\ [\Psi \mid s = \varepsilon, s' = \varepsilon]$ (by $N\langle r/y\rangle \doteq M \in A\ [\Psi \mid s = s']$), and
3. $\mathsf{ghcom}_A^{r\rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\ [\Psi, y \mid s = \varepsilon, s' = \overline{\varepsilon}]$ by inductive hypothesis (1), $\mathsf{ghcom}_A \doteq N_i \in A\ [\Psi, y \mid s = \varepsilon, s' = \overline{\varepsilon}, \xi_i]$ by inductive hypothesis (3), and $(\mathsf{ghcom}_A)\langle r/y\rangle \doteq M \in A\ [\Psi, y \mid s = \varepsilon, s' = \overline{\varepsilon}]$ by inductive hypothesis (2).

The remaining equations are immediate. To check $\mathsf{hcom}_A \in A\ [\Psi]$ it suffices to observe that $T_\varepsilon \in A\ [\Psi, z \mid s = \varepsilon]$ (by the above); $T_\varepsilon \doteq N\langle z/y\rangle \in A\ [\Psi, z \mid s = \varepsilon, s = s']$ (by the $s' = \varepsilon$ face of T_ε); $T_\varepsilon \doteq N_i\langle z/y\rangle \in A\ [\Psi, z \mid s = \varepsilon, \xi_i]$ (by the ξ_i face of T_ε); $T_\varepsilon\langle r/z\rangle \doteq M \in A\ [\Psi \mid s = \varepsilon]$ (by $r = z\langle r/z\rangle$ in T_ε); and the $\overrightarrow{s = \varepsilon}$ faces ensure the composition is valid. Part (1) follows by repeating this argument on the right side, and parts (2–3) follow by Definition 4.29. □

Theorem 4.35 (Generalized heterogeneous composition). *If* $A \doteq B\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, y]$,

1. $M \doteq M' \in A\langle r/y\rangle\ [\Psi]$,
2. $N_i \doteq N'_j \in A\ [\Psi, y \mid \xi_i, \xi_j]$ *for any* i, j, *and*
3. $N_i\langle r/y\rangle \doteq M \in A\langle r/y\rangle\ [\Psi \mid \xi_i]$ *for any* i,

then

1. $\mathsf{gcom}_{y.A}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{gcom}_{y.B}^{r\rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in A\langle r'/y\rangle\ [\Psi]$;
2. *if* $r = r'$ *then* $\mathsf{gcom}_{y.A}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in A\langle r/y\rangle\ [\Psi]$; *and*

3. *if* ξ_i *then* $\mathsf{gcom}^{r \rightsquigarrow r'}_{y.A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in A\langle r'/y\rangle\ [\Psi]$.

Proof. Our implementation of gcom by ghcom and coe mirrors exactly our implementation of com by hcom and coe. The proof is therefore identical to that of Theorem 4.33, appealing to Theorem 4.34 instead of Definition 4.29. □

4.4 Semantics of types

We now establish the basic properties of each type former in Cartesian cubical type theory. Many of these properties are the standard rules of type theory (as in Figure 2.4) generalized to arbitrary Ψ. Others govern novel type formers—path types, $\mathbb{S}^1$, V-types, and formal composites of types—or specify when types are Kan.

Unlike in Section 2.5, formation and introduction rules do not follow directly from the definitions of Figure 4.3, but require value-coherence of each type's underlying Ψ-PER. The *Kan formation* rules require us to verify that the operational semantics of hcom and coe at each type satisfy the conditions of Definition 4.29. We prove elimination and computation rules by coherent expansion, checking that the interval substitution instances of each elimination form compute coherently on introduction forms. Finally, uniqueness rules follow from induction on the value elements of each type, as in Idealized Nuprl. We prove only the closed, binary form of each rule, as one can mechanically extend these to open terms by commuting term substitutions past constructors, using the fact that judgments are closed under interval substitutions.

We present the rules of Cartesian cubical type theory in the appendices of this dissertation, both as a computational type theory (in the style of Nuprl or **RedPRL**) in Appendix A, and as an intensional type theory (in the style of Agda, Coq, or **redtt**) in Appendix B. (See Section 2.4 for a discussion of computational and intensional type theories.) The former exposes particularities of our computational semantics, whereas we expect the latter to admit denotational semantics in Cartesian cubical sets similar to those of Angiuli et al. [Ang+19]. The results of this section establish consistency (Theorem 4.58) of both type theories, as well as canonicity for the computational calculus (Theorems 4.63 and 4.77).

4.4.1 Dependent functions

Recall from Figure 4.3 that for $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$,

$$\tau_i^\kappa(\Psi, (a{:}A) \to B, (a{:}A') \to B', \{(\lambda a.N, \lambda a.N') \mid a : A \gg N \doteq N' \in B\ [\Psi]\})$$

if and only if $A \sim A' \downarrow \alpha \in \tau_i^\kappa\ [\Psi]$, $a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau_i^\kappa\ [\Psi]$, $\mathsf{Coh}(\alpha)$, and $\mathsf{CohFam}(\beta)$; expanding definitions, these in turn hold if and only if $\tau_i^\kappa \models (A \doteq A'\ \mathsf{type_{pre}}\ [\Psi])$ and $\tau_i^\kappa \models$

$(a : A \gg B \doteq B'\ \mathsf{type_{pre}}\ [\Psi])$. Moreover, $(a{:}A) \to B \sim (a{:}A') \to B' \downarrow [\![(a{:}A) \to B]\!] \in \tau_i^\kappa\ [\Psi]$ because interval substitutions preserve judgments and $(a{:}A) \to B\ \mathsf{val}_{\Box}$.

In the remainder of this section, all judgments are relative to τ_i^κ. Relative to $\tau_\omega^{\mathsf{pre}}$, Rule 4.36 is the ordinary formation rule for dependent functions. In our proof of Rule 4.103, however, we will invoke Rule 4.36 at arbitrary τ_i^κ in order to observe that every universe $\mathcal{U}_i^\kappa$ is closed under dependent functions.

Rule 4.36 (Pretype formation)**.** *If* $A \doteq A'\ \mathsf{type_{pre}}\ [\Psi]$ *and* $a : A \gg B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$ *then* $(a{:}A) \to B \doteq (a{:}A') \to B'\ \mathsf{type_{pre}}\ [\Psi]$.

Proof. We have already observed that $(a{:}A) \to B \sim (a{:}A') \to B' \downarrow [\![(a{:}A) \to B]\!] \in \tau_i^\kappa\ [\Psi]$; it remains to show $\mathsf{Coh}([\![(a{:}A) \to B]\!])$. Suppose $\psi : \Psi' \to \Psi$ and $[\![(a{:}A) \to B]\!]_\psi(\lambda a.N, \lambda a.N')$, in which case $a : A\psi \gg N \doteq N' \in B\psi\ [\Psi']$. We must show $\lambda a.N \sim \lambda a.N' \in [\![(a{:}A) \to B]\!]\psi\ [\Psi']$. By $\lambda a.N\ \mathsf{val}_{\Box}$, it suffices to see that for all $\psi' : \Psi'' \to \Psi'$, $[\![(a{:}A) \to B]\!]_{\psi\psi'}(\lambda a.N\psi', \lambda a.N'\psi')$, which holds by $a{:}A\psi\psi' \gg N\psi' {\doteq} N'\psi' \in B\psi\psi'\ [\Psi'']$. □

Because we have defined the value elements of $(a{:}A) \to B$ according to its introduction principle, value-coherence amounts to the introduction rule itself.

Rule 4.37 (Introduction)**.** *If* $a : A \gg M \doteq M' \in B\ [\Psi]$ *then* $\lambda a.M \doteq \lambda a.M' \in (a{:}A) \to B\ [\Psi]$.

Proof. The presuppositions of our hypothesis and Rule 4.36 imply $(a{:}A) \to B\ \mathsf{type_{pre}}\ [\Psi]$. The result follows by $\mathsf{Coh}([\![(a{:}A) \to B]\!])$ at id_Ψ. □

Lemma 4.38. *If* $M \in (a{:}A) \to B\ [\Psi]$ *and* $N \in A\ [\Psi]$ *then* $M \Downarrow \lambda a.O$ *and* $M\ N \doteq O[N/a] \in B[N/a]\ [\Psi]$.

Proof. The presupposition of our first hypothesis, $(a{:}A) \to B\ \mathsf{type_{pre}}\ [\Psi]$, implies $a : A \gg B\ \mathsf{type_{pre}}\ [\Psi]$; the presupposition $B[N/a]\ \mathsf{type_{pre}}\ [\Psi]$ follows immediately.

For all $\psi : \Psi' \to \Psi$, we know $M\psi \Downarrow \lambda a.O_\psi$ and $[\![(a{:}A) \to B]\!]_\psi(\lambda a.O_{\mathsf{id}_\Psi}\psi, \lambda a.O_\psi)$, and therefore $a : A\psi \gg O_{\mathsf{id}_\Psi}\psi \doteq O_\psi \in B\psi\ [\Psi']$. We apply coherent expansion (Lemma 4.18) to $(M\ N)$, $B[N/a]\ \mathsf{type_{pre}}\ [\Psi]$, and $\{O_\psi[N\psi/a]\}_\psi^{\Psi'}$, using $M\psi\ N\psi \longmapsto^* (\lambda a.O_\psi)\ N\psi \longmapsto O_\psi[N\psi/a]$ and $O_\psi[N\psi/a] \doteq (O_{\mathsf{id}_\Psi}[N/a])\psi \in B\psi[N\psi/a]\ [\Psi']$. We conclude that $M\ N \doteq O_{\mathsf{id}_\Psi}[N/a] \in B[N/a]\ [\Psi]$, as desired. □

Rule 4.39 (Elimination)**.** *If* $M {\doteq} M' \in (a{:}A) \to B\ [\Psi]$ *and* $N {\doteq} N' \in A\ [\Psi]$ *then* $M\ N {\doteq} M'\ N' \in B[N/a]\ [\Psi]$.

Proof. By Lemma 4.38, $M \Downarrow \lambda a.O$, $M' \Downarrow \lambda a.O'$, $M\ N \doteq O[N/a] \in B[N/a]\ [\Psi]$, and $M'\ N' \doteq O'[N'/a] \in B[N'/a]\ [\Psi]$. By Lemma 4.16, $M \doteq \lambda a.O \in (a{:}A) \to B\ [\Psi]$ and $M' \doteq \lambda a.O' \in (a{:}A) \to B\ [\Psi]$, and so by $[\![(a{:}A) \to B]\!](\lambda a.O, \lambda a.O')$, $a : A \gg O \doteq O' \in B\ [\Psi]$. We conclude $O[N/a] \doteq O'[N'/a] \in B[N/a]\ [\Psi]$ and $B[N/a] \doteq B[N'/a]\ \mathsf{type_{pre}}\ [\Psi]$, and the result follows by symmetry, transitivity, and Lemma 4.12. □

Rule 4.40 (Uniqueness). *If* $M \in (a{:}A) \to B\ [\Psi]$ *then* $M \doteq \lambda a.M\ a \in (a{:}A) \to B\ [\Psi]$.

Proof. By Lemma 4.16, $M \Downarrow \lambda a.O$ and $M \doteq \lambda a.O \in (a{:}A) \to B\ [\Psi]$; by transitivity and Rule 4.37 it suffices to show $a : A \gg O \doteq M\ a \in B\ [\Psi]$, that is, if $\psi : \Psi' \to \Psi$ and $N \doteq N' \in A\psi\ [\Psi']$ then $O\psi[N/a] \doteq M\psi\ N' \in B\psi[N/a]\ [\Psi']$. By Lemma 4.38, $O_\psi[N'/a] \doteq M\psi\ N' \in B\psi[N'/a]\ [\Psi']$, where $M\psi \Downarrow \lambda a.O_\psi$. The result follows by $B\psi[N/a] \doteq B\psi[N'/a]\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$ and $a : A\psi \gg O_{\mathsf{id}_\Psi}\psi \doteq O_\psi \in B\psi\ [\Psi']$, the latter by $[\![(a{:}A) \to B]\!]_\psi(\lambda a.O\psi, \lambda a.O_\psi)$. □

Computation rules that correspond to cubically-stable evaluation steps follow directly from head expansion; as in Idealized Nuprl, we omit such proofs, and omit such rules from Appendix A. We include the following proof for illustrative purposes:

Rule 4.41 (Computation). *If* $a : A \gg M \in B\ [\Psi]$ *and* $N \in A\ [\Psi]$ *then* $(\lambda a.M)\ N \doteq M[N/a] \in B[N/a]\ [\Psi]$.

Proof. By Lemma 4.32, $M[N/a] \in B[N/a]\ [\Psi]$, and $(\lambda a.M)\ N \longmapsto_{\Box} M[N/a]$. □

Finally, we must show that the dependent function type of two Kan types is again Kan. In our proof of Rule 4.102, we will invoke Rule 4.42 in order to observe that the (inductively defined collection of) elements of $\mathcal{U}_i^{\mathsf{Kan}}$ are Kan types.

Rule 4.42 (Kan type formation). *If* $A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ *and* $a : A \gg B \doteq B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ *then* $(a{:}A) \to B \doteq (a{:}A') \to B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. Rule 4.36 establishes the presupposition $(a{:}A) \to B \doteq (a{:}A') \to B'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$. It remains to check the Kan conditions of Definition 4.29; we begin with homogeneous composition. Suppose $\psi : \Psi' \to \Psi$, $\overrightarrow{\xi_i}$ is valid,

1. $M \doteq M' \in (a{:}A\psi) \to B\psi\ [\Psi']$,
2. $N_i \doteq N_j' \in (a{:}A\psi) \to B\psi\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and
3. $N_i\langle r/y\rangle \doteq M \in (a{:}A\psi) \to B\psi\ [\Psi' \mid \xi_i]$ for any i; show

$\mathsf{hcom}^{r \rightsquigarrow r'}_{(a:A\psi)\to B\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r \rightsquigarrow r'}_{(a:A'\psi)\to B'\psi}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}) \in (a{:}A\psi) \to B\psi\ [\Psi']$. By Lemma 4.32 on both sides and Rule 4.37, it suffices to show

$$a : A\psi \gg \mathsf{hcom}^{r \rightsquigarrow r'}_{B\psi}(M\ a; \overrightarrow{\xi_i \hookrightarrow y.N_i\ a}) \doteq \mathsf{hcom}^{r \rightsquigarrow r'}_{B'\psi}(M'\ a; \overrightarrow{\xi_i \hookrightarrow y.N_i'\ a}) \in B\psi\ [\Psi']$$

which is to say that for any $\psi' : \Psi'' \to \Psi'$ and $N \doteq N' \in A\psi\psi'\ [\Psi'']$,

$$\begin{aligned} &\mathsf{hcom}^{r\psi \rightsquigarrow r'\psi}_{B\psi\psi'[N/a]}(M\psi'\ N; \overrightarrow{\xi_i\psi' \hookrightarrow y.N_i\psi'\ N}) \\ \doteq\ &\mathsf{hcom}^{r\psi \rightsquigarrow r'\psi}_{B'\psi\psi'[N'/a]}(M'\psi'\ N'; \overrightarrow{\xi_i\psi' \hookrightarrow y.N_i'\psi'\ N'}) \in B\psi\psi'[N/a]\ [\Psi'']. \end{aligned}$$

We have $B\psi\psi'[N/a] \doteq B'\psi\psi'[N'/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'']$ by $a : A \gg B \doteq B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, so the result follows from Definition 4.29 if we show that $\overrightarrow{\xi_i\psi'}$ is valid,

1. $M\psi'\ N \doteq M'\psi'\ N' \in B\psi\psi'[N/a]\ [\Psi'']$,
2. $N_i\psi'\ N \doteq N_j'\psi'\ N' \in B\psi\psi'[N/a]\ [\Psi'', y \mid \xi_i\psi', \xi_j\psi']$ for any i, j, and
3. $N_i\langle r/y\rangle\psi'\ N \doteq M\psi'\ N' \in B\psi\psi'[N/a]\ [\Psi'' \mid \xi_i\psi']$ for any i.

These follow from our hypotheses and a restricted elimination rule (obtained from Rule 4.39 and Definition 4.27)—if $M \doteq M' \in (a{:}A) \to B\ [\Psi \mid \Xi]$ and $N \doteq N' \in A\ [\Psi \mid \Xi]$ then $M\ N \doteq M'\ N' \in B[N/a]\ [\Psi \mid \Xi]$.

Next, show that if $r = r'$ then $\mathsf{hcom}^{r\rightsquigarrow r'}_{(a:A\psi)\to B\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in (a{:}A\psi) \to B\psi\ [\Psi']$. By Lemma 4.32 on the left and Rule 4.40 on the right, it suffices to show

$$\lambda a.\mathsf{hcom}^{r\rightsquigarrow r'}_{B\psi}(M\ a; \overrightarrow{\xi_i \hookrightarrow y.N_i\ a}) \doteq \lambda a.M\ a \in (a{:}A\psi) \to B\psi\ [\Psi'].$$

By Rule 4.37, it suffices to show that for any $\psi' : \Psi'' \to \Psi'$ and $N \doteq N' \in A\psi\psi'\ [\Psi'']$,

$$\mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{B\psi\psi'[N/a]}(M\psi'\ N; \overrightarrow{\xi_i\psi' \hookrightarrow y.N_i\psi'\ N}) \doteq M\psi'\ N' \in B\psi\psi'[N/a]\ [\Psi'].$$

By $B\psi\psi'[N/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'']$ and $r = r'$ on the left, it suffices to show $M\psi'\ N \doteq M\psi'\ N' \in B\psi\psi'[N/a]\ [\Psi'']$, which holds by Rule 4.39.

The final hcom property states that if ξ_i holds then $\mathsf{hcom}^{r\rightsquigarrow r'}_{(a:A\psi)\to B\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in (a{:}A\psi) \to B\psi\ [\Psi']$. As before, by Lemma 4.32 on the left, Rule 4.40 on the right, and Rule 4.37, show for any $\psi' : \Psi'' \to \Psi'$ and $N \doteq N' \in A\psi\psi'\ [\Psi'']$ that

$$\mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{B\psi\psi'[N/a]}(M\psi'\ N; \overrightarrow{\xi_i\psi' \hookrightarrow y.N_i\psi'\ N}) \doteq N_i\langle r'/y\rangle\psi'\ N' \in B\psi\psi'[N/a]\ [\Psi'].$$

This follows by $B\psi\psi'[N/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'']$ and $\xi_i\psi'$ on the left, and Rule 4.39.

Now we must show that dependent functions also support coercion. Suppose $\psi : (\Psi', x) \to \Psi$ and $M \doteq M' \in (a{:}A\psi\langle r/x\rangle) \to B\psi\langle r/x\rangle\ [\Psi']$, and show $\mathsf{coe}^{r\rightsquigarrow r'}_{x.(a:A\psi)\to B\psi}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{x.(a:A'\psi)\to B'\psi}(M') \in (a{:}A\psi\langle r'/x\rangle) \to B\psi\langle r'/x\rangle\ [\Psi']$. By Lemma 4.32 on both sides and Rule 4.37, we must show for any $\psi' : \Psi'' \to \Psi'$ and $N \doteq N' \in A\psi\psi'\langle r'\psi'/x\rangle\ [\Psi'']$ that

$$\begin{aligned}
&\mathsf{coe}^{r\psi'\rightsquigarrow r'\psi'}_{x.B\psi\psi'[\mathsf{coe}^{r'\psi'\rightsquigarrow x}_{x.A\psi\psi'}(N)/a]}(M\psi'\ \mathsf{coe}^{r'\psi'\rightsquigarrow r\psi'}_{x.A\psi\psi'}(N)) \\
&\doteq \mathsf{coe}^{r\psi'\rightsquigarrow r'\psi'}_{x.B'\psi\psi'[\mathsf{coe}^{r'\psi'\rightsquigarrow x}_{x.A'\psi\psi'}(N')/a]}(M'\psi'\ \mathsf{coe}^{r'\psi'\rightsquigarrow r\psi'}_{x.A'\psi\psi'}(N')) \in B\psi\psi'\langle r'\psi'/x\rangle[N/a]\ [\Psi''].
\end{aligned}$$

By $A\psi\psi' \doteq A'\psi\psi'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'', x]$, $\mathsf{coe}^{r'\psi'\rightsquigarrow x}_{x.A\psi\psi'}(N) \doteq \mathsf{coe}^{r'\psi'\rightsquigarrow x}_{x.A'\psi\psi'}(N') \in A\psi\psi'\langle r'\psi'/x\rangle\ [\Psi''];$ therefore, the above instances of $B\psi\psi'$ and $B'\psi\psi'$ are equal Kan types. By Rule 4.39,

$$M\psi'\ \mathsf{coe}^{r'\psi'\rightsquigarrow r\psi'}_{x.A\psi\psi'}(N) \doteq M'\psi'\ \mathsf{coe}^{r'\psi'\rightsquigarrow r\psi'}_{x.A'\psi\psi'}(N') \in B\psi\psi'\langle r\psi'/x\rangle[\mathsf{coe}^{r'\psi'\rightsquigarrow r\psi'}_{x.A\psi\psi'}(N)/a]\ [\Psi'']$$

so the above coe are equal in $B\psi\psi'\langle r'\psi'/x\rangle[\mathsf{coe}^{r'\psi'\rightsquigarrow r'\psi'}_{x.A\psi\psi'}(N)/a]$. The result follows by Lemma 4.12 and $\mathsf{coe}^{r'\psi'\rightsquigarrow r'\psi'}_{x.A\psi\psi'}(N) \doteq N \in A\psi\psi'\langle r'\psi'/x\rangle\ [\Psi'']$.

Finally, show that if $r = r'$ then $\mathsf{coe}^{r\rightsquigarrow r'}_{x.(a:A\psi)\to B\psi}(M) \doteq M \in (a{:}A\psi\langle r'/x\rangle) \to B\psi\langle r'/x\rangle\ [\Psi']$. By Lemma 4.32 on the left, Rule 4.40 on the right, and Rule 4.37, it suffices to show for any $\psi' : \Psi'' \to \Psi'$ and $N \doteq N' \in A\psi\psi'\langle r'\psi'/x\rangle\ [\Psi'']$ that

$$\mathsf{coe}^{r\psi'\rightsquigarrow r'\psi'}_{x.B\psi\psi'[\mathsf{coe}^{r'\psi'\rightsquigarrow x}_{x.A\psi\psi'}(N)/a]}(M\psi'\ \mathsf{coe}^{r'\psi'\rightsquigarrow r\psi'}_{x.A\psi\psi'}(N)) \doteq M\psi'\ N' \in B\psi\psi'\langle r'\psi'/x\rangle[N/a]\ [\Psi''].$$

By $r\psi' = r'\psi'$, $A\psi\psi'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'', x]$, Rule 4.39, and $B\psi\psi'[\mathsf{coe}^{r'\psi'\rightsquigarrow x}_{x.A\psi\psi'}(N)/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi'', x]$, it suffices to show $M\psi'\ N \doteq M\psi'\ N' \in B\psi\psi'\langle r'\psi'/x\rangle[N/a]\ [\Psi'']$. Rule 4.39 completes the proof. □

4.4.2 *Dependent pairs*

For $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$,

$\tau^\kappa_i(\Psi, (a{:}A)\times B, (a{:}A')\times B', \{(\langle M, N\rangle, \langle M', N'\rangle) \mid M \doteq M' \in A\ [\Psi] \wedge N \doteq N' \in B[M/a]\ [\Psi]\})$

if and only if $A \sim A' \downarrow \alpha \in \tau^\kappa_i\ [\Psi]$, $a : \alpha \triangleright B \sim B' \downarrow \beta \in \tau^\kappa_i\ [\Psi]$, $\mathsf{Coh}(\alpha)$, and $\mathsf{CohFam}(\beta)$; expanding definitions, these in turn hold if and only if $\tau^\kappa_i \models (A \doteq A'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi])$ and $\tau^\kappa_i \models (a : A \gg B \doteq B'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi])$. In the remainder of this section, all judgments are relative to τ^κ_i.

Rule 4.43 (Pretype formation). *If* $A \doteq A'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ *and* $a : A \gg B \doteq B'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ *then* $(a{:}A) \times B \doteq (a{:}A') \times B'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. First, $(a{:}A) \times B \sim (a{:}A') \times B' \downarrow \llbracket(a{:}A) \to B\rrbracket \in \tau^\kappa_i\ [\Psi]$ because $(a{:}A) \times B\ \mathsf{val}_{\square}$ and interval substitutions preserve judgments. It remains to show $\mathsf{Coh}(\llbracket(a{:}A) \times B\rrbracket)$. Suppose $\psi : \Psi' \to \Psi$ and $\llbracket(a{:}A) \times B\rrbracket_\psi(\langle M, N\rangle, \langle M', N'\rangle)$. Then $M \doteq M' \in A\psi\ [\Psi']$ and $N \doteq N' \in B\psi[M/a]\ [\Psi']$; because $\langle M, N\rangle\ \mathsf{val}_{\square}$, $\langle M, N\rangle \sim \langle M', N'\rangle \in \llbracket(a{:}A) \times B\rrbracket\psi\ [\Psi]$. □

Rule 4.44 (Introduction). *If* $a{:}A \gg B\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, $M \doteq M' \in A\ [\Psi]$, *and* $N \doteq N' \in B[M/a]\ [\Psi]$, *then* $\langle M, N\rangle \doteq \langle M', N'\rangle \in (a{:}A) \times B\ [\Psi]$.

Proof. Immediate by Rule 4.43. (Note that the hypothesis $a : A \gg B\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ does not follow from the presupposition $B[M/a]\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.) □

Rule 4.45 (Elimination). *If* $P \doteq P' \in (a{:}A) \times B\ [\Psi]$ *then* $\mathsf{fst}(P) \doteq \mathsf{fst}(P') \in A\ [\Psi]$ *and* $\mathsf{snd}(P) \doteq \mathsf{snd}(P') \in B[\mathsf{fst}(P)/a]\ [\Psi]$.

Proof. For any $\psi : \Psi' \to \Psi$, $P\psi \Downarrow \langle M_\psi, N_\psi\rangle$, $M_\psi \in A\psi\ [\Psi']$, and $N_\psi \in B\psi[M_\psi/a]\ [\Psi']$. For part (1), apply coherent expansion to $\mathsf{fst}(P)$ with family $\{M_\psi\}^{\Psi'}_{\psi}$; then $(M_{\mathsf{id}_\Psi})\psi \doteq M_\psi \in A\psi\ [\Psi']$ by $P \in (a{:}A) \times B\ [\Psi]$ at id_Ψ, ψ. By Lemma 4.18, $\mathsf{fst}(P) \doteq M_{\mathsf{id}_\Psi} \in A\ [\Psi]$, and part (1) follows by $M_{\mathsf{id}_\Psi} \doteq M'_{\mathsf{id}_\Psi} \in A\ [\Psi]$ and a symmetric argument on the right side.

For part (2), apply coherent expansion to $\mathsf{snd}(P)$ with family $\{N_\psi\}^{\Psi'}_{\psi}$. We have $(N_{\mathsf{id}_\Psi})\psi \doteq N_\psi \in B\psi[(M_{\mathsf{id}_\Psi})\psi/a]\ [\Psi']$ by $P \in (a{:}A)\times B\ [\Psi]$ at id_Ψ, ψ, so by Lemma 4.18, $\mathsf{snd}(P) \doteq N_{\mathsf{id}_\Psi} \in B[M_{\mathsf{id}_\Psi}/a]\ [\Psi]$. Part (2) follows by $B[M_{\mathsf{id}_\Psi}/a] \doteq B[\mathsf{fst}(P)/a]\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ (by $a : A \gg B \doteq B'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ and $M_{\mathsf{id}_\Psi} \doteq \mathsf{fst}(P) \in A\ [\Psi]$), $N_{\mathsf{id}_\Psi} \doteq N'_{\mathsf{id}_\Psi} \in B[M_{\mathsf{id}_\Psi}/a]\ [\Psi]$, and a symmetric argument on the right side. □

Rule 4.46 (Uniqueness). *If* $P \in (a{:}A) \times B\ [\Psi]$ *then* $P \doteq \langle \mathsf{fst}(P), \mathsf{snd}(P)\rangle \in (a{:}A) \times B\ [\Psi]$.

Proof. By Lemma 4.16, $P \Downarrow \langle M, N\rangle$, $P \doteq \langle M, N\rangle \in (a{:}A) \times B\ [\Psi]$, $M \in A\ [\Psi]$, and $N \in B[M/a]\ [\Psi]$. By Rule 4.44 and Lemma 4.15 and transitivity, we show $[\![A]\!]^{\Downarrow}(M, \mathsf{fst}(P))$ and $[\![B[M/a]]\!]^{\Downarrow}(N, \mathsf{snd}(P))$. This is immediate by $\mathsf{fst}(P) \longmapsto^* \mathsf{fst}(\langle M, N\rangle) \longmapsto M$ and $\mathsf{snd}(P) \longmapsto^* \mathsf{snd}(\langle M, N\rangle) \longmapsto N$. □

Rule 4.47 (Kan type formation). *If* $A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ *and* $a : A \gg B \doteq B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ *then* $(a{:}A) \times B \doteq (a{:}A') \times B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. We begin with homogeneous composition. Suppose $\psi : \Psi' \to \Psi$, $\overrightarrow{\xi_i}$ is valid,

1. $M \doteq M' \in (a{:}A\psi) \times B\psi\ [\Psi']$,
2. $N_i \doteq N'_j \in (a{:}A\psi) \times B\psi\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and
3. $N_i\langle r/y\rangle \doteq M \in (a{:}A\psi) \times B\psi\ [\Psi' \mid \xi_i]$ for any i; show

$\mathsf{hcom}^{r\rightsquigarrow r'}_{(a:A\psi)\times B\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{(a:A'\psi)\times B'\psi}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in (a{:}A\psi) \times B\psi\ [\Psi']$. By Lemma 4.32 on both sides and Rule 4.44, it suffices to show (the binary forms of)

$$\mathsf{hcom}^{r\rightsquigarrow r'}_{A\psi}(\mathsf{fst}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{fst}(N_i)}) \in A\psi\ [\Psi']$$
$$\mathsf{com}^{r\rightsquigarrow r'}_{z.B\psi[F/a]}(\mathsf{snd}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{snd}(N_i)}) \in B\psi[\mathsf{hcom}_{A\psi}/a]\ [\Psi']$$
$$\text{where } F := \mathsf{hcom}^{r\rightsquigarrow z}_{A\psi}(\mathsf{fst}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{fst}(N_i)}).$$

We have $\mathsf{hcom}_{A\psi} \in A\psi\ [\Psi']$ and $F \in A\psi\ [\Psi', z]$ by $A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ and Rule 4.45. We establish $\mathsf{com}_{z.B\psi[F/a]} \in B\psi[\mathsf{hcom}_{A\psi}/a]\ [\Psi']$ by Theorem 4.33, using $B\psi[F/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', z]$, $F\langle r'/z\rangle = \mathsf{hcom}_{A\psi}$,

1. $\mathsf{snd}(M) \in B\psi[F\langle r/z\rangle/a]\ [\Psi']$ by $F\langle r/z\rangle \doteq \mathsf{fst}(M) \in A\psi\ [\Psi']$ and Rule 4.45,
2. $N_i \doteq N_j \in B\psi[F\langle y/z\rangle/a]\ [\Psi', y \mid \xi_i, \xi_j]$ by $F\langle y/z\rangle \doteq \mathsf{fst}(N_i) \in A\psi\ [\Psi', y \mid \xi_i]$ and Rule 4.45, and

3. $\mathsf{snd}(N_i\langle r/y\rangle) \doteq \mathsf{snd}(M) \in B\psi[F\langle r/z\rangle/a]\ [\Psi' \mid \xi_i]$ by $F\langle r/z\rangle \doteq \mathsf{fst}(M) \in A\psi\ [\Psi']$ and Rule 4.45.

Next, we must show that if $r = r'$ then $\mathsf{hcom}_{(a:A\psi)\times B\psi} \doteq M \in (a{:}A\psi) \times B\psi\ [\Psi']$. By Lemma 4.32, $\mathsf{hcom}_{(a:A\psi)\times B\psi} \doteq \langle \mathsf{hcom}_{A\psi}, \mathsf{com}_{z.B\psi[F/a]}\rangle \in (a{:}A\psi)\times B\psi\ [\Psi']$. By Definition 4.29 and Theorem 4.33, $\mathsf{hcom}_{A\psi} \doteq \mathsf{fst}(M) \in A\psi\ [\Psi']$, $\mathsf{com}_{z.B\psi[F/a]} \doteq \mathsf{snd}(M) \in B\psi[F\langle r/z\rangle/a]\ [\Psi']$, and $B\psi[F\langle r/z\rangle/a] \doteq B\psi[\mathsf{fst}(M)/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$. The result follows by Rule 4.46.

Finally, show that if ξ_i, $\mathsf{hcom}_{(a:A\psi)\times B\psi} \doteq N_i\langle r'/y\rangle \in (a{:}A\psi) \times B\psi\ [\Psi']$; this follows from $\mathsf{hcom}_{A\psi} \doteq \mathsf{fst}(N_i\langle r'/y\rangle) \in A\psi\ [\Psi']$, $\mathsf{com}_{z.B\psi[F/a]} \doteq \mathsf{snd}(N_i\langle r'/y\rangle) \in B\psi[F\langle r'/z\rangle/a]\ [\Psi']$, and $B\psi[F\langle r'/z\rangle/a] \doteq B\psi[\mathsf{fst}(N_i\langle r'/y\rangle)/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$.

For coercion, suppose that $\psi : (\Psi', x) \to \Psi$ and $M \doteq M' \in ((a{:}A\psi)\times B\psi)\langle r/x\rangle\ [\Psi']$, and show $\mathsf{coe}^{r\rightsquigarrow r'}_{x.(a:A\psi)\times B\psi}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{x.(a:A'\psi)\times B'\psi}(M') \in ((a{:}A\psi) \times B\psi)\langle r'/x\rangle\ [\Psi']$. By Lemma 4.32 and Rule 4.44, it suffices to show (the binary form of)

$$\mathsf{coe}^{r\rightsquigarrow r'}_{x.A\psi}(\mathsf{fst}(M)) \in A\psi\langle r'/x\rangle\ [\Psi']$$
$$\mathsf{coe}^{r\rightsquigarrow r'}_{x.B\psi[\mathsf{coe}^{r\rightsquigarrow x}_{x.A\psi}(\mathsf{fst}(M))/a]}(\mathsf{snd}(M)) \in B\psi\langle r'/x\rangle[\mathsf{coe}^{r\rightsquigarrow r'}_{x.A\psi}(\mathsf{fst}(M))/a]\ [\Psi']$$

We know that $\mathsf{coe}^{r\rightsquigarrow r'}_{x.A\psi}(\mathsf{fst}(M)) \in A\psi\langle r'/x\rangle\ [\Psi']$ and $B\psi[\mathsf{coe}^{r\rightsquigarrow x}_{x.A\psi}(\mathsf{fst}(M))/a]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', x]$ by $A\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', x]$, $a : A\psi \gg B\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', x]$, and Rule 4.45. We also know that $\mathsf{snd}(M) \in B\psi\langle r/x\rangle[\mathsf{fst}(M)/a]\ [\Psi']$ and $(\mathsf{coe}^{r\rightsquigarrow x}_{x.A\psi}(\mathsf{fst}(M)))\langle r/x\rangle \doteq \mathsf{fst}(M) \in A\langle r/x\rangle\ [\Psi']$, so $\mathsf{coe}_{x.B\psi[\dots/a]} \in B\psi\langle r'/x\rangle[\mathsf{coe}^{r\rightsquigarrow r'}_{x.A\psi}(\mathsf{fst}(M))/a]\ [\Psi']$ and the result follows.

Finally, show that if $r = r'$ then $\mathsf{coe}^{r\rightsquigarrow r}_{x.(a:A\psi)\times B\psi}(M) \doteq M \in ((a{:}A\psi) \times B\psi)\langle r/x\rangle\ [\Psi']$. By Lemma 4.32 and Rules 4.44 and 4.46, this follows from $\mathsf{coe}_{x.A\psi} \doteq \mathsf{fst}(M) \in A\psi\langle r/x\rangle\ [\Psi']$ and $\mathsf{coe}_{x.B\psi[\dots/a]} \doteq \mathsf{snd}(M) \in B\psi\langle r/x\rangle[\mathsf{fst}(M)/a]\ [\Psi']$. □

4.4.3 *Paths*

For $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$, $\tau^\kappa_i(\Psi, \mathsf{Path}_{x.A}(P_0, P_1), \mathsf{Path}_{x.A'}(P'_0, P'_1), \varphi)$ where

$$\varphi = \{(\langle x\rangle M, \langle x\rangle M') \mid (M \sim M' \in \alpha\ [\Psi, x]) \land (\forall\varepsilon.M\langle\varepsilon/x\rangle \sim P_\varepsilon \in \alpha\langle\varepsilon/x\rangle\ [\Psi])\}$$

if and only if $A \sim A' \downarrow \alpha \in \tau^\kappa_i\ [\Psi, x]$, $\mathsf{Coh}(\alpha)$, and $P_\varepsilon \sim P'_\varepsilon \in \alpha\langle\varepsilon/x\rangle\ [\Psi]$ for $\varepsilon \in \{0, 1\}$, or equivalently, $\tau^\kappa_i \models (A \doteq A'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi, x])$ and $\tau^\kappa_i \models (P_\varepsilon \doteq P'_\varepsilon \in A\langle\varepsilon/x\rangle\ [\Psi])$ for $\varepsilon \in \{0, 1\}$. Relative to τ^κ_i:

Rule 4.48 (Pretype formation). *If* $A \doteq A'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi, x]$ *and* $P_\varepsilon \doteq P'_\varepsilon \in A\langle\varepsilon/x\rangle\ [\Psi]$ *for* $\varepsilon \in \{0, 1\}$*, then* $\mathsf{Path}_{x.A}(P_0, P_1) \doteq \mathsf{Path}_{x.A'}(P'_0, P'_1)\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. We have $\mathsf{Path}_{x.A}(P_0, P_1) \sim \mathsf{Path}_{x.A'}(P'_0, P'_1) \downarrow \llbracket\mathsf{Path}_{x.A}(P_0, P_1)\rrbracket \in \tau^\kappa_i\ [\Psi]$ because $\mathsf{Path}_{x.A}(P_0, P_1)\ \mathsf{val}_{\Box}$. It remains to show $\mathsf{Coh}(\llbracket\mathsf{Path}_{x.A}(P_0, P_1)\rrbracket)$. Suppose that $\psi : \Psi' \to \Psi$ and $\llbracket\mathsf{Path}_{x.A}(P_0, P_1)\rrbracket_\psi(\langle x\rangle M, \langle x\rangle M')$. Then $M \doteq M' \in A\psi\ [\Psi', x]$ and $M\langle\varepsilon/x\rangle \doteq P_\varepsilon\psi \in A\psi\langle\varepsilon/x\rangle\ [\Psi']$; by $\langle x\rangle M\ \mathsf{val}_{\Box}$, $\langle x\rangle M \sim \langle x\rangle M' \in \llbracket\mathsf{Path}_{x.A}(P_0, P_1)\rrbracket\psi\ [\Psi]$. □

Rule 4.49 (Introduction). *If $M \doteq M' \in A\ [\Psi, x]$ and $M\langle \varepsilon/x\rangle \doteq P_\varepsilon \in A\langle \varepsilon/x\rangle\ [\Psi]$ for $\varepsilon \in \{0, 1\}$, then $\langle x\rangle M \doteq \langle x\rangle M' \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$.*

Proof. Immediate by Rule 4.48. □

Rule 4.50 (Elimination).

1. *If $M \doteq M' \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$ then $M@r \doteq M'@r \in A\langle r/x\rangle\ [\Psi]$.*
2. *If $M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$ then $M@\varepsilon \doteq P_\varepsilon \in A\langle \varepsilon/x\rangle\ [\Psi]$.*

Proof. Apply coherent expansion to $M@r$ with family $\{M_\psi\langle r\psi/x\rangle \mid M\psi \Downarrow \langle x\rangle M_\psi\}_\psi^{\Psi'}$. By $M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$ at id_Ψ, ψ we know $(M_{\mathsf{id}_\Psi})\psi \doteq M_\psi \in A\psi\ [\Psi', x]$, so $(M_{\mathsf{id}_\Psi})\psi\langle r\psi/x\rangle \doteq M_\psi\langle r\psi/x\rangle \in A\langle r/x\rangle\psi\ [\Psi']$. Thus by Lemma 4.18, $M@r \doteq M_{\mathsf{id}_\Psi}\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi]$; part (1) follows by the same argument on the right side and $M_{\mathsf{id}_\Psi} \doteq M'_{\mathsf{id}_\Psi} \in A\ [\Psi, x]$. Part (2) follows from $M@\varepsilon \doteq M_{\mathsf{id}_\Psi}\langle \varepsilon/x\rangle \in A\langle \varepsilon/x\rangle\ [\Psi]$ and $M_{\mathsf{id}_\Psi}\langle \varepsilon/x\rangle \doteq P_\varepsilon \in A\langle \varepsilon/x\rangle\ [\Psi]$. □

Rule 4.51 (Uniqueness). *If $M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$, $M \doteq \langle x\rangle(M@x) \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$.*

Proof. By Lemma 4.16, $M \Downarrow \langle x\rangle N$ and $M \doteq \langle x\rangle N \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$. By Rule 4.50, $M@x \doteq (\langle x\rangle N)@x \in A\ [\Psi, x]$, so by Lemma 4.32 on the right, $M@x \doteq N \in A\ [\Psi, x]$. By Rule 4.49, $\langle x\rangle(M@x) \doteq \langle x\rangle N \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]$, and the result follows by transitivity. □

Rule 4.52 (Kan type formation). *If $A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x]$ and $P_\varepsilon \doteq P'_\varepsilon \in A\langle \varepsilon/x\rangle\ [\Psi]$ for $\varepsilon \in \{0, 1\}$, then* $\mathsf{Path}_{x.A}(P_0, P_1) \doteq \mathsf{Path}_{x.A'}(P'_0, P'_1)\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. We begin with homogeneous composition. Suppose $\psi : \Psi' \to \Psi$, $\overrightarrow{\xi_i}$ is valid,

1. $M \doteq M' \in \mathsf{Path}_{x.A\psi}(P_0\psi, P_1\psi)\ [\Psi']$,
2. $N_i \doteq N'_j \in \mathsf{Path}_{x.A\psi}(P_0\psi, P_1\psi)\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and
3. $N_i\langle r/y\rangle \doteq M \in \mathsf{Path}_{x.A\psi}(P_0\psi, P_1\psi)\ [\Psi' \mid \xi_i]$ for any i; show

$$\begin{aligned}
&\mathsf{hcom}^{r\rightsquigarrow r'}_{(\mathsf{Path}_{x.A}(P_0,P_1))\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \\
\doteq\ &\mathsf{hcom}^{r\rightsquigarrow r'}_{(\mathsf{Path}_{x.A'}(P'_0,P'_1))\psi}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in (\mathsf{Path}_{x.A}(P_0, P_1))\psi\ [\Psi'].
\end{aligned}$$

By Lemma 4.32 and Rule 4.49 on both sides it suffices to show

$$\begin{aligned}
&\mathsf{hcom}^{r\rightsquigarrow r'}_{A\psi}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow \_.P_\varepsilon\psi}, \overrightarrow{\xi_i \hookrightarrow y.N_i@x}) \\
\doteq\ &\mathsf{hcom}^{r\rightsquigarrow r'}_{A'\psi}(M'@x; \overrightarrow{x = \varepsilon \hookrightarrow \_.P'_\varepsilon\psi}, \overrightarrow{\xi_i \hookrightarrow y.N'_i@x}) \in A\psi\ [\Psi', x]
\end{aligned}$$

and $(\mathsf{hcom}_{A\psi})\langle \varepsilon/x\rangle \doteq P_\varepsilon\psi \in A\psi\langle \varepsilon/x\rangle\ [\Psi']$. By our hypotheses and Rule 4.50,

1. $M@x \doteq M'@x \in A\psi\ [\Psi', x]$,
2. $P_\varepsilon\psi \doteq P'_\varepsilon\psi \in A\psi\ [\Psi', x \mid x = \varepsilon]$ and $P_\varepsilon\psi \doteq M@x \in A\psi\ [\Psi', x \mid x = \varepsilon]$,
3. $N_i@x \doteq N'_j@x \in A\psi\ [\Psi', x, y \mid \xi_i, \xi_j]$, $N_i@x \doteq P'_\varepsilon\psi \in A\psi\ [\Psi', x, y \mid \xi_i, x = \varepsilon]$, and $N_i\langle r/y\rangle@x \doteq M@x \in A\psi\ [\Psi', x \mid \xi_i]$,

so $\mathsf{hcom}_{A\psi} \doteq \mathsf{hcom}_{A'\psi} \in A\psi\ [\Psi', x]$ and $(\mathsf{hcom}_{A\psi})\langle\varepsilon/x\rangle \doteq P_\varepsilon\psi \in A\psi\ [\Psi]$.

Next, show if $r = r'$, $\mathsf{hcom}^{r\rightsquigarrow r'}_{(\mathsf{Path}_{x.A}(P_0,P_1))\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in (\mathsf{Path}_{x.A}(P_0, P_1))\psi\ [\Psi']$. By Rule 4.49 and Definition 4.29 the left side equals $\langle x\rangle(M@x)$; the result follows by Rule 4.51.

Finally, if ξ_i, $\mathsf{hcom}^{r\rightsquigarrow r'}_{(\mathsf{Path}_{x.A}(P_0,P_1))\psi}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in (\mathsf{Path}_{x.A}(P_0, P_1))\psi\ [\Psi']$. The left side equals $\langle x\rangle(N_i\langle r'/y\rangle@x)$; the result follows by Rule 4.51.

For coercion, let $\psi : (\Psi', y) \to \Psi$ and $M \doteq M' \in (\mathsf{Path}_{x.A}(P_0, P_1))\psi\langle r/y\rangle\ [\Psi']$; show $\mathsf{coe}^{r\rightsquigarrow r'}_{y.(\mathsf{Path}_{x.A}(P_0,P_1))\psi}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{y.(\mathsf{Path}_{x.A'}(P'_0,P'_1))\psi}(M') \in (\mathsf{Path}_{x.A}(P_0, P_1))\psi\langle r'/y\rangle\ [\Psi']$. Applying Lemma 4.32 on both sides and then Rule 4.49, it suffices to show

$$\mathsf{com}^{r\rightsquigarrow r'}_{y.A\psi}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow y.P_\varepsilon\psi}) \doteq \mathsf{com}^{r\rightsquigarrow r'}_{y.A'\psi}(M'@x; \overrightarrow{x = \varepsilon \hookrightarrow y.P'_\varepsilon\psi}) \in A\psi\langle r'/y\rangle\ [\Psi', x]$$

and $(\mathsf{com}_{y.A\psi})\langle\varepsilon/x\rangle \doteq P_\varepsilon\psi\langle r'/y\rangle \in A\psi\langle r'/y\rangle\langle\varepsilon/x\rangle\ [\Psi']$. By our hypotheses and Rule 4.50, $M@x \doteq M'@x \in A\psi\langle r/y\rangle\ [\Psi', x]$, $P_\varepsilon\psi \doteq P'_\varepsilon\psi \in A\psi\ [\Psi', x, y \mid x = \varepsilon]$, and $P_\varepsilon\psi\langle r/y\rangle \doteq M@x \in A\psi\langle r/y\rangle\ [\Psi', x \mid x = \varepsilon]$, so by Theorem 4.33, $\mathsf{com}_{y.A\psi} \doteq \mathsf{com}_{y.A'\psi} \in A\psi\langle r'/y\rangle\ [\Psi', x]$ and $(\mathsf{com}_{y.A\psi})\langle\varepsilon/x\rangle \doteq P_\varepsilon\psi\langle r'/y\rangle \in A\psi\langle r'/y\rangle\langle\varepsilon/x\rangle\ [\Psi']$.

Finally, show that if $r = r'$, $\mathsf{coe}^{r\rightsquigarrow r'}_{y.(\mathsf{Path}_{x.A}(P_0,P_1))\psi}(M) \doteq M \in (\mathsf{Path}_{x.A}(P_0, P_1))\psi\langle r'/y\rangle\ [\Psi']$. Again the left side equals $\langle x\rangle(M@x)$, and Rule 4.51 completes the proof. □

The above proofs generalize straightforwardly to extension types (Section 3.5).

4.4.4 *Equalities*

For $i \in \{0, 1, \dots, \omega\}$, $\tau^{\mathsf{pre}}_i(\Psi, \mathsf{Eq}_A(M, N), \mathsf{Eq}_{A'}(M', N'), \{(\star, \star) \mid M \sim N \in \alpha\ [\Psi]\})$ if and only if $A \sim A' \downarrow \alpha \in \tau^{\mathsf{pre}}_i\ [\Psi]$, $\mathsf{Coh}(\alpha)$, $M \sim M' \in \alpha\ [\Psi]$, and $N \sim N' \in \alpha\ [\Psi]$; or equivalently, $\tau^{\mathsf{pre}}_i \models (A \doteq A'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi])$, $\tau^{\mathsf{pre}}_i \models (M \doteq M' \in A\ [\Psi])$, and $\tau^{\mathsf{pre}}_i \models (N \doteq N' \in A\ [\Psi])$. Relative to τ^{pre}_i:

Rule 4.53 (Pretype formation)**.** *If* $A \doteq A'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, $M \doteq M' \in A\ [\Psi]$, *and* $N \doteq N' \in A\ [\Psi]$, *then* $\mathsf{Eq}_A(M, N) \doteq \mathsf{Eq}_{A'}(M', N')\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. Again, $\mathsf{Eq}_A(M, N) \sim \mathsf{Eq}_{A'}(M', N') \downarrow [\![\mathsf{Eq}_A(M, N)]\!] \in \tau^{\mathsf{pre}}_i\ [\Psi]$ by $\mathsf{Eq}_A(M, N)\ \mathsf{val}_{\square}$. To see $\mathsf{Coh}([\![\mathsf{Eq}_A(M, N)]\!])$, suppose $\psi : \Psi' \to \Psi$ and $[\![\mathsf{Eq}_A(M, N)]\!]_\psi(\star, \star)$. Then $M\psi \doteq N\psi \in A\psi\ [\Psi']$, and $\star \sim \star \in [\![\mathsf{Eq}_A(M, N)]\!]\psi\ [\Psi]$ holds by $\star\ \mathsf{val}_{\square}$. □

Rule 4.54 (Introduction). *If* $M \doteq N \in A\ [\Psi]$ *then* $\star \in \mathsf{Eq}_A(M, N)\ [\Psi]$.

Proof. Immediate by Rule 4.53. □

Rule 4.55 (Elimination). *If* $E \in \mathsf{Eq}_A(M, N)\ [\Psi]$ *then* $M \doteq N \in A\ [\Psi]$.

Proof. Then $[\![\mathsf{Eq}_A(M, N)]\!]^{\Downarrow}(E, E)$ so $E \Downarrow \star$ and $M \doteq N \in A\ [\Psi]$. □

Rule 4.56 (Uniqueness). *If* $E \in \mathsf{Eq}_A(M, N)\ [\Psi]$ *then* $E \doteq \star \in \mathsf{Eq}_A(M, N)\ [\Psi]$.

Proof. Immediate by Lemma 4.16. □

As we observed at the beginning of this chapter, equality types are not generally Kan, because coercion in $x.\mathsf{Eq}_A(M, N)$ requires all paths in A to be trivial. (In Chapter 5 we will discuss the converse—if all paths in A are trivial in an appropriate sense, then $x.\mathsf{Eq}_A(M, N)$ supports coercion.) On the other hand, one can always homogeneously compose:

$$\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{Eq}_A(M,N)}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}) := \star$$

because M', N_i' equal $\star$ by Rule 4.56.

4.4.5 Void

For $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$, $\tau_i^{\kappa}(\Psi, \mathsf{void}, \mathsf{void}, \{\})$. Relative to τ_i^{κ}:

Rule 4.57 (Pretype formation). $\mathsf{void}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. By $\mathsf{void}\ \mathsf{val}_{\square}$, $\mathsf{void} \sim \mathsf{void} \downarrow [\![\mathsf{void}]\!] \in \tau_i^{\kappa}\ [\Psi]$ where every $[\![\mathsf{void}]\!]_{\Psi}$ is empty; $\mathsf{Coh}([\![\mathsf{void}]\!])$ holds vacuously. □

Theorem 4.58 (Consistency). *There is no* M *such that* $M \in \mathsf{void}\ [\Psi]$.

Proof. Suppose $M \in \mathsf{void}\ [\Psi]$; then $[\![\mathsf{void}]\!]^{\Downarrow}_{\Psi}(M, M)$, which is impossible. □

Theorem 4.58 implies the absurdity rule in Appendix A—if $\Gamma \gg M \in \mathsf{void}\ [\Psi]$ then $\Gamma \gg N \in A\ [\Psi]$—because the premise only holds if no $\gamma \sim \gamma' \in \Gamma\ [\Psi]$ exist.

Rule 4.59 (Kan type formation). $\mathsf{void}\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. Each Kan condition supposes that $M \doteq M' \in \mathsf{void}\ [\Psi']$, and therefore holds vacuously by Theorem 4.58. □

4.4.6 *Booleans*

For $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$, $\tau_i^\kappa(\Psi, \mathsf{bool}, \mathsf{bool}, \{(\mathsf{true}, \mathsf{true}), (\mathsf{false}, \mathsf{false})\})$. Relative to τ_i^κ:

Rule 4.60 (Pretype formation)**.** $\mathsf{bool}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. Once again, $\mathsf{bool} \sim \mathsf{bool} \downarrow [\![\mathsf{bool}]\!] \in \tau_i^\kappa\ [\Psi]$ by $\mathsf{bool}\ \mathsf{val}_{\Box}$. For $\mathrm{Coh}([\![\mathsf{bool}]\!])$, we must show $\mathsf{true} \sim \mathsf{true} \in [\![\mathsf{bool}]\!]\ [\Psi]$ and $\mathsf{false} \sim \mathsf{false} \in [\![\mathsf{bool}]\!]\ [\Psi]$, which hold because $\mathsf{true}\ \mathsf{val}_{\Box}$, $\mathsf{false}\ \mathsf{val}_{\Box}$, and for all Ψ', $[\![\mathsf{bool}]\!]_{\Psi'}(\mathsf{true}, \mathsf{true})$ and $[\![\mathsf{bool}]\!]_{\Psi'}(\mathsf{false}, \mathsf{false})$. □

Rule 4.61 (Introduction)**.** $\mathsf{true} \in \mathsf{bool}\ [\Psi]$ *and* $\mathsf{false} \in \mathsf{bool}\ [\Psi]$.

Proof. Immediate by Rule 4.60. □

Rule 4.62 (Elimination)**.** *If* $b : \mathsf{bool} \gg A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$*,* $M \doteq M' \in \mathsf{bool}\ [\Psi]$*,* $T \doteq T' \in A[\mathsf{true}/b]\ [\Psi]$*, and* $F \doteq F' \in A[\mathsf{false}/b]\ [\Psi]$*, then* $\mathsf{if}(M; T, F) \doteq \mathsf{if}(M'; T', F') \in A[M/b]\ [\Psi]$*.*

Proof. Apply coherent expansion on the left with family $\{\mathsf{if}(M_\psi; T\psi, F\psi) \mid M\psi \Downarrow M_\psi\}_\psi^{\Psi'}$. We must show $\mathsf{if}(M_\psi; T\psi, F\psi) \doteq \mathsf{if}((M_{\mathsf{id}_\Psi})\psi; T\psi, F\psi) \in A\psi[M\psi/b]\ [\Psi']$. By the definition of $[\![\mathsf{bool}]\!]_{\Psi'}$, either $M_\psi = \mathsf{true}$ or $M_\psi = \mathsf{false}$; in either case $M_{\mathsf{id}_\Psi} = M_\psi$ because $[\![\mathsf{bool}]\!]^{\Downarrow}_{\Psi'}((M_{\mathsf{id}_\Psi})\psi, M_\psi)$. Suppose both equal true; show $\mathsf{if}(\mathsf{true}; T\psi, F\psi) \in A\psi[M\psi/b]\ [\Psi']$. By Lemma 4.32 and $T\psi \in A\psi[\mathsf{true}/b]\ [\Psi']$ (by hypothesis) it suffices to show $A\psi[M\psi/b] \doteq A\psi[\mathsf{true}/b]\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$, which follows from $M\psi \doteq \mathsf{true} \in \mathsf{bool}\ [\Psi']$ (using Lemma 4.16). The $M_\psi = \mathsf{false}$ case is symmetric.

We conclude by Lemma 4.18 that $\mathsf{if}(M; T, F) \doteq \mathsf{if}(M_{\mathsf{id}_\Psi}; T, F) \in A[M/b]\ [\Psi]$. By transitivity, Lemma 4.16, and the same argument on the right, it suffices to show $\mathsf{if}(M_{\mathsf{id}_\Psi}; T, F) \doteq \mathsf{if}(M'_{\mathsf{id}_\Psi}; T', F') \in A[M_{\mathsf{id}_\Psi}/b]\ [\Psi]$. By $M \doteq M' \in \mathsf{bool}\ [\Psi]$, either $M_{\mathsf{id}_\Psi} = M'_{\mathsf{id}_\Psi} = \mathsf{true}$ or $M_{\mathsf{id}_\Psi} = M'_{\mathsf{id}_\Psi} = \mathsf{false}$, and in either case the result follows by Lemma 4.32. □

Our computational type theory (Appendix A) contains an untyped evaluation rule (Lemma 4.16) and therefore satisfies canonicity. Unlike in Chapter 2, that rule applies only to *closed* terms, because unstable evaluation does not commute with substitution.

Theorem 4.63 (Canonicity I)**.** *If* $M \in \mathsf{bool}\ [\Psi]$ *then either* $M \Downarrow \mathsf{true}$ *and* $M \doteq \mathsf{true} \in \mathsf{bool}\ [\Psi]$ *or* $M \Downarrow \mathsf{false}$ *and* $M \doteq \mathsf{false} \in \mathsf{bool}\ [\Psi]$*. Furthermore, the rules of* Appendix A *suffice to derive these equations.*

Proof. By $M \sim M \in \mathsf{bool}\ [\Psi]$ at $\mathsf{id}_\Psi, \mathsf{id}_\Psi$, either $M \Downarrow \mathsf{true}$ or $M \Downarrow \mathsf{false}$; the desired equation follows by Lemma 4.16 (a rule of Appendix A) and determinacy of evaluation. □

Rule 4.64 (Kan type formation)**.** $\mathsf{bool}\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. For homogeneous composition, suppose that $\overrightarrow{\xi_i}$ is valid,

1. $M \doteq M' \in \mathsf{bool}\ [\Psi']$,

2. $N_i \doteq N'_j \in \mathsf{bool}\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and

3. $N_i\langle r/y\rangle \doteq M \in \mathsf{bool}\ [\Psi' \mid \xi_i]$ for any i; show

$\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in \mathsf{bool}\ [\Psi']$. This is immediate by Lemma 4.32 on both sides, because $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \longmapsto_{\Box} M$ and $M \doteq M' \in \mathsf{bool}\ [\Psi']$. By the same argument, when $r = r'$, $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in \mathsf{bool}\ [\Psi']$. Next, suppose ξ_i holds, and show $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in \mathsf{bool}\ [\Psi']$. The left equals M by Lemma 4.32, which by hypothesis equals $N_i\langle r/y\rangle$; therefore, it suffices to show $N_i\langle r/y\rangle \doteq N_i\langle r'/y\rangle \in \mathsf{bool}\ [\Psi']$. By Theorem 4.63, either $N_i \doteq \mathsf{true} \in \mathsf{bool}\ [\Psi', y]$ or $N_i \doteq \mathsf{false} \in \mathsf{bool}\ [\Psi', y]$; in both cases, $N_i\langle r/y\rangle \doteq N_i\langle r'/y\rangle \in \mathsf{bool}\ [\Psi']$.

For coercion, show if $M \doteq M' \in \mathsf{bool}\ [\Psi']$ that $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{bool}}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{bool}}(M') \in \mathsf{bool}\ [\Psi']$. This holds by Lemma 4.32 on both sides, because $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{bool}}(M) \longmapsto_{\Box} M$ and $M \doteq M' \in \mathsf{bool}\ [\Psi']$. By the same argument, when $r = r'$, $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{bool}}(M) \doteq M \in \mathsf{bool}\ [\Psi']$. □

4.4.7 Natural numbers

For $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$, $\tau^\kappa_i(\Psi, \mathsf{nat}, \mathsf{nat}, \mathbb{N}_\Psi)$ where $\mathbb{N}$ is the least context-indexed relation with $\mathbb{N}_\Psi(\mathsf{z}, \mathsf{z})$ and $\mathbb{N}_\Psi(\mathsf{s}(M), \mathsf{s}(M'))$ whenever $M \sim M' \in \mathbb{N}\ [\Psi]$. Relative to τ^κ_i:

Rule 4.65 (Pretype formation)**.** $\mathsf{nat}\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. By $\mathsf{nat}\ \mathsf{val}_{\Box}$, $\mathsf{nat} \sim \mathsf{nat} \downarrow \mathbb{N} \in \tau^\kappa_i\ [\Psi]$. To prove $\mathrm{Coh}(\mathbb{N})$, we must show $\mathsf{z} \sim \mathsf{z} \in \mathbb{N}\ [\Psi']$ (immediate by $\mathsf{z}\ \mathsf{val}_{\Box}$) and $\mathsf{s}(M) \sim \mathsf{s}(M') \in \mathbb{N}\ [\Psi']$ when $M \sim M' \in \mathbb{N}\ [\Psi']$ (which holds by $\mathsf{s}(M)\ \mathsf{val}_{\Box}$ and $M\psi \sim M'\psi \in \mathbb{N}\ [\Psi'']$ for all $\psi : \Psi' \to \Psi$). □

Rule 4.66 (Introduction)**.** $\mathsf{z} \in \mathsf{nat}\ [\Psi]$ *and if* $M \doteq M' \in \mathsf{nat}\ [\Psi]$ *then* $\mathsf{s}(M) \doteq \mathsf{s}(M') \in \mathsf{nat}\ [\Psi]$.

Proof. By Rule 4.65. □

Rule 4.67 (Elimination)**.** *If* $n : \mathsf{nat} \gg A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, $M \doteq M' \in \mathsf{nat}\ [\Psi]$, $Z \doteq Z' \in A[\mathsf{z}/n]\ [\Psi]$, *and* $n : \mathsf{nat}, a : A \gg S \doteq S' \in A[\mathsf{s}(n)/n]\ [\Psi]$, *then* $\mathsf{natrec}(M; Z, n.a.S) \doteq \mathsf{natrec}(M'; Z', n.a.S') \in A[M/n]\ [\Psi]$.

Proof. In essence, we "proceed by induction on $- \sim - \in \mathbb{N}\ [-]$," a relation obtained by lifting the context-indexed relation

$$\mathbb{N} := \mu R.(\{(\Psi, \mathsf{z}, \mathsf{z})\} \cup \{(\Psi, \mathsf{s}(M), \mathsf{s}(M')) \mid M \sim M' \in R\ [\Psi]\})$$

from values to terms (via various interval substitutions). More precisely, we show (1) the elimination rule lifts from values to terms, (2) the elimination rule holds for values, and thus (3) the elimination rule holds for terms.

Define $\Phi_\Psi(M_0, M_0')$ to hold when $\mathbb{N}_\Psi(M_0, M_0')$ and for all $n : \mathsf{nat} \gg A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, $Z \doteq Z' \in A[\mathsf{z}/n]\ [\Psi]$, and $n{:}\mathsf{nat}, a{:}A \gg S \doteq S' \in A[\mathsf{s}(n)/n]\ [\Psi]$, we have $\mathsf{natrec}(M_0; Z, n.a.S) \doteq \mathsf{natrec}(M_0'; Z', n.a.S') \in A[M_0/n]\ [\Psi]$.

1. If $M \sim M' \in \Phi\ [\Psi]$ then the elimination rule holds for M, M'.

   We have $M \sim M' \in \mathbb{N}\ [\Psi]$ by $\Phi \subseteq \mathbb{N}$ and monotonicity of candidate judgments. Apply coherent expansion to $\mathsf{natrec}(M; Z, n.a.S)$ at $A[M/n]\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$ with family $\{\mathsf{natrec}(M_\psi; Z\psi, n.a.S\psi) \mid M\psi \Downarrow M_\psi\}^{\Psi'}_{\psi}$. Then for all $\psi : \Psi' \to \Psi$ we obtain $\mathsf{natrec}(M_\psi; Z\psi, n.a.S\psi) \in A\psi[M_\psi/n]\ [\Psi']$ from $\Phi^{\Downarrow}_{\Psi}(M, M')$ (by $M \sim M' \in \Phi\ [\Psi]$). We must show

   $$\mathsf{natrec}(M_\psi; Z\psi, n.a.S\psi) \doteq \mathsf{natrec}((M_{\mathsf{id}_\Psi})\psi; Z\psi, n.a.S\psi) \in A\psi[M_\psi/n]\ [\Psi']$$

   but by Lemma 4.15 and $(M_{\mathsf{id}_\Psi})\psi \doteq M_\psi \in \mathsf{nat}\ [\Psi']$ it suffices to show these natrecs are related by $[\![A\psi[M_\psi/n]]\!]^{\Downarrow}$, which follows from $\Phi^{\Downarrow}_{\Psi'}((M_{\mathsf{id}_\Psi})\psi, M_\psi)$.

2. If $\mathbb{N}_\Psi(M_0, M_0')$ then $\Phi_\Psi(M_0, M_0')$.

   Let N be the function of which $\mathbb{N}$ is the least pre-fixed point; we prove Φ is also a pre-fixed point of N (that is, $N(\Phi) \subseteq \Phi$). Suppose $N(\Phi)_\Psi(M_0, M_0')$. Then either:

   a) $M_0 = M_0' = \mathsf{z}$.
   Show $\mathsf{natrec}(\mathsf{z}; Z, n.a.S) \doteq \mathsf{natrec}(\mathsf{z}; Z', n.a.S') \in A[\mathsf{z}/n]\ [\Psi]$, which is immediate by $Z \doteq Z' \in A[\mathsf{z}/n]\ [\Psi]$ and Lemma 4.32 on both sides.

   b) $M_0 = \mathsf{s}(M)$, $M_0' = \mathsf{s}(M')$, and $M \sim M' \in \Phi\ [\Psi]$.
   Show $\mathsf{natrec}(\mathsf{s}(M); Z, n.a.S) \doteq \mathsf{natrec}(\mathsf{s}(M'); Z', n.a.S') \in A[\mathsf{s}(M)/n]\ [\Psi]$. By Lemma 4.32 on both sides, it suffices to show

   $$\begin{aligned} &S[M/n][\mathsf{natrec}(M; Z, n.a.S)/a] \\ \doteq\ &S'[M'/n][\mathsf{natrec}(M'; Z', n.a.S')/a] \in A[\mathsf{s}(M)/n]\ [\Psi]. \end{aligned}$$

   We have $M \doteq M' \in \mathsf{nat}\ [\Psi]$ and $\mathsf{natrec}(M; Z, n.a.S) \doteq \mathsf{natrec}(M'; Z', n.a.S') \in A[M/n]\ [\Psi]$ by $M \sim M' \in \Phi\ [\Psi]$, so the result follows by $n{:}\mathsf{nat}, a{:}A \gg S \doteq S' \in A[\mathsf{s}(n)/n]\ [\Psi]$.

3. Assume $M \sim M' \in \mathbb{N}\ [\Psi]$; by monotonicity and $\mathbb{N} \subseteq \Phi$, $M \sim M' \in \Phi\ [\Psi]$. Thus the elimination rule holds for M, M', completing the proof. □

Rule 4.68 (Kan type formation)**.** $\mathsf{nat}\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. Identical to Rule 4.64. □

The existence property also holds in our computational type theory (Appendix A), following the same argument as in Idealized Nuprl (Theorem 2.42).

4.4.8 Circle

For $i \in \{0, 1, \dots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$, $\tau_i^\kappa(\Psi, \mathbb{S}^1, \mathbb{S}^1, \mathbb{C}_\Psi)$ where $\mathbb{C}$ is the least context-indexed relation with:

1. $\mathbb{C}_\Psi(\mathsf{base}, \mathsf{base})$,

2. $\mathbb{C}_{(\Psi,x)}(\mathsf{loop}_x, \mathsf{loop}_x)$, and

3. $\mathbb{C}_\Psi(\mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}), \mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}))$ whenever

   a) $r \neq r'$, $\neg\xi_i$ for all i, and $\overrightarrow{\xi_i}$ is valid,
   b) $M \sim M' \in \mathbb{C}\ [\Psi]$,
   c) $N_i\psi \sim N_j'\psi \in \mathbb{C}\ [\Psi']$ for all i, j and $\psi : \Psi' \to (\Psi, y)$ satisfying ξ_i, ξ_j, and
   d) $N_i\langle r/y\rangle\psi \sim M\psi \in \mathbb{C}\ [\Psi']$ for all i and $\psi : \Psi' \to \Psi$ satisfying ξ_i.

By $\mathbb{S}^1$ $\mathsf{val}_{\Box}$ it is immediate that $\mathbb{S}^1 \sim \mathbb{S}^1 \downarrow \mathbb{C} \in \tau_i^\kappa\ [\Psi]$. Relative to τ_i^κ:

Lemma 4.69. *If $\overrightarrow{\xi_i}$ is valid,*

1. *$M \sim M' \in \mathbb{C}\ [\Psi]$,*

2. *$N_i\psi \sim N_j'\psi \in \mathbb{C}\ [\Psi']$ for all i, j and $\psi : \Psi' \to (\Psi, y)$ satisfying ξ_i, ξ_j, and*

3. *$N_i\langle r/y\rangle\psi \sim M\psi \in \mathbb{C}\ [\Psi']$ for all i and $\psi : \Psi' \to \Psi$ satisfying ξ_i,*

then $\mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \sim \mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}) \in \mathbb{C}\ [\Psi]$.

Proof. Abbreviating the above hcoms L and R, we must show for any $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$ that $L\psi_1 \Downarrow L_1$, $R\psi_1 \Downarrow R_1$, and $\mathbb{C}_{\Psi_2}^{\Downarrow}$ relates $L_1\psi_2$, $L\psi_1\psi_2$, $R_1\psi_2$, and $R\psi_1\psi_2$. We proceed by cases on the first step taken by $L\psi_1$ and $L\psi_1\psi_2$.

1. $r\psi_1 = r'\psi_1$.

   Then $L\psi_1 \longmapsto_{\Box} M\psi_1$, $R\psi_1 \longmapsto_{\Box} M'\psi_1$, and the result follows from assumption (1).

2. $r\psi_1 \neq r'\psi_1$, $\xi_j\psi_1$ holds (and $\neg\xi_i\psi_1$ for all $i < j$), and $r\psi_1\psi_2 = r'\psi_1\psi_2$.

   Then $L\psi_1 \longmapsto N_j\langle r'/y\rangle\psi_1$, $L\psi_1\psi_2 \longmapsto M\psi_1\psi_2$, $R\psi_1 \longmapsto N'_j\langle r'/y\rangle\psi_1$, and $R\psi_1\psi_2 \longmapsto M'\psi_1\psi_2$. Because ψ_1 satisfies ξ_j, by (2) and (3), $N_j\psi_1 \sim N'_j\psi_1 \in \mathbb{C}\ [\Psi_1, y]$ and $N_j\langle r/y\rangle\psi_1 \sim M\psi_1 \in \mathbb{C}\ [\Psi_1]$. By the former at $\langle r'\psi_1/y\rangle, \psi_2$, $\mathbb{C}^{\Downarrow}_{\Psi_2}(N_j\langle r'/y\rangle\psi_1\psi_2, L_1\psi_2)$ and $\mathbb{C}^{\Downarrow}_{\Psi_2}(L_1\psi_2, R_1\psi_2)$. By the latter at $\psi_2, \mathsf{id}_{\Psi_2}$, $\mathbb{C}^{\Downarrow}_{\Psi_2}(N_j\langle r/y\rangle\psi_1\psi_2, M\psi_1\psi_2)$; by transitivity and $r\psi_1\psi_2 = r'\psi_1\psi_2$ we have $\mathbb{C}^{\Downarrow}_{\Psi_2}(L_1\psi_2, L\psi_1\psi_2)$. Finally, by (1), $\mathbb{C}^{\Downarrow}_{\Psi_2}(L\psi_1\psi_2, R\psi_1\psi_2)$.

3. $r\psi_1 \neq r'\psi_1$, $\xi_i\psi_1$ holds (and this is the least such i), $r\psi_1\psi_2 \neq r'\psi_1\psi_2$, and $\xi_j\psi_1\psi_2$ holds (and this is the least such $j \leq i$).

   Then $L\psi_1 \longmapsto N_i\langle r'/y\rangle\psi_1$, $L\psi_1\psi_2 \longmapsto N_j\langle r'/y\rangle\psi_1\psi_2$, $R\psi_1 \longmapsto N'_i\langle r'/y\rangle\psi_1$, and $R\psi_1\psi_2 \longmapsto N'_j\langle r'/y\rangle\psi_1\psi_2$. In this case, $\langle r'/y\rangle\psi_1\psi_2$ satisfies ξ_i, ξ_j; the result holds because $- \sim - \in \mathbb{C}\ [\Psi_2]$ relates $N_i\langle r'/y\rangle\psi_1\psi_2$, $N_j\langle r'/y\rangle\psi_1\psi_2$, $N'_i\langle r'/y\rangle\psi_1\psi_2$, and $N'_j\langle r'/y\rangle\psi_1\psi_2$.

4. $r\psi_1 \neq r'\psi_1$, $\neg\xi_i\psi_1$ for all i, and $r\psi_1\psi_2 = r'\psi_1\psi_2$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2 \longmapsto M\psi_1\psi_2$, $R\psi_1\ \mathsf{val}$, and $R\psi_1\psi_2 \longmapsto M'\psi_1\psi_2$. In this case, $L_1\psi_2 = L\psi_1\psi_2$ and $R_1\psi_2 = R\psi_1\psi_2$, so the result follows by (1).

5. $r\psi_1 \neq r'\psi_1$, $\neg\xi_i\psi_1$ for all i, $r\psi_1\psi_2 \neq r'\psi_1\psi_2$, and $\xi_j\psi_1\psi_2$ holds (the least such j).

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2 \longmapsto N_j\langle r'/y\rangle\psi_1\psi_2$, $R\psi_1\ \mathsf{val}$, and $R\psi_1\psi_2 \longmapsto N'_j\langle r'/y\rangle\psi_1\psi_2$. The result follows because $L_1\psi_2 = L\psi_1\psi_2$, $R_1\psi_2 = R\psi_1\psi_2$, and because $\langle r'/y\rangle\psi_1\psi_2$ satisfies ξ_j, $N_j\langle r'/y\rangle\psi_1\psi_2 \sim N'_j\langle r'/y\rangle\psi_1\psi_2 \in \mathbb{C}\ [\Psi_2]$.

6. $r\psi_1 \neq r'\psi_1$, $\neg\xi_i\psi_1$ for all i, $r\psi_1\psi_2 \neq r'\psi_1\psi_2$, and $\neg\xi_j\psi_1\psi_2$ for all j.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2\ \mathsf{val}$, $R\psi_1\ \mathsf{val}$, and $R\psi_1\psi_2\ \mathsf{val}$; it suffices to show $\mathbb{C}_{\Psi_2}(L\psi_1\psi_2, R\psi_1\psi_2)$. We know $\overrightarrow{\xi_i\psi_1\psi_2}$ is valid and $r_i\psi_1\psi_2 \neq r'_i\psi_1\psi_2$ for all i, so the result follows immediately by the third clause of the definition of $\mathbb{C}$. □

Rule 4.70 (Pretype formation). $\mathbb{S}^1\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. We must show $\mathsf{Coh}(\mathbb{C})$. There are three cases:

1. $\mathsf{base} \sim \mathsf{base} \in \mathbb{C}\ [\Psi]$.

   Immediate because $\mathsf{base}\ \mathsf{val}_{\square}$.

2. $\mathsf{loop}_x \sim \mathsf{loop}_x \in \mathbb{C}\ [\Psi, x]$.

   Let $\psi_1 : \Psi_1 \to (\Psi, x)$ and $\psi_2 : \Psi_2 \to \Psi_1$; show $\mathsf{loop}_{x\psi_1} \Downarrow M_1$ and $\mathbb{C}^{\Downarrow}_{\Psi_2}(M_1\psi_2, \mathsf{loop}_{x\psi_1\psi_2})$. If $x\psi_1 = \varepsilon$ then $M_1 = \mathsf{base}$, $\mathsf{loop}_{x\psi_1\psi_2} \longmapsto \mathsf{base}$, and $\mathbb{C}_{\Psi_2}(\mathsf{base}, \mathsf{base})$. If instead $x\psi_1 = x'$ and $x'\psi_2 = \varepsilon$, then $M_1 = \mathsf{loop}_{x'}$, $\mathsf{loop}_{x'\psi_2} \longmapsto \mathsf{base}$, $\mathsf{loop}_{x\psi_1\psi_2} \longmapsto \mathsf{base}$, and $\mathbb{C}_{\Psi_2}(\mathsf{base},\mathsf{base})$. Otherwise, $x\psi_1 = x'$ and $x'\psi_2 = x''$, so $M_1 = \mathsf{loop}_{x'}$ and $\mathbb{C}_{\Psi_2}(\mathsf{loop}_{x''},\mathsf{loop}_{x''})$.

3. $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}) \sim \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M';\overrightarrow{\xi_i\hookrightarrow y.N_i'}) \in \mathbb{C}\ [\Psi]$ where…

   This is a special case of Lemma 4.69 in which $r \neq r'$ and $\neg\xi_i$ for all i. □

Rule 4.71 (Introduction). $\mathsf{base} \in \mathbb{S}^1\ [\Psi]$, $\mathsf{loop}_\varepsilon \doteq \mathsf{base} \in \mathbb{S}^1\ [\Psi]$, *and* $\mathsf{loop}_r \in \mathbb{S}^1\ [\Psi]$.

Proof. The first follows from $\mathsf{Coh}(\mathbb{C})$, the second from $\mathsf{loop}_\varepsilon \longmapsto_{\Box} \mathsf{base}$ and Lemma 4.32, and the third from $\mathsf{Coh}(\mathbb{C})$ if $r = x$ and Lemma 4.32 if $r = \varepsilon$. □

Rule 4.72 (Kan type formation). $\mathbb{S}^1\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. For homogeneous composition, suppose $\overrightarrow{\xi_i}$ is valid,

1. $M \doteq M' \in \mathbb{S}^1\ [\Psi']$,

2. $N_i \doteq N_j' \in \mathbb{S}^1\ [\Psi', y \mid \xi_i, \xi_j]$ for any i, j, and

3. $N_i\langle r/y\rangle \doteq M \in \mathbb{S}^1\ [\Psi' \mid \xi_i]$ for any i; show

$\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M';\overrightarrow{\xi_i\hookrightarrow y.N_i'}) \in \mathbb{S}^1\ [\Psi']$, which is immediate by Lemma 4.69. Next, we must show if $r = r'$ that $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}) \doteq M \in \mathbb{S}^1\ [\Psi']$; this is immediate by $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}) \longmapsto_{\Box} M$ and Lemma 4.32. Finally, show if ξ_i holds that $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in \mathbb{S}^1\ [\Psi']$. Each side is an element of $\mathbb{S}^1$, so by Lemma 4.16 it suffices to show $\mathbb{C}^{\Downarrow}_{\Psi'}(\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M;\overrightarrow{\xi_i\hookrightarrow y.N_i}), N_i\langle r'/y\rangle)$. If $r = r'$ then $\mathsf{hcom} \longmapsto M$ and the result follows by $N_i\langle r/y\rangle \doteq M \in \mathbb{S}^1\ [\Psi' \mid \xi_i]$, because $\mathsf{id}_{\Psi'}$ satisfies ξ_i. Otherwise, let ξ_j be the first true equation. Then $\mathsf{hcom} \longmapsto N_j\langle r'/y\rangle$ and the result follows from $N_i \doteq N_j \in \mathbb{S}^1\ [\Psi', y \mid \xi_i, \xi_j]$.

For coercion, suppose that $M \doteq M' \in \mathbb{S}^1\ [\Psi']$ and show $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathbb{S}^1}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathbb{S}^1}(M') \in \mathbb{S}^1\ [\Psi']$. This is immediate by $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathbb{S}^1}(M) \longmapsto_{\Box} M$ and Lemma 4.32 on both sides. Similarly, if $r = r'$ then $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathbb{S}^1}(M) \doteq M \in \mathbb{S}^1\ [\Psi']$ by Lemma 4.32 on the left. □

Rule 4.73 (Computation).

1. *If* $P \in B\ [\Psi]$ *then* $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{base}; P, x.L) \doteq P \in B\ [\Psi]$.

2. *If* $L \in B\ [\Psi, x]$ *and* $L\langle\varepsilon/x\rangle \doteq P \in B\langle\varepsilon/x\rangle\ [\Psi]$ *for* $\varepsilon \in \{0,1\}$, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_r; P, x.L) \doteq L\langle r/x\rangle \in B\langle r/x\rangle\ [\Psi]$.

Proof. Like previous computation rules, part (1) is immediate by Lemma 4.32; in contrast, part (2) holds only under certain typing hypotheses, and not by untyped computation.

If $r = \varepsilon$, part (2) holds by Lemma 4.32 and $L\langle\varepsilon/x\rangle \doteq P \in B\langle\varepsilon/x\rangle\ [\Psi]$. If $r = y$, apply coherent expansion to the left with $\{P\psi \mid y\psi = \varepsilon\}^{\Psi'}_{\psi} \cup \{L\psi\langle z/x\rangle \mid y\psi = z\}^{\Psi'}_{\psi}$. The id_Ψ element of this family is $L\langle y/x\rangle$; when $y\psi = \varepsilon$, $L\langle y/x\rangle\psi \doteq P\psi \in B\langle y/x\rangle\psi\ [\Psi']$ (by $\langle y/x\rangle\psi = \langle\varepsilon/x\rangle\psi$), and when $y\psi = z$, $L\langle y/x\rangle\psi \doteq L\psi\langle z/x\rangle \in B\langle y/x\rangle\psi\ [\Psi']$ (by $\psi\langle z/x\rangle = \langle y/x\rangle\psi$). Thus by Lemma 4.18, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_y; P, x.L) \doteq L\langle y/x\rangle \in B\langle y/x\rangle\ [\Psi]$. □

To prove the elimination rule, as in Rule 4.67, we proceed by induction on $\mathbb{C}$, which we defined in Section 4.2 as the least pre-fixed point of a function we will notate C. Our inductive hypothesis $\Phi_\Psi(M_0, M_0')$ holds when:

1. $\mathbb{C}_\Psi(M_0, M_0')$ and
2. The elimination rule holds for M_0 and M_0'. That is: for all $c : \mathbb{S}^1 \gg A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, $P \doteq P' \in A[\mathsf{base}/c]\ [\Psi]$, $L \doteq L' \in A[\mathsf{loop}_x/c]\ [\Psi, x]$, and $L\langle\varepsilon/x\rangle \doteq P \in A[\mathsf{base}/c]\ [\Psi]$ for $\varepsilon \in \{0,1\}$, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M_0; P, x.L) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(M_0'; P', x.L') \in A[M_0/c]\ [\Psi]$.

Lemma 4.74. *If $M \sim M' \in \Phi\ [\Psi]$ then whenever $c : \mathbb{S}^1 \gg A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, $P \doteq P' \in A[\mathsf{base}/c]\ [\Psi]$, $L \doteq L' \in A[\mathsf{loop}_x/c]\ [\Psi, x]$, and $L\langle\varepsilon/x\rangle \doteq P \in A[\mathsf{base}/c]\ [\Psi]$ for $\varepsilon \in \{0,1\}$, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(M'; P', x.L') \in A[M/c]\ [\Psi]$.*

Proof. Apply coherent expansion to the left with family $\{\mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M_\psi; P\psi, x.L\psi) \mid M\psi \Downarrow M_\psi\}^{\Psi'}_{\psi}$, by showing that

$$\mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M_\psi; P\psi, x.L\psi) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}((M_{\mathsf{id}_\Psi})\psi; P\psi, x.L\psi) \in (A[M/c])\psi\ [\Psi'].$$

The left is an element of this type by $\Phi_{\Psi'}(M_\psi, M_\psi)$ and $A\psi[M_\psi/c] \doteq A\psi[M\psi/c]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$ (by $M_\psi \doteq M\psi \in \mathbb{S}^1\ [\Psi']$); the right is an element by $\Phi_\Psi(M_{\mathsf{id}_\Psi}, M_{\mathsf{id}_\Psi})$ and $A[M_{\mathsf{id}_\Psi}/c] \doteq A[M/c]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$. The equality follows by $(M_{\mathsf{id}_\Psi})\psi \Downarrow M_2$, $\Phi_{\Psi'}(M_\psi, M_2)$, and Lemma 4.16. Thus, by Lemma 4.18, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M_{\mathsf{id}_\Psi}; P, x.L) \in A[M/c]\ [\Psi]$.

By a symmetric argument on the right, $A[M/c] \doteq A'[M'/c]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ (by $M \doteq M' \in \mathbb{S}^1\ [\Psi]$), and transitivity, it remains only to establish that $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M_{\mathsf{id}_\Psi}; P, x.L) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(M'_{\mathsf{id}_\Psi}; P', x.L') \in A[M/c]\ [\Psi]$; this holds by $\Phi_\Psi(M_{\mathsf{id}_\Psi}, M'_{\mathsf{id}_\Psi})$ and $A[M_{\mathsf{id}_\Psi}/c] \doteq A[M/c]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$. □

Lemma 4.75. *If $C(\Phi)_\Psi(M_0, M_0')$ then $\Phi_\Psi(M_0, M_0')$.*

Proof. We must show that $\mathbb{C}_\Psi(M_0, M_0')$, and that if $c : \mathbb{S}^1 \gg A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, $P \doteq P' \in A[\mathsf{base}/c]\ [\Psi]$, $L \doteq L' \in A[\mathsf{loop}_x/c]\ [\Psi, x]$, and $L\langle\varepsilon/x\rangle \doteq P \in A[\mathsf{base}/c]\ [\Psi]$ for $\varepsilon \in \{0,1\}$, then $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M_0; P, x.L) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(M_0'; P', x.L') \in A[M_0/c]\ [\Psi]$. There are three cases:

1. $C(\Phi)_\Psi(\mathsf{base}, \mathsf{base})$.

   Then $\mathbb{C}_\Psi(\mathsf{base}, \mathsf{base})$ by definition, and the elimination rule holds by Lemma 4.32.

2. $C(\Phi)_{(\Psi,y)}(\mathsf{loop}_y, \mathsf{loop}_y)$.

   Then $\mathbb{C}_{(\Psi,y)}(\mathsf{loop}_y, \mathsf{loop}_y)$ by definition, and the elimination rule holds by Rule 4.73 on both sides (setting $B = A[\mathsf{loop}_x/c]$, and by $A[\mathsf{loop}_x/c]\langle\varepsilon/x\rangle \doteq A[\mathsf{base}/c]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$) and $L\langle y/x\rangle \doteq L'\langle y/x\rangle \in A[\mathsf{loop}_y/c]\ [\Psi]$.

3. $C(\Phi)_\Psi(\mathsf{hcom}_{\mathbb{S}^1}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}), \mathsf{hcom}_{\mathbb{S}^1}^{r\rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.N_i'}))$ where

   a) $r \neq r'$, $\neg\xi_i$ for all i, and $\overrightarrow{\xi_i}$ is valid,

   b) $M \sim M' \in \Phi\ [\Psi]$,

   c) $N_i\psi \sim N_j'\psi \in \Phi\ [\Psi']$ for all i, j and $\psi : \Psi' \to (\Psi, y)$ satisfying ξ_i, ξ_j, and

   d) $N_i\langle r/y\rangle\psi \sim M\psi \in \Phi\ [\Psi']$ for all i and $\psi : \Psi' \to \Psi$ satisfying ξ_i.

   By construction, $\Phi \subseteq \mathbb{C}$, so by monotonicity of candidate judgments, $\mathbb{C}_\Psi$ relates the above hcoms. By Lemma 4.74 and $M \sim M' \in \Phi\ [\Psi]$, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M, \dots) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(M', \dots) \in A[M/c]\ [\Psi]$. For all ψ satisfying ξ_i, ξ_j, $N_i\psi \sim N_j'\psi \in \Phi\ [\Psi']$, so by Lemma 4.74, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(N_i, \dots) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(N_j', \dots) \in A[N_i/c]\ [\Psi, y \mid \xi_i, \xi_j]$. Similarly, $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(N_i\langle r/y\rangle) \in A[M/c]\ [\Psi \mid \xi_i]$.

   Apply coherent expansion to $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{hcom}_{\mathbb{S}^1}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}); P, x.L)$ at type $A[\mathsf{hcom}_{\mathbb{S}^1}^{r\rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})/c]\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$ with family:

$$\begin{cases} \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M\psi; P\psi, x.L\psi) & r\psi = r'\psi \\ \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(N_j\langle r'/y\rangle\psi; P\psi, x.L\psi) & r\psi \neq r'\psi, \text{ least } j \text{ s.t. } \xi_j\psi \\ \mathsf{com}_{z.A\psi[F/c]}^{r\psi\rightsquigarrow r'\psi}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M\psi; P\psi, x.L\psi); \overrightarrow{\xi_i\psi \hookrightarrow y.T_i}) & \text{otherwise} \\ \quad F := \mathsf{hcom}_{\mathbb{S}^1}^{r\psi\rightsquigarrow z}(M\psi; \overrightarrow{\xi_i\psi \hookrightarrow y.N_i\psi}) & \\ \quad T_i := \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(N_i\psi; P\psi, x.L\psi) & \end{cases}$$

   We must check all three cross-cases, noting that id_Ψ falls in the third category above. In the first case, when $r\psi = r'\psi$:

$$\mathsf{com}_{z.A\psi[F/c]}^{r\psi\rightsquigarrow r'\psi}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M\psi); \overrightarrow{\xi_i\psi \hookrightarrow y.T_i}) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M\psi) \in A\psi[\mathsf{hcom}\psi/c]\ [\Psi']$$

   This holds by Theorem 4.33, $A\psi[\mathsf{hcom}\psi/c] = A\psi[F/c]\langle r'\psi/z\rangle$, $A\psi[F/c]\langle r'\psi/z\rangle \doteq A[M/c]\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$, and $A\psi[F/c] \doteq A[N_i\langle z/y\rangle/c]\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', z \mid \xi_i\psi]$. In the second case, when $r\psi \neq r'\psi$, $\xi_j\psi$ holds, and $\neg\xi_i\psi$ for $i < j$:

$$\begin{aligned}&\mathsf{com}^{r\psi\rightsquigarrow r'\psi}_{z.A\psi[F/c]}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M\psi); \overrightarrow{\xi_i\psi \hookrightarrow y.T_i})\\ &\doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(N_j\langle r'/y\rangle\psi) \in A\psi[\mathsf{hcom}\psi/c]\ [\Psi']\end{aligned}$$

As before, this holds by Theorem 4.33. Finally, when $r\psi \neq r'\psi$ and $\neg\xi_i\psi$ for all i:

$$\mathsf{com}^{r\psi\rightsquigarrow r'\psi}_{z.A\psi[F/c]}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A\psi}(M\psi); \overrightarrow{\xi_i\psi \hookrightarrow y.T_i}) \in A\psi[\mathsf{hcom}\psi/c]\ [\Psi']$$

Once again this holds by Theorem 4.33. Therefore, by Lemma 4.18:

$$\begin{aligned}&\mathsf{com}^{r\rightsquigarrow r'}_{z.A\psi[F/c]}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L); \overrightarrow{\xi_i \hookrightarrow y.\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(N_i; P, x.L)})\\ &\doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathbb{S}^1}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}); P, x.L) \in A[\mathsf{hcom}/c]\ [\Psi].\end{aligned}$$

By transitivity and a symmetric argument on the right, it suffices to show that two coms are equal, which follows from Theorem 4.33. □

Rule 4.76 (Elimination). *If* $c : \mathbb{S}^1 \gg A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, $M \doteq M' \in \mathbb{S}^1\ [\Psi]$, $P \doteq P' \in A[\mathsf{base}/c]\ [\Psi]$, $L \doteq L' \in A[\mathsf{loop}_x/c]\ [\Psi, x]$, *and* $L\langle\varepsilon/x\rangle \doteq P \in A[\mathsf{base}/c]\ [\Psi]$ *for* $\varepsilon \in \{0, 1\}$, *then* $\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L) \doteq \mathbb{S}^1\text{-}\mathsf{elim}_{c.A'}(M'; P', x.L') \in A[M/c]\ [\Psi]$.

Proof. By Lemma 4.75, Φ is a pre-fixed point of C, and therefore $\mathbb{C} \subseteq \Phi$ (because $\mathbb{C}$ is its least pre-fixed point). By monotonicity of the candidate judgments, $M \sim M' \in \Phi\ [\Psi]$; the result follows from Lemma 4.74. □

By restricting homogeneous composition to *valid* shapes $\overrightarrow{\xi_i}$, we obtain a sharper canonicity result than Cohen et al. [CCHM18], in which 0-dimensional elements of higher inductive types evaluate to constructors, not homogeneous compositions.

Theorem 4.77 (Canonicity II). *If* $M \in \mathbb{S}^1\ [\cdot]$ *then* $M \Downarrow \mathsf{base}$ *and* $M \doteq \mathsf{base} \in \mathbb{S}^1\ [\cdot]$*; moreover, the rules of Appendix A derive these equations.*

Proof. By definition, $\mathbb{C}^{\Downarrow}_{\cdot}(M, M)$. The only generator of $\mathbb{C}_{\cdot}$ is base, because there exist no valid shapes $\overrightarrow{\xi_i}$ with $\mathsf{FI}(\overrightarrow{\xi_i}) = \cdot$ and $\neg\xi_i$ for all i. (By Definition 4.28, all valid shapes in an empty interval context must contain either $0 = 0$ or $1 = 1$.) Therefore $M \Downarrow \mathsf{base}$, and the desired equation follows by Lemma 4.16 (which is a rule of Appendix A) and determinacy of evaluation. □

4.4.9 V-types

Throughout this section, we use the following abbreviations:

$$\begin{aligned}\mathsf{Fiber}(A,B,F,M) &:= (a{:}A)\times \mathsf{Path}_{\_.B}(F\ a, M)\\ \mathsf{isContr}(C) &:= (c{:}C)\times((c'{:}C)\to \mathsf{Path}_{\_.C}(c',c))\\ \mathsf{Equiv}(A,B) &:= (f{:}A\to B)\times((b{:}B)\to \mathsf{isContr}(\mathsf{Fiber}(A,B,f,b)))\end{aligned}$$

For $i\in\{0,1,\dots,\omega\}$ and $\kappa\in\{\mathsf{pre},\mathsf{Kan}\}$, $\tau_i^\kappa((\Psi,x),\mathsf{V}_x(A,B,E),\mathsf{V}_x(A',B',E'),\varphi)$ if and only if $\tau_i^\kappa \models (A \doteq A'\ \mathsf{type_{pre}}\ [\Psi,x \mid x=0])$, $\tau_i^\kappa \models (B \doteq B'\ \mathsf{type_{pre}}\ [\Psi,x])$, $\tau_i^\kappa \models (E \doteq E' \in \mathsf{Equiv}(A,B)\ [\Psi,x \mid x=0])$, and $\varphi(\mathsf{Vin}_x(M,N),\mathsf{Vin}_x(M',N'))$ whenever

1. $\tau_i^\kappa \models (N \doteq N' \in B\ [\Psi,x])$,
2. $\tau_i^\kappa \models (M \doteq M' \in A\ [\Psi,x \mid x=0])$, and
3. $\tau_i^\kappa \models (\mathsf{fst}(E)\ M \doteq N \in B\ [\Psi,x \mid x=0])$.

(Note that $\tau_i^\kappa \models (\mathsf{Equiv}(A,B)\ \mathsf{type_{pre}}\ [\Psi,x \mid x=0])$ by the formation, introduction, and elimination rules for dependent functions, dependent pairs, and paths.) Relative to τ_i^κ:

Rule 4.78 (Pretype formation)**.**

1. *If* A $\mathsf{type_{pre}}\ [\Psi]$ *then* $\mathsf{V}_0(A,B,E) \doteq A\ \mathsf{type_{pre}}\ [\Psi]$.
2. *If* B $\mathsf{type_{pre}}\ [\Psi]$ *then* $\mathsf{V}_1(A,B,E) \doteq B\ \mathsf{type_{pre}}\ [\Psi]$.
3. *If* $A \doteq A'\ \mathsf{type_{pre}}\ [\Psi \mid r=0]$, $B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$, *and* $E \doteq E' \in \mathsf{Equiv}(A,B)\ [\Psi \mid r=0]$, *then* $\mathsf{V}_r(A,B,E) \doteq \mathsf{V}_r(A',B',E')\ \mathsf{type_{pre}}\ [\Psi]$.

Proof. Parts (1–2) hold by Lemma 4.32. For part (3), we must first show $\mathsf{V}_r(A,B,E) \sim \mathsf{V}_r(A',B',E') \downarrow \_ \in \tau_i^\kappa\ [\Psi]$, which is to say that, abbreviating these terms L and R, for all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2 : \Psi_2 \to \Psi_1$, $L\psi_1 \Downarrow L_1$, $R\psi_1 \Downarrow R_1$, $(\tau_i^\kappa)^{\Downarrow}(\Psi_2, L_1\psi_2, L\psi_1\psi_2, \_)$, $(\tau_i^\kappa)^{\Downarrow}(\Psi_2, R_1\psi_2, R\psi_1\psi_2, \_)$, and $(\tau_i^\kappa)^{\Downarrow}(\Psi_2, L_1\psi_2, R_1\psi_2, \_)$. We proceed by cases on the first step taken by $L\psi_1$ and $L\psi_1\psi_2$.

1. $r\psi_1 = 0$.

   Then $L\psi_1 \longmapsto_{\square} A\psi_1$, $R\psi_1 \longmapsto_{\square} A'\psi_1$, and the result follows by $A\psi_1 \doteq A'\psi_1\ \mathsf{type_{pre}}\ [\Psi_1]$.

2. $r\psi_1 = 1$.

   Then $L\psi_1 \longmapsto_{\square} B\psi_1$, $R\psi_1 \longmapsto_{\square} B'\psi_1$, and the result follows by $B \doteq B'\ \mathsf{type_{pre}}\ [\Psi]$.

3. $r\psi_1 = x$ and $r\psi_1\psi_2 = 0$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2 \longmapsto A\psi_1\psi_2$, $R\psi_1\ \mathsf{val}$, $R\psi_1\psi_2 \longmapsto A'\psi_1\psi_2$, and the result follows by $A\psi_1\psi_2 \doteq A'\psi_1\psi_2\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_2]$.

4. $r\psi_1 = x$ and $r\psi_1\psi_2 = 1$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2 \longmapsto B\psi_1\psi_2$, $R\psi_1\ \mathsf{val}$, $R\psi_1\psi_2 \longmapsto B'\psi_1\psi_2$, and the result follows by $B \doteq B'\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

5. $r\psi_1 = x$ and $r\psi_1\psi_2 = x'$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2\ \mathsf{val}$, $R\psi_1\ \mathsf{val}$, $R\psi_1\psi_2\ \mathsf{val}$, and by $A\psi_1\psi_2 \doteq A'\psi_1\psi_2\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_2 \mid x' = 0]$, $B\psi_1\psi_2 \doteq B'\psi_1\psi_2\ \mathsf{type}_{\mathsf{pre}}\ [\Psi_2]$, and $E\psi_1\psi_2 \doteq E'\psi_1\psi_2 \in \mathsf{Equiv}(A\psi_1\psi_2, B\psi_1\psi_2)\ [\Psi_2 \mid x' = 0]$, we have $\tau_i^\kappa(\Psi_2, \mathsf{V}_{x'}(A\psi_1\psi_2, B\psi_1\psi_2, E\psi_1\psi_2), \mathsf{V}_{x'}(A'\psi_1\psi_2, B'\psi_1\psi_2, E'\psi_1\psi_2), \_)$.

It remains only to prove $\mathsf{Coh}(\llbracket \mathsf{V}_r(A,B,E)\rrbracket)$. For any $\psi : \Psi' \to \Psi$, suppose that $\llbracket \mathsf{V}_{r\psi}(A\psi, B\psi, E\psi)\rrbracket(M_0, N_0)$, and show $M_0 \sim N_0 \in \llbracket \mathsf{V}_{r\psi}(A\psi, B\psi, E\psi)\rrbracket\ [\Psi']$. If $r\psi = 0$, coherence follows from $\llbracket \mathsf{V}_0(A\psi, B\psi, E\psi)\rrbracket = \llbracket A\psi\rrbracket$; if $r\psi = 1$, coherence follows from $\llbracket \mathsf{V}_1(A\psi, B\psi, E\psi)\rrbracket = \llbracket B\psi\rrbracket$. Otherwise, $\llbracket \mathsf{V}_x(A\psi, B\psi, E\psi)\rrbracket(\mathsf{Vin}_x(M,N), \mathsf{Vin}_x(M',N'))$, or equivalently, $N \doteq N' \in B\psi\ [\Psi']$, $M \doteq M' \in A\psi\ [\Psi' \mid x = 0]$, and $\mathsf{fst}(E\psi)\ M \doteq N \in B\psi\ [\Psi' \mid x = 0]$. We proceed by cases on the first step taken by the ψ_1 and $\psi_1\psi_2$ instances of the left:

1. $x\psi_1 = 0$.

   Then $L\psi_1 \longmapsto_{\square} M\psi_1$, $R\psi_1 \longmapsto_{\square} M'\psi_1$, and the result follows by $\llbracket \mathsf{V}_0(A\psi\psi_1, \dots)\rrbracket = \llbracket A\psi\psi_1\rrbracket$ and $M\psi_1 \doteq M'\psi_1 \in A\psi\psi_1\ [\Psi_1]$.

2. $x\psi_1 = 1$.

   Then $L\psi_1 \longmapsto_{\square} N\psi_1$, $R\psi_1 \longmapsto_{\square} N'\psi_1$, and the result follows by $\llbracket \mathsf{V}_1(A\psi\psi_1, \dots)\rrbracket = \llbracket B\psi\psi_1\rrbracket$ and $N \doteq N' \in B\psi\ [\Psi']$.

3. $x\psi_1 = x'$ and $x\psi_1\psi_2 = 0$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2 \longmapsto M\psi_1\psi_2$, $R\psi_1\ \mathsf{val}$, $R\psi_1\psi_2 \longmapsto M'\psi_1\psi_2$, and the result follows by $\llbracket \mathsf{V}_0(A\psi\psi_1\psi_2, \dots)\rrbracket = \llbracket A\psi\psi_1\psi_2\rrbracket$ and $M\psi_1\psi_2 \doteq M'\psi_1\psi_2 \in A\psi\psi_1\psi_2\ [\Psi_2]$.

4. $x\psi_1 = x'$ and $x\psi_1\psi_2 = 1$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2 \longmapsto N\psi_1\psi_2$, $R\psi_1\ \mathsf{val}$, $R\psi_1\psi_2 \longmapsto N'\psi_1\psi_2$, and the result follows by $\llbracket \mathsf{V}_1(A\psi\psi_1\psi_2, \dots)\rrbracket = \llbracket B\psi\psi_1\psi_2\rrbracket$ and $N \doteq N' \in B\psi\ [\Psi']$.

5. $x\psi_1 = x'$ and $x\psi_1\psi_2 = x''$.

   Then $L\psi_1\ \mathsf{val}$, $L\psi_1\psi_2\ \mathsf{val}$, $R\psi_1\ \mathsf{val}$, $R\psi_1\psi_2\ \mathsf{val}$, and by $N\psi_1\psi_2 \doteq N'\psi_1\psi_2 \in B\psi\psi_1\psi_2\ [\Psi_2]$, $M\psi_1\psi_2 \doteq M'\psi_1\psi_2 \in A\psi\psi_1\psi_2\ [\Psi_2 \mid x'' = 0]$, and $\mathsf{fst}(E\psi\psi_1\psi_2)\ M\psi_1\psi_2 \doteq N\psi_1\psi_2 \in B\psi\ [\Psi_2 \mid x'' = 0]$, $\llbracket \mathsf{V}_{x''}(A\psi\psi_1\psi_2, \dots)\rrbracket(\mathsf{Vin}_{x''}(M\psi_1\psi_2, N\psi_1\psi_2), \mathsf{Vin}_{x''}(M'\psi_1\psi_2, N'\psi_1\psi_2))$. □

Rule 4.79 (Introduction)**.**

1. *If* $M \in A\ [\Psi]$ *then* $\mathsf{Vin}_0(M, N) \doteq M \in A\ [\Psi]$.

2. *If* $N \in B\ [\Psi]$ *then* $\mathsf{Vin}_1(M, N) \doteq N \in B\ [\Psi]$.

3. *If* $M \doteq M' \in A\ [\Psi \mid r = 0]$, $N \doteq N' \in B\ [\Psi]$, $E \in \mathsf{Equiv}(A, B)\ [\Psi \mid r = 0]$, *and* $\mathsf{fst}(E)\ M \doteq N \in B\ [\Psi \mid r = 0]$, *then* $\mathsf{Vin}_r(M, N) \doteq \mathsf{Vin}_r(M', N') \in \mathsf{V}_r(A, B, E)\ [\Psi]$.

Proof. Parts (1–2) hold by Lemma 4.32. For part (3), if $r = 0$ or $r = 1$, the result follows from parts (1–2) and Rule 4.78. If $r = x$, the result is immediate by $\mathsf{Coh}(\llbracket \mathsf{V}_x(A, B, E) \rrbracket)$. □

Rule 4.80 (Elimination)**.**

1. *If* $M \in A\ [\Psi]$ *and* $F \in A \to B\ [\Psi]$, *then* $\mathsf{Vproj}_0(M, F) \doteq F\ M \in B\ [\Psi]$.

2. *If* $M \in B\ [\Psi]$ *then* $\mathsf{Vproj}_1(M, F) \doteq M \in B\ [\Psi]$.

3. *If* $M \doteq M' \in \mathsf{V}_r(A, B, E)\ [\Psi]$ *and* $F \doteq \mathsf{fst}(E) \in A \to B\ [\Psi \mid r = 0]$, *then* $\mathsf{Vproj}_r(M, F) \doteq \mathsf{Vproj}_r(M', \mathsf{fst}(E)) \in B\ [\Psi]$.

Proof. Parts (1–2) hold by Lemma 4.32. For part (3), if $r = 0$ or $r = 1$, the result follows from parts (1–2), Rule 4.39, and Rule 4.78. If $r = x$, apply coherent expansion to the left with family

$$\begin{cases} F\psi\ M\psi & x\psi = 0 \\ M\psi & x\psi = 1 \\ N_\psi & x\psi = x', M\psi \Downarrow \mathsf{Vin}_{x'}(O_\psi, N_\psi) \end{cases}$$

where $O_\psi \in A\psi\ [\Psi' \mid x' = 0]$, $N_\psi \in B\psi\ [\Psi']$, and $\mathsf{fst}(E\psi)\ O_\psi \doteq N_\psi \in B\psi\ [\Psi' \mid x' = 0]$. First, show that if $x\psi = 0$, $F\psi\ M\psi \doteq (N_{\mathsf{id}_\Psi})\psi \in B\psi\ [\Psi']$. By Lemma 4.16, $M \doteq \mathsf{Vin}_x(O_{\mathsf{id}_\Psi}, N_{\mathsf{id}_\Psi}) \in \mathsf{V}_x(A, B, E)\ [\Psi]$, so by Rule 4.79, $M\psi \doteq (O_{\mathsf{id}_\Psi})\psi \in A\psi\ [\Psi']$. By assumption, $F\psi \doteq \mathsf{fst}(E\psi) \in A\psi \to B\psi\ [\Psi']$. This case is completed by Rule 4.39 and $\mathsf{fst}(E\psi)\ (O_{\mathsf{id}_\Psi})\psi \doteq (N_{\mathsf{id}_\Psi})\psi \in B\psi\ [\Psi']$. Next, show that if $x\psi = 1$, $M\psi \doteq (N_{\mathsf{id}_\Psi})\psi \in B\psi\ [\Psi']$. This case is immediate by Rule 4.79 and $M \doteq \mathsf{Vin}_x(O_{\mathsf{id}_\Psi}, N_{\mathsf{id}_\Psi}) \in \mathsf{V}_x(A, B, E)\ [\Psi]$ under ψ. Finally, show that if $x\psi = x'$, $N_\psi \doteq (N_{\mathsf{id}_\Psi})\psi \in B\psi\ [\Psi']$. By $M \in \mathsf{V}_x(A, B, E)\ [\Psi]$ under id_Ψ, ψ we have $\llbracket \mathsf{V}_x(A, B, E) \rrbracket_\psi(\mathsf{Vin}_{x'}(O_\psi, N_\psi), \mathsf{Vin}_{x'}((O_{\mathsf{id}_\Psi})\psi, (N_{\mathsf{id}_\Psi})\psi))$, completing this case.

By Lemma 4.18 we conclude $\mathsf{Vproj}_x(M, F) \doteq N_{\mathsf{id}_\Psi} \in B\ [\Psi]$, and by a symmetric argument, $\mathsf{Vproj}_x(M', \mathsf{fst}(E)) \doteq N'_{\mathsf{id}_\Psi} \in B\ [\Psi]$. We complete the proof with transitivity and $N_{\mathsf{id}_\Psi} \doteq N'_{\mathsf{id}_\Psi} \in B\ [\Psi]$ by $\llbracket \mathsf{V}_x(A, B, E) \rrbracket(\mathsf{Vin}_x(O_{\mathsf{id}_\Psi}, N_{\mathsf{id}_\Psi}), \mathsf{Vin}_x(O'_{\mathsf{id}_\Psi}, N'_{\mathsf{id}_\Psi}))$. □

Rule 4.81 (Computation)**.** *If* $M \in A\ [\Psi \mid r = 0]$, $N \in B\ [\Psi]$, $F \in A \to B\ [\Psi \mid r = 0]$, *and* $F\ M \doteq N \in B\ [\Psi \mid r = 0]$, *then* $\mathsf{Vproj}_r(\mathsf{Vin}_r(M, N), F) \doteq N \in B\ [\Psi]$.

Proof. If $r = 0$ then by Lemma 4.32 it suffices to show $F\ \mathsf{Vin}_0(M, N) \doteq N \in B\ [\Psi]$, which follows from Rules 4.39 and 4.79 and our hypothesis $F\ M \doteq N \in B\ [\Psi]$. If $r = 1$ the result holds by Lemma 4.32. If $r = x$ we apply coherent expansion to the left with family

$$\begin{cases} F\psi\ \mathsf{Vin}_0(M\psi, N\psi) & x\psi = 0 \\ N\psi & x\psi = 1 \text{ or } x\psi = x' \end{cases}$$

If $x\psi = 0$ then $F\psi\ \mathsf{Vin}_0(M\psi, N\psi) \doteq N\psi \in B\psi\ [\Psi']$ by Rules 4.39 and 4.79 and $F\ M \doteq N \in B\ [\Psi \mid x = 0]$. If $x\psi \neq 0$ then $N\psi \in B\psi\ [\Psi']$ and the result follows by Lemma 4.18. □

Rule 4.82 (Uniqueness). *If* $N \in \mathsf{V}_r(A, B, E)\ [\Psi]$ *and* $M \doteq N \in A\ [\Psi \mid r = 0]$, *then* $\mathsf{Vin}_r(M, \mathsf{Vproj}_r(N, \mathsf{fst}(E))) \doteq N \in \mathsf{V}_r(A, B, E)\ [\Psi]$.

Proof. If $r = 0$ or $r = 1$ the result is immediate by Lemma 4.32 and Rule 4.78. If $r = x$ then by Lemma 4.16, $N \doteq \mathsf{Vin}_x(M', P') \in \mathsf{V}_x(A, B, E)\ [\Psi]$ where $M' \in A\ [\Psi \mid x = 0]$, $P' \in B\ [\Psi]$, and $\mathsf{fst}(E)\ M' \doteq P' \in B\ [\Psi \mid x = 0]$. By Rule 4.79 it suffices to show that $M \doteq M' \in A\ [\Psi \mid x = 0]$, $\mathsf{Vproj}_x(N, \mathsf{fst}(E)) \doteq P' \in B\ [\Psi]$, and $\mathsf{fst}(E)\ M' \doteq P' \in B\ [\Psi \mid x = 0]$ (which is immediate). To show $M \doteq M' \in A\ [\Psi \mid x = 0]$ it suffices to prove $N \doteq M' \in A\ [\Psi \mid x = 0]$, which follows from $N \doteq \mathsf{Vin}_x(M', P') \in \mathsf{V}_x(A, B, E)\ [\Psi]$ and Rules 4.78 and 4.79. To show $\mathsf{Vproj}_x(N, \mathsf{fst}(E)) \doteq P' \in B\ [\Psi]$, by Rule 4.80 it suffices to check $\mathsf{Vproj}_x(\mathsf{Vin}_x(M', P'), \mathsf{fst}(E)) \doteq P' \in B\ [\Psi]$, which holds by Rule 4.81. □

Lemma 4.83. *If* $A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid x = 0]$, $B \doteq B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, $E \doteq E' \in \mathsf{Equiv}(A, B)\ [\Psi \mid x = 0]$, $\overrightarrow{\xi_i}$ *is valid,*

1. $M \doteq M' \in \mathsf{V}_x(A, B, E)\ [\Psi]$,
2. $N_i \doteq N'_j \in \mathsf{V}_x(A, B, E)\ [\Psi, y \mid \xi_i, \xi_j]$ *for any* i, j, *and*
3. $N_i\langle r/y\rangle \doteq M \in \mathsf{V}_x(A, B, E)\ [\Psi \mid \xi_i]$ *for any* i,

then

1. $\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{V}_x(A,B,E)}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{V}_x(A',B',E')}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in \mathsf{V}_x(A, B, E)\ [\Psi]$;
2. *if* $r = r'$ *then* $\mathsf{hcom}^{r \rightsquigarrow r}_{\mathsf{V}_x(A,B,E)}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in \mathsf{V}_x(A, B, E)\ [\Psi]$; *and*
3. *if* ξ_i *then* $\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{V}_x(A,B,E)}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in \mathsf{V}_x(A, B, E)\ [\Psi]$.

Proof. For part (1), apply coherent expansion to $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{V}_x(A,B,E)}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i})$ with family

$$\begin{cases} \mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{A\psi}(M\psi;\overrightarrow{\xi_i\psi \hookrightarrow y.N_i\psi}) & x\psi = 0 \\ \mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{B\psi}(M\psi;\overrightarrow{\xi_i\psi \hookrightarrow y.N_i\psi}) & x\psi = 1 \\ (\mathsf{Vin}_x(O\langle r'/y\rangle, \mathsf{hcom}^{r\rightsquigarrow r'}_{B}(\mathsf{Vproj}_x(M,\mathsf{fst}(E)); \overrightarrow{T})))\psi & x\psi = x' \\ \quad O := \mathsf{hcom}^{r\rightsquigarrow y}_{A}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) & \\ \quad \overrightarrow{T} := \overrightarrow{\xi_i \hookrightarrow y.\mathsf{Vproj}_x(N_i,\mathsf{fst}(E))}, & \\ \qquad\quad x = 0 \hookrightarrow y.\mathsf{fst}(E)\ O, & \\ \qquad\quad x = 1 \hookrightarrow y.\mathsf{hcom}^{r\rightsquigarrow y}_{B}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) & \end{cases}$$

Consider $\psi = \mathsf{id}_\Psi$. Using rules for dependent functions, dependent types, and univalence:

1. $O \in A\ [\Psi, y \mid x = 0]$ and $O\langle r/y\rangle \doteq M \in A\ [\Psi \mid x = 0]$ (by $\mathsf{V}_x(A,B,E) \doteq A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi \mid x = 0]$).

2. $\mathsf{Vproj}_x(M,\mathsf{fst}(E)) \in B\ [\Psi]$ where $\mathsf{Vproj}_x(M,\mathsf{fst}(E)) \doteq \mathsf{fst}(E)\ M \in B\ [\Psi \mid x = 0]$ and $\mathsf{Vproj}_x(M,\mathsf{fst}(E)) \doteq M \in B\ [\Psi \mid x = 1]$.

3. $\mathsf{Vproj}_x(N_i,\mathsf{fst}(E)) \doteq \mathsf{Vproj}_x(N_j,\mathsf{fst}(E)) \in B\ [\Psi, y \mid \xi_i, \xi_j]$ and $\mathsf{Vproj}_x(M,\mathsf{fst}(E)) \doteq \mathsf{Vproj}_x(N_i\langle r/y\rangle,\mathsf{fst}(E)) \in B\ [\Psi \mid \xi_i]$.

4. $\mathsf{fst}(E)\ O \in B\ [\Psi, y \mid x = 0]$, $\mathsf{fst}(E)\ O \doteq \mathsf{Vproj}_x(N_i,\mathsf{fst}(E)) \in B\ [\Psi, y \mid x = 0, \xi_i]$ (both equal $\mathsf{fst}(E)\ N_i$), and $\mathsf{fst}(E)\ O\langle r/y\rangle \doteq \mathsf{Vproj}_x(M,\mathsf{fst}(E)) \in B\ [\Psi \mid x = 0]$ (both equal $\mathsf{fst}(E)\ M$).

5. $\mathsf{hcom}^{r\rightsquigarrow y}_{B}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \in B\ [\Psi, y \mid x = 1]$ (by $\mathsf{V}_x(A,B,E) \doteq B\ \mathsf{type}_{\mathsf{pre}}\ [\Psi \mid x = 1]$), $\mathsf{hcom}^{r\rightsquigarrow y}_{B}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{Vproj}_x(N_i,\mathsf{fst}(E)) \in B\ [\Psi, y \mid x = 1, \xi_i]$ (both equal N_i), and $\mathsf{hcom}^{r\rightsquigarrow r}_{B}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{Vproj}_x(M,\mathsf{fst}(E)) \in B\ [\Psi \mid x = 1]$ (both equal M).

6. By the above, $\mathsf{hcom}^{r\rightsquigarrow r'}_{B}(\mathsf{Vproj}_x(M,\mathsf{fst}(E));\overrightarrow{T}) \in B\ [\Psi]$ and $\mathsf{hcom}_B \doteq \mathsf{fst}(E)\ O\langle r'/y\rangle \in B\ [\Psi \mid x = 0]$, so $\mathsf{Vin}_x(O\langle r'/y\rangle, \mathsf{hcom}^{r\rightsquigarrow r'}_{B}(\mathsf{Vproj}_x(M,\mathsf{fst}(E));\overrightarrow{T})) \in \mathsf{V}_x(A,B,E)\ [\Psi]$.

When $x\psi = x'$, coherence is immediate. When $x\psi = 0$, $\mathsf{Vin}_0(O\langle r'\psi/y\rangle, \dots) \doteq \mathsf{hcom}_{A\psi} \in A\psi\ [\Psi']$ as required. When $x\psi = 1$, $\mathsf{Vin}_1(\dots, \mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{B\psi}(\dots;\overrightarrow{T})) \doteq \mathsf{hcom}_{B\psi} \in B\psi\ [\Psi']$ as required. Part (1) follows by Lemma 4.18 and a symmetric argument on the right.

For part (2), show $\mathsf{Vin}_x(O\langle r'/y\rangle, \mathsf{hcom}^{r\rightsquigarrow r'}_{B}(\mathsf{Vproj}_x(M,\mathsf{fst}(E));\overrightarrow{T})) \doteq M \in \mathsf{V}_x(A,B,E)\ [\Psi]$ when $r = r'$. By the above, $\mathsf{Vin}_x(\dots) \doteq \mathsf{Vin}_x(M, \mathsf{Vproj}_x(M,\mathsf{fst}(E))) \in \mathsf{V}_x(A,B,E)\ [\Psi]$, so the result follows by Rule 4.82.

For part (3), we must show $\mathsf{Vin}_x(O\langle r'/y\rangle, \mathsf{hcom}_B^{r\rightsquigarrow r'}(\mathsf{Vproj}_x(M, \mathsf{fst}(E)); \overrightarrow{T})) \doteq N_i\langle r'/y\rangle \in \mathsf{V}_x(A, B, E)\ [\Psi]$ when ξ_i. But we have $\mathsf{Vin}_x(\dots) \doteq \mathsf{Vin}_x(N_i\langle r'/y\rangle, \mathsf{Vproj}_x(N_i\langle r'/y\rangle, \mathsf{fst}(E))) \in \mathsf{V}_x(A, B, E)\ [\Psi]$, so the result again follows by Rule 4.82. □

Lemma 4.84. *If $x \neq y$,*

1. $A \doteq A'\ \mathsf{type_{Kan}}\ [\Psi, y \mid x = 0]$,
2. $B \doteq B'\ \mathsf{type_{Kan}}\ [\Psi, y]$,
3. $E \doteq E' \in \mathsf{Equiv}(A, B)\ [\Psi, y \mid x = 0]$, *and*
4. $M \doteq M' \in (\mathsf{V}_x(A, B, E))\langle r/y\rangle\ [\Psi]$,

then

1. $\mathsf{coe}_{y.\mathsf{V}_x(A,B,E)}^{r\rightsquigarrow r'}(M) \doteq \mathsf{coe}_{y.\mathsf{V}_x(A',B',E')}^{r\rightsquigarrow r'}(M') \in (\mathsf{V}_x(A, B, E))\langle r'/y\rangle\ [\Psi]$ *and*
2. $\mathsf{coe}_{y.\mathsf{V}_x(A,B,E)}^{r\rightsquigarrow r}(M) \doteq M \in (\mathsf{V}_x(A, B, E))\langle r/y\rangle\ [\Psi]$.

Proof. We apply coherent expansion to $\mathsf{coe}_{y.\mathsf{V}_x(A,B,E)}^{r\rightsquigarrow r'}(M)$ with family

$$\begin{cases} \mathsf{coe}_{y.A\psi}^{r\psi\rightsquigarrow r'\psi}(M\psi) & x\psi = 0 \\ \mathsf{coe}_{y.B\psi}^{r\psi\rightsquigarrow r'\psi}(M\psi) & x\psi = 1 \\ (\mathsf{Vin}_x(\mathsf{coe}_{y.A}^{r\rightsquigarrow r'}(M), \mathsf{com}_{y.B}^{r\rightsquigarrow r'}(\mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)); \overrightarrow{T})))\psi & x\psi = x' \\ \quad \overrightarrow{T} := x = 0 \hookrightarrow y.\mathsf{fst}(E)\ \mathsf{coe}_{y.A}^{r\rightsquigarrow y}(M), & \\ \qquad\qquad x = 1 \hookrightarrow y.\mathsf{coe}_{y.B}^{r\rightsquigarrow y}(M) & \end{cases}$$

Consider $\psi = \mathsf{id}_\Psi$.

1. $\mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)) \in B\langle r/y\rangle\ [\Psi]$ (by our hypothesis $M \in \mathsf{V}_x(A\langle r/y\rangle, \dots)\ [\Psi]$), with boundary $\mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)) \doteq \mathsf{fst}(E\langle r/y\rangle)\ M \in B\langle r/y\rangle\ [\Psi \mid x = 0]$ and $\mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)) \doteq M \in B\langle r/y\rangle\ [\Psi \mid x = 1]$.

2. $\mathsf{fst}(E)\ \mathsf{coe}_{y.A}^{r\rightsquigarrow y}(M) \in B\ [\Psi, y \mid x = 0]$ because $\mathsf{fst}(E) \in A \to B\ [\Psi, y \mid x = 0]$ and $\mathsf{coe}_{y.A}^{r\rightsquigarrow y}(M) \in A\ [\Psi, y \mid x = 0]$ (by $M \in A\langle r/y\rangle\ [\Psi \mid x = 0]$). Under $\langle r/y\rangle$ this $\doteq \mathsf{fst}(E\langle r/y\rangle)\ M \in B\langle r/y\rangle\ [\Psi \mid x = 0]$.

3. $\mathsf{coe}_{y.B}^{r\rightsquigarrow y}(M) \in B\ [\Psi, y \mid x = 1]$ (by $M \in B\langle r/y\rangle\ [\Psi \mid x = 1]$) and $\mathsf{coe}_{y.B}^{r\rightsquigarrow r}(M) \doteq M \in B\langle r/y\rangle\ [\Psi \mid x = 1]$.

4. Therefore $\mathsf{com}_{y.B} \in B\langle r'/y\rangle\ [\Psi]$, $\mathsf{com}_{y.B} \doteq \mathsf{fst}(E\langle r'/y\rangle)\ \mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M) \in B\langle r'/y\rangle\ [\Psi \mid x = 0]$, and $\mathsf{com}_{y.B} \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{y.B}(M) \in B\langle r'/y\rangle\ [\Psi \mid x = 1]$. It follows that $\mathsf{Vin}_x(\dots) \in \mathsf{V}_x(A\langle r'/y\rangle, B\langle r'/y\rangle, E\langle r'/y\rangle)\ [\Psi]$.

When $x\psi = x'$, coherence is immediate. When $x\psi = 0$, $\mathsf{Vin}_0(\mathsf{coe}^{r\psi\rightsquigarrow r'\psi}_{y.A\psi}(M\psi), \dots) \doteq \mathsf{coe}^{r\psi\rightsquigarrow r'\psi}_{y.A\psi}(M\psi) \in A\psi\langle r'\psi/y\rangle\ [\Psi']$. When $x\psi = 1$, we have $\mathsf{Vin}_1(\dots) \doteq \mathsf{coe}^{r\psi\rightsquigarrow r'\psi}_{y.B\psi}(M\psi) \in B\psi\langle r'\psi/y\rangle\ [\Psi']$. Part (1) follows by Lemma 4.18 and a symmetric argument on the right.

For part (2),

$$\begin{aligned}
\mathsf{coe}^{r\rightsquigarrow r}_{y.\mathsf{V}_x(A,B,E)}(M) &\doteq \mathsf{Vin}_x(\mathsf{coe}^{r\rightsquigarrow r}_{y.A}(M), \mathsf{com}^{r\rightsquigarrow r}_{y.B}(\mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)); \overrightarrow{T}))\\
&\doteq \mathsf{Vin}_x(M, \mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)))\\
&\doteq M \in (\mathsf{V}_x(A,B,E))\langle r/y\rangle\ [\Psi]. \qquad \square
\end{aligned}$$

Lemma 4.85. *If*

1. $A \doteq A'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x \mid x = 0]$,
2. $B \doteq B'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x]$,
3. $E \doteq E' \in \mathsf{Equiv}(A, B)\ [\Psi, x \mid x = 0]$, *and*
4. $M \doteq M' \in (\mathsf{V}_x(A,B,E))\langle r/x\rangle\ [\Psi]$,

then

1. $\mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{V}_x(A,B,E)}(M) \doteq \mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{V}_x(A',B',E')}(M') \in (\mathsf{V}_x(A,B,E))\langle r'/x\rangle\ [\Psi]$ *and*
2. $\mathsf{coe}^{r\rightsquigarrow r}_{x.\mathsf{V}_x(A,B,E)}(M) \doteq M \in (\mathsf{V}_x(A,B,E))\langle r/x\rangle\ [\Psi]$.

Proof. For part (1), by Lemma 4.32 on both sides, it suffices to show (the binary form of) $\mathsf{Vin}_{r'}(\mathsf{fst}(O), P) \in (\mathsf{V}_x(A,B,E))\langle r'/x\rangle\ [\Psi]$, where

$$\begin{gathered}
N := \mathsf{coe}^{r\rightsquigarrow r'}_{x.B}(\mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle)))\\
F := \mathsf{Fiber}(A\langle r'/x\rangle, B\langle r'/x\rangle, \mathsf{fst}(E\langle r'/x\rangle), N)\\
C := \mathsf{snd}(E\langle r'/x\rangle)\ N\\
O := \mathsf{hcom}^{1\rightsquigarrow 0}_{F}(\mathsf{fst}(C); r = 0 \hookrightarrow z.(\mathsf{snd}(C)\ \langle M, \langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M)\rangle)@z, r = 1 \hookrightarrow \_.\mathsf{fst}(C))\\
P := \mathsf{hcom}^{1\rightsquigarrow 0}_{B\langle r'/x\rangle}(N; r' = 0 \hookrightarrow z.\mathsf{snd}(O)@z, r' = 1 \hookrightarrow \_.N, T)\\
T := r = r' \hookrightarrow \_.\mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle))
\end{gathered}$$

1. $N \in B\langle r'/x\rangle\ [\Psi]$ by $\mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle)) \in B\langle r/x\rangle\ [\Psi]$.

2. F $\mathsf{type_{Kan}}$ $[\Psi]$.

3. $C \in \mathsf{isContr}(F)$ $[\Psi \mid r' = 0]$ by $\mathsf{snd}(E) \in (b{:}B) \to \mathsf{isContr}(\mathsf{Fiber}(A, B, \mathsf{fst}(E), b))$ $[\Psi, x \mid x = 0]$.

4. $O \in F$ $[\Psi \mid r' = 0]$ because

   a) $(\mathsf{snd}(C)\ \langle M, \langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M)\rangle)@z \in F$ $[\Psi, z \mid r' = 0, r = 0]$ because we have $\mathsf{snd}(C)\ \langle M, \dots\rangle \in \mathsf{Path}_{\_.F}(\langle M, \langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M)\rangle, \mathsf{fst}(C))$ $[\Psi \mid r' = 0, r = 0]$, by

      i. $\mathsf{snd}(C) \in (c'{:}F) \to \mathsf{Path}_{\_.F}(c', \mathsf{fst}(C))$ $[\Psi \mid r' = 0]$,
      ii. $M \in A\langle 0/x\rangle$ $[\Psi \mid r = 0]$ by $(\mathsf{V}_x(A, B, E))\langle r/x\rangle \doteq A\langle 0/x\rangle$ $\mathsf{type_{Kan}}$ $[\Psi \mid r = 0]$,
      iii. $\mathsf{fst}(E\langle 0/x\rangle)\ M \in B\langle 0/x\rangle$ $[\Psi \mid r = 0]$,
      iv. $\langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M) \in \mathsf{Path}_{\_.B\langle 0/x\rangle}(\mathsf{fst}(E\langle 0/x\rangle)\ M, N)$ $[\Psi \mid r' = 0, r = 0]$ by $N \doteq \mathsf{fst}(E\langle 0/x\rangle)\ M \in B\langle 0/x\rangle$ $[\Psi \mid r' = 0, r = 0]$, and
      v. $F \doteq (a{:}A\langle 0/x\rangle) \times \mathsf{Path}_{\_.B\langle 0/x\rangle}(\mathsf{fst}(E\langle 0/x\rangle)\ a, N)$ $\mathsf{type_{Kan}}$ $[\Psi \mid r' = 0]$.

   b) $\mathsf{fst}(C) \in F$ $[\Psi \mid r' = 0]$, and

   c) $\mathsf{fst}(C) \doteq (\mathsf{snd}(C)\ \langle M, \dots\rangle)@1 \in F$ $[\Psi \mid r' = 0, r = 0]$ by the right endpoint specified above.

5. $P \in B\langle r'/x\rangle$ $[\Psi]$ because

   a) $\mathsf{snd}(O)@z \in B\langle r'/x\rangle$ $[\Psi, z \mid r' = 0]$,

   b) $N \in B\langle r'/x\rangle$ $[\Psi \mid r' = 1]$,

   c) $\mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle)) \in B\langle r'/x\rangle$ $[\Psi \mid r = r']$,

   d) $N \in B\langle r'/x\rangle$ $[\Psi]$,

   e) $N \doteq \mathsf{snd}(O)@1 \in B\langle r'/x\rangle$ $[\Psi \mid r' = 0]$ by the right endpoint specified in F,

   f) $N \doteq \mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle)) \in B\langle r'/x\rangle$ $[\Psi \mid r = r']$, and

   g) $\mathsf{snd}(O)@z \doteq \mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle)) \in B\langle r'/x\rangle$ $[\Psi, z \mid r' = 0, r = r']$ because $O \doteq (\mathsf{snd}(C)\ \langle M, \dots\rangle)@0 \doteq \langle M, \langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M)\rangle$ (by $r = 0$ and the specified left endpoint) and thus $\mathsf{snd}(O)@z \doteq \mathsf{fst}(E\langle 0/x\rangle)\ M$.

Then $\mathsf{Vin}_{r'}(\mathsf{fst}(O), P) \in (\mathsf{V}_x(A, B, E))\langle r'/x\rangle$ $[\Psi]$ because $\mathsf{fst}(O) \in A\langle r'/x\rangle$ $[\Psi \mid r' = 0]$, $P \in B\langle r'/x\rangle$ $[\Psi]$, and $\mathsf{fst}(E\langle r'/x\rangle)\ \mathsf{fst}(O) \doteq P \in B\langle r'/x\rangle$ $[\Psi \mid r' = 0]$ (by $P \doteq \mathsf{snd}(O)@0 \in B\langle r'/x\rangle$ $[\Psi \mid r' = 0]$ and the specified left endpoint), completing part (1).

Part (2) follows from Rule 4.82, $P \doteq \mathsf{Vproj}_r(M, \mathsf{fst}(E\langle r/x\rangle)) \in B\langle r'/x\rangle$ $[\Psi \mid r = r']$, and $\mathsf{fst}(O) \doteq M \in A\langle r'/x\rangle$ $[\Psi \mid r = r', r' = 0]$ (by $O \doteq (\mathsf{snd}(C)\ \langle M, \dots\rangle)@0 \doteq \langle M, \dots\rangle$). □

In the proof of Lemma 4.85, we include the $r = 1$ face of O and the $r' = 1$ face of P only to ensure all compositions are valid; one could instead use ghcom (Theorem 4.34).

Rule 4.86 (Kan type formation).

1. *If* A $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$ *then* $\mathsf{V}_0(A,B,E) \doteq A$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$.
2. *If* B $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$ *then* $\mathsf{V}_1(A,B,E) \doteq B$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$.
3. *If* $A \doteq A'$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi \mid r = 0]$, $B \doteq B'$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$, *and* $E \doteq E' \in \mathsf{Equiv}(A,B)$ $[\Psi \mid r = 0]$, *then* $\mathsf{V}_r(A,B,E) \doteq \mathsf{V}_r(A',B',E')$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$.

Proof. Parts (1–2) follow from Lemma 4.32. Part (3) requires homogeneous composition and coercion; for homogeneous composition, suppose we are given $\psi : \Psi' \to \Psi$ and a valid composition scenario in $\mathsf{V}_{r\psi}(A\psi, B\psi, E\psi)$. If $r\psi = 0$ (resp., 1) then the composition is in $A\psi$ (resp., $B\psi$) and homogeneous composition follows from $A\psi \doteq A'\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi']$ (resp., $B\psi \doteq B'\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi']$). Otherwise, $r\psi = x$ and homogeneous composition exists by Lemma 4.83 at $A\psi \doteq A'\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi' \mid x = 0]$, $B\psi \doteq B'\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi']$, and $E\psi \doteq E'\psi \in \mathsf{Equiv}(A\psi, B\psi)$ $[\Psi' \mid x = 0]$.

For coercion, suppose $\psi : (\Psi', x) \to \Psi$ and $M \doteq M' \in (\mathsf{V}_r(A,B,E))\psi\langle s/x\rangle$ $[\Psi']$. If $r\psi = 0$ (resp., 1), coercion follows from $A\psi \doteq A'\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi', x]$ (resp., $B\psi \doteq B'\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi', x]$). If $r\psi = x$, coercion follows from Lemma 4.85; otherwise, it follows from Lemma 4.84. □

4.4.10 *Compositions of pretypes*

As discussed in Section 3.4, in order for the pretype universe (resp., Kan type universe) to be Kan, we must equip homogeneous compositions of pretypes (resp., Kan types) with the structure of a pretype (resp., Kan type). Let us say that $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ are *(equal) pretype compositions* $r \rightsquigarrow r'$ whenever $\overrightarrow{\xi_i}$ is valid,

1. $A \doteq A'$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi]$,
2. $B_i \doteq B'_j$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi, y \mid \xi_i, \xi_j]$ for any i, j, and
3. $B_i\langle r/y\rangle \doteq A$ $\mathsf{type}_{\mathsf{pre}}$ $[\Psi \mid \xi_i]$ for any i.

For $i \in \{0, 1, \dots, \omega\}$, $\tau_i^{\mathsf{pre}}(\Psi, \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}), \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A'; \overrightarrow{\xi_i \hookrightarrow y.B'_i}), \{\})$ when $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ are equal pretype compositions $r \rightsquigarrow r'$ relative to τ_i^{pre} with $r \neq r'$ and $\neg\xi_i$ for all i. Then, relative to τ_i^{pre}:

Rule 4.87 (Pretype formation). *If* $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ *and* $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ *are equal pretype compositions* $r \rightsquigarrow r'$, *then*

1. $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A;\overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A';\overrightarrow{\xi_i \hookrightarrow y.B'_i})\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$,

2. *if* $r = r'$ *then* $\mathsf{hcom}^{r\rightsquigarrow r}_{\mathcal{U}^{\mathsf{pre}}_k}(A;\overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, *and*

3. *if* ξ_i *then* $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{pre}}_k}(A;\overrightarrow{\xi_i \hookrightarrow B_i}) \doteq B_i\langle r'/y\rangle\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. We abbreviate these $\mathsf{hcom}_{\mathcal{U}^{\mathsf{pre}}_k}$ terms L and R. Part (2) holds by Lemma 4.32. For part (3), if $r = r'$, the result holds by Lemma 4.32 and $B_i\langle r/y\rangle \doteq A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi \mid \xi_i]$. Otherwise, there is a least j such that ξ_j holds. Apply coherent expansion to L with family

$$\begin{cases} A\psi & r\psi = r'\psi \\ B_j\langle r'/y\rangle\psi & r\psi \neq r'\psi,\ \xi_j\psi, \text{ and } \forall l < j.\neg\xi_l\psi. \end{cases}$$

If $r\psi = r'\psi$ then $B_j\langle r/y\rangle\psi \doteq A\psi\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$. If $r\psi \neq r'\psi$, there is some least l such that $\xi_l\psi$; then $B_j\langle r'/y\rangle\psi \doteq B_l\langle r'/y\rangle\psi\ \mathsf{type}_{\mathsf{pre}}\ [\Psi']$. By Lemma 4.17, $L \doteq B_j\langle r'/y\rangle\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, and we complete part (3) with $B_i\langle r'/y\rangle \doteq B_j\langle r'/y\rangle\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

For part (1), we conclude $L \sim R \downarrow [\![L]\!] \in \tau^{\mathsf{pre}}_i\ [\Psi]$ by the same argument as in Lemma 4.69, because the definition and operational semantics of $\mathsf{hcom}_{\mathbb{S}^1}$ and $\mathsf{hcom}_{\mathcal{U}^{\mathsf{pre}}_k}$ are identical. To see $\mathsf{Coh}([\![L]\!])$, suppose $[\![L]\!]_\psi(M_0, N_0)$ for any $\psi : \Psi' \to \Psi$. If $r\psi = r'\psi$, coherence follows by part (2); if $\xi_i\psi$, coherence follows by part (3); otherwise, $[\![L]\!]_\psi$ is the empty relation, which contradicts $[\![L]\!]_\psi(M_0, N_0)$. □

4.4.11 *Compositions of Kan types*

For $i \in \{0, 1, \ldots, \omega\}$ and $\kappa \in \{\mathsf{pre}, \mathsf{Kan}\}$,

$$\tau^\kappa_i(\Psi, \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{\xi_i \hookrightarrow y.B_i}), \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A';\overrightarrow{\xi_i \hookrightarrow y.B'_i}), \varphi)$$

when $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ are equal pretype compositions $r \rightsquigarrow r'$ relative to τ^{Kan}_i with $r \neq r'$ and $\neg\xi_i$ for all i, and $\varphi(\mathsf{box}^{r\rightsquigarrow r'}(M;\overrightarrow{\xi_i \hookrightarrow N_i}), \mathsf{box}^{r\rightsquigarrow r'}(M';\overrightarrow{\xi_i \hookrightarrow N'_i}))$ whenever

1. $\tau^{\mathsf{Kan}}_i \models (M \doteq M' \in A\ [\Psi])$,
2. $\tau^{\mathsf{Kan}}_i \models (N_i \doteq N'_j \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i, \xi_j])$ for all i, j, and
3. $\tau^{\mathsf{Kan}}_i \models (M \doteq \mathsf{coe}^{r'\rightsquigarrow r}_{y.B_i}(N_i) \in A\ [\Psi \mid \xi_i])$ for all i.

Note that τ^{pre}_i is closed under $\mathsf{hcom}_{\mathcal{U}^{\mathsf{Kan}}_k}(-)$ of types in τ^{Kan}_i, because $[\![\mathsf{hcom}_{\mathcal{U}^{\mathsf{Kan}}_k}(-)]\!]$ is only value-coherent for Kan types, and we require $\tau^{\mathsf{Kan}}_i \subseteq \tau^{\mathsf{pre}}_i$. In this section, we therefore prove rules relative only to Kan types in τ_i^{Kan}; in Section 4.4.12 we will show that all pretypes in τ_i^{Kan} are Kan. Let us say that $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ are *(equal) Kan type compositions $r \rightsquigarrow r'$* whenever $\overrightarrow{\xi_i}$ is valid,

1. $A \doteq A'$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$,
2. $B_i \doteq B'_j$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi, y \mid \xi_i, \xi_j]$ for any i, j, and
3. $B_i\langle r/y\rangle \doteq A$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi \mid \xi_i]$ for any i.

Lemma 4.88. *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ are equal Kan type compositions $r \rightsquigarrow r'$, then*

1. $\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \sim \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A'; \overrightarrow{\xi_i \hookrightarrow y.B'_i}) \downarrow \_ \in \tau_i^{\mathsf{Kan}}$ $[\Psi]$,
2. *if $r = r'$ then* $\mathsf{hcom}^{r \rightsquigarrow r}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq A$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$, *and*
3. *if ξ_i then* $\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow B_i}) \doteq B_i\langle r'/y\rangle$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$.

Proof. As in Rule 4.87, part (1) follows from the argument of Lemma 4.69. Part (2) holds by Lemma 4.32. For part (3), if $r = r'$, the result follows by Lemma 4.32 and $B_i\langle r/y\rangle \doteq A$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi \mid \xi_i]$. Otherwise, there is a least j such that ξ_j holds. Apply coherent expansion to the left with family

$$\begin{cases} A\psi & r\psi = r'\psi \\ B_j\langle r'/y\rangle\psi & r\psi \neq r'\psi,\ \xi_j\psi, \text{ and } \forall l < j.\neg\xi_l\psi. \end{cases}$$

If $r\psi = r'\psi$ then $B_j\langle r/y\rangle\psi \doteq A\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi']$. If $r\psi \neq r'\psi$, there is some least $l \leq j$ such that $\xi_l\psi$; then $B_j\langle r'/y\rangle\psi \doteq B_l\langle r'/y\rangle\psi$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi']$. By Lemma 4.31, $\mathsf{hcom}_{\mathcal{U}_k^{\mathsf{Kan}}} \doteq B_j\langle r'/y\rangle$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$, and part (3) follows by $B_i\langle r'/y\rangle \doteq B_j\langle r'/y\rangle$ $\mathsf{type}_{\mathsf{Kan}}$ $[\Psi]$. □

Lemma 4.89. *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ is a Kan type composition $r \rightsquigarrow r'$,*

1. $M \doteq M' \in A$ $[\Psi]$,
2. $N_i \doteq N'_j \in B_i\langle r'/y\rangle$ $[\Psi \mid \xi_i, \xi_j]$ *for any i, j, and*
3. $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i) \doteq M \in A$ $[\Psi \mid \xi_i]$ *for any i,*

then $\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \sim \mathsf{box}^{r \rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow N'_i}) \in [\![\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})]\!]$ $[\Psi]$.

Proof. We focus on the unary case; the binary case follows similarly. For any $\psi_1 : \Psi_1 \longrightarrow \Psi$ and $\psi_2 : \Psi_2 \longrightarrow \Psi_1$ we must show $\mathsf{box}\psi_1 \Downarrow X_1$ and $[\![\mathsf{hcom}]\!]^{\Downarrow}_{\psi_1\psi_2}(X_1\psi_2, \mathsf{box}\psi_1\psi_2)$. We proceed by cases on the first step taken by $\mathsf{box}\psi_1$ and $\mathsf{box}\psi_1\psi_2$.

1. $r\psi_1 = r'\psi_1$.

   Then $\mathsf{box}\psi_1 \longmapsto_{\square} M\psi_1$, $\llbracket \mathsf{hcom} \rrbracket_{\psi_1\psi_2} = \llbracket A \rrbracket_{\psi_1\psi_2}$ by Lemma 4.88, and by $M \in A\ [\Psi]$, $\llbracket A \rrbracket^{\Downarrow}_{\psi_1\psi_2}(X_1\psi_2, M\psi_1\psi_2)$.

2. $r\psi_1 \neq r'\psi_1$, $\xi_j\psi_1$ holds (where this is the least such j), and $r\psi_1\psi_2 = r'\psi_1\psi_2$.

   Then $\mathsf{box}\psi_1 \longmapsto N_j\psi_1$, $\mathsf{box}\psi_1\psi_2 \longmapsto M\psi_1\psi_2$, and again $\llbracket \mathsf{hcom} \rrbracket_{\psi_1\psi_2} = \llbracket A \rrbracket_{\psi_1\psi_2}$ by Lemma 4.88. By $B_j\langle r'/y\rangle\psi_1\psi_2 \doteq A\psi_1\psi_2\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi_2]$ and $N_j\psi_1 \in B_j\langle r'/y\rangle\psi_1\ [\Psi_1]$ at $\mathsf{id}_{\Psi_1}, \psi_2$ we have $\llbracket A \rrbracket^{\Downarrow}_{\psi_1\psi_2}(X_1\psi_2, N_j\psi_1\psi_2)$. We also have $\llbracket A \rrbracket^{\Downarrow}_{\psi_1\psi_2}(N_j\psi_1\psi_2, M\psi_1\psi_2)$ by $(\mathsf{coe}^{r'\rightsquigarrow r}_{y.B_j}(N_j))\psi_1\psi_2 \doteq M\psi_1\psi_2 \in A\psi_1\psi_2\ [\Psi_2]$ and $(\mathsf{coe}^{r'\rightsquigarrow r}_{y.B_j}(N_j))\psi_1\psi_2 \doteq N_j\psi_1\psi_2 \in A\psi_1\psi_2\ [\Psi_2]$; the result follows by transitivity.

3. $r\psi_1 \neq r'\psi_1$, $\xi_i\psi_1$ holds (least such), $r\psi_1\psi_2 \neq r'\psi_1\psi_2$, and $\xi_j\psi_1\psi_2$ holds (least such).

   Then $\mathsf{box}\psi_1 \longmapsto N_i\psi_1$, $\mathsf{box}\psi_1\psi_2 \longmapsto N_j\psi_1\psi_2$, and $\llbracket \mathsf{hcom} \rrbracket_{\psi_1\psi_2} = \llbracket B_i\langle r'/y\rangle \rrbracket_{\psi_1\psi_2}$ by Lemma 4.88. The result follows by $N_i\psi_1 \in B_i\langle r'/y\rangle\psi_1\ [\Psi_1]$ and $N_i\psi_1\psi_2 \doteq N_j\psi_1\psi_2 \in B_i\langle r'/y\rangle\psi_1\psi_2\ [\Psi_2]$.

4. $r\psi_1 \neq r'\psi_1$, $\neg\xi_i\psi_1$ for all i, and $r\psi_1\psi_2 = r'\psi_1\psi_2$.

   Then $\mathsf{box}\psi_1\ \mathsf{val}$, $\mathsf{box}\psi_1\psi_2 \longmapsto M\psi_1\psi_2$, $\llbracket \mathsf{hcom} \rrbracket_{\psi_1\psi_2} = \llbracket A \rrbracket_{\psi_1\psi_2}$ by Lemma 4.88, and the result follows by $M \in A\ [\Psi]$.

5. $r\psi_1 \neq r'\psi_1$, $\neg\xi_i\psi_1$ for all i, $r\psi_1\psi_2 \neq r'\psi_1\psi_2$, and $\xi_j\psi_1\psi_2$ holds (least such).

   Then $\mathsf{box}\psi_1\ \mathsf{val}$, $\mathsf{box}\psi_1\psi_2 \longmapsto N_j\psi_1\psi_2$, $\llbracket \mathsf{hcom} \rrbracket_{\psi_1\psi_2} = \llbracket B_i\langle r'/y\rangle \rrbracket_{\psi_1\psi_2}$ by Lemma 4.88, and the result follows by $N_j\psi_1\psi_2 \in B_i\langle r'/y\rangle\psi_1\psi_2\ [\Psi_2]$.

6. $r\psi_1 \neq r'\psi_1$, $\neg\xi_i\psi_1$ for all i, and $r\psi_1\psi_2 \neq r'\psi_1\psi_2$, and $\neg\xi_j\psi_1\psi_2$ for all j.

   Then $\mathsf{box}\psi_1\ \mathsf{val}$ and $\mathsf{box}\psi_1\psi_2\ \mathsf{val}$; the result follows by the definition of $\llbracket \mathsf{hcom} \rrbracket$. □

Rule 4.90 (Pretype formation). *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B'_i}$ are equal Kan type compositions $r \rightsquigarrow r'$, then*

1. $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A'; \overrightarrow{\xi_i \hookrightarrow y.B'_i})\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$,
2. *if $r = r'$ then* $\mathsf{hcom}^{r\rightsquigarrow r}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq A\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$, *and*
3. *if ξ_i then* $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow B_i}) \doteq B_i\langle r'/y\rangle\ \mathsf{type}_{\mathsf{pre}}\ [\Psi]$.

Proof. Parts (2–3) hold immediately by Lemma 4.88. For part (1), by Lemma 4.88 it suffices to show $\mathsf{Coh}(\llbracket \mathsf{hcom} \rrbracket)$. Let $\psi : \Psi' \to \Psi$ and $\llbracket \mathsf{hcom} \rrbracket_{\psi}(M_0, N_0)$. If $r\psi = r'\psi$ then $M_0 \sim N_0 \in \llbracket \mathsf{hcom} \rrbracket\psi\ [\Psi']$ by $\llbracket \mathsf{hcom} \rrbracket\psi = \llbracket A \rrbracket\psi$ and $\mathsf{Coh}(\llbracket A \rrbracket)$. If $\xi_i\psi$ holds for some i, then $M_0 \sim N_0 \in \llbracket \mathsf{hcom} \rrbracket\psi\ [\Psi']$ by $\llbracket \mathsf{hcom} \rrbracket\psi = \llbracket B_i\langle r'/y\rangle\psi \rrbracket$ and $\mathsf{Coh}(\llbracket B_i\langle r'/y\rangle\psi \rrbracket)$. If $r\psi \neq r'\psi$ and $\neg\xi_i\psi$ for all i, then M_0 and N_0 are boxes and the result holds by Lemma 4.89. □

Rule 4.91 (Introduction). *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ is a Kan type composition $r \rightsquigarrow r'$,*

1. $M \doteq M' \in A\ [\Psi]$,
2. $N_i \doteq N_j' \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i, \xi_j]$ *for any i, j, and*
3. $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i) \doteq M \in A\ [\Psi \mid \xi_i]$ *for any i,*

then

1. $\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \doteq \mathsf{box}^{r \rightsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow N_i'}) \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_l}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi]$;
2. *if $r = r'$ then* $\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \doteq M \in A\ [\Psi]$*; and*
3. *if ξ_i then* $\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \doteq N_i \in B_i\langle r'/y\rangle\ [\Psi]$.

Proof. Part (1) holds by Lemma 4.89 and Rule 4.90; part (2) holds by Lemma 4.32. For part (3), if $r = r'$, the result holds by Lemma 4.32. Otherwise, there is a least j such that ξ_j holds, and we apply coherent expansion to the left with family

$$\begin{cases} M\psi & r\psi = r'\psi \\ N_k\psi & r\psi \neq r'\psi,\ \xi_k\psi, \text{ and } \forall k' < k.\neg\xi_{k'}\psi. \end{cases}$$

If $r\psi = r'\psi$ then $M\psi \doteq N_j\psi \in B_i\langle r'/y\rangle\psi\ [\Psi']$ by $M\psi \doteq (\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(N_j))\psi \in A\psi\ [\Psi']$, $(\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(N_j))\psi \doteq N_j\psi \in B_i\langle r'/y\rangle\psi\ [\Psi']$, and $B_i\langle r'/y\rangle\psi \doteq A\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$. If $r\psi \neq r'\psi$ then $N_k\psi \doteq N_j\psi \in B_i\langle r'/y\rangle\psi\ [\Psi']$ by $N_k\psi \doteq N_j\psi \in B_j\langle r'/y\rangle\psi\ [\Psi']$ and $B_i\psi \doteq B_j\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', y]$. Thus by Lemma 4.31 we have $\mathsf{hcom} \doteq N_j \in B_i\langle r'/y\rangle\ [\Psi]$, and part (3) follows by $N_j \doteq N_i \in B_i\langle r'/y\rangle\ [\Psi]$. □

Rule 4.92 (Elimination). *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A, \overrightarrow{\xi_i \hookrightarrow y.B_i'}$ are equal Kan type compositions $r \rightsquigarrow r'$ and $M \doteq M' \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_l}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi]$, then*

1. $\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq \mathsf{cap}^{r \leftsquigarrow r'}(M'; \overrightarrow{\xi_i \hookrightarrow y.B_i'}) \in A\ [\Psi]$;
2. *if $r = r'$ then* $\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq M \in A\ [\Psi]$*; and*
3. *if ξ_i then* $\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(M) \in A\ [\Psi]$.

Proof. Part (2) holds by Lemma 4.32 and Rule 4.90. For part (3), if $r = r'$ then the result holds by part (2), $B_i\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, y]$, and $B_i\langle r/y\rangle \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$. Otherwise, $r \neq r'$ and there is a least j such that ξ_j holds. Apply coherent expansion to the left with family

$$\begin{cases} M\psi & r\psi = r'\psi \\ \mathsf{coe}^{r'\psi \rightsquigarrow r\psi}_{y.B_k\psi}(M\psi) & r\psi \neq r'\psi,\ \xi_k\psi, \text{ and } \forall i < k.\neg\xi_i\psi \end{cases}$$

When $r\psi = r'\psi$, $(\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(M))\psi \doteq M\psi \in A\psi\ [\Psi']$ by $M \in B_j\langle r'/y\rangle\ [\Psi]$ (by Rule 4.90), $B_j\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, y]$, and $B_j\langle r/y\rangle\psi \doteq A\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$. When $r\psi \neq r'\psi$ and $\xi_k\psi$ where k is the least such, we have $(\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(M))\psi \doteq \mathsf{coe}^{r'\psi \rightsquigarrow r\psi}_{y.B_k\psi}(M\psi) \in A\psi\ [\Psi']$ by $B_j\psi \doteq B_k\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi', y]$ and $B_j\langle r/y\rangle\psi \doteq A\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$. Thus $\mathsf{cap} \doteq \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(M) \in A\ [\Psi]$ by Lemma 4.18, and part (3) follows by $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(M) \doteq \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(M) \in A\ [\Psi]$.

For part (1), if $r = r'$ or ξ_i then the result follows by parts (2–3). If $r \neq r'$ and $\neg\xi_i$ for all i, then for any $\psi : \Psi' \to \Psi$, $M\psi \doteq \mathsf{box}^{r \rightsquigarrow r'}(O_\psi; \overrightarrow{\xi_i\psi \hookrightarrow N_{i,\psi}}) \in \mathsf{hcom}\psi\ [\Psi']$ by Lemma 4.16. Apply coherent expansion to the left with family

$$\begin{cases} M\psi & r\psi = r'\psi \\ \mathsf{coe}^{r'\psi \rightsquigarrow r\psi}_{y.B_j\psi}(M\psi) & r\psi \neq r'\psi,\ \xi_j\psi, \text{ and } \forall i < j.\neg\xi_i\psi \\ O_\psi & r\psi \neq r'\psi \text{ and } \forall i.\neg\xi_i\psi \\ & \quad \text{where } M\psi \Downarrow \mathsf{box}^{r\psi \rightsquigarrow r'\psi}(O_\psi; \overrightarrow{\xi_i\psi \hookrightarrow N_{i,\psi}}). \end{cases}$$

When $r\psi = r'\psi$, $M\psi \doteq (O_{\mathsf{id}_\Psi})\psi \in A\psi\ [\Psi']$ by $M\psi \doteq \mathsf{box}\psi \in \mathsf{hcom}\psi\ [\Psi']$, $\mathsf{hcom}\psi \doteq A\psi\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi']$ (by Rule 4.90), and $\mathsf{box}\psi \doteq (O_{\mathsf{id}_\Psi})\psi \in \mathsf{hcom}\psi\ [\Psi']$ (by Rule 4.91). When $r\psi \neq r'\psi$ and $\xi_j\psi$ where j is the least such, $(O_{\mathsf{id}_\Psi})\psi \doteq \mathsf{coe}^{r'\psi \rightsquigarrow r\psi}_{y.B_j\psi}(M\psi) \in A\psi\ [\Psi']$ by $M\psi \doteq (N_{j,\mathsf{id}_\Psi})\psi \in B_j\langle r'/y\rangle\psi\ [\Psi']$ (by Rules 4.90 and 4.91) and $O_{\mathsf{id}_\Psi} \doteq \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_j}(N_{j,\mathsf{id}_\Psi}) \in A\ [\Psi \mid \xi_j]$. When $r\psi \neq r'\psi$ and $\neg\xi_i\psi$ for all i, $(O_{\mathsf{id}_\Psi})\psi \doteq O_\psi \in A\psi\ [\Psi]$ by

$$[\![\mathsf{hcom}]\!]((\mathsf{box}^{r \rightsquigarrow r'}(O_{\mathsf{id}_\Psi}; \overrightarrow{\xi_i \hookrightarrow N_{i,\mathsf{id}_\Psi}}))\psi, \mathsf{box}^{r\psi \rightsquigarrow r'\psi}(O_\psi; \overrightarrow{\xi_i\psi \hookrightarrow N_{i,\psi}}))$$

(by $M \in \mathsf{hcom}\ [\Psi]$ at id_Ψ, ψ). Therefore $\mathsf{cap} \doteq O_{\mathsf{id}_\Psi} \in A\ [\Psi]$ by Lemma 4.18, and part (1) follows by a symmetric argument on the right. □

Rule 4.93 (Computation). *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ is a Kan type composition $r \rightsquigarrow r'$,*

1. $M \doteq M' \in A\ [\Psi]$,
2. $N_i \doteq N'_j \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i, \xi_j]$ *for any i, j, and*
3. $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i) \doteq M \in A\ [\Psi \mid \xi_i]$ *for any i,*

then $\mathsf{cap}^{r \leftsquigarrow r'}(\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}); \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq M \in A\ [\Psi]$.

Proof. Both terms are elements of A by Rules 4.91 and 4.93, so by Lemma 4.15 it suffices to show $[\![A]\!]^{\Downarrow}(\mathsf{cap}, M)$. If $r = r'$ then $\mathsf{cap} \longmapsto \mathsf{box} \longmapsto M$ and $[\![A]\!]^{\Downarrow}(M, M)$. If $r \neq r'$ and ξ_i holds where i is the least such, then $\mathsf{cap} \longmapsto \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(\mathsf{box})$, and $[\![A]\!]^{\Downarrow}(\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(\mathsf{box}), M)$ by $\mathsf{box} \doteq N_i \in B_i\langle r'/y\rangle\ [\Psi]$ and $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i) \doteq M \in A\ [\Psi]$. If $r \neq r'$ and $\neg\xi_i$ for all i, then $\mathsf{cap} \longmapsto M$ and $[\![A]\!]^{\Downarrow}(M, M)$. □

Rule 4.94 (Uniqueness). *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ is a Kan type composition $r \rightsquigarrow r'$ and $M \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi]$, then*

$$\mathsf{box}^{r \rightsquigarrow r'}(\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}); \overrightarrow{\xi_i \hookrightarrow M}) \doteq M \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi].$$

Proof. By $\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \in A\ [\Psi]$ (by Rule 4.92), $M \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i]$ (by Rule 4.90), $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(M) \doteq \mathsf{cap} \in A\ [\Psi \mid \xi_i]$ (by Rule 4.92), and Rule 4.91, $\mathsf{box} \in \mathsf{hcom}\ [\Psi]$. Thus, by Lemma 4.15 it suffices to show $[\![\mathsf{hcom}]\!]^{\Downarrow}(\mathsf{box}, M)$. If $r = r'$ then $\mathsf{box} \longmapsto \mathsf{cap} \longmapsto M$ and $[\![\mathsf{hcom}]\!]^{\Downarrow}(M, M)$. If $r \neq r'$ and ξ_i, then $\mathsf{box} \longmapsto M$ and $[\![\mathsf{hcom}]\!]^{\Downarrow}(M, M)$. If $r \neq r'$ and $\neg\xi_i$ for all i, then $M \Downarrow \mathsf{box}^{r \rightsquigarrow r'}(O; \overrightarrow{\xi_i \hookrightarrow N_i})$ and $M \doteq \mathsf{box}^{r \rightsquigarrow r'}(O; \overrightarrow{\xi_i \hookrightarrow N_i}) \in \mathsf{hcom}\ [\Psi]$. The result follows by transitivity and Rule 4.91:

1. $\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq O \in A\ [\Psi]$ by Lemma 4.15 and $\mathsf{cap} \longmapsto^* O$,
2. $M \doteq N_i \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i]$ by $M \doteq \mathsf{box}^{r \rightsquigarrow r'}(O; \overrightarrow{\xi_i \hookrightarrow N_i}) \in \mathsf{hcom}\ [\Psi]$ and Rule 4.91, and
3. $\mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(M) \doteq \mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \in A\ [\Psi \mid \xi_i]$ by Rule 4.92 as before. □

Lemma 4.95. *If $A, \overrightarrow{s_j = s'_j \hookrightarrow z.B_j}$ and $A', \overrightarrow{s_j = s'_j \hookrightarrow z.B'_j}$ are equal Kan type compositions $s \rightsquigarrow s'$ and, writing $H := \mathsf{hcom}^{s \rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{s_j = s'_j \hookrightarrow z.B_j})$, if $\overrightarrow{\xi_i}$ is valid,*

1. *$M \doteq M' \in H\ [\Psi]$,*
2. *$N_i \doteq N'_{i'} \in H\ [\Psi, y \mid \xi_i, \xi_{i'}]$ for any i, i', and*
3. *$N_i\langle r/y\rangle \doteq M \in H\ [\Psi \mid \xi_i]$ for any i,*

then

1. *$\mathsf{hcom}^{r \rightsquigarrow r'}_{H}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{hcom}^{s \rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A'; \overrightarrow{s_j = s'_j \hookrightarrow z.B'_j})}(M'; \overrightarrow{\xi_i \hookrightarrow y.N'_i}) \in H\ [\Psi]$;*
2. *if $r = r'$ then $\mathsf{hcom}^{r \rightsquigarrow r}_{H}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq M \in H\ [\Psi]$; and*
3. *if ξ_i then $\mathsf{hcom}^{r \rightsquigarrow r'}_{H}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq N_i\langle r'/y\rangle \in H\ [\Psi]$.*

Proof. If $s = s'$ or $s_j = s'_j$ for some j, the results are immediate by parts (2–3) of Lemma 4.88. Otherwise, $s \neq s'$ and $s_j \neq s'_j$ for all j; apply coherent expansion to hcom_H with the family defined in Figure 4.4. Consider $\psi = \mathsf{id}_\Psi$.

$$\begin{cases}
\mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{A\psi}(M\psi;\overrightarrow{\xi_i\psi \hookrightarrow y.N_i\psi}) & s\psi = s'\psi \\
\mathsf{hcom}^{r\psi\rightsquigarrow r'\psi}_{B_j\langle s'/z\rangle\psi}(M\psi;\overrightarrow{\xi_i\psi \hookrightarrow y.N_i\psi}) & s\psi \neq s'\psi,\ \text{least } s_j\psi = s'_j\psi \\
(\mathsf{box}^{s\rightsquigarrow s'}(Q;\overrightarrow{s_j = s'_j \hookrightarrow P_j\langle r'/y\rangle}))\psi & s\psi \neq s'\psi,\ \forall j.s_j\psi \neq s'_j\psi \\
\quad O_i := \mathsf{cap}^{s\leftsquigarrow s'}(N_i;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j}) & \\
\quad P_j := \mathsf{hcom}^{r\rightsquigarrow y}_{B_j\langle s'/z\rangle}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) & \\
\quad Q := \mathsf{hcom}^{r\rightsquigarrow r'}_{A}(\mathsf{cap}^{s\leftsquigarrow s'}(M;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j});\overrightarrow{T}) & \\
\quad \overrightarrow{T} := \overrightarrow{\xi_i \hookrightarrow y.O_i}, \overrightarrow{s_j = s'_j \hookrightarrow y.\mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(P_j)}, & \\
\qquad\quad s = s' \hookrightarrow y.\mathsf{hcom}^{r\rightsquigarrow y}_{A}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) &
\end{cases}$$

Figure 4.4: Family of reducts in Lemma 4.95.

1. $O_i \doteq O_{i'} \in A\ [\Psi, y \mid \xi_i, \xi_{i'}]$ for all i, i', by $N_i \doteq N_{i'} \in H\ [\Psi, y \mid \xi_i, \xi_{i'}]$.

2. $P_j \doteq P_{j'} \in B_j\langle s'/z\rangle\ [\Psi, y \mid s_j = s'_j, s_{j'} = s'_{j'}]$ for all j, j', by $H \doteq B_j\langle s'/z\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid s_j = s'_j]$ and $B_j\langle s'/z\rangle \doteq B_{j'}\langle s'/z\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid s_j = s'_j, s_{j'} = s'_{j'}]$.

3. $Q \in A\ [\Psi]$ by

   a) $O_i \doteq O_{i'} \in A\ [\Psi, y \mid \xi_i, \xi_{i'}]$ for all i, i',

   b) $\mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(P_j) \doteq \mathsf{coe}^{s'\rightsquigarrow s}_{z.B_{j'}}(P_{j'}) \in A\ [\Psi, y \mid s_j = s'_j, s_{j'} = s'_{j'}]$ for all j, j', by $P_j \doteq P_{j'} \in B_j\langle s'/z\rangle\ [\Psi, y \mid s_j = s'_j, s_{j'} = s'_{j'}]$, $B_j \doteq B_{j'}\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, z \mid s_j = s'_j, s_{j'} = s'_{j'}]$, and $B_j\langle s/z\rangle \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid s_j = s'_j]$,

   c) $\mathsf{hcom}^{r\rightsquigarrow y}_{A}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\ [\Psi, y \mid s = s']$ by $H \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid s = s']$,

   d) $\mathsf{cap}^{s\leftsquigarrow s'}(M;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j}) \in A\ [\Psi]$ by $M \in H\ [\Psi]$,

   e) $O_i\langle r/y\rangle \doteq \mathsf{cap}^{s\leftsquigarrow s'}(M;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j}) \in A\ [\Psi \mid \xi_i]$ for all i, by $N_i\langle r/y\rangle \doteq M \in H\ [\Psi \mid \xi_i]$,

   f) $\mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(P_j\langle r/y\rangle) \doteq \mathsf{cap}^{s\leftsquigarrow s'}(M;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j}) \in A\ [\Psi \mid s_j = s'_j]$ for all j, by $P_j\langle r/y\rangle \doteq M \in A\ [\Psi \mid s_j = s'_j]$ and $\mathsf{cap}^{s\leftsquigarrow s'}(M;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j}) \doteq \mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(M) \in A\ [\Psi \mid s_j = s'_j]$,

   g) $\mathsf{hcom}^{r\rightsquigarrow r}_{A}(M;\overrightarrow{\xi_i \hookrightarrow y.N_i}) \doteq \mathsf{cap}^{s\leftsquigarrow s'}(M;\overrightarrow{s_j = s'_j \hookrightarrow z.B_j}) \in A\ [\Psi \mid s = s']$ because both $\doteq M$,

   h) $O_i \doteq \mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(P_j) \in A\ [\Psi, y \mid \xi_i, s_j = s'_j]$ for all i, j because both $\doteq \mathsf{coe}^{s'\rightsquigarrow s}_{z.B_j}(N_i)$,

i) $O_i \doteq \mathsf{hcom}_A^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\ [\Psi, y \mid \xi_i, s = s']$ for all i because both $\doteq N_i$, and

j) $\mathsf{coe}_{z.B_j}^{s' \rightsquigarrow s}(P_j) \doteq \mathsf{hcom}_A^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\ [\Psi, y \mid s_j = s_j', s = s']$ for all j, by $\mathsf{coe}_{z.B_j}^{s' \rightsquigarrow s}(P_j) \doteq \mathsf{hcom}_{B_j\langle s'/z\rangle}^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\ [\Psi, y \mid s_j = s_j', s = s']$ and $B_j\langle s'/z\rangle \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid s_j = s_j', s = s']$.

4. $\mathsf{box}^{s \rightsquigarrow s'}(Q; \overrightarrow{s_j = s_j' \hookrightarrow P_j\langle r'/y\rangle}) \in H\ [\Psi]$ because $Q \in A\ [\Psi]$, $P_j\langle r'/y\rangle \doteq P_{j'}\langle r'/y\rangle \in B_j\langle s'/z\rangle\ [\Psi \mid s_j = s_j', s_{j'} = s_{j'}']$ for all j, j', and $\mathsf{coe}_{z.B_j}^{s' \rightsquigarrow s}(P_j\langle r'/y\rangle) \doteq Q \in A\ [\Psi \mid s_j = s_j']$ for all j.

When $s\psi \neq s'\psi$ and $s_j\psi \neq s_j'\psi$ for all j, coherence is immediate. When $s\psi = s'\psi$, $\mathsf{box}\psi \doteq Q\psi \doteq (\mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}))\psi$. When $s\psi \neq s'\psi$ and $s_j\psi = s_j'\psi$ for the least such j, $\mathsf{box}\psi \doteq P_j\langle r'/y\rangle\psi = (\mathsf{hcom}_{B_j\langle s'/z\rangle}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}))\psi$. By Lemma 4.18, $\mathsf{hcom}_H \doteq \mathsf{box} \in H\ [\Psi]$; part (1) follows by a symmetric argument on the right.

For part (2), if $r = r'$ then $Q \doteq \mathsf{cap}^{s \leftsquigarrow s'}(M; \overrightarrow{s_j = s_j' \hookrightarrow z.B_j}) \in A\ [\Psi]$ and $P_j\langle r'/y\rangle \doteq M \in B_j\langle s'/z\rangle\ [\Psi \mid s_j = s_j']$ for all j, so $\mathsf{box}^{s \rightsquigarrow s'}(Q; \overrightarrow{s_j = s_j' \hookrightarrow P_j\langle r'/y\rangle}) \doteq M \in H\ [\Psi]$ by Rule 4.94.

For part (3), if ξ_i then $Q \doteq O_i\langle r'/y\rangle = \mathsf{cap}^{s \leftsquigarrow s'}(N_i\langle r'/y\rangle; \overrightarrow{s_j = s_j' \hookrightarrow z.B_j})$ and $P_j\langle r'/y\rangle \doteq N_i\langle r'/y\rangle$ for all j, so $\mathsf{box} \doteq N_i\langle r'/y\rangle \in H\ [\Psi]$ by Rule 4.94. □

Lemma 4.96. *If* $A, \overrightarrow{\xi_i \hookrightarrow z.B_i}$ *and* $A', \overrightarrow{\xi_i \hookrightarrow z.B_i'}$ *are equal Kan type compositions* $r \rightsquigarrow r'$ *in* (Ψ, x) *and, writing* $H := \mathsf{hcom}_{\mathcal{U}_l^{\mathsf{Kan}}}^{s \rightsquigarrow s'}(A; \overrightarrow{\xi_i \hookrightarrow z.B_i})$, *we have* $M \doteq M' \in H\langle r/x\rangle\ [\Psi]$, *then*

1. $\mathsf{coe}_{x.H}^{r \rightsquigarrow r'}(M) \doteq \mathsf{coe}_{x.\mathsf{hcom}_{\mathcal{U}_l^{\mathsf{Kan}}}^{s \rightsquigarrow s'}(A'; \overrightarrow{\xi_i \hookrightarrow z.B_i'})}^{r \rightsquigarrow r'}(M') \in H\langle r'/x\rangle\ [\Psi]$ *and*

2. *if* $r = r'$ *then* $\mathsf{coe}_{x.H}^{r \rightsquigarrow r'}(M) \doteq M \in H\langle r'/x\rangle\ [\Psi]$.

Proof. If $s = s'$ or ξ_i for some i, the results are immediate by parts (2–3) of Lemma 4.88. Otherwise, $s \neq s'$ and $\neg\xi_i$ for all i; apply coherent expansion to $\mathsf{coe}_{x.H}^{r \rightsquigarrow r'}(M)$ with the family defined in Figure 4.5. Consider $\psi = \mathrm{id}_\Psi$.

1. $N_i \doteq N_j \in B_i\ [\Psi, x, z \mid \xi_i\langle r/x\rangle, \xi_j\langle r/x\rangle]$ for all i, j by $M \in B_i\langle s'/z\rangle\langle r/x\rangle\ [\Psi \mid \xi_i\langle r/x\rangle]$ (by $M \in H\langle r/x\rangle\ [\Psi]$ and $H \doteq B_i\langle s'/z\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x \mid \xi_i]$) and $B_i \doteq B_j\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x, z \mid \xi_i, \xi_j]$.

2. $O \in A\langle r/x\rangle\ [\Psi, z]$ by

a) $(\mathsf{cap}^{s \leftsquigarrow s'}(M; \overrightarrow{\xi_i \hookrightarrow z.B_i}))\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi]$ by $M \in H\langle r/x\rangle\ [\Psi]$,

$$\begin{cases}
\mathsf{coe}^{r\psi\rightsquigarrow r'\psi}_{x.A\psi}(M\psi) & s\psi = s'\psi \\
\mathsf{coe}^{r\psi\rightsquigarrow r'\psi}_{x.B_i\langle s'/z\rangle\psi}(M\psi) & s\psi \neq s'\psi, \text{least } \xi_i\psi \\
(\mathsf{box}^{s\rightsquigarrow s'}(R; \overrightarrow{\xi_i \hookrightarrow Q_i\langle s'/z\rangle}))\langle r'/x\rangle\psi & s\psi \neq s'\psi, \forall i.\neg\xi_i\psi \\
\quad N_i := \mathsf{coe}^{s'\rightsquigarrow z}_{z.B_i}(\mathsf{coe}^{r\rightsquigarrow x}_{x.B_i\langle s'/z\rangle}(M)) & \\
\quad O := (\mathsf{hcom}^{s'\rightsquigarrow z}_{A}(\mathsf{cap}^{s\leftrightsquigarrow s'}(M; \overrightarrow{\xi_i \hookrightarrow z.B_i}); \overrightarrow{\xi_i \hookrightarrow z.\mathsf{coe}^{z\rightsquigarrow s}_{z.B_i}(N_i)}))\langle r/x\rangle & \\
\quad P := \mathsf{gcom}^{r\rightsquigarrow r'}_{x.A}(O\langle s\langle r/x\rangle/z\rangle; \overrightarrow{\xi_i \hookrightarrow x.N_i\langle s/z\rangle}|_{(x\#\xi_i)}, T) & \\
\quad T := s = s' \hookrightarrow x.\mathsf{coe}^{r\rightsquigarrow x}_{x.A}(M)|_{(x\#s,s')} & \\
\quad Q_k := \mathsf{gcom}^{s\langle r'/x\rangle\rightsquigarrow z}_{z.B_k\langle r'/x\rangle}(P; \overrightarrow{\xi_i \hookrightarrow z.N_i\langle r'/x\rangle}|_{(x\#\xi_i)}, r = r' \hookrightarrow z.N_k\langle r'/x\rangle) & \\
\quad R := \mathsf{hcom}^{s\rightsquigarrow s'}_{A}(P; \overrightarrow{\xi_i \hookrightarrow z.\mathsf{coe}^{z\rightsquigarrow s}_{z.B_i}(Q_i)}, r = r' \hookrightarrow z.O) &
\end{cases}$$

Figure 4.5: Family of reducts in Lemma 4.96.

b) $\mathsf{coe}^{z\rightsquigarrow s\langle r/x\rangle}_{z.B_i\langle r/x\rangle}(N_i\langle r/x\rangle) \doteq \mathsf{coe}^{z\rightsquigarrow s\langle r/x\rangle}_{z.B_j\langle r/x\rangle}(N_j\langle r/x\rangle) \in A\langle r/x\rangle\ [\Psi, z \mid \xi_i\langle r/x\rangle, \xi_j\langle r/x\rangle]$ for all i, j by $B_i\langle s/z\rangle\langle r/x\rangle \doteq A\langle r/x\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid \xi_i\langle r/x\rangle]$, and

c) $(\mathsf{cap}^{s\leftrightsquigarrow s'}(M; \overrightarrow{\xi_i \hookrightarrow z.B_i}))\langle r/x\rangle \doteq (\mathsf{coe}^{z\rightsquigarrow s}_{z.B_i}(N_i))\langle s'/z\rangle\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi \mid \xi_i\langle r/x\rangle]$ for all i because $\mathsf{cap}\langle r/x\rangle \doteq (\mathsf{coe}^{s'\rightsquigarrow s}_{z.B_i}(M))\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi \mid \xi_i\langle r/x\rangle]$ and $N_i\langle s'/z\rangle\langle r/x\rangle \doteq M \in B_i\langle s'/z\rangle\langle r/x\rangle\ [\Psi \mid \xi_i\langle r/x\rangle]$.

3. $P \in A\langle r'/x\rangle\ [\Psi]$ by

a) $O\langle s\langle r/x\rangle/z\rangle \in A\langle r/x\rangle\ [\Psi]$,

b) $N_i\langle s/z\rangle \doteq N_j\langle s/z\rangle \in A\ [\Psi, x \mid \xi_i, \xi_j]$ for all i, j such that $x \# \xi_i, \xi_j$ by $N_i\langle s/z\rangle \doteq N_j\langle s/z\rangle \in B_i\langle s/z\rangle\ [\Psi, x \mid \xi_i\langle r/x\rangle, \xi_j\langle r/x\rangle]$ and $B_i\langle s/z\rangle \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x \mid \xi_i]$,

c) $\mathsf{coe}^{r\rightsquigarrow x}_{x.A}(M) \in A\ [\Psi, x \mid s = s']$ if $x \# s, s'$ by $H\langle r/x\rangle \doteq A\langle r/x\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid s\langle r/x\rangle = s'\langle r/x\rangle]$,

d) $O\langle s\langle r/x\rangle/z\rangle \doteq N_i\langle s/z\rangle\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi \mid \xi_i]$ for all i such that $x \# \xi_i$ by $O\langle s\langle r/x\rangle/z\rangle \doteq (\mathsf{coe}^{z\rightsquigarrow s}_{z.B_i}(N_i))\langle s/z\rangle\langle r/x\rangle \doteq N_i\langle s/z\rangle\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi \mid \xi_i\langle r/x\rangle]$,

e) $O\langle s\langle r/x\rangle/z\rangle \doteq (\mathsf{coe}^{r\rightsquigarrow x}_{x.A}(M))\langle r/x\rangle \in A\langle r/x\rangle\ [\Psi \mid s = s']$ if $x \# s, s'$ because $O\langle s\langle r/x\rangle/z\rangle = O\langle s/z\rangle \doteq \mathsf{cap}\langle r/x\rangle \doteq M \in A\langle r/x\rangle\ [\Psi \mid s = s']$, and

f) $N_i\langle s/z\rangle \doteq \mathsf{coe}^{r\rightsquigarrow x}_{x.A}(M) \in A\ [\Psi, x \mid \xi_i, s = s']$ for all i such that $x \# \xi_i, s, s'$ by $N_i\langle s/z\rangle \doteq \mathsf{coe}^{r\rightsquigarrow x}_{x.B_i\langle s'/z\rangle}(M) \in B_i\langle s'/z\rangle\ [\Psi, x \mid \xi_i, s = s']$ and $B_i\langle s'/z\rangle \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x \mid \xi_i, s = s']$.

4. $Q_k \doteq Q_{k'} \in B_k\langle r'/x\rangle\ [\Psi, z \mid \xi_k\langle r'/x\rangle, \xi_{k'}\langle r'/x\rangle]$ for all k, k' by

a) $P \in B_k\langle s/z\rangle\langle r'/x\rangle\ [\Psi \mid \xi_k\langle r'/x\rangle]$ by $A \doteq B_k\langle s/z\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x \mid \xi_k]$,

b) $N_i\langle r'/x\rangle \doteq N_j\langle r'/x\rangle \in B_k\langle r'/x\rangle\ [\Psi, z \mid \xi_k\langle r'/x\rangle, \xi_i, \xi_j]$ for all i, j such that $x \mathbin{\#} \xi_i, \xi_j$ by $N_i \doteq N_j \in B_i\ [\Psi, x, z \mid \xi_i, \xi_j]$ and $B_i \doteq B_k\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, x, z \mid \xi_i, \xi_k]$,

c) $N_k\langle r'/x\rangle \doteq N_{k'}\langle r'/x\rangle \in B_k\langle r'/x\rangle\ [\Psi, z \mid \xi_k\langle r'/x\rangle, \xi_{k'}\langle r'/x\rangle]$,

d) $P \doteq N_i\langle s/z\rangle\langle r'/x\rangle \in B_k\langle s/z\rangle\langle r'/x\rangle\ [\Psi \mid \xi_k\langle r'/x\rangle, \xi_i]$ for all i such that $x \mathbin{\#} \xi_i$ by $P \doteq N_i\langle s/z\rangle\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi \mid \xi_i]$ and $A\langle r'/x\rangle \doteq B_k\langle s/z\rangle\langle r'/x\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid \xi_k\langle r'/x\rangle]$,

e) $P \doteq N_k\langle s/z\rangle\langle r'/x\rangle \in B_k\langle s/z\rangle\langle r'/x\rangle\ [\Psi \mid \xi_k\langle r'/x\rangle, r = r']$ by $P \doteq O\langle s\langle r/x\rangle/z\rangle \doteq (\mathrm{coe}_{z.B_k}^{z \rightsquigarrow s}(N_k))\langle s\langle r/x\rangle/z\rangle\langle r/x\rangle \in A\langle r'/x\rangle\ [\Psi \mid \xi_k\langle r'/x\rangle, r = r']$, and

f) $N_i\langle r'/x\rangle \doteq N_k\langle r'/x\rangle \in B_k\langle r'/x\rangle\ [\Psi, z \mid \xi_k\langle r'/x\rangle, \xi_i, r = r']$ for all i such that $x \mathbin{\#} \xi_i$.

5. $R\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi]$ by

a) $P \in A\langle r'/x\rangle\ [\Psi]$,

b) $\mathrm{coe}_{z.B_i\langle r'/x\rangle}^{z \rightsquigarrow s\langle r'/x\rangle}(Q_i) \doteq \mathrm{coe}_{z.B_j\langle r'/x\rangle}^{z \rightsquigarrow s\langle r'/x\rangle}(Q_j) \in A\langle r'/x\rangle\ [\Psi, z \mid \xi_i\langle r'/x\rangle, \xi_j\langle r'/x\rangle]$ for all i, j by $B_i \doteq B_j\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi, z, x \mid \xi_i, \xi_j]$ and $B_i\langle s/z\rangle\langle r'/x\rangle \doteq A\langle r'/x\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid \xi_i\langle r'/x\rangle]$,

c) $O \in A\langle r'/x\rangle\ [\Psi, z \mid r = r']$,

d) $P \doteq (\mathrm{coe}_{z.B_i}^{z \rightsquigarrow s}(Q_i))\langle s/z\rangle\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi \mid \xi_i\langle r'/x\rangle]$ for all i by $Q_i\langle s/z\rangle\langle r'/x\rangle \doteq P \in B_i\langle s/z\rangle\langle r'/x\rangle\ [\Psi \mid \xi_i\langle r'/x\rangle]$ and $B_i\langle s/z\rangle\langle r'/x\rangle \doteq A\langle r'/x\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi \mid \xi_i\langle r'/x\rangle]$,

e) $P \doteq O\langle s/z\rangle\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi \mid r = r']$ by $O\langle s/z\rangle\langle r'/x\rangle = O\langle s\langle r'/x\rangle/z\rangle$, and

f) $(\mathrm{coe}_{z.B_i}^{z \rightsquigarrow s}(Q_i))\langle r'/x\rangle \doteq O\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi, z \mid \xi_i\langle r'/x\rangle, r = r']$ for all i by $O\langle r'/x\rangle = O \doteq (\mathrm{coe}_{z.B_i}^{z \rightsquigarrow s}(N_i))\langle r/x\rangle \in A\langle r'/x\rangle\ [\Psi, z \mid \xi_i\langle r'/x\rangle]$ and $Q_i\langle r'/x\rangle \doteq \mathrm{N}_i\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi, z \mid \xi_i\langle r'/x\rangle, r = r']$.

6. $\mathrm{box}^{s\langle r'/x\rangle \rightsquigarrow s'\langle r'/x\rangle}(R\langle r'/x\rangle; \overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow Q_i\langle s'\langle r'/x\rangle/z\rangle}) \in H\langle r'/x\rangle\ [\Psi]$ by

a) $R\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi]$,

b) $Q_i\langle s'\langle r'/x\rangle/z\rangle \doteq Q_j\langle s'\langle r'/x\rangle/z\rangle \in B_i\langle s'/z\rangle\langle r'/x\rangle\ [\Psi \mid \xi_i\langle r'/x\rangle, \xi_j\langle r'/x\rangle]$ for all i, j, and

c) $(\mathrm{coe}_{z.B_i}^{s' \rightsquigarrow s}(Q_i\langle s'/z\rangle))\langle r'/x\rangle \doteq R\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi \mid \xi_i\langle r'/x\rangle]$ for all i because $R\langle r'/x\rangle \doteq (\mathrm{coe}_{z.B_i}^{z \rightsquigarrow s}(Q_i))\langle s'/z\rangle\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi \mid \xi_i\langle r'/x\rangle]$.

Consider $\psi : \Psi' \to \Psi$. When $s\psi \neq s'\psi$ and $\neg\xi_i\psi$ for all i, coherence is immediate. When $s\psi = s'\psi$, then by $s \neq s'$, we must have $x \mathbin{\#} s, s'$ and thus $s\langle r'/x\rangle\psi = s'\langle r'/x\rangle\psi$

also. Thus $\mathsf{box}\langle r'/x\rangle\psi \doteq R\langle r'/x\rangle\psi \doteq P\langle r'/x\rangle\psi \doteq (\mathsf{coe}^{r\rightsquigarrow x}_{x.A}(M))\langle r'/x\rangle\psi \in A\langle r'/x\rangle\ [\Psi']$ as required. When $s\psi \neq s'\psi$ and $\xi_i\psi$ for the least such i, again $x \mathrel{\#} \xi_i$ and $\mathsf{box}\langle r'/x\rangle\psi \doteq Q_i\langle s'/z\rangle\langle r'/x\rangle\psi \doteq N_i\langle s'/z\rangle\langle r'/x\rangle\psi \doteq (\mathsf{coe}^{r\rightsquigarrow r'}_{x.B_i\langle s'/z\rangle}(M))\psi \in A\langle r'/x\rangle\ [\Psi']$. By Lemma 4.18, $\mathsf{coe}^{r\rightsquigarrow r'}_{x.H}(M) \doteq \mathsf{box}\langle r'/x\rangle \in H\langle r'/x\rangle\ [\Psi]$; part (1) follows by a symmetric argument.

For part (2), if $r = r'$ then $R\langle r'/x\rangle \doteq O\langle s'/z\rangle\langle r'/x\rangle \doteq (\mathsf{cap}^{s\leftsquigarrow s'}(M; \overrightarrow{\xi_i \hookrightarrow z.B_i}))\langle r'/x\rangle \in A\langle r'/x\rangle\ [\Psi]$ and $Q_i\langle s'/z\rangle\langle r'/x\rangle \doteq N_k\langle s'/z\rangle\langle r'/x\rangle \doteq M \in B_i\langle s'/z\rangle\langle r'/x\rangle\ [\Psi \mid \xi_i\langle r'/x\rangle]$ for all i, so $\mathsf{box}\langle r'/x\rangle \doteq M \in H\langle r'/x\rangle\ [\Psi]$ by Rule 4.94, and part (2) follows by transitivity. □

Rule 4.97 (Kan type formation). *If $A, \overrightarrow{\xi_i \hookrightarrow y.B_i}$ and $A', \overrightarrow{\xi_i \hookrightarrow y.B_i'}$ are equal Kan type compositions $r \rightsquigarrow r'$, then*

1. $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A'; \overrightarrow{\xi_i \hookrightarrow y.B_i'})\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$,
2. *if $r = r'$ then* $\mathsf{hcom}^{r\rightsquigarrow r}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq A\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, *and*
3. *if ξ_i then* $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A; \overrightarrow{\xi_i \hookrightarrow B_i}) \doteq B_i\langle r'/y\rangle\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$.

Proof. We already showed parts (2–3) in Lemma 4.88. For part (1), the hcom conditions hold by Lemma 4.95 instantiated at all $\psi : \Psi' \to \Psi$ instances of this Kan type composition; the coe conditions hold by Lemma 4.96 at the $\psi : (\Psi', x) \to \Psi$ instances. □

4.4.12 *Universes*

For $i \in \{0, 1, \dots\}$ and $j \in \{i+1, i+2, \dots, \omega\}$:

$$\tau^{\kappa'}_j(\Psi, \mathcal{U}^\kappa_i, \mathcal{U}^\kappa_i, \{(A_0, B_0) \mid \exists\varphi.\tau^\kappa_i(\Psi, A_0, B_0, \varphi)\})$$

Rule 4.98 (Pretype formation). *For $i \in \{0, 1, \dots\}$ and $j \in \{i+1, i+2, \dots, \omega\}$, $\tau^{\kappa'}_j \models (\mathcal{U}^\kappa_i\ \mathsf{type}_{\mathsf{pre}}\ [\Psi])$.*

Proof. We have $\mathcal{U}^\kappa_i \sim \mathcal{U}^\kappa_i \Downarrow \llbracket\mathcal{U}^\kappa_i\rrbracket \in \tau^{\kappa'}_j\ [\Psi]$ by $\mathcal{U}^\kappa_i\ \mathsf{val}_{\square}$ and the definition of $\tau^{\kappa'}_j$. To prove $\mathrm{Coh}(\llbracket\mathcal{U}^\kappa_i\rrbracket)$, suppose that $\llbracket\mathcal{U}^\kappa_i\rrbracket_{\Psi'}(A_0, B_0)$ holds and show $A_0 \sim B_0 \in \llbracket\mathcal{U}^\kappa_i\rrbracket\ [\Psi']$. Then $\tau^\kappa_i(\Psi', A_0, B_0, \varphi)$ for some φ, and thus $A_0 \sim B_0 \Downarrow \alpha \in \tau^\kappa_i\ [\Psi']$ for some α by the value-coherence property of τ^κ_i (Theorem 4.9), completing our proof. □

Rule 4.99 (Cumulativity).

1. *If $\tau^{\mathsf{pre}}_\omega \models (A \doteq B \in \mathcal{U}^\kappa_i\ [\Psi])$ and $i \leq j$ then $\tau^{\mathsf{pre}}_\omega \models (A \doteq B \in \mathcal{U}^\kappa_j\ [\Psi])$.*
2. *If $\tau^{\mathsf{pre}}_\omega \models (A \doteq B \in \mathcal{U}^{\mathsf{Kan}}_i\ [\Psi])$ then $\tau^{\mathsf{pre}}_\omega \models (A \doteq B \in \mathcal{U}^{\mathsf{pre}}_i\ [\Psi])$.*

Proof. For part (1), by Theorem 4.10, $\tau_i^\kappa \subseteq \tau_j^\kappa$ and therefore $[\![\mathcal{U}_i^\kappa]\!] \subseteq [\![\mathcal{U}_j^\kappa]\!]$. The result follows by monotonicity of the candidate judgments. For part (2), by Theorem 4.10, $\tau_i^{\mathsf{Kan}} \subseteq \tau_i^{\mathsf{pre}}$ and the result follows similarly. □

As in Chapter 2, we have defined the elements of $\mathcal{U}_i^\kappa$ inductively; therefore, to see that the elements of $\mathcal{U}_i^{\mathsf{Kan}}$ are Kan types we must induct on the definition of τ_i^{Kan}.

Lemma 4.100. *Let* $\Phi^\kappa(\tau) = \{(\Psi, A_0, B_0, \varphi) \mid \tau \models (A_0 \doteq B_0 \ \mathsf{type}_\kappa \ [\Psi])\}$*. If* τ *is a cubical type system and* $A \sim B \downarrow \alpha \in \Phi^\kappa(\tau)\ [\Psi]$ *then* $\tau \models (A \doteq B \ \mathsf{type}_\kappa \ [\Psi])$*.*

Proof. We apply coherent expansion to A with family $\{A_\psi^{\Psi'} \mid A\psi \Downarrow A_\psi^{\Psi'}\}_\psi^{\Psi'}$. By our hypothesis at $\psi, \mathsf{id}_{\Psi'}$ we have $\tau \models (A_\psi^{\Psi'} \ \mathsf{type}_\kappa \ [\Psi'])$; we must show $\tau \models (A_\psi^{\Psi'} \doteq (A_{\mathsf{id}_\Psi}^{\Psi})\psi \ \mathsf{type}_\kappa \ [\Psi'])$.

Suppose $\kappa = \mathsf{pre}$. We already know $A_\psi^{\Psi'}$ and $(A_{\mathsf{id}_\Psi}^{\Psi})\psi$ are pretypes, so by Lemma 4.14 it suffices to show that their values are equal pretypes, which follows from our hypothesis at id_Ψ, ψ. Thus, by Lemma 4.17, $\tau \models (A \doteq A_0 \ \mathsf{type}_{\mathsf{pre}} \ [\Psi])$ where $A \Downarrow A_0$.

Suppose $\kappa = \mathsf{Kan}$. Both $A_\psi^{\Psi'}$ and $(A_{\mathsf{id}_\Psi}^{\Psi})\psi$ are Kan, so by Lemma 4.30 it suffices to show their values under all $\psi' : \Psi'' \to \Psi'$ are equally Kan, which follows from our hypothesis at ψ, ψ' and $\mathsf{id}_\Psi, \psi\psi'$. Thus, by Lemma 4.31, $\tau \models (A \doteq A_0 \ \mathsf{type}_{\mathsf{Kan}} \ [\Psi])$ where $A \Downarrow A_0$.

In each case, we similarly obtain $\tau \models (B \doteq B_0 \ \mathsf{type}_\kappa \ [\Psi])$ where $B \Downarrow B_0$, and the result follows from $\tau \models (A_0 \doteq B_0 \ \mathsf{type}_\kappa \ [\Psi])$ by our hypothesis at $\mathsf{id}_\Psi, \mathsf{id}_\Psi$. □

Lemma 4.101.

1. *If* $\tau_\omega^{\mathsf{pre}} \models (A \doteq B \in \mathcal{U}_i^{\mathsf{Kan}} \ [\Psi])$ *then* $\tau_i^{\mathsf{Kan}} \models (A \doteq B \ \mathsf{type}_{\mathsf{Kan}} \ [\Psi])$*.*
2. *If* $\tau_\omega^{\mathsf{pre}} \models (A \doteq B \in \mathcal{U}_i^{\mathsf{pre}} \ [\Psi])$ *then* $\tau_i^{\mathsf{pre}} \models (A \doteq B \ \mathsf{type}_{\mathsf{pre}} \ [\Psi])$*.*

Proof. Let $\Phi_i^\kappa = \{(\Psi, A_0, B_0, \varphi) \mid \tau_i^\kappa \models (A_0 \doteq B_0 \ \mathsf{type}_\kappa \ [\Psi])\}$. Our proof proceeds by mutual strong induction on i.

1. We show $K(\nu_i, \Phi_i^{\mathsf{Kan}}) \subseteq \Phi_i^{\mathsf{Kan}}$; then $\tau_i^{\mathsf{Kan}} \subseteq \Phi_i^{\mathsf{Kan}}$, so $A \sim B \downarrow \alpha \in \Phi_i^{\mathsf{Kan}}\ [\Psi]$, and part (1) follows from Lemma 4.100. We check each type former separately. Suppose $\textsc{Fun}(\Phi_i^{\mathsf{Kan}})(\Psi, (a{:}A) \to B, (a{:}A') \to B', \varphi)$. Then $A \sim A' \downarrow \alpha \in \Phi_i^{\mathsf{Kan}}\ [\Psi]$, which by Lemma 4.100 implies $\tau_i^{\mathsf{Kan}} \models (A \doteq A' \ \mathsf{type}_{\mathsf{Kan}} \ [\Psi])$; similarly, $\tau_i^{\mathsf{Kan}} \models (a : A \gg B \doteq B' \ \mathsf{type}_{\mathsf{Kan}} \ [\Psi])$. By Rule 4.42, $\tau_i^{\mathsf{Kan}} \models ((a{:}A) \to B \doteq (a{:}A') \to B' \ \mathsf{type}_{\mathsf{Kan}} \ [\Psi])$.

   Other cases are similar except for universes, where we must show for $j < i$ that $\tau_i^{\mathsf{Kan}} \models (\mathcal{U}_j^\kappa \ \mathsf{type}_{\mathsf{Kan}} \ [\Psi])$. To see that $\mathcal{U}_j^{\mathsf{pre}}$ admits homogeneous composition, suppose we are given a composition scenario in $\mathcal{U}_j^{\mathsf{pre}}$ relative to τ_i^{Kan} (and hence, by monotonicity, relative also to $\tau_\omega^{\mathsf{pre}}$). By inductive hypothesis (2), this is a composition of pretypes in τ_j^{pre}. By Rule 4.87, $\mathsf{hcom}_{\mathcal{U}_j^{\mathsf{pre}}}$ of this scenario is a pretype in $\tau_\omega^{\mathsf{pre}}$. Therefore $\tau_i^{\mathsf{Kan}} \models (\mathsf{hcom}_{\mathcal{U}_j^{\mathsf{pre}}} \in \mathcal{U}_j^{\mathsf{pre}} \ [\Psi])$ as required. Composition in $\mathcal{U}_j^{\mathsf{Kan}}$ follows by a similar argument, appealing instead to inductive hypothesis (1) and Rule 4.90. Coercion for universes is immediate by $\mathsf{coe}^{r \rightsquigarrow r'}_{x.\mathcal{U}^{\kappa}_{j}}(A) \longmapsto_{\square} A$.

2. We show $P(\nu_i, \tau^{\mathsf{Kan}}_i, \Phi^{\mathsf{pre}}_i) \subseteq \Phi^{\mathsf{pre}}_i$; then $\tau^{\mathsf{pre}}_i \subseteq \Phi^{\mathsf{pre}}_i$, and part (2) follows from Lemma 4.100. Most cases are analogous to part (1), with a few exceptions: for HKan, we appeal to inductive hypothesis (1) and Rule 4.90, and for UPre and UKan we obtain $\tau^{\mathsf{pre}}_i \models (\mathcal{U}^{\kappa}_j \ \mathsf{type}_{\mathsf{pre}} \ [\Psi])$ immediately for all $j < i$ by Rule 4.98. □

Rule 4.102 (Elimination). *If* $\tau^{\mathsf{pre}}_{\omega} \models (A \doteq B \in \mathcal{U}^{\kappa}_i \ [\Psi])$ *then* $\tau^{\mathsf{pre}}_{\omega} \models (A \doteq B \ \mathsf{type}_{\kappa} \ [\Psi])$.

Proof. By Lemma 4.101, $\tau^{\kappa}_i \models (A \doteq B \ \mathsf{type}_{\kappa} \ [\Psi])$; we obtain $\tau^{\mathsf{pre}}_{\omega} \models (A \doteq B \ \mathsf{type}_{\kappa} \ [\Psi])$ from $\tau^{\kappa}_i \subseteq \tau^{\mathsf{pre}}_{\omega}$ (by Theorem 4.10) and monotonicity of the pretype and Kan type judgments. □

Rule 4.103 (Introduction). *Relative to* $\tau^{\mathsf{pre}}_{\omega}$*:*

1. *If* $A \doteq A' \in \mathcal{U}^{\kappa}_i \ [\Psi]$ *and* $a : A \gg B \doteq B' \in \mathcal{U}^{\kappa}_i \ [\Psi]$ *then* $(a{:}A) \to B \doteq (a{:}A') \to B' \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
2. *If* $A \doteq A' \in \mathcal{U}^{\kappa}_i \ [\Psi]$ *and* $a : A \gg B \doteq B' \in \mathcal{U}^{\kappa}_i \ [\Psi]$ *then* $(a{:}A) \times B \doteq (a{:}A') \times B' \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
3. *If* $A \doteq A' \in \mathcal{U}^{\kappa}_i \ [\Psi, x]$ *and* $P_{\varepsilon} \doteq P'_{\varepsilon} \in A\langle \varepsilon/x \rangle \ [\Psi]$ *for* $\varepsilon \in \{0, 1\}$, *then* $\mathsf{Path}_{x.A}(P_0, P_1) \doteq \mathsf{Path}_{x.A'}(P'_0, P'_1) \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
4. *If* $A \doteq A' \in \mathcal{U}^{\mathsf{pre}}_i \ [\Psi]$, $M \doteq M' \in A \ [\Psi]$, *and* $N \doteq N' \in A \ [\Psi]$, *then* $\mathsf{Eq}_A(M, N) \doteq \mathsf{Eq}_{A'}(M', N') \in \mathcal{U}^{\mathsf{pre}}_i \ [\Psi]$.
5. $\mathsf{void} \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
6. $\mathsf{bool} \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
7. $\mathsf{nat} \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
8. $\mathbb{S}^1 \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
9. *If* $A \doteq A' \in \mathcal{U}^{\kappa}_i \ [\Psi \mid r = 0]$, $B \doteq B' \in \mathcal{U}^{\kappa}_i \ [\Psi]$, *and* $E \doteq E' \in \mathsf{Equiv}(A, B) \ [\Psi \mid r = 0]$, *then* $\mathsf{V}_r(A, B, E) \doteq \mathsf{V}_r(A', B', E') \in \mathcal{U}^{\kappa}_i \ [\Psi]$.
10. *If* $j > i$ *then* $\mathcal{U}^{\kappa}_i \in \mathcal{U}^{\kappa'}_j \ [\Psi]$.

Proof. We establish the presuppositions of parts (1–4) and (9) by Rule 4.102. In part (1), for example, from Rule 4.102 and $\tau^{\mathsf{pre}}_{\omega} \models (A \in \mathcal{U}^{\kappa}_j \ [\Psi])$ we obtain the presupposition $\tau^{\mathsf{pre}}_{\omega} \models (A \ \mathsf{type}_{\kappa} \ [\Psi])$ of $\tau^{\mathsf{pre}}_{\omega} \models (a : A \gg B \doteq B' \in \mathcal{U}^{\kappa}_j \ [\Psi])$.

For part (1), by $\tau^{\mathsf{pre}}_{\omega} \models (A \doteq A' \in \mathcal{U}^{\kappa}_j \ [\Psi])$ and Lemma 4.101, $\tau^{\kappa}_j \models (A \doteq A' \ \mathsf{type}_{\kappa} \ [\Psi])$. By $\tau^{\mathsf{pre}}_{\omega} \models (a : A \gg B \doteq B' \in \mathcal{U}^{\kappa}_j \ [\Psi])$, Lemma 4.101, and the fact that $\tau^{\mathsf{pre}}_{\omega}$ and τ^{κ}_j give rise to equal $[\![A]\!]$ (by Theorem 4.10 and the unicity property of type systems), $\tau_j^\kappa \models (a : A \gg B \doteq B'\ \mathsf{type}_\kappa\ [\Psi])$. Therefore, by Rule 4.36, $(a{:}A) \to B \sim (a{:}A') \to B' \downarrow \_ \in \tau_j^\kappa\ [\Psi]$ and thus $(a{:}A) \to B \sim (a{:}A') \to B' \in [\![\mathcal{U}_j^\kappa]\!]\ [\Psi]$, completing part (1). Parts (2–10) are analogous. □

Rule 4.104 (Kan type formation)**.** $\tau_\omega^{\mathsf{pre}} \models (\mathcal{U}_i^\kappa\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi])$.

Proof. By Rule 4.103, $\tau_\omega^{\mathsf{pre}} \models (\mathcal{U}_i^\kappa \in \mathcal{U}_{i+1}^{\mathsf{Kan}}\ [\Psi])$; the result follows by Rule 4.102. □

Theorem 4.105 (Univalence)**.** *There exist terms* $\mathsf{ua}_{\square}$ *and* $\mathsf{ua}\beta_{\square}$ *such that, relative to* $\tau_\omega^{\mathsf{pre}}$, *and for all* $i \in \{0, 1, \dots\}$*:*

$$\mathsf{ua}_{\square} \in (A\ B{:}\mathcal{U}_i^{\mathsf{Kan}}) \to \mathsf{Equiv}(A, B) \to \mathsf{Path}_{\_.\mathcal{U}_i^{\mathsf{Kan}}}(A, B)\ [\cdot]$$
$$\mathsf{ua}\beta_{\square} \in (A\ B{:}\mathcal{U}_i^{\mathsf{Kan}}) \to (e{:}\mathsf{Equiv}(A, B)) \to (a{:}A) \to \mathsf{Path}_{\_.B}(\mathsf{coe}_{x.(\mathsf{ua}_{\square}\ e)@x}^{0 \rightsquigarrow 1}(a), \mathsf{fst}(e)\ a)\ [\cdot]$$

Therefore, every $\mathcal{U}_i^{\mathsf{Kan}}$ *is a univalent universe [Lic16].*

Proof. Define:

$$\mathsf{ua}_{\square}\ A\ B\ e := \langle x \rangle \mathsf{V}_x(A, B, e)$$
$$\mathsf{ua}\beta_{\square}\ A\ B\ e\ a := \langle x \rangle \mathsf{coe}_{\_.B}^{x \rightsquigarrow 1}(\mathsf{fst}(e)\ a)$$

Using previously-proven rules, one can routinely verify the required typing judgments. The only obstacle is to see that the left endpoint of $\mathsf{ua}\beta_{\square}$ is correct, which requires unfolding the definition of coercion in V-types. Using the variable names of Figure 4.2:

$$\begin{aligned}
\mathsf{coe}_{x.(\mathsf{ua}_{\square}\ e)@x}^{0 \rightsquigarrow 1}(a) &\doteq \mathsf{coe}_{x.\mathsf{V}_x(A,B,e)}^{0 \rightsquigarrow 1}(a) \\
&\doteq \mathsf{Vin}_1(\mathsf{fst}(O), P) \\
&\doteq \mathsf{hcom}_B^{1 \rightsquigarrow 0}(N; \dots, 1 = 1 \hookrightarrow \_.N, \dots) \\
&\doteq \mathsf{coe}_{\_.B}^{0 \rightsquigarrow 1}(\mathsf{Vproj}_0(a, \mathsf{fst}(e))) \\
&\doteq \mathsf{coe}_{\_.B}^{0 \rightsquigarrow 1}(\mathsf{fst}(e)\ a) \in B\ [\cdot]
\end{aligned}$$

□

5 Conclusion

In the Univalent Foundations we accept Gödel's results not simply as correct ones but as *natural* ones. We know that any foundation will be incomplete and we are not concerned with the impossible task of finding one complete foundation. Instead, we are concerned with creating a practical foundation which we can use now and establishing a process which can ensure a healthy [growth] and transformation of this foundation in the future.

—Vladimir Voevodsky, *Paul Bernays Lectures* [Voe14, p. 3.22]

In this dissertation, we contribute a Cartesian cubical type theory which gives constructive meaning to higher-dimensional concepts—univalence and higher inductive types—first introduced in Book HoTT [UF13]. Cubical type theories are organized around *interval variables* representing abstract hypercubes as syntax, and *uniform Kan operations* expressing closure properties of those hypercubes; compared to De Morgan cubical type theory [CCHM18], our type theory has a simpler interval but more complex Kan operations.

We present our type theory through the lens of its computational semantics *à la* Nuprl [All87], defining each type as a behavioral predicate on programs drawn from an untyped λ-calculus augmented with interval variables and Kan operations. Because we successfully reconstruct core features of Book HoTT, we argue that *higher-dimensional types classify higher-dimensional programs extensionally according to their behaviors.* These classifications match expectations—for instance, Boolean programs all evaluate to true or false (Theorem 4.63). In fact, using a novel *validity* condition on Kan compositions, we obtain a sharper characterization of programs of higher inductive type (Theorem 4.77) than Cohen et al. [CCHM18].

The author's collaborators have already built on results presented in this dissertation. Cavallo and Harper [CH19a] extend our syntax and computational semantics with a schema for indexed higher inductive types, including identity types, truncations, and many other type formers. Separately, Cavallo and Harper [CH19b] extend our type theory with internal parametricity primitives affirming that type quantification is uniform and not *ad hoc* [Rey83]. Internal parametricity implies anti-classical principles such as the refutation of the law of excluded middle, and may simplify difficult coherence theorems for indexed higher inductive types, such as associativity of smash products.

Although this dissertation adopts an operational perspective, Angiuli et al. [Ang+19] show that the same constructions give rise to a denotational semantics of univalent type theory in Cartesian cubical sets. Cavallo, Mörtberg, and Swan [CMS19] have recently generalized the Cartesian and De Morgan constructions, obtaining a denotational semantics for univalent type theory that coincides with prior work when extended with diagonal equations ($x = z$) or connections, respectively.

Our experiences developing the **RedPRL** [Ang+18] and **redtt** [Red18] proof assistants have suggested various extensions to Chapter 4, including extension types (Section 3.5) and a refinement of our two-level system supporting hcom types, coe types, and discrete Kan types in addition to pretypes and Kan types. Although we need both homogeneous composition and coercion to recover weak identity elimination (as discussed in Section 3.2), we can achieve a finer analysis by considering types with only one of these operations.

For instance, homogeneous composition in $(a{:}A) \to B$ does not require both A and B to be Kan as Appendix A suggests, but only that B admits homogeneous composition; similarly, coercion in $(a{:}A) \to B$ requires only coercion in both A and B. In contrast, coercion in $\mathsf{Path}_{x.A}(M, N)$ does require A to admit both composition and coercion. Types with only one of two Kan operations do arise in practice—notably, $\mathsf{Eq}_A(M, N)$ always admits homogeneous composition.

On the other hand, coercion in $\mathsf{Eq}_A(M, N)$ implies that $(a\ a'{:}A) \to \mathsf{Path}_{\_.A}(a, a') \to \mathsf{Eq}_A(a, a')$, and is therefore impossible in general. It is, however, possible at types with sufficiently trivial path structure—more precisely, types that are constant in all interval variables, and whose elements are constant in all interval variables (Definition 5.1). We call such types *discrete Kan*; they are closed under standard type formers, including bool and nat. We therefore obtain a systematic account of those types whose strict equality types are Kan, an analysis lacking in previous two-level type theories [Voe13; ACK16; ACK17].

Definition 5.1 (Discrete Kan types). $A \doteq B\ \mathsf{type}_{\mathsf{disc}}\ [\Psi]$, presupposing $A \doteq B\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi]$, when for all $\psi_1 : \Psi_1 \to \Psi$ and $\psi_2, \psi_2' : \Psi_2 \to \Psi_1$,

1. $A\psi_1\psi_2 \doteq B\psi_1\psi_2'\ \mathsf{type}_{\mathsf{Kan}}\ [\Psi_2]$ and
2. for all $M \in A\psi_1\ [\Psi_1]$, $M\psi_2 \doteq M\psi_2' \in A\psi_1\psi_2\ [\Psi_2]$.

We close with remarks on several core problems in dependent type theory.

Relation to spaces The *Homotopy Type Theory* book [UF13] establishes basic results of algebraic topology for the axiomatic notion of space available in Book HoTT. At the time, it was unclear how general this notion of space really was, and we had no models of many higher inductive types. Much has changed since 2013. Shulman [Shu19] proves that most of Book HoTT has models in all Grothendieck ∞-toposes, which serve as general settings for homotopy theory. Cubical type theories, in addition to giving constructive meaning to univalence, provide models of parametrized higher inductive types [CHM18; CH19a].

Despite various advantages of cubical type theories over Book HoTT, it remains unknown whether cubical type theories yield the standard notion of space. One approach to this problem is to formulate the uniform Kan operations of cubical type theory in the language of Quillen model categories and establish a Quillen equivalence with simplicial sets. The former has been completed for De Morgan cubes by Sattler [Sat17], for Cartesian cubes by Coquand [Coq18] and Awodey [Awo19], and for both by Cavallo, Mörtberg, and Swan [CMS19]. Unfortunately, the latter step of proving an equivalence fails for both De Morgan and Cartesian cubes [Coq18]. But not all is lost: Awodey, Cavallo, Coquand, Riehl, and Sattler have recently proposed *equivariant uniform Kan operations*, a variation of the Cartesian Kan operations for which the equivalence holds.

Metatheoretic equality This dissertation wrestles with many notions of equality in both object theory and metatheory. In Chapters 2 and 4, we define typehood and membership judgments that respect a powerful, type-sensitive equality judgment, using partial equivalence relations defined directly on untyped λ-terms. In Chapter 4, typehood and membership also respect interval substitutions, even though evaluation does not.

Although our consideration of untyped λ-terms is specific to computational semantics, similar issues arise in standard canonicity proofs for intensional type theories. Huber's [Hub18] operational semantics is defined only on well-typed terms, but is nevertheless finer than judgmental equality; like us, he expends significant effort proving that (on well-typed terms) it commutes with interval substitution in an appropriate sense. In both cases, one experiences a tension between the judgments of type theory, which respect judgmental equality, and evaluation, which does not.

Resolving this tension, type theorists have recently developed canonicity and normalization proofs that consider only well-typed terms modulo judgmental equality [Shu15; Coq19; KHS19; SAG19]. These proofs use *Artin gluing* to construct proof-relevant logical relations as a model of type theory *qua* algebraic structure. In gluing models for canonicity,

$$[\![\cdot \vdash M : \mathsf{bool}]\!] := (M = \mathsf{true}) + (M = \mathsf{false})$$

where $[\![\cdot \vdash \mathsf{true} : \mathsf{bool}]\!] := \mathsf{inl} \star$ and $[\![\cdot \vdash \mathsf{false} : \mathsf{bool}]\!] := \mathsf{inr} \star$. One computes the value of a closed Boolean M by reading the tag bit of $[\![M]\!]$; disjointness of true and false follows from $\mathsf{inl} \star \neq \mathsf{inr} \star$ and the fact that $[\![-]\!]$ respects equality [Coq19].

We expect gluing techniques to be a particularly significant advance for cubical type theories [CHS19; SAG19], where they sidestep entirely the need to establish coherence of evaluation and interval substitution, as in Chapter 4 and Huber [Hub18].

Transportational equality Most type theories describe two differing notions of equality, which Voevodsky [Voe16] calls *substitutional* and *transportational*. Substitutional equalities are silent: elements can be replaced directly by their equals. In intensional type theories, substitutional equality is definitional, and includes computation rules for all types and uniqueness rules for dependent function and pair types. In (Idealized) Nuprl, a consequence of equality reflection is that all provable equations are substitutional, as is the case in standard (informal) mathematical practice.

Transportational equalities require coercions: properties of elements can be translated into corresponding properties of their equals. In Book HoTT and many proof assistants, *all* non-definitional equations are transportational, including mundane equations like the laws of arithmetic. Because of dependency, these term-level coercions appear also in types, causing headaches for users who must coerce between different coercions, *et cetera.* But such bureaucracy is necessary for the weaker notion of equality induced by univalence or higher inductive types. Equalities generated by univalence, for instance, are necessarily transportational because an element of $A \times B$ is not literally also an element of $B \times A$.

In the author's opinion, there is still much more to understand about rich transportational equality in type theory. First, univalence and higher inductive types have countless applications beyond synthetic homotopy theory. Notably, they provide direct type-theoretic constructions of quotients and free algebraic structures on sets [UF13, Section 6.11]. In intensional type theories, such constructions previously required passing to setoids [Hof95]. Extensional type theories are compatible with quotients [Ana+14], but passing to a quotient A/R there erases information: one cannot recover a proof of $R(a, b)$ from an equality $\mathsf{Eq}_{A/R}(\mathsf{in}(a), \mathsf{in}(b))$, a useful property known as *effectivity* [Mai99].[1] These basic applications of higher-dimensional structure have been investigated primarily in non-cubical type theories [VAG+; Bau+17]; we suggest further experimentation in cubical type theories, where these constructions are also computationally well-behaved.

In computer science, univalence establishes that type-theoretic constructions respect equivalence, allowing for code reuse in both programming [VMA19] and proving [TTS18]. Higher inductive types realize the decades-old dream of data types with laws, or quotiented algebraic data types [Tur85; BGW17], encode computational effects such as partiality [ADK17], and capture abstract interfaces and their implementations [Ang+16]. Again, many of these applications emerged before they could be realized in cubical programming languages; we look forward to experimenting with them in cubical Agda and **redtt**.

Although homotopy type theorists primarily consider univalence and higher inductive types, other transportational equalities are also useful in type theory. Birkedal et al. [Bir+18] consider a cubical type theory with transportational equalities corresponding to type isomorphisms in the topos of trees; these isomorphisms arise in the study of guarded recursion, but can unfold infinitely when added to type theory as substitutional equalities. Finally, directed type theories, so named for their *non-invertible* transportational equalities, have many potential applications: for mathematicians as a language for $(\infty, 1)$-categories

[1] One can obtain effective quotients by adding axioms, but these lack any obvious computational interpretation [Mai99, Section 5].

[RS17], and for programming language theorists as a framework for defining languages whose syntax automatically respects substitution [LH11]. We are hopeful that researchers will develop a directed type theory with canonicity.

Substitutional equality Substitutional equality is a double-edged sword. On one hand, type theories with strong substitutional equalities have more well-typed terms and fewer coercions cluttering types; on the other hand, it is harder to establish judgments of such type theories, because their terms and types omit more necessary steps of reasoning. It is here that the constructive nature of type theory plays an essential role in practice, although the specifics differ significantly between intensional and computational type theories.

Substitutional equality of intensional type theories is decidable by calculational means; proof assistants can therefore always reconstruct the reason why a judgment holds, or determine conclusively that it does not hold. In earlier theories, such as the calculus of constructions, substitutional equality coincides with untyped β-conversion and is decided by comparing normal forms [CH88]. Modern *normalization by evaluation* algorithms also account for many type-sensitive equations, including uniqueness principles for dependent function and pair types, by interleaving reduction and type-driven expansion [ACD07].

In type theories with equality reflection, substitutional equality is not decidable; instead, a proof assistant must at times appeal to users for equality reasoning. However, it is essential for proof assistants to automate *many* substitutional equalities, lest users be required to manually invoke all computation principles, congruence principles, unfoldings of definitions, *et cetera*. Without any automation, substitutional equality takes on the character of transportational equality—paradoxically, one must always coerce explicitly!

Indeed, users of modern Nuprl rely heavily on judgments being closed under arbitrary contraction of β-redexes [How89; AR14], which removes typing obligations from computation and congruence principles but requires a non-standard semantics for open judgments (see Section 2.6). Even so, Nuprl cannot β-reduce automatically: reduction may obfuscate subgoals requiring user input, and may not even terminate!

Our experience in **RedPRL** has shown that the Nuprl approach is significantly less effective for cubical type theory, because necessary simplification steps often engender many typing obligations. For example, to replace $\mathsf{hcom}^{0\rightsquigarrow 0}_{(a:A)\to B}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})$ by M, users must prove that $(a{:}A) \to B$ is a Kan type (and hence that B respects equality in A), that M and every N_i are elements of $(a{:}A) \to B$, and that M, N_i are pairwise equal where they overlap. Unlike in Nuprl, only *closed* **RedPRL** judgments respect evaluation (Lemma 4.16), a property sufficient for canonicity (Theorem 4.63) but not helpful to users of a proof assistant, who typically work under hypotheses.

The author concludes that canonicity theorems are a standard measure of robustness for substitutional equality of closed terms at base type, but say little about usability of proof assistants, which is instead correlated with having good facilities for establishing substitutional equality of *open terms* and at *higher type*. The nature of such facilities is of course highly sensitive to the choice of substitutional equality: Nuprl provides tactics for untyped reduction, whereas Coq and Agda silently decide a weaker definitional equality.

As for cubical computational type theory, there are multiple paths forward. One is to find modifications to our computational semantics justifying unstable reduction for open terms in **RedPRL**, just as Nuprl relies on modifications to Martin-Löf's meaning explanations proposed by Allen [All87, Chapter 8] and Howe [How89].

Another possibility is to design an implementation strategy for equality reflection that does not rely on untyped reduction. The most promising work in this direction is the Andromeda proof assistant [Bau+16], which aims to implement Voevodsky's Homotopy Type System [Voe13]. Andromeda uses bidirectional type checking modulo equational queries that are implemented by algebraic effects and handlers [BP15]. However, most type theorists consider equality reflection unworkable, and often approximate it by postulating function extensionality and uniqueness of identity proofs [Hof95].

A third approach, explored in **redtt** and Appendix B, is to regard our computational semantics as the extraction semantics of an intensional calculus. In this approach, a natural next step is to develop and prove correct a normalization by evaluation algorithm for Cartesian cubical type theory; another is to add strict equality types to **redtt**. We have already touched on multiple approaches suitable for two-level type theory: Nuprl's (less effective in the cubical setting), Andromeda's (promising but not yet fully realized), and even postulating axioms (which disrupts canonicity) [ACK16].

The author proposes another strict equality connective in XTT [SAG19], an intensional cubical type theory with a strict variant of path types. Path types are already in many ways an improvement on traditional identity types, which lack function extensionality and do not directly express type-heterogeneous equations. In XTT, any two paths with the same endpoints are moreover definitionally equal. Therefore, while equality is still mediated by coercions, one never needs to mediate between two coercions. Although XTT (and its forebear observational type theory [AMS07]) are not the end of the story [SAG19, Section 2.3], we are optimistic about the future of type-theoretic equality connectives.

In closing, while the discourse around dependent type theory often centers on its relative merits as a foundation of mathematics, the author finds it difficult to nominate any one type theory for consideration. Instead, we suggest that type theories are more akin to programming languages—certainly, some are more considered than others, but there is no reason why one language should be ideal for all tasks. As Voevodsky [Voe14] suggested, we should focus on practicality, not completeness, and rely on our experience gained to inform our future type-theoretic endeavors.

A

Computational-style rules

This appendix proposes rules for Cartesian cubical computational type theory. As discussed in Section 2.4, *computational type theories* such as Nuprl [Con+85] and Idealized Nuprl (Figure 2.4) are deductive systems for establishing judgments in a computational semantics, characterized by their inclusion of equality reflection and untyped computation rules.

Readers can experiment with Cartesian cubical computational type theory in the **RedPRL** proof assistant [Ang+18].[1] Notably, **RedPRL** derives less benefit from untyped computation rules than Nuprl: cubically-unstable evaluation is sound only in an empty context (Lemma 4.16), and path endpoint projections (Rule 4.50) require typing information.

The judgments below were defined in Section 4.3, with the exception of $\Downarrow$, $\longmapsto_{\square}$, and $\psi : \Psi' \longrightarrow \Psi$, defined in Section 4.1. Contexts Ψ and Ξ are unordered, and equations in Ξ are symmetric; κ ranges over $\{\mathsf{pre}, \mathsf{Kan}\}$ and ε ranges over $\{0, 1\}$. Each rule is annotated with the lemma containing its proof, and all judgments are relative to the cubical type system $\tau_\omega^{\mathsf{pre}}$ defined in Section 4.2. We suppress ambient hypotheses in these rules for clarity; every judgment implicitly contains additional hypotheses Γ except when specifically noted. Metavariables marked with † abbreviate large terms defined in Figure 4.2.

Structural rules

$$\frac{A \in \mathcal{U}_i^\kappa\ [\Psi]}{a : A \gg a \in A\ [\Psi]}\ (4.23) \qquad \frac{M \doteq M' \in B\ [\Psi] \qquad A \in \mathcal{U}_i^\kappa\ [\Psi]}{a : A \gg M \doteq M' \in B\ [\Psi]}\ (4.24)$$

$$\frac{M \doteq M' \in A\ [\Psi] \qquad \psi : \Psi' \longrightarrow \Psi}{M\psi \doteq M'\psi \in A\psi\ [\Psi']}\ (4.13) \qquad \frac{M \doteq M' \in A\ [\Psi] \qquad A \doteq A' \in \mathcal{U}_i^\kappa\ [\Psi]}{M \doteq M' \in A'\ [\Psi]}\ (4.12)$$

$$\frac{a : A \gg M \doteq M' \in B\ [\Psi] \qquad N \doteq N' \in A\ [\Psi]}{M[N/a] \doteq M'[N'/a] \in B[N/a]\ [\Psi]}\ (4.25)$$

$$\frac{M' \in A\ [\Psi] \qquad M \longmapsto_{\square}^{*} M'}{M \doteq M' \in A\ [\Psi]}\ (4.32) \qquad \frac{\cdot \gg M \in A\ [\Psi] \qquad M \Downarrow M_0}{\cdot \gg M \doteq M_0 \in A\ [\Psi]}\ (4.16)$$

[1] There are some discrepancies between this appendix and **RedPRL**: the latter interleaves term and interval variables in a single context, and uses sequent calculus–style left rules rather than elimination rules.

$$\frac{M \doteq M' \in A\,[\Psi]}{M \doteq M' \in A\,[\Psi \mid \cdot]}\;(4.27) \qquad \frac{M \doteq M' \in A\,[\Psi \mid \Xi]}{M \doteq M' \in A\,[\Psi \mid \Xi, r = r]}\;(4.27)$$

$$\frac{}{M \doteq M' \in A\,[\Psi \mid \Xi, 0 = 1]}\;(4.27) \qquad \frac{M\langle r/x\rangle \doteq M'\langle r/x\rangle \in A\langle r/x\rangle\,[\Psi \mid \Xi\langle r/x\rangle]}{M \doteq M' \in A\,[\Psi, x \mid \Xi, x = r]}\;(4.27)$$

Kan operations

$$\frac{(r = 0), (r = 1) \in \overrightarrow{\xi_i}}{\overrightarrow{\xi_i}\ \mathsf{valid}}\;(4.28) \qquad \frac{(r = r) \in \overrightarrow{\xi_i}}{\overrightarrow{\xi_i}\ \mathsf{valid}}\;(4.28)$$

$$\frac{A \in \mathcal{U}_k^{\mathsf{Kan}}\,[\Psi, x] \qquad M \in A\langle r/x\rangle\,[\Psi]}{\begin{array}{c}\mathsf{coe}_{x.A}^{r \rightsquigarrow r'}(M) \in A\langle r'/x\rangle\,[\Psi] \\ \doteq M \quad \text{when } r = r'\end{array}}\;(4.29)$$

$$\frac{\begin{array}{ll} & \overrightarrow{\xi_i}\ \mathsf{valid} \\ & A \in \mathcal{U}_k^{\mathsf{Kan}}\,[\Psi] \\ & M \in A\,[\Psi] \\ (\forall i, j) & N_i \doteq N_j \in A\,[\Psi, y \mid \xi_i, \xi_j] \\ (\forall i) & N_i\langle r/y\rangle \doteq M \in A\,[\Psi \mid \xi_i]\end{array}}{\begin{array}{l}\mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\,[\Psi] \\ \quad \doteq \begin{cases} M & \text{when } r = r' \\ N_i\langle r'/y\rangle & \text{when } \xi_i \end{cases}\end{array}}\;(4.29)$$

$$\frac{\begin{array}{ll} & \overrightarrow{\xi_i}\ \mathsf{valid} \\ & A \in \mathcal{U}_k^{\mathsf{Kan}}\,[\Psi, y] \\ & M \in A\langle r/y\rangle\,[\Psi] \\ (\forall i, j) & N_i \doteq N_j \in A\,[\Psi, y \mid \xi_i, \xi_j] \\ (\forall i) & N_i\langle r/y\rangle \doteq M \in A\langle r/y\rangle\,[\Psi \mid \xi_i]\end{array}}{\begin{array}{l}\mathsf{com}_{y.A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\langle r'/y\rangle\,[\Psi] \\ \quad \doteq \begin{cases} M & \text{when } r = r' \\ N_i\langle r'/y\rangle & \text{when } \xi_i \end{cases}\end{array}}\;(4.33)$$

$$\frac{\begin{array}{ll} & A \in \mathcal{U}_k^{\mathsf{Kan}}\,[\Psi] \\ & M \in A\,[\Psi] \\ (\forall i, j) & N_i \doteq N_j \in A\,[\Psi, y \mid \xi_i, \xi_j] \\ (\forall i) & N_i\langle r/y\rangle \doteq M \in A\,[\Psi \mid \xi_i]\end{array}}{\begin{array}{l}\mathsf{ghcom}_A^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\,[\Psi] \\ \quad \doteq \begin{cases} M & \text{when } r = r' \\ N_i\langle r'/y\rangle & \text{when } \xi_i \end{cases}\end{array}}\;(4.34)$$

$$\frac{\begin{array}{ll} & A \in \mathcal{U}_k^{\mathsf{Kan}}\,[\Psi, y] \\ & M \in A\langle r/y\rangle\,[\Psi] \\ (\forall i, j) & N_i \doteq N_j \in A\,[\Psi, y \mid \xi_i, \xi_j] \\ (\forall i) & N_i\langle r/y\rangle \doteq M \in A\langle r/y\rangle\,[\Psi \mid \xi_i]\end{array}}{\begin{array}{l}\mathsf{gcom}_{y.A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \in A\langle r'/y\rangle\,[\Psi] \\ \quad \doteq \begin{cases} M & \text{when } r = r' \\ N_i\langle r'/y\rangle & \text{when } \xi_i \end{cases}\end{array}}\;(4.35)$$

Other rules

$$\frac{A \in \mathcal{U}_i^{\kappa}\ [\Psi] \qquad a : A \gg B \in \mathcal{U}_i^{\kappa}\ [\Psi]}{(a{:}A) \to B \in \mathcal{U}_i^{\kappa}\ [\Psi]}\ (4.36) \qquad \frac{a : A \gg M \in B\ [\Psi]}{\lambda a.M \in (a{:}A) \to B\ [\Psi]}\ (4.37)$$

$$\frac{M \in (a{:}A) \to B\ [\Psi] \qquad N \in A\ [\Psi]}{M\ N \in B[N/a]\ [\Psi]}\ (4.39) \qquad \frac{M \in (a{:}A) \to B\ [\Psi]}{M \doteq \lambda a.M\ a \in (a{:}A) \to B\ [\Psi]}\ (4.40)$$

$$\frac{A \in \mathcal{U}_i^{\kappa}\ [\Psi] \qquad a : A \gg B \in \mathcal{U}_i^{\kappa}\ [\Psi]}{(a{:}A) \times B \in \mathcal{U}_i^{\kappa}\ [\Psi]}\ (4.43) \qquad \frac{\begin{array}{c} a : A \gg B \in \mathcal{U}_i^{\kappa}\ [\Psi] \\ M \in A\ [\Psi] \qquad N \in B[M/a]\ [\Psi] \end{array}}{\langle M, N\rangle \in (a{:}A) \times B\ [\Psi]}\ (4.44)$$

$$\frac{P \in (a{:}A) \times B\ [\Psi]}{\mathsf{fst}(P) \in A\ [\Psi]}\ (4.45) \qquad \frac{P \in (a{:}A) \times B\ [\Psi]}{\mathsf{snd}(P) \in B[\mathsf{fst}(P)/a]\ [\Psi]}\ (4.45)$$

$$\frac{P \in (a{:}A) \times B\ [\Psi]}{P \doteq \langle \mathsf{fst}(P), \mathsf{snd}(P)\rangle \in (a{:}A) \times B\ [\Psi]}\ (4.46)$$

$$\frac{\begin{array}{c} A \in \mathcal{U}_i^{\kappa}\ [\Psi, x] \\ P_0 \in A\langle 0/x\rangle\ [\Psi] \\ P_1 \in A\langle 1/x\rangle\ [\Psi] \end{array}}{\mathsf{Path}_{x.A}(P_0, P_1) \in \mathcal{U}_i^{\kappa}\ [\Psi]}\ (4.48) \qquad \frac{\begin{array}{c} M \in A\ [\Psi, x] \\ M\langle 0/x\rangle \doteq P_0 \in A\langle 0/x\rangle\ [\Psi] \\ M\langle 1/x\rangle \doteq P_1 \in A\langle 1/x\rangle\ [\Psi] \end{array}}{\langle x\rangle M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]}\ (4.49)$$

$$\frac{M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]}{M@r \in A\langle r/x\rangle\ [\Psi]}\ (4.50) \qquad \frac{M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]}{M@\varepsilon \doteq P_\varepsilon \in A\langle \varepsilon/x\rangle\ [\Psi]}\ (4.50)$$

$$\frac{M \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]}{M \doteq \langle x\rangle(M@x) \in \mathsf{Path}_{x.A}(P_0, P_1)\ [\Psi]}\ (4.51)$$

$$\frac{A \in \mathcal{U}_i^{\mathsf{pre}}\ [\Psi] \qquad M \in A\ [\Psi] \qquad N \in A\ [\Psi]}{\mathsf{Eq}_A(M, N) \in \mathcal{U}_i^{\mathsf{pre}}\ [\Psi]}\ (4.53) \qquad \frac{M \doteq N \in A\ [\Psi]}{\star \in \mathsf{Eq}_A(M, N)\ [\Psi]}\ (4.54)$$

$$\frac{E \in \mathsf{Eq}_A(M, N)\ [\Psi]}{M \doteq N \in A\ [\Psi]}\ (4.55) \qquad \frac{E \in \mathsf{Eq}_A(M, N)\ [\Psi]}{E \doteq \star \in \mathsf{Eq}_A(M, N)\ [\Psi]}\ (4.56)$$

$$\frac{}{\mathsf{void} \in \mathcal{U}^{\kappa}_{i}\ [\Psi]}\ (4.57) \qquad\qquad \frac{M \in \mathsf{void}\ [\Psi]}{N \in A\ [\Psi]}\ (4.58)$$

$$\frac{}{\mathsf{bool} \in \mathcal{U}^{\kappa}_{i}\ [\Psi]}\ (4.60) \qquad \frac{}{\mathsf{true} \in \mathsf{bool}\ [\Psi]}\ (4.61) \qquad \frac{}{\mathsf{false} \in \mathsf{bool}\ [\Psi]}\ (4.61)$$

$$\frac{\begin{array}{c} b : \mathsf{bool} \gg A \in \mathcal{U}^{\kappa}_{i}\ [\Psi] \\ M \in \mathsf{bool}\ [\Psi] \qquad T \in A[\mathsf{true}/b]\ [\Psi] \qquad F \in A[\mathsf{false}/b]\ [\Psi] \end{array}}{\mathsf{if}(M; T, F) \in A[M/b]\ [\Psi]}\ (4.62)$$

$$\frac{}{\mathsf{nat} \in \mathcal{U}^{\kappa}_{i}\ [\Psi]}\ (4.65) \qquad \frac{}{\mathsf{z} \in \mathsf{nat}\ [\Psi]}\ (4.66) \qquad \frac{M \in \mathsf{nat}\ [\Psi]}{\mathsf{s}(M) \in \mathsf{nat}\ [\Psi]}\ (4.66)$$

$$\frac{\begin{array}{c} n : \mathsf{nat} \gg A \in \mathcal{U}^{\kappa}_{i}\ [\Psi] \\ M \in \mathsf{nat}\ [\Psi] \qquad Z \in A[\mathsf{z}/n]\ [\Psi] \qquad n : \mathsf{nat}, a : A \gg S \in A[\mathsf{s}(n)/n]\ [\Psi] \end{array}}{\mathsf{natrec}(M; Z, n.a.S) \in A[M/n]\ [\Psi]}\ (4.67)$$

$$\frac{}{\mathbb{S}^1 \in \mathcal{U}^{\kappa}_{i}\ [\Psi]}\ (4.70) \qquad \frac{}{\mathsf{base} \in \mathbb{S}^1\ [\Psi]}\ (4.71) \qquad \frac{}{\mathsf{loop}_r \in \mathbb{S}^1\ [\Psi]}\ (4.71)$$

$$\frac{\begin{array}{rl} & c : \mathbb{S}^1 \gg A \in \mathcal{U}^{\mathsf{Kan}}_{i}\ [\Psi] \\ & M \in \mathbb{S}^1\ [\Psi] \\ & P \in A[\mathsf{base}/c]\ [\Psi] \\ & L \in A[\mathsf{loop}_x/c]\ [\Psi, x] \\ (\forall \varepsilon) & L\langle \varepsilon/x \rangle \doteq P \in A[\mathsf{base}/c]\ [\Psi] \end{array}}{\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L) \in A[M/c]\ [\Psi]}\ (4.76)$$

$$\frac{L \in B\ [\Psi, x] \qquad (\forall \varepsilon)\ L\langle \varepsilon/x \rangle \doteq P \in B\langle \varepsilon/x \rangle\ [\Psi]}{\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_r; P, x.L) \doteq L\langle r/x \rangle \in B\langle r/x \rangle\ [\Psi]}\ (4.73)$$

$$\frac{\begin{array}{rl} & c : \mathbb{S}^1 \gg A \in \mathcal{U}_i^{\mathsf{Kan}}\ [\Psi] \\ & P \in A[\mathsf{base}/c]\ [\Psi] \\ & L \in A[\mathsf{loop}_x/c]\ [\Psi, x] \\ (\forall \varepsilon) & L\langle \varepsilon/x \rangle \doteq P \in A[\mathsf{base}/c]\ [\Psi] \end{array} \qquad \begin{array}{rl} & \overrightarrow{\xi_i}\ \mathsf{valid} \\ & M \in \mathbb{S}^1\ [\Psi] \\ (\forall i, j) & N_i \doteq N_j \in \mathbb{S}^1\ [\Psi, y \mid \xi_i, \xi_j] \\ (\forall i) & N_i\langle r/y \rangle \doteq M \in \mathbb{S}^1\ [\Psi \mid \xi_i] \end{array}}{\begin{array}{c} \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}); P, x.L) \\ \doteq \mathsf{com}^{r \rightsquigarrow r'}_{z.A[\mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow z}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})/c]}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; P, x.L); \overrightarrow{\xi_i \hookrightarrow y.\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(N_i; P, x.L)}) \\ \in A[\mathsf{hcom}_{\mathbb{S}^1}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})/c]\ [\Psi] \end{array}} \tag{4.75}$$

$$\mathsf{isContr}(C) := (c{:}C) \times ((c'{:}C) \to \mathsf{Path}_{\_.C}(c', c))$$

$$\mathsf{Equiv}(A, B) := (f{:}A \to B) \times ((b{:}B) \to \mathsf{isContr}((a{:}A) \times \mathsf{Path}_{\_.B}(f\ a, b)))$$

$$\frac{\begin{array}{c} A \in \mathcal{U}_i^{\kappa}\ [\Psi \mid r = 0] \\ B \in \mathcal{U}_i^{\kappa}\ [\Psi] \\ E \in \mathsf{Equiv}(A, B)\ [\Psi \mid r = 0] \end{array}}{\mathsf{V}_r(A, B, E) \in \mathcal{U}_i^{\kappa}\ [\Psi]} \tag{4.78}$$

$$\frac{\begin{array}{c} M \in A\ [\Psi \mid r = 0] \\ N \in B\ [\Psi] \\ E \in \mathsf{Equiv}(A, B)\ [\Psi \mid r = 0] \\ \mathsf{fst}(E)\ M \doteq N \in B\ [\Psi \mid r = 0] \end{array}}{\mathsf{Vin}_r(M, N) \in \mathsf{V}_r(A, B, E)\ [\Psi]} \tag{4.79}$$

$$\frac{M \in \mathsf{V}_r(A, B, E)\ [\Psi]}{\mathsf{Vproj}_r(M, \mathsf{fst}(E)) \in B\ [\Psi]} \tag{4.80}$$

$$\frac{N \in B\ [\Psi] \qquad F \in A \to B\ [\Psi \mid r = 0] \qquad \begin{array}{c} M \in A\ [\Psi \mid r = 0] \\ F\ M \doteq N \in B\ [\Psi \mid r = 0] \end{array}}{\mathsf{Vproj}_r(\mathsf{Vin}_r(M, N), F) \doteq N \in B\ [\Psi]} \tag{4.81}$$

$$\frac{N \in \mathsf{V}_r(A, B, E)\ [\Psi]}{\mathsf{Vin}_r(N, \mathsf{Vproj}_r(N, \mathsf{fst}(E))) \doteq N \in \mathsf{V}_r(A, B, E)\ [\Psi]} \tag{4.82}$$

$$\frac{\begin{array}{rl} & \overrightarrow{\xi_i}\ \mathsf{valid} \\ & M \in \mathsf{V}_x(A, B, E)\ [\Psi] \\ (\forall i, j) & N_i \doteq N_j \in \mathsf{V}_x(A, B, E)\ [\Psi, y \mid \xi_i, \xi_j] \\ (\forall i) & N_i\langle r/y \rangle \doteq M \in \mathsf{V}_x(A, B, E)\ [\Psi \mid \xi_i] \end{array}}{\begin{array}{c} \mathsf{hcom}_{\mathsf{V}_x(A,B,E)}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \\ \doteq \mathsf{Vin}_x(O^{\dagger}\langle r'/y \rangle, \mathsf{hcom}_B^{r \rightsquigarrow r'}(\mathsf{Vproj}_x(M, \mathsf{fst}(E)); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{Vproj}_x(N_i, \mathsf{fst}(E))}, \overrightarrow{T}^{\dagger})) \\ \in \mathsf{V}_x(A, B, E)\ [\Psi] \end{array}} \tag{4.83}$$

$$\frac{\begin{array}{c} A \in \mathcal{U}_i^{\mathsf{Kan}}\ [\Psi, x, y \mid x = 0] \qquad B \in \mathcal{U}_i^{\mathsf{Kan}}\ [\Psi, x, y] \\ E \in \mathsf{Equiv}(A, B)\ [\Psi, x, y \mid x = 0] \qquad M \in \mathsf{V}_x(A\langle r/y\rangle, B\langle r/y\rangle, E\langle r/y\rangle)\ [\Psi, x] \end{array}}{\begin{array}{c} \mathsf{coe}^{r \rightsquigarrow r'}_{y.\mathsf{V}_x(A,B,E)}(M) \doteq \mathsf{Vin}_x(\mathsf{coe}^{r \rightsquigarrow r'}_{y.A}(M), \mathsf{com}^{r \rightsquigarrow r'}_{y.B}(\mathsf{Vproj}_x(M, \mathsf{fst}(E\langle r/y\rangle)); \overrightarrow{T}^{\dagger})) \\ \in \mathsf{V}_x(A\langle r'/y\rangle, B\langle r'/y\rangle, E\langle r'/y\rangle)\ [\Psi, x] \end{array}} \ (4.84)$$

$$\frac{i < j}{\mathcal{U}_i^{\kappa} \in \mathcal{U}_j^{\kappa'}\ [\Psi]} \ (4.103) \qquad \frac{A \in \mathcal{U}_i^{\kappa}\ [\Psi]}{A \in \mathcal{U}_{i+1}^{\kappa}\ [\Psi]} \ (4.99) \qquad \frac{A \in \mathcal{U}_i^{\mathsf{Kan}}\ [\Psi]}{A \in \mathcal{U}_i^{\mathsf{pre}}\ [\Psi]} \ (4.99)$$

$$\frac{\begin{array}{llll} & \overrightarrow{\xi_i}\ \mathsf{valid} & & \\ & A \in \mathcal{U}_k^{\mathsf{Kan}}\ [\Psi] & & M \in A\ [\Psi] \\ (\forall i, j) & B_i \doteq B_j \in \mathcal{U}_k^{\mathsf{Kan}}\ [\Psi, y \mid \xi_i, \xi_j] & (\forall i, j) & N_i \doteq N_j \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i, \xi_j] \\ (\forall i) & B_i\langle r/y\rangle \doteq A \in \mathcal{U}_k^{\mathsf{Kan}}\ [\Psi \mid \xi_i] & (\forall i) & \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i) \doteq M \in A\ [\Psi \mid \xi_i] \end{array}}{\begin{array}{c} \mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi] \\ \doteq N_i \quad \text{when } \xi_i \end{array}} \ (4.91)$$

$$\frac{M \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi]}{\begin{array}{c} \mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) \in A\ [\Psi] \\ \doteq \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(M) \quad \text{when } \xi_i \end{array}} \ (4.92)$$

$$\frac{\begin{array}{llll} & \overrightarrow{\xi_i}\ \mathsf{valid} & & \\ & A \in \mathcal{U}_k^{\mathsf{Kan}}\ [\Psi] & & M \in A\ [\Psi] \\ (\forall i, j) & B_i \doteq B_j \in \mathcal{U}_k^{\mathsf{Kan}}\ [\Psi, y \mid \xi_i, \xi_j] & (\forall i, j) & N_i \doteq N_j \in B_i\langle r'/y\rangle\ [\Psi \mid \xi_i, \xi_j] \\ (\forall i) & B_i\langle r/y\rangle \doteq A \in \mathcal{U}_k^{\mathsf{Kan}}\ [\Psi \mid \xi_i] & (\forall i) & \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i) \doteq M \in A\ [\Psi \mid \xi_i] \end{array}}{\mathsf{cap}^{r \leftsquigarrow r'}(\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}); \overrightarrow{\xi_i \hookrightarrow y.B_i}) \doteq M \in A\ [\Psi]} \ (4.93)$$

$$\frac{M \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi]}{\mathsf{box}^{r \rightsquigarrow r'}(\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}); \overrightarrow{\xi_i \hookrightarrow M}) \doteq M \in \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathcal{U}_k^{\mathsf{Kan}}}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [\Psi]} \ (4.94)$$

$$\frac{\begin{array}{ll} & \overrightarrow{r_i = r_i'}\ \mathsf{valid} \\ & M \in \mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{s_j = s_j' \hookrightarrow z.B_j})\ [\Psi] \\ (\forall i,j) & N_i \doteq N_j \in \mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{s_j = s_j' \hookrightarrow z.B_j})\ [\Psi, y \mid r_i = r_i', r_j = r_j'] \\ (\forall i) & N_i\langle r/y\rangle \doteq M \in \mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{s_j = s_j' \hookrightarrow z.B_j})\ [\Psi \mid r_i = r_i'] \end{array}}{\begin{array}{l} \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{s_j=s_j'\hookrightarrow z.B_j})}(M;\overrightarrow{r_i = r_i' \hookrightarrow y.N_i}) \doteq \mathsf{box}^{s\rightsquigarrow s'}(Q^{\dagger};\overrightarrow{s_j = s_j' \hookrightarrow P_j^{\dagger}\langle r'/y\rangle}) \\ \qquad \in \mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{s_j = s_j' \hookrightarrow z.B_j})\ [\Psi] \end{array}} \tag{4.95}$$

$$\frac{\begin{array}{ll} & \overrightarrow{\xi_i}\ \mathsf{valid} \\ & A \in \mathcal{U}^{\mathsf{Kan}}_k\ [\Psi, x] \\ (\forall i,j) & B_i \doteq B_j \in \mathcal{U}^{\mathsf{Kan}}_k\ [\Psi, x, z \mid \xi_i, \xi_j] \\ (\forall i) & B_i\langle r/z\rangle \doteq A \in \mathcal{U}^{\mathsf{Kan}}_k\ [\Psi, x \mid \xi_i] \\ & M \in \mathsf{hcom}^{s\langle r/x\rangle\rightsquigarrow s'\langle r/x\rangle}_{\mathcal{U}^{\mathsf{Kan}}_k}(A\langle r/x\rangle;\overrightarrow{\xi_i\langle r/x\rangle \hookrightarrow z.B_i\langle r/x\rangle})\ [\Psi] \end{array}}{\begin{array}{l} \mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{hcom}^{s\rightsquigarrow s'}_{\mathcal{U}^{\mathsf{Kan}}_k}(A;\overrightarrow{\xi_i\hookrightarrow z.B_i})}(M) \doteq \mathsf{box}^{s\langle r'/x\rangle\rightsquigarrow s'\langle r'/x\rangle}(H^{\dagger};\overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow Q_i^{\dagger}\langle s'\langle r'/x\rangle/z\rangle}) \\ \qquad \in \mathsf{hcom}^{s\langle r'/x\rangle\rightsquigarrow s'\langle r'/x\rangle}_{\mathcal{U}^{\mathsf{Kan}}_k}(A\langle r'/x\rangle;\overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow z.B_i\langle r'/x\rangle})\ [\Psi] \end{array}} \tag{4.96}$$

B

Intensional-style rules

This appendix proposes rules for *intensional* Cartesian cubical type theory. As discussed in Section 2.4, intensional type theories, such as those implemented by Agda [Agda] and Coq [Coq], are deductive systems whose equality judgments (and therefore, typing judgments) admit decision procedures.[1] We conjecture but have not proven decidability or even canonicity for the system of rules below. Readers can experiment with similar intensional Cartesian cubical type theories in the **redtt** proof assistant [Red18], and can consult Angiuli et al. [Ang+19] for the denotational semantics of a related deductive system.

We intend these rules as a general reference for Cartesian cubical type theory, so we have omitted pretypes (and hence, strict equality types) and our validity condition on composition shapes (and hence, generalized composition). Conversely, we have added level annotations to our typehood judgment [Coq19] and type annotations to elimination forms (and an eliminator $\mathsf{abort}(-)$ for void), ensuring that our rules are generalized algebraic and thus admit a category of models [Car86].

In order to admit decidable equality judgments, we have eliminated several instances of equality reflection. First, $\mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{nat}}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})$ must compute by recursion on M, N_i rather than generically to M; otherwise, whenever $P : \mathsf{Path}_{\_.\mathsf{nat}}(P_0, P_1)$, the composition $\mathsf{hcom}^{0\rightsquigarrow 1}_{\mathsf{nat}}(P_0; 0 = 0 \hookrightarrow y.P@y)$ equals both P_0 (by the rule for $\mathsf{hcom}_{\mathsf{nat}}$) and P_1 (by $0 = 0$), implying $P_0 = P_1$ judgmentally. Secondly, we omit the equality reflection rule itself. In the future, one might consider alternative formulations of strict equality compatible with decidable judgmental equality; possibilities include observational type theory [AMS07], XTT [SAG19], or even simply postulating an axiom for uniqueness of identity proofs (at the expense of canonicity) [ACK16].

Judgments

The judgments of intensional Cartesian cubical type theory range over the syntactic sorts presented in Figure B.1. Following XTT [SAG19], we gather interval variables and constraints in a single context Ψ, now representing an object of the *augmented Cartesian cube category* (which adjoins to $\square$ an initial object interpreting inconsistent contexts).

1. $\Psi\ \mathsf{cube}_+$ when Ψ is an augmented Cartesian cube.

2. $\Psi \mid r : \mathbb{I}$, presupposing $\Psi\ \mathsf{cube}_+$, when r is an interval term in Ψ.

[1] Historically, intensional equality judgments were generated by β rules and decided by untyped means; modern decision procedures are type-sensitive and admit various η rules [ACD07].

$$
\begin{array}{lrcl}
\textit{Augmented cubes} & \Psi & := & \cdot \mid \Psi, x \mid \Psi, \xi \\
\textit{Contexts} & \Gamma & := & \cdot \mid \Gamma, a : A \\
\textit{Interval terms} & r, s & := & x \mid \varepsilon \\
\textit{Interval constants} & \varepsilon & := & 0 \mid 1 \\
\textit{Interval equations} & \xi & := & r = r' \\
\textit{Universe levels} & i, j & := & n \in \mathbb{N} \\
\textit{Types} & A, B & := & (a{:}A) \to B \mid (a{:}A) \times B \mid \mathsf{Path}_{x.A}(M, N) \mid \mathsf{void} \mid \mathsf{bool} \mid \mathsf{nat} \\
& & \mid & \mathbb{S}^1 \mid \mathsf{V}_r(A, B, M) \mid \mathsf{Type}_i \\
\textit{Terms} & M, N & := & \lambda a.M \mid \mathsf{app}_{a:A.B}(M, N) \mid \langle M, N \rangle \mid \mathsf{fst}_{a:A.B}(M) \mid \mathsf{snd}_{a:A.B}(M) \\
& & \mid & \langle x \rangle M \mid \mathsf{iapp}_{x.A}(M, r) \mid \mathsf{abort}_A(M) \mid \mathsf{true} \mid \mathsf{false} \\
& & \mid & \mathsf{if}_{b.A}(M; N_0, N_1) \mid \mathsf{z} \mid \mathsf{s}(M) \mid \mathsf{natrec}_{n.A}(M; N_0, n.a.N_1) \\
& & \mid & \mathsf{base} \mid \mathsf{loop}_r \mid \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; N_0, x.N_1) \mid \mathsf{Vin}_r(M, N) \\
& & \mid & \mathsf{Vproj}_r^{A,B}(M, N) \mid \mathsf{coe}_{x.A}^{r \rightsquigarrow r'}(M) \mid \mathsf{hcom}_A^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \\
& & \mid & \mathsf{com}_{y.A}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \mid \mathsf{box}_{A, \overrightarrow{y.B_i}}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) \\
& & \mid & \mathsf{cap}_A^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i})
\end{array}
$$

Figure B.1: Grammar of intensional Cartesian cubical type theory.

3. $\Psi \mid r = r' : \mathbb{I}$, presupposing $\Psi \mid r, r' : \mathbb{I}$, when r, r' are equal interval terms in Ψ.

4. $\Psi \mid \Gamma$ ctx, presupposing Ψ cube$_+$, when Γ is a context in Ψ.

5. $\Psi \mid \Gamma \vdash A\ \mathsf{type}_i$, presupposing $\Psi \mid \Gamma$ ctx, when A is a type at level i in Γ.

6. $\Psi \mid \Gamma \vdash A = B\ \mathsf{type}_i$, presupposing $\Psi \mid \Gamma \vdash A, B\ \mathsf{type}_i$, when A, B are equal types.

7. $\Psi \mid \Gamma \vdash M : A$, presupposing $\Psi \mid \Gamma \vdash A\ \mathsf{type}_i$, when M is an element of A in Γ.

8. $\Psi \mid \Gamma \vdash M = N : A$, presupposing $\Psi \mid \Gamma \vdash M, N : A$, when M, N are equal elements.

In the following rules, we omit premises to equational rules, most type and term annotations (writing $M\ N$ for $\mathsf{app}_{a:A.B}(M, N)$, $M@r$ for $\mathsf{iapp}_{x.A}(M, r)$, *et cetera*), and all congruence rules. Moreover, we adopt the following notations:

1. $\Psi \mid \Gamma \vdash \mathcal{J}$ abbreviates any typehood, membership, or equality judgment.

2. $\Psi \mid \Gamma \vdash A\ \mathsf{type}_i\ [\overrightarrow{\xi_i \hookrightarrow B_i}]$ abbreviates $\Psi \mid \Gamma \vdash A\ \mathsf{type}_i$ and $\overrightarrow{\Psi, \xi_i \mid \Gamma \vdash A = B\ \mathsf{type}_i}$.

3. $\Psi \mid \Gamma \vdash M : A\ [\overrightarrow{\xi_i \hookrightarrow N_i}]$ abbreviates $\Psi \mid \Gamma \vdash M : A$ and $\overrightarrow{\Psi, \xi_i \mid \Gamma \vdash M = N_i : A}$.

Structural rules

$$\frac{}{\cdot\ \mathsf{cube}_+} \qquad \frac{\Psi\ \mathsf{cube}_+}{\Psi, x\ \mathsf{cube}_+} \qquad \frac{\Psi\ \mathsf{cube}_+ \qquad \Psi \mid r, r' : \mathbb{I}}{\Psi, r = r'\ \mathsf{cube}_+}$$

$$\frac{}{\Psi \mid 0 : \mathbb{I}} \qquad \frac{}{\Psi \mid 1 : \mathbb{I}} \qquad \frac{\Psi \ni x}{\Psi \mid x : \mathbb{I}} \qquad \frac{\Psi \ni r = r'}{\Psi \mid r = r' : \mathbb{I}}$$

$$\frac{\Psi\ \mathsf{cube}_+}{\Psi \mid \cdot\ \mathsf{ctx}} \qquad \frac{\Psi \mid \Gamma\ \mathsf{ctx} \qquad \Psi \mid \Gamma \vdash A\ \mathsf{type}_i}{\Psi \mid \Gamma, a : A\ \mathsf{ctx}} \qquad \frac{\Gamma \ni a : A}{\Psi \mid \Gamma \vdash a : A}$$

$$\frac{\Psi \mid 0 = 1 : \mathbb{I}}{\Psi \mid \Gamma \vdash \mathcal{J}} \qquad \frac{\Psi \mid \Gamma \vdash A = A'\ \mathsf{type}_i \qquad \Psi \mid \Gamma \vdash M : A}{\Psi \mid \Gamma \vdash M : A'}$$

$$\frac{\Psi \mid \Gamma \vdash A\ \mathsf{type}_i \qquad i < j}{\Psi \mid \Gamma \vdash A\ \mathsf{type}_j} \qquad \frac{\Psi \mid \Gamma \vdash A\ \mathsf{type}_i}{\Psi \mid \Gamma \vdash A : \mathsf{Type}_i} \qquad \frac{\Psi \mid \Gamma \vdash A : \mathsf{Type}_i}{\Psi \mid \Gamma \vdash A\ \mathsf{type}_i}$$

$$\frac{\Psi \mid \Gamma \vdash A = A'\ \mathsf{type}_i}{\Psi \mid \Gamma \vdash A = A' : \mathsf{Type}_i} \qquad \frac{\Psi \mid \Gamma \vdash A = A' : \mathsf{Type}_i}{\Psi \mid \Gamma \vdash A = A'\ \mathsf{type}_i}$$

Kan operations

$$\frac{\Psi \mid r, r' : \mathbb{I} \qquad \Psi, x \mid \Gamma \vdash A\ \mathsf{type}_i \qquad \Psi \mid \Gamma \vdash M : A\langle r/x\rangle}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r\rightsquigarrow r'}_{x.A}(M) : A\langle r'/x\rangle\ [r = r' \hookrightarrow M]}$$

$$\frac{\Psi \mid r, r' : \mathbb{I} \qquad \Psi \mid \Gamma \vdash M : A \qquad \overrightarrow{\Psi, \xi_i, y \mid \Gamma \vdash N_i : A\ [y = r \hookrightarrow M, \overrightarrow{\xi_j \hookrightarrow N_j}]}}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r\rightsquigarrow r'}_{A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) : A\ [r = r' \hookrightarrow M, \overrightarrow{\xi_i \hookrightarrow N_i\langle r'/y\rangle}]}$$

$$\frac{\begin{array}{c}\Psi \mid r, r' : \mathbb{I} \qquad \Psi, y \mid \Gamma \vdash A\ \mathsf{type}_k \\ \Psi \mid \Gamma \vdash M : A\langle r/y\rangle \qquad \overrightarrow{\Psi, \xi_i, y \mid \Gamma \vdash N_i : A\ [y = r \hookrightarrow M, \overrightarrow{\xi_j \hookrightarrow N_j}]}\end{array}}{\Psi \mid \Gamma \vdash \mathsf{com}^{r\rightsquigarrow r'}_{y.A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) : A\langle r'/y\rangle\ [r = r' \hookrightarrow M, \overrightarrow{\xi_i \hookrightarrow N_i\langle r'/y\rangle}]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{com}^{r\rightsquigarrow r'}_{y.A}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) = \mathsf{hcom}^{r\rightsquigarrow r'}_{A\langle r'/y\rangle}(\mathsf{coe}^{r\rightsquigarrow r'}_{y.A}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{coe}^{y\rightsquigarrow r'}_{y.A}(N_i)}) : A\langle r'/y\rangle}$$

Other rules

$$\frac{\Psi \mid \Gamma \vdash A\ \mathsf{type}_i \qquad \Psi \mid \Gamma, a : A \vdash B\ \mathsf{type}_i}{\Psi \mid \Gamma \vdash (a{:}A) \to B\ \mathsf{type}_i} \qquad \frac{\Psi \mid \Gamma, a : A \vdash M : B}{\Psi \mid \Gamma \vdash \lambda a.M : (a{:}A) \to B}$$

$$\frac{\Psi \mid \Gamma \vdash M : (a{:}A) \to B \qquad \Psi \mid \Gamma \vdash N : A}{\Psi \mid \Gamma \vdash \mathsf{app}_{a:A.B}(M, N) : B[N/a]} \qquad \frac{}{\Psi \mid \Gamma \vdash (\lambda a.M)\ N = M[N/a] : B[N/a]}$$

$$\frac{}{\Psi \mid \Gamma \vdash M = \lambda a.M\ a : (a{:}A) \to B}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r \rightsquigarrow r'}_{x.(a:A)\to B}(M) = \lambda a.\mathsf{coe}^{r \rightsquigarrow r'}_{x.B[\mathsf{coe}^{r' \rightsquigarrow x}_{x.A}(a)/a]}(M\ \mathsf{coe}^{r' \rightsquigarrow r}_{x.A}(a)) : (a{:}A\langle r'/x\rangle) \to B\langle r'/x\rangle}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r \rightsquigarrow r'}_{(a:A)\to B}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) = \lambda a.\mathsf{hcom}^{r \rightsquigarrow r'}_{B}(M\ a; \overrightarrow{\xi_i \hookrightarrow y.N_i\ a}) : (a{:}A) \to B}$$

$$\frac{\begin{array}{c}\Psi \mid \Gamma \vdash A\ \mathsf{type}_i \\ \Psi \mid \Gamma, a : A \vdash B\ \mathsf{type}_i\end{array}}{\Psi \mid \Gamma \vdash (a{:}A) \times B\ \mathsf{type}_i} \qquad \frac{\begin{array}{c}\Psi \mid \Gamma, a : A \vdash B\ \mathsf{type}_i \\ \Psi \mid \Gamma \vdash M : A \\ \Psi \mid \Gamma \vdash N : B[M/a]\end{array}}{\Psi \mid \Gamma \vdash \langle M, N\rangle : (a{:}A) \times B} \qquad \frac{\Psi \mid \Gamma \vdash M : (a{:}A) \times B}{\Psi \mid \Gamma \vdash \mathsf{fst}_{a:A.B}(M) : A}$$

$$\frac{\Psi \mid \Gamma \vdash M : (a{:}A) \times B}{\Psi \mid \Gamma \vdash \mathsf{snd}_{a:A.B}(M) : B[\mathsf{fst}(M)/a]} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{fst}(\langle M, N\rangle) = M : A}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{snd}(\langle M, N\rangle) = N : B[M/a]} \qquad \frac{}{\Psi \mid \Gamma \vdash M = \langle \mathsf{fst}(M), \mathsf{snd}(M)\rangle : (a{:}A) \times B}$$

$$\frac{\widetilde{M_0}[x] := \mathsf{coe}^{r \rightsquigarrow x}_{x.A}(\mathsf{fst}(M))}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r \rightsquigarrow r'}_{x.(a:A)\times B}(M) = \langle \widetilde{M_0}[r'], \mathsf{coe}^{r \rightsquigarrow r'}_{x.B[\widetilde{M_0}[x]/a]}(\mathsf{snd}(M))\rangle : (a{:}A\langle r'/x\rangle) \times B\langle r'/x\rangle}$$

$$\frac{\begin{array}{c}\widetilde{M_0}[z] := \mathsf{hcom}^{r \rightsquigarrow z}_{A}(\mathsf{fst}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{fst}(N_i)}) \\ \widetilde{M_1} := \mathsf{com}^{r \rightsquigarrow r'}_{z.B[\widetilde{M_0}[z]/a]}(\mathsf{snd}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{snd}(N_i)})\end{array}}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r \rightsquigarrow r'}_{(a:A)\times B}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) = \langle \widetilde{M_0}[r'], \widetilde{M_1}\rangle : (a{:}A) \times B}$$

$$\frac{\Psi, x \mid \Gamma \vdash A\ \mathsf{type}_i \qquad \overrightarrow{\Psi, x, x = \varepsilon \mid \Gamma \vdash N_\varepsilon : A}}{\Psi \mid \Gamma \vdash \mathsf{Path}_{x.A}(N_0, N_1)\ \mathsf{type}_i} \qquad \frac{\Psi, x \mid \Gamma \vdash M : A\ [\overrightarrow{x = \varepsilon \hookrightarrow N_\varepsilon}]}{\Psi \mid \Gamma \vdash \langle x\rangle M : \mathsf{Path}_{x.A}(N_0, N_1)}$$

$$\frac{\Psi \mid r : \mathbb{I} \qquad \Psi \mid \Gamma \vdash M : \mathsf{Path}_{x.A}(N_0, N_1)}{\Psi \mid \Gamma \vdash \mathsf{iapp}_{x.A}(M, r) : A\langle r/x\rangle\ [\overrightarrow{r = \varepsilon \hookrightarrow N_\varepsilon}]} \qquad \frac{}{\Psi \mid \Gamma \vdash (\langle x\rangle M)@r = M\langle r/x\rangle : A\langle r/x\rangle}$$

$$\frac{}{\Psi \mid \Gamma \vdash M = \langle x\rangle(M@x) : \mathsf{Path}_{x.A}(N_0, N_1)}$$

$$\frac{\widetilde{M} := \langle x\rangle\mathsf{com}^{r\rightsquigarrow r'}_{y.A}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow y.N_\varepsilon})}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r\rightsquigarrow r'}_{y.\mathsf{Path}_{x.A}(N_0,N_1)}(M) = \widetilde{M} : \mathsf{Path}_{x.A\langle r'/y\rangle}(N_0\langle r'/y\rangle, N_1\langle r'/y\rangle)}$$

$$\frac{\widetilde{M} := \langle x\rangle\mathsf{hcom}^{r\rightsquigarrow r'}_{A}(M@x; \overrightarrow{x = \varepsilon \hookrightarrow \_.N_\varepsilon}, \overrightarrow{\xi_i \hookrightarrow y.Q_i@x})}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{Path}_{x.A}(N_0,N_1)}(M; \overrightarrow{\xi_i \hookrightarrow y.Q_i}) = \widetilde{M} : \mathsf{Path}_{x.A}(N_0, N_1)}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{void}\ \mathsf{type}_i} \qquad \frac{\Psi \mid \Gamma \vdash A\ \mathsf{type}_i \qquad \Psi \mid \Gamma \vdash M : \mathsf{void}}{\Psi \mid \Gamma \vdash \mathsf{abort}_A(M) : A}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{bool}\ \mathsf{type}_i} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{true} : \mathsf{bool}} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{false} : \mathsf{bool}}$$

$$\frac{\begin{array}{c}\Psi \mid \Gamma, b : \mathsf{bool} \vdash A\ \mathsf{type}_i \\ \Psi \mid \Gamma \vdash M : \mathsf{bool} \\ \Psi \mid \Gamma \vdash N_0 : A[\mathsf{true}/b] \\ \Psi \mid \Gamma \vdash N_1 : A[\mathsf{false}/b]\end{array}}{\Psi \mid \Gamma \vdash \mathsf{if}_{b.A}(M; N_0, N_1) : A[M/b]} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{if}(\mathsf{true}; N_0, N_1) = N_0 : A[\mathsf{true}/b]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{if}(\mathsf{false}; N_0, N_1) = N_1 : A[\mathsf{false}/b]} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r\rightsquigarrow r'}_{x.\mathsf{bool}}(M) = M : \mathsf{bool}}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(\mathsf{true}; \overrightarrow{\xi_i \hookrightarrow y.\mathsf{true}}) = \mathsf{true} : \mathsf{bool}}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r\rightsquigarrow r'}_{\mathsf{bool}}(\mathsf{false}; \overrightarrow{\xi_i \hookrightarrow y.\mathsf{false}}) = \mathsf{false} : \mathsf{bool}}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{nat}\ \mathsf{type}_i} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{z} : \mathsf{nat}} \qquad \frac{\Psi \mid \Gamma \vdash M : \mathsf{nat}}{\Psi \mid \Gamma \vdash \mathsf{s}(M) : \mathsf{nat}}$$

$$\frac{\begin{array}{c}\Psi \mid \Gamma, n : \mathsf{nat} \vdash A\ \mathsf{type}_i \\ \Psi \mid \Gamma \vdash M : \mathsf{nat} \qquad \Psi \mid \Gamma \vdash N_0 : A[\mathsf{z}/n] \qquad \Psi \mid \Gamma, n : \mathsf{nat}, a : A \vdash N_1 : A[\mathsf{s}(n)/n]\end{array}}{\Psi \mid \Gamma \vdash \mathsf{natrec}_{n.A}(M; N_0, n.a.N_1) : A[M/n]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{natrec}(\mathsf{z}; N_0, n.a.N_1) = N_0 : A[\mathsf{z}/n]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{natrec}(\mathsf{s}(M); N_0, n.a.N_1) = N_1[M, \mathsf{natrec}(M; N_0, n.a.N_1)/n, a] : A[\mathsf{s}(M)/n]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r \rightsquigarrow r'}_{x.\mathsf{nat}}(M) = M : \mathsf{nat}} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{nat}}(\mathsf{z}; \overrightarrow{\xi_i \hookrightarrow y.\mathsf{z}}) = \mathsf{z} : \mathsf{nat}}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{nat}}(\mathsf{s}(M); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{s}(N_i)}) = \mathsf{s}(\mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{nat}}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i})) : \mathsf{nat}}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathbb{S}^1\ \mathsf{type}_i} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{base} : \mathbb{S}^1} \qquad \frac{\Psi \mid r : \mathbb{I}}{\Psi \mid \Gamma \vdash \mathsf{loop}_r : \mathbb{S}^1\ [\overrightarrow{r = \varepsilon \hookrightarrow \mathsf{base}}]}$$

$$\frac{\begin{array}{c}\Psi \mid \Gamma, c : \mathbb{S}^1 \vdash A\ \mathsf{type}_i \\ \Psi \mid \Gamma \vdash M : \mathbb{S}^1 \qquad \Psi \mid \Gamma \vdash N_0 : A[\mathsf{base}/c] \qquad \Psi, x \mid \Gamma \vdash N_1 : A[\mathsf{loop}_x/c]\ [\overrightarrow{x = \varepsilon \hookrightarrow N_0}]\end{array}}{\Psi \mid \Gamma \vdash \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; N_0, x.N_1) : A[M/c]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r \rightsquigarrow r'}_{x.\mathbb{S}^1}(M) = M : \mathbb{S}^1} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{base}; N_0, x.N_1) = N_0 : A[\mathsf{base}/c]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\mathsf{loop}_r; N_0, x.N_1) = N_1\langle r/x \rangle : A[\mathsf{loop}_r/c]}$$

$$\frac{\begin{array}{c}\widetilde{M}[z] := \mathsf{hcom}^{r \rightsquigarrow z}_{\mathbb{S}^1}(M; \overrightarrow{\xi_i \hookrightarrow y.Q_i}) \\ \widetilde{H} := \mathsf{com}^{r \rightsquigarrow r'}_{z.A[\widetilde{M}[z]/c]}(\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(M; N_0, x.N_1); \overrightarrow{\xi_i \hookrightarrow y.\mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(Q_i; N_0, x.N_1)})\end{array}}{\Psi \mid \Gamma \vdash \mathbb{S}^1\text{-}\mathsf{elim}_{c.A}(\widetilde{M}[r']; N_0, x.N_1) = \widetilde{H} : A[\widetilde{M}[r']/c]}$$

$$\mathsf{isContr}(C) := (c{:}C) \times ((c'{:}C) \to \mathsf{Path}_{\_.C}(c', c))$$

$$\mathsf{Equiv}(A, B) := (f{:}A \to B) \times ((b{:}B) \to \mathsf{isContr}((a{:}A) \times \mathsf{Path}_{\_.B}(f\ a, b)))$$

$$\frac{\Psi, r = 0 \mid \Gamma \vdash A\ \mathsf{type}_i \qquad \Psi \mid \Gamma \vdash B\ \mathsf{type}_i \qquad \Psi, r = 0 \mid \Gamma \vdash E : \mathsf{Equiv}(A, B)}{\Psi \mid \Gamma \vdash \mathsf{V}_r(A, B, E)\ \mathsf{type}_i\ [r = 0 \hookrightarrow A, r = 1 \hookrightarrow B]}$$

$$\frac{\begin{array}{c}\Psi, r = 0 \mid \Gamma \vdash M : A \\ \Psi, r = 0 \mid \Gamma \vdash E : \mathsf{Equiv}(A, B) \qquad \Psi \mid \Gamma \vdash N : B\ [r = 0 \hookrightarrow \mathsf{fst}(E)\ M]\end{array}}{\Psi \mid \Gamma \vdash \mathsf{Vin}_r(M, N) : \mathsf{V}_r(A, B, E)\ [r = 0 \hookrightarrow M, r = 1 \hookrightarrow N]}$$

$$\frac{\Psi \mid \Gamma \vdash M : \mathsf{V}_r(A, B, E)}{\Psi \mid \Gamma \vdash \mathsf{Vproj}_r^{A,B}(M, E) : B\ [r = 0 \hookrightarrow \mathsf{fst}(E)\ M, r = 1 \hookrightarrow M]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{Vproj}_r(\mathsf{Vin}_r(M, N), E) = N : B} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{Vin}_r(N, \mathsf{Vproj}_r(N, E)) = N : \mathsf{V}_r(A, B, E)}$$

$$\frac{\begin{array}{c}\widetilde{N} := \mathsf{coe}_{x.B}^{r \rightsquigarrow r'}(\mathsf{Vproj}_r(M, E\langle r/x\rangle)) \\ \widetilde{F} := (a{:}A\langle r'/x\rangle) \times \mathsf{Path}_{\_.B\langle r'/x\rangle}(\mathsf{fst}(E\langle r'/x\rangle)\ a, \widetilde{N}) \qquad \widetilde{C} := \mathsf{snd}(E\langle r'/x\rangle)\ \widetilde{N} \\ \widetilde{O} := \mathsf{hcom}_{\widetilde{F}}^{1 \rightsquigarrow 0}(\mathsf{fst}(\widetilde{C}); r = 0 \hookrightarrow z.(\mathsf{snd}(\widetilde{C})\ \langle M, \langle\_\rangle(\mathsf{fst}(E\langle 0/x\rangle)\ M)\rangle)@z) \\ \widetilde{P} := \mathsf{hcom}_{B\langle r'/x\rangle}^{1 \rightsquigarrow 0}(\widetilde{N}; r' = 0 \hookrightarrow z.\mathsf{snd}(\widetilde{O})@z, r = r' \hookrightarrow \_.\mathsf{Vproj}_r(M, E\langle r/x\rangle))\end{array}}{\Psi \mid \Gamma \vdash \mathsf{coe}_{x.\mathsf{V}_x(A,B,E)}^{r \rightsquigarrow r'}(M) = \mathsf{Vin}_{r'}(\mathsf{fst}(\widetilde{O}), \widetilde{P}) : \mathsf{V}_{r'}(A\langle r'/x\rangle, B\langle r'/x\rangle, E\langle r'/x\rangle)}$$

$$\frac{\begin{array}{c}\widetilde{T} := x = 0 \hookrightarrow y.\mathsf{fst}(E)\ \mathsf{coe}_{y.A}^{r \rightsquigarrow y}(M), x = 1 \hookrightarrow y.\mathsf{coe}_{y.B}^{r \rightsquigarrow y}(M) \\ \widetilde{C} := \mathsf{Vin}_x(\mathsf{coe}_{y.A}^{r \rightsquigarrow r'}(M), \mathsf{com}_{y.B}^{r \rightsquigarrow r'}(\mathsf{Vproj}_x(M, E\langle r/y\rangle); \widetilde{T}))\end{array}}{\Psi \mid \Gamma \vdash \mathsf{coe}_{y.\mathsf{V}_x(A,B,E)}^{r \rightsquigarrow r'}(M) = \widetilde{C} : \mathsf{V}_x(A\langle r'/y\rangle, B\langle r'/y\rangle, E\langle r'/y\rangle)}$$

$$\frac{\begin{array}{c}\widetilde{O}[y] := \mathsf{hcom}_A^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \\ \widetilde{T} := x = 0 \hookrightarrow y.\mathsf{fst}(E)\ \widetilde{O}[y], x = 1 \hookrightarrow y.\mathsf{hcom}_B^{r \rightsquigarrow y}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) \\ \widetilde{H} := \mathsf{Vin}_x(\widetilde{O}[r'], \mathsf{hcom}_B^{r \rightsquigarrow r'}(\mathsf{Vproj}_x(M, E); \overrightarrow{\xi_i \hookrightarrow y.\mathsf{Vproj}_x(N_i, E)}, \widetilde{T}))\end{array}}{\Psi \mid \Gamma \vdash \mathsf{hcom}_{\mathsf{V}_x(A,B,E)}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.N_i}) = \widetilde{H} : \mathsf{V}_x(A, B, E)}$$

$$\frac{i < j}{\Psi \mid \Gamma \vdash \mathsf{Type}_i\ \mathsf{type}_j} \qquad \frac{}{\Psi \mid \Gamma \vdash \mathsf{coe}_{x.\mathsf{Type}_i}^{r \rightsquigarrow r'}(A) = A : \mathsf{Type}_i}$$

$$\frac{\begin{array}{c}\Psi \mid r, r' : \mathbb{I} \qquad \Psi \mid \Gamma \vdash A : \mathsf{Type}_k \qquad \overrightarrow{\Psi, \xi_i, y \mid \Gamma \vdash B_i : \mathsf{Type}_k\ [y = r \hookrightarrow A, \overrightarrow{\xi_j \hookrightarrow B_j}]} \\ \overrightarrow{\Psi, \xi_i \mid \Gamma \vdash N_i : B_i\langle r'/y\rangle\ [\overrightarrow{\xi_j \hookrightarrow N_j}]} \qquad \Psi \mid \Gamma \vdash M : A\ [\overrightarrow{\xi_i \hookrightarrow \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(N_i)}]\end{array}}{\Psi \mid \Gamma \vdash \mathsf{box}^{r \rightsquigarrow r'}_{\overrightarrow{A, y.B_i}}(M; \overrightarrow{\xi_i \hookrightarrow N_i}) : \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{Type}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})\ [r = r' \hookrightarrow M, \overrightarrow{\xi_i \hookrightarrow N_i}]}$$

$$\frac{\Psi \mid \Gamma \vdash M : \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{Type}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})}{\Psi \mid \Gamma \vdash \mathsf{cap}^{r \leftsquigarrow r'}_{A}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}) : A\ [r = r' \hookrightarrow M, \overrightarrow{\xi_i \hookrightarrow \mathsf{coe}^{r' \rightsquigarrow r}_{y.B_i}(M)}]}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{cap}^{r \leftsquigarrow r'}(\mathsf{box}^{r \rightsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow N_i}); \overrightarrow{\xi_i \hookrightarrow y.B_i}) = M : A}$$

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{box}^{r \rightsquigarrow r'}(\mathsf{cap}^{r \leftsquigarrow r'}(M; \overrightarrow{\xi_i \hookrightarrow y.B_i}); \overrightarrow{\xi_i \hookrightarrow M}) = M : \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{Type}_k}(A; \overrightarrow{\xi_i \hookrightarrow y.B_i})}$$

$$\frac{\begin{array}{c}\widetilde{N_i}[w, z] := \mathsf{coe}^{s'\langle w/x\rangle \rightsquigarrow z}_{z.B_i\langle w/x\rangle}(\mathsf{coe}^{r \rightsquigarrow w}_{x.B_i\langle s'/z\rangle}(M)) \\ \widetilde{T} := \overrightarrow{\xi_i\langle r/x\rangle \hookrightarrow z.\mathsf{coe}^{z \rightsquigarrow s\langle r/x\rangle}_{z.B_i\langle r/x\rangle}(\mathsf{coe}^{s'\langle r/x\rangle \rightsquigarrow z}_{z.B_i\langle r/x\rangle}(M))} \\ \widetilde{O}[z] := \mathsf{hcom}^{s'\langle r/x\rangle \rightsquigarrow z}_{A\langle r/x\rangle}(\mathsf{cap}^{s\langle r/x\rangle \leftsquigarrow s'\langle r/x\rangle}(M; \overrightarrow{\xi_i\langle r/x\rangle \hookrightarrow z.B_i\langle r/x\rangle}); \widetilde{T}) \\ \widetilde{P} := \mathsf{com}^{r \rightsquigarrow r'}_{x.A}(\widetilde{O}[s\langle r/x\rangle]; \overrightarrow{\xi_i \hookrightarrow x.\widetilde{N_i}[x, s]}|_{(x \# \xi_i)}, s = s' \hookrightarrow x.\mathsf{coe}^{r \rightsquigarrow x}_{x.A}(M)|_{(x \# s, s')}) \\ \widetilde{Q_k}[z] := \mathsf{com}^{s\langle r'/x\rangle \rightsquigarrow z}_{z.B_k\langle r'/x\rangle}(\widetilde{P}; \overrightarrow{\xi_i \hookrightarrow z.\widetilde{N_i}[r', z]}|_{(x \# \xi_i)}, r = r' \hookrightarrow z.\mathsf{coe}^{s'\langle r'/x\rangle \rightsquigarrow z}_{z.B_k\langle r'/x\rangle}(M)) \\ \widetilde{H} := \mathsf{hcom}^{s\langle r'/x\rangle \rightsquigarrow s'\langle r'/x\rangle}_{A\langle r'/x\rangle}(\widetilde{P}; \overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow z.\mathsf{coe}^{z \rightsquigarrow s\langle r'/x\rangle}_{z.B_i\langle r'/x\rangle}(\widetilde{Q_i}[z])}, r = r' \hookrightarrow z.\widetilde{O}[z]) \\ \widetilde{C} := \mathsf{box}^{s\langle r'/x\rangle \rightsquigarrow s'\langle r'/x\rangle}(\widetilde{H}; \overrightarrow{\xi_i\langle r'/x\rangle \hookrightarrow \widetilde{Q_i}[s'\langle r'/x\rangle]})\end{array}}{\Psi \mid \Gamma \vdash \mathsf{coe}^{r \rightsquigarrow r'}_{x.\mathsf{hcom}^{s \rightsquigarrow s'}_{\mathsf{Type}_j}(A; \overrightarrow{\xi_i \hookrightarrow z.B_i})}(M) = \widetilde{C} : (\mathsf{hcom}^{s \rightsquigarrow s'}_{\mathsf{Type}_j}(A; \overrightarrow{\xi_i \hookrightarrow z.B_i}))\langle r'/x\rangle}$$

$$\frac{\begin{array}{c}\widetilde{O_i} := \mathsf{cap}^{s \leftsquigarrow s'}(N_i; \overrightarrow{\xi_j \hookrightarrow z.B_j}) \qquad \widetilde{P_j}[y] := \mathsf{hcom}^{r \rightsquigarrow y}_{B_j\langle s'/z\rangle}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i}) \\ \widetilde{T} := \overrightarrow{r_i = r'_i \hookrightarrow y.\widetilde{O_i}}, \overrightarrow{\xi_j \hookrightarrow y.\mathsf{coe}^{s' \rightsquigarrow s}_{z.B_j}(\widetilde{P_j}[y])}, s = s' \hookrightarrow y.\mathsf{hcom}^{r \rightsquigarrow y}_{A}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i}) \\ \widetilde{Q} := \mathsf{hcom}^{r \rightsquigarrow r'}_{A}(\mathsf{cap}^{s \leftsquigarrow s'}(M; \overrightarrow{\xi_j \hookrightarrow z.B_j}); \widetilde{T}) \qquad \widetilde{H} := \mathsf{box}^{s \rightsquigarrow s'}(\widetilde{Q}; \overrightarrow{\xi_j \hookrightarrow \widetilde{P_j}[r']})\end{array}}{\Psi \mid \Gamma \vdash \mathsf{hcom}^{r \rightsquigarrow r'}_{\mathsf{hcom}^{s \rightsquigarrow s'}_{\mathsf{Type}_k}(A; \overrightarrow{\xi_j \hookrightarrow z.B_j})}(M; \overrightarrow{r_i = r'_i \hookrightarrow y.N_i}) = \widetilde{H} : \mathsf{hcom}^{s \rightsquigarrow s'}_{\mathsf{Type}_k}(A; \overrightarrow{\xi_j \hookrightarrow z.B_j})}$$

Bibliography

[Abe13] Andreas Abel. “Normalization by Evaluation: Dependent Types and Impredicativity”. Habilitation thesis. Ludwig-Maximilians-Universität München, 2013. URL: http://www2.tcs.ifi.lmu.de/~abel/habil.pdf.

[ACD07] Andreas Abel, Thierry Coquand, and Peter Dybjer. “Normalization by Evaluation for Martin-Löf Type Theory with Typed Equality Judgements”. In: *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*. 2007, pp. 3–12. DOI: 10.1109/LICS.2007.33.

[ACK16] Thorsten Altenkirch, Paolo Capriotti, and Nicolai Kraus. “Extending Homotopy Type Theory with Strict Equality”. In: *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*. Ed. by Jean-Marc Talbot and Laurent Regnier. Vol. 62. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 21:1–21:17. ISBN: 978-3-95977-022-4. DOI: 10.4230/LIPIcs.CSL.2016.21.

[ACK17] Danil Annenkov, Paolo Capriotti, and Nicolai Kraus. *Two-Level Type Theory and Applications*. Preprint. 2017. arXiv: 1705.03307 [cs.LO].

[ADK17] Thorsten Altenkirch, Nils Anders Danielsson, and Nicolai Kraus. “Partiality, Revisited”. In: *Foundations of Software Science and Computation Structures (FoSSaCS 2017)*. Ed. by Javier Esparza and Andrzej S. Murawski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 534–549. ISBN: 978-3-662-54458-7. DOI: 10.1007/978-3-662-54458-7_31.

[AFH17] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. *Computational Higher Type Theory III: Univalent Universes and Exact Equality*. Preprint. 2017. arXiv: 1712.01800 [cs.LO].

[AFH18] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. “Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities”. In: *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*. Ed. by Dan Ghica and Achim Jung. Vol. 119. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 6:1–6:17. ISBN: 978-3-95977-088-0. DOI: 10.4230/LIPIcs.CSL.2018.6.

[Agda] The Agda Development Team. *The Agda Programming Language*. 2018. URL: http://wiki.portal.chalmers.se/agda/pmwiki.php.

[AH17] Carlo Angiuli and Robert Harper. *Computational Higher Type Theory II: Dependent Cubical Realizability.* Preprint. 2017. arXiv: 1606.09638 [cs.LO].

[AH18] Carlo Angiuli and Robert Harper. "Meaning explanations at higher dimension". In: *Indagationes Mathematicae* 29.1 (2018). L.E.J. Brouwer, fifty years later, pp. 135–149. ISSN: 0019-3577. DOI: 10.1016/j.indag.2017.07.010.

[AHW17] Carlo Angiuli, Robert Harper, and Todd Wilson. "Computational Higher-Dimensional Type Theory". In: *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages.* POPL 2017. Paris, France: ACM, 2017, pp. 680–693. ISBN: 978-1-4503-4660-3. DOI: 10.1145/3009837.3009861.

[All87] Stuart F. Allen. "A Non-Type-Theoretic Semantics for Type-Theoretic Language". PhD thesis. Cornell University, 1987. URL: https://ecommons.cornell.edu/handle/1813/6706.

[AMS07] Thorsten Altenkirch, Conor McBride, and Wouter Swierstra. "Observational Equality, Now!" In: *Proceedings of the 2007 Workshop on Programming Languages Meets Program Verification.* PLPV '07. Freiburg, Germany: ACM, 2007, pp. 57–68. ISBN: 978-1-59593-677-6. DOI: 10.1145/1292597.1292608.

[Ana+14] Abhishek Anand, Mark Bickford, Robert L. Constable, and Vincent Rahli. "A Type Theory with Partial Equivalence Relations as Types". Extended abstract at *20th International Conference on Types for Proofs and Programs (TYPES 2014).* 2014. URL: http://www.nuprl.org/documents/Anand/ATypeTheoryWithPartialEquivalenceRelationsAsTypes.pdf.

[Ane+17] Mathieu Anel, Georg Biedermann, Eric Finster, and André Joyal. *A Generalized Blakers–Massey Theorem.* Preprint. 2017. arXiv: 1703.09050 [math.AT].

[Ang+14] Carlo Angiuli, Edward Morehouse, Daniel R. Licata, and Robert Harper. "Homotopical Patch Theory". In: *Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming.* ICFP '14. Gothenburg, Sweden: ACM, 2014, pp. 243–256. ISBN: 978-1-4503-2873-9. DOI: 10.1145/2628136.2628158.

[Ang+16] Carlo Angiuli, Edward Morehouse, Daniel R. Licata, and Robert Harper. "Homotopical patch theory". In: *Journal of Functional Programming* 26 (2016). Special issue dedicated to ICFP 2014. DOI: 10.1017/S0956796816000198.

[Ang+18] Carlo Angiuli, Evan Cavallo, Kuen-Bang Hou (Favonia), Robert Harper, and Jonathan Sterling. "The RedPRL Proof Assistant (Invited Paper)". In: *13th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP 2018).* Ed. by Frédéric Blanqui and Giselle Reis. Vol. 274. Electronic Proceedings in Theoretical Computer Science. Oxford, UK, 2018. DOI: 10.4204/EPTCS.274.1.

[Ang+19] Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. “Syntax and Models of Cartesian Cubical Type Theory”. Preprint. 2019. URL: https://github.com/dlicata335/cart-cube.

[App+17] Andrew W. Appel, Lennart Beringer, Adam Chlipala, Benjamin C. Pierce, Zhong Shao, Stephanie Weirich, and Steve Zdancewic. “Position paper: the science of deep specification”. In: *Philosophical Transactions of the Royal Society of London Series A* 375.2104 (2017). ISSN: 1364-503X. DOI: 10.1098/rsta.2016.0331.

[AR14] Abhishek Anand and Vincent Rahli. “Towards a Formally Verified Proof Assistant”. In: *Interactive Theorem Proving*. Ed. by Gerwin Klein and Ruben Gamboa. ITP 2014. Vienna, Austria: Springer International Publishing, 2014, pp. 27–44. ISBN: 978-3-319-08970-6. DOI: 10.1007/978-3-319-08970-6_3.

[AW09] Steve Awodey and Michael A. Warren. “Homotopy theoretic models of identity types”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 146.1 (2009), pp. 45–55. ISSN: 0305-0041. DOI: 10.1017/S0305004108001783.

[Awo18] Steve Awodey. “A cubical model of homotopy type theory”. In: *Annals of Pure and Applied Logic* 169.12 (2018). Logic Colloquium 2015, pp. 1270–1294. ISSN: 0168-0072. DOI: 10.1016/j.apal.2018.08.002.

[Awo19] Steve Awodey. “A Quillen model structure on the category of cartesian cubical sets”. Preprint. 2019. URL: https://github.com/awodey/math/.

[Bau+16] Andrej Bauer, Gaëtan Gilbert, Philipp Haselwarter, Matija Pretnar, and Christopher A. Stone. “Design and Implementation of the Andromeda proof assistant”. Abstract at *22nd International Conference on Types for Proofs and Programs (TYPES 2016)*. 2016. URL: http://www.types2016.uns.ac.rs/images/abstracts/bauer2.pdf.

[Bau+17] Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Michael Shulman, Matthieu Sozeau, and Bas Spitters. “The HoTT Library: A Formalization of Homotopy Type Theory in Coq”. In: *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs*. CPP 2017. Paris, France: ACM, 2017, pp. 164–172. ISBN: 978-1-4503-4705-1. DOI: 10.1145/3018610.3018615.

[BC15] Marc Bezem and Thierry Coquand. “A Kripke model for simplicial sets”. In: *Theoretical Computer Science* 574 (2015), pp. 86–91. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2015.01.035.

[BCH14] Marc Bezem, Thierry Coquand, and Simon Huber. “A Model of Type Theory in Cubical Sets”. In: *19th International Conference on Types for Proofs and Programs (TYPES 2013)*. Ed. by Ralph Matthes and Aleksy Schubert. Vol. 26. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014, pp. 107–128. ISBN: 978-3-939897-72-9. DOI: 10.4230/LIPIcs.TYPES.2013.107.

[BCH18] Marc Bezem, Thierry Coquand, and Simon Huber. “The Univalence Axiom in Cubical Sets”. In: *Journal of Automated Reasoning* (2018). ISSN: 1573-0670. DOI: 10.1007/s10817-018-9472-6.

[BCM15] Jean-Philippe Bernardy, Thierry Coquand, and Guilhem Moulin. “A Presheaf Model of Parametric Type Theory”. In: *Electronic Notes in Theoretical Computer Science*. MFPS XXXI 319 (2015). 31st Conference on the Mathematical Foundations of Programming Semantics, pp. 67–82. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2015.12.006.

[Bee85] Michael J. Beeson. *Foundations of Constructive Mathematics*. Vol. 6. A Series of Modern Surveys in Mathematics. Springer-Verlag Berlin Heidelberg, 1985. ISBN: 978-3-642-68954-3. DOI: 10.1007/978-3-642-68952-9.

[Bek84] Hans Bekić. “Definable operations in general algebras, and the theory of automata and flowcharts”. In: *Programming Languages and Their Definition: H. Bekič (1936–1982)*. Ed. by C. B. Jones. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 30–55. ISBN: 978-3-540-38933-0. DOI: 10.1007/BFb0048939.

[BG11] Benno van den Berg and Richard Garner. “Types are weak ω-groupoids”. In: *Proceedings of the London Mathematical Society* 102.2 (2011), pp. 370–394. DOI: 10.1112/plms/pdq026.

[BGW17] Henning Basold, Herman Geuvers, and Niels van der Weide. “Higher Inductive Types in Programming”. In: *Journal of Universal Computer Science* 23.1 (2017), pp. 63–88. DOI: 10.3217/jucs-023-01-0063.

[Bir+18] Lars Birkedal, Aleš Bizjak, Ranald Clouston, Hans Bugge Grathwohl, Bas Spitters, and Andrea Vezzosi. “Guarded Cubical Type Theory”. In: *Journal of Automated Reasoning* (2018). ISSN: 1573-0670. DOI: 10.1007/s10817-018-9471-7.

[BLM15] Sylvie Boldo, Catherine Lelay, and Guillaume Melquiond. “Coquelicot: A User-Friendly Library of Real Analysis for Coq”. In: *Mathematics in Computer Science* 9.1 (2015), pp. 41–62. ISSN: 1661-8289. DOI: 10.1007/s11786-014-0181-1.

[BM17] Ulrik Buchholtz and Edward Morehouse. “Varieties of Cubical Sets”. In: *16th International Conference on Relational and Algebraic Methods in Computer Science*. Ed. by Peter Höfner, Damien Pous, and Georg Struth. RAMiCS ’17. Lyon, France: Springer International Publishing, 2017, pp. 77–92. ISBN: 978-3-319-57418-9. DOI: 10.1007/978-3-319-57418-9_5.

[Boo71] George Boolos. “The Iterative Conception of Set”. In: *The Journal of Philosophy* 68.8 (1971), pp. 215–231. ISSN: 0022362X. DOI: 10.2307/2025204.

[BP15] Andrej Bauer and Matija Pretnar. “Programming with algebraic effects and handlers”. In: *Journal of Logical and Algebraic Methods in Programming* 84.1 (2015). Special Issue on Domains X, International workshop on Domain Theory and applications, Swansea, 5–7 September, 2011, pp. 108–123. ISSN: 2352-2208. DOI: 10.1016/j.jlamp.2014.02.001.

[Bru16] Guillaume Brunerie. “On the homotopy groups of spheres in homotopy type theory”. PhD thesis. Université Nice Sophia Antipolis, 2016. URL: http://arxiv.org/abs/1606.05916.

[Bru+18] Guillaume Brunerie, Kuen-Bang Hou (Favonia), Evan Cavallo, Tim Baumann, Eric Finster, Jesper Cockx, Christian Sattler, Chris Jeris, Michael Shulman, et al. *Homotopy Type Theory in Agda*. 2018. URL: https://github.com/HoTT/HoTT-Agda.

[BT17] Simon Boulier and Nicolas Tabareau. *Model structure on the universe in a two level type theory*. Preprint. 2017. URL: https://hal.archives-ouvertes.fr/hal-01579822.

[Car86] John Cartmell. “Generalised algebraic theories and contextual categories”. In: *Annals of Pure and Applied Logic* 32 (1986), pp. 209–243. ISSN: 0168-0072. DOI: 10.1016/0168-0072(86)90053-9.

[CCHM18] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. “Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom”. In: *21st International Conference on Types for Proofs and Programs (TYPES 2015)*. Ed. by Tarmo Uustalu. Vol. 69. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 5:1–5:34. ISBN: 978-3-95977-030-9. DOI: 10.4230/LIPIcs.TYPES.2015.5.

[CH18] Evan Cavallo and Robert Harper. *Computational Higher Type Theory IV: Inductive Types*. Preprint. 2018. arXiv: 1801.01568 [cs.LO].

[CH19a] Evan Cavallo and Robert Harper. “Higher Inductive Types in Cubical Computational Type Theory”. In: *Proceedings of the ACM on Programming Languages* 3.POPL (2019), 1:1–1:27. ISSN: 2475-1421. DOI: 10.1145/3290314.

[CH19b] Evan Cavallo and Robert Harper. *Parametric Cubical Type Theory*. Preprint. 2019. arXiv: 1901.00489 [cs.LO].

[CH88] Thierry Coquand and Gérard Huet. “The Calculus of Constructions”. In: *Information and Computation* 76.2 (1988), pp. 95–120. ISSN: 0890-5401. DOI: 10.1016/0890-5401(88)90005-3.

[CHM18] Thierry Coquand, Simon Huber, and Anders Mörtberg. “On Higher Inductive Types in Cubical Type Theory”. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS ’18. Oxford, United Kingdom: ACM, 2018, pp. 255–264. ISBN: 978-1-4503-5583-4. DOI: 10.1145/3209108.3209197.

[CHS19] Thierry Coquand, Simon Huber, and Christian Sattler. “Homotopy Canonicity for Cubical Type Theory”. In: *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*. Ed. by Herman Geuvers. Vol. 131. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 11:1–11:23. ISBN: 978-3-95977-107-8. DOI: 10.4230/LIPIcs.FSCD.2019.11.

[Cis06] Denis-Charles Cisinski. *Les préfaisceaux comme modèles des types d’homotopie*. Vol. 308. Astérisque. Société Mathématique de France, 2006. ISBN: 978-2-85629-225-9. DOI: 10.24033/ast.715.

[CM16] Thierry Coquand and Bassel Mannaa. “The Independence of Markov’s Principle in Type Theory”. In: *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*. Ed. by Delia Kesner and Brigitte Pientka. Vol. 52. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 17:1–17:18. ISBN: 978-3-95977-010-1. DOI: 10.4230/LIPIcs.FSCD.2016.17.

[CMS19] Evan Cavallo, Anders Mörtberg, and Andrew W. Swan. “Unifying Cubical Models of Univalent Type Theory”. Preprint. 2019. URL: https://github.com/mortberg/gen-cart.

[Con+85] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development Environment*. Prentice-Hall, 1985. URL: http://www.nuprl.org/book/.

[Coq] The Coq Development Team. *The Coq Proof Assistant*. 2018. URL: https://www.coq.inria.fr.

[Coq14] Thierry Coquand. “Variation on Cubical sets”. Unpublished note. 2014. URL: http://www.cse.chalmers.se/~coquand/diag.pdf.

[Coq18] Thierry Coquand. *Quillen model structure.* Email to Homotopy Type Theory mailing list. 2018. URL: https://groups.google.com/d/msg/homotopytypetheory/RQkLWZ_83kQ/tAyb3zYTBQAJ.

[Coq19] Thierry Coquand. “Canonicity and normalization for dependent type theory”. In: *Theoretical Computer Science* 777 (2019). In memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part I, pp. 184–191. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2019.01.015.

[Cra98] Karl Fredrick Crary. “Type-Theoretic Methodology for Practical Programming Languages”. PhD thesis. Cornell University, 1998. URL: https://ecommons.cornell.edu/handle/1813/7353.

[Cur93] P.-L. Curien. “Substitution up to isomorphism”. In: *Fundamenta Informaticae* 19.1-2 (1993). Special issue: lambda calculus and type theory, pp. 51–85. ISSN: 0169-2968.

[Dag13] Pierre-Évariste Dagand. “Reusability and Dependent Types”. PhD thesis. University of Strathclyde, 2013. URL: https://pages.lip6.fr/Pierre-Evariste.Dagand/stuffs/thesis-2011-phd/thesis.pdf.

[DP02] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order.* Second Edition. Cambridge University Press, 2002. ISBN: 0-521-78451-4. DOI: 10.1017/CBO9780511809088.

[DRB17] Floris van Doorn, Jakob von Raumer, and Ulrik Buchholtz. “Homotopy Type Theory in Lean”. In: *Interactive Theorem Proving*. Ed. by Mauricio Ayala-Rincón and César A. Muñoz. ITP 2017. Brasília, Brazil: Springer, 2017, pp. 479–495. ISBN: 978-3-319-66107-0. DOI: 10.1007/978-3-319-66107-0_30.

[Dum77] Michael Dummett. *Elements of Intuitionism.* Vol. 39. Oxford Logic Guides. Oxford University Press, 1977. ISBN: 9780198505242.

[Dyb00] Peter Dybjer. “A General Formulation of Simultaneous Inductive-Recursive Definitions in Type Theory”. In: *The Journal of Symbolic Logic* 65.2 (2000), pp. 525–549. ISSN: 00224812. DOI: 10.2307/2586554.

[Dyb96] Peter Dybjer. “Internal type theory”. In: *Types for Proofs and Programs (TYPES 1995).* Ed. by Stefano Berardi and Mario Coppo. Vol. 1158. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 120–134. ISBN: 978-3-540-70722-6. DOI: 10.1007/3-540-61780-9_66.

[GG08] Nicola Gambino and Richard Garner. “The identity type weak factorisation system”. In: *Theoretical Computer Science* 409.1 (2008), pp. 94–109. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2008.08.030.

[Gon08] Georges Gonthier. "Formal Proof—The Four-Color Theorem". In: *Notices of the American Mathematical Society* 55.11 (2008), pp. 1382–1393. ISSN: 0002-9920. URL: https://www.ams.org/notices/200811/tx081101382p.pdf.

[Gon+13] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. "A Machine-Checked Proof of the Odd Order Theorem". In: *Interactive Theorem Proving*. Ed. by Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie. ITP 2013. Rennes, France: Springer Berlin Heidelberg, 2013, pp. 163–179. ISBN: 978-3-642-39634-2. DOI: 10.1007/978-3-642-39634-2_14.

[Gra09] Johan Georg Granström. "Reference and Computation in Intuitionistic Type Theory". PhD thesis. Uppsala University, 2009. URL: https://intuitionistic.files.wordpress.com/2010/07/theses_published_uppsala.pdf.

[Har16] Robert Harper. *Practical Foundations for Programming Languages*. Second Edition. Cambridge University Press, 2016. ISBN: 9781107150300. DOI: 10.1017/CBO9781316576892.

[Har92] Robert Harper. "Constructing Type Systems over an Operational Semantics". In: *Journal of Symbolic Computation* 14.1 (1992), pp. 71–84. ISSN: 0747-7171. DOI: 10.1016/0747-7171(92)90026-Z.

[HF15] Robert Harper and Kuen-Bang Hou (Favonia). *A Note on the Uniform Kan Condition in Nominal Cubical Sets*. Preprint. 2015. arXiv: 1501.05691 [math.LO].

[Hof95] Martin Hofmann. "Extensional concepts in intensional type theory". PhD thesis. University of Edinburgh, 1995. URL: http://www.lfcs.inf.ed.ac.uk/reports/95/ECS-LFCS-95-327/.

[How80] William A. Howard. "The formulae-as-types notion of construction". In: *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Ed. by Jonathan P. Seldin and J. Roger Hindley. Academic Press, 1980, pp. 479–490. ISBN: 978-0-12-349050-6.

[How89] Douglas J. Howe. "Equality in Lazy Computation Systems". In: *Proceedings of the Fourth Annual IEEE Symposium on Logic in Computer Science*. LICS 1989. Pacific Grove, CA, USA: IEEE Computer Society Press, 1989, pp. 198–203. DOI: 10.1109/LICS.1989.39174.

[How91] Douglas J. Howe. "On Computational Open-Endedness in Martin-Löf's Type Theory". In: *Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science*. LICS 1991. Amsterdam, The Netherlands: IEEE Computer Society Press, 1991, pp. 162–172. DOI: 10.1109/LICS.1991.151641.

[HS94] Douglas J. Howe and Scott D. Stoller. “An Operational Approach to Combining Classical Set Theory and Functional Programming Languages”. In: *Theoretical Aspects of Computer Software*. Ed. by Masami Hagiya and John C. Mitchell. Vol. 789. Lecture Notes in Computer Science. Sendai, Japan: Springer Berlin Heidelberg, 1994, pp. 36–55. ISBN: 978-3-540-48383-0. DOI: 10.1007/3-540-57887-0_89.

[HS98] Martin Hofmann and Thomas Streicher. “The groupoid interpretation of type theory”. In: *Twenty-Five Years of Constructive Type Theory*. Ed. by Giovanni Sambin and Jan Smith. Vol. 36. Oxford Logic Guides. Oxford University Press, 1998, pp. 83–111.

[Hub18] Simon Huber. “Canonicity for Cubical Type Theory”. In: *Journal of Automated Reasoning* (2018). ISSN: 1573-0670. DOI: 10.1007/s10817-018-9469-1.

[Hur95] Antonius J. C. Hurkens. “A simplification of Girard’s paradox”. In: *Typed Lambda Calculi and Applications*. Ed. by Mariangiola Dezani-Ciancaglini and Gordon Plotkin. TLCA 1995. Edinburgh, United Kingdom: Springer Berlin Heidelberg, 1995, pp. 266–278. ISBN: 978-3-540-49178-1. DOI: 10.1007/BFb0014058.

[Kan55] Daniel M. Kan. “Abstract Homotopy. I”. In: *Proceedings of the National Academy of Sciences of the United States of America* 41.12 (1955), pp. 1092–1096. ISSN: 00278424. URL: http://www.jstor.org/stable/89108.

[KHS19] Ambrus Kaposi, Simon Huber, and Christian Sattler. “Gluing for Type Theory”. In: *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*. Ed. by Herman Geuvers. Vol. 131. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 25:1–25:19. ISBN: 978-3-95977-107-8. DOI: 10.4230/LIPIcs.FSCD.2019.25.

[KL16] Krzysztof Kapulkin and Peter LeFanu Lumsdaine. *The Simplicial Model of Univalent Foundations (after Voevodsky)*. Preprint. 2016. arXiv: 1211.2851 [math.LO].

[KPB15] Neelakantan R. Krishnaswami, Pierre Pradic, and Nick Benton. “Integrating Linear and Dependent Types”. In: *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’15. Mumbai, India: ACM, 2015, pp. 17–30. ISBN: 978-1-4503-3300-9. DOI: 10.1145/2676726.2676969.

[KS15] Nicolai Kraus and Christian Sattler. “Higher Homotopies in a Hierarchy of Univalent Universes”. In: *ACM Transactions on Computational Logic* 16.2 (2015), 18:1–18:12. ISSN: 1529-3785. DOI: 10.1145/2729979.

[KS19] Krzysztof Kapulkin and Karol Szumiło. “Internal languages of finitely complete $(\infty, 1)$-categories”. In: *Selecta Mathematica* 25.2 (2019). ISSN: 1420-9020. DOI: 10.1007/s00029-019-0480-0.

[LB14] Daniel R. Licata and Guillaume Brunerie. *A Cubical Type Theory*. Talk at Oxford Homotopy Type Theory Workshop. 2014. URL: http://dlicata.web.wesleyan.edu/pubs/lb14cubical/lb14cubes-oxford.pdf.

[Ler09] Xavier Leroy. “Formal Verification of a Realistic Compiler”. In: *Communications of the ACM* 52.7 (2009), pp. 107–115. ISSN: 0001-0782. DOI: 10.1145/1538788.1538814.

[LH11] Daniel R. Licata and Robert Harper. “2-Dimensional Directed Type Theory”. In: *Electronic Notes in Theoretical Computer Science* 276 (2011). Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVII), pp. 263–289. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2011.09.026.

[LH12] Daniel R. Licata and Robert Harper. “Canonicity for 2-Dimensional Type Theory”. In: *Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’12. Philadelphia, PA, USA: ACM, 2012, pp. 337–348. ISBN: 978-1-4503-1083-3. DOI: 10.1145/2103656.2103697.

[Lic16] Dan Licata. *Weak univalence with “beta” implies full univalence*. Email to Homotopy Type Theory mailing list. 2016. URL: https://groups.google.com/d/msg/homotopytypetheory/j2KBIvDw53s/YTDK4D0NFQAJ.

[Lic+18] Daniel R. Licata, Ian Orton, Andrew M. Pitts, and Bas Spitters. “Internal Universes in Models of Homotopy Type Theory”. In: *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*. Ed. by Hélène Kirchner. Vol. 108. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 22:1–22:17. ISBN: 978-3-95977-077-4. DOI: 10.4230/LIPIcs.FSCD.2018.22.

[LS19] Peter LeFanu Lumsdaine and Michael Shulman. “Semantics of higher inductive types”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* (2019). DOI: 10.1017/S030500411900015X.

[Lum09] Peter LeFanu Lumsdaine. “Weak ω-Categories from Intensional Type Theory”. In: *9th International Conference on Typed Lambda Calculi and Applications*. Ed. by Pierre-Louis Curien. TLCA 2009. Brasilia, Brazil: Springer Berlin Heidelberg, 2009, pp. 172–187. ISBN: 978-3-642-02273-9. DOI: 10.1007/978-3-642-02273-9_14.

[Mai99] Maria Emilia Maietti. "About Effective Quotients in Constructive Type Theory". In: *Types for Proofs and Programs (TYPES 1998)*. Ed. by Thorsten Altenkirch, Bernhard Reus, and Wolfgang Naraschewski. Vol. 1657. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 166–178. ISBN: 978-3-540-48167-6. DOI: 10.1007/3-540-48167-2_12.

[ML13] Per Martin-Löf. "Verificationism Then and Now". In: *Judgement and the Epistemic Foundation of Logic*. Ed. by Maria van der Schaar. Dordrecht: Springer Netherlands, 2013, pp. 3–14. ISBN: 978-94-007-5137-8. DOI: 10.1007/978-94-007-5137-8_1.

[ML75a] Per Martin-Löf. "About Models for Intuitionistic Type Theories and the Notion of Definitional Equality". In: *Proceedings of the Third Scandinavian Logic Symposium*. Ed. by Stig Kanger. Vol. 82. Studies in Logic and the Foundations of Mathematics. Elsevier, 1975, pp. 81–109. DOI: 10.1016/S0049-237X(08)70727-4.

[ML75b] Per Martin-Löf. "An Intuitionistic Theory of Types: Predicative Part". In: *Logic Colloquium '73*. Ed. by H. E. Rose and J. C. Shepherdson. Vol. 80. Studies in Logic and the Foundations of Mathematics. North-Holland, 1975, pp. 73–118. DOI: 10.1016/S0049-237X(08)71945-1.

[ML82] Per Martin-Löf. "Constructive mathematics and computer programming". In: *Logic, Methodology and Philosophy of Science VI, Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science, Hannover 1979*. Ed. by L. Jonathan Cohen, Jerzy Łoś, Helmut Pfeiffer, and Klaus-Peter Podewski. Vol. 104. Studies in Logic and the Foundations of Mathematics. North-Holland, 1982, pp. 153–175. DOI: 10.1016/S0049-237X(09)70189-2.

[ML84] Per Martin-Löf. *Intuitionistic type theory. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980*. Vol. 1. Studies in Proof Theory. Bibliopolis, 1984. ISBN: 88-7088-105-9.

[Mou+15] Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. "The Lean Theorem Prover (System Description)". In: *Automated Deduction – CADE-25*. Ed. by Amy P. Felty and Aart Middeldorp. Berlin, Germany: Springer International Publishing, 2015, pp. 378–388. ISBN: 978-3-319-21401-6. DOI: 10.1007/978-3-319-21401-6_26.

[NPS90] Bengt Nordström, Kent Petersson, and Jan Smith. *Programming in Martin-Löf's Type Theory*. Oxford University Press, 1990. URL: http://www.cse.chalmers.se/research/group/logic/book/.

[Oos08] Jaap van Oosten. *Realizability: An Introduction to its Categorical Side*. Vol. 152. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 2008. ISBN: 9780444515841.

[OP16] Ian Orton and Andrew M. Pitts. "Axioms for Modelling Cubical Type Theory in a Topos". In: *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*. Ed. by Jean-Marc Talbot and Laurent Regnier. Vol. 62. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 24:1–24:19. ISBN: 978-3-95977-022-4. DOI: 10.4230/LIPIcs.CSL.2016.24.

[Par15] Jason Parker. "Duality between Cubes and Bipointed Sets". Master's thesis. Carnegie Mellon University, 2015.

[Pit13] Andrew M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Vol. 57. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2013. ISBN: 9781107017788. DOI: 10.1017/CBO9781139084673.

[Pit15] Andrew M. Pitts. "Nominal Presentation of Cubical Sets Models of Type Theory". In: *20th International Conference on Types for Proofs and Programs (TYPES 2014)*. Ed. by H. Herbelin, P. Letouzey, and M. Sozeau. Vol. 39. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 202–220. ISBN: 978-3-939897-88-0. DOI: 10.4230/LIPIcs.TYPES.2014.202.

[Plo81] Gordon D. Plotkin. "A Structural Approach to Operational Semantics". DAIMI FN-19, Computer Science Department, Aarhus University. 1981.

[Pra65] Dag Prawitz. *Natural Deduction: A Proof-Theoretical Study*. Vol. 3. Stockholm Studies in Philosophy. Almquist & Wiksell, 1965.

[RB16] Vincent Rahli and Mark Bickford. "A Nominal Exploration of Intuitionism". In: *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*. CPP 2016. St. Petersburg, FL, USA: ACM, 2016, pp. 130–141. ISBN: 978-1-4503-4127-1. DOI: 10.1145/2854065.2854077.

[RBA13] Vincent Rahli, Mark Bickford, and Abhishek Anand. "Formal Program Optimization in Nuprl Using Computational Equivalence and Partial Types". In: *Interactive Theorem Proving*. Ed. by Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie. ITP 2013. Rennes, France: Springer Berlin Heidelberg, 2013, pp. 261–278. ISBN: 978-3-642-39634-2. DOI: 10.1007/978-3-642-39634-2_20.

[Red16] The RedPRL Development Team. *The* ***RedPRL*** *Proof Assistant*. 2016–2018. URL: http://www.redprl.org/.

[Red18] The RedPRL Development Team. *The* **redtt** *Proof Assistant*. 2018. URL: https://github.com/RedPRL/redtt/.

[Rey83] John C. Reynolds. “Types, Abstraction and Parametric Polymorphism”. In: *Information Processing ’83: Proceedings of the IFIP 9th World Computer Congress*. Ed. by R. E. A. Mason. North-Holland, 1983, pp. 513–523. ISBN: 0444867295.

[RS17] Emily Riehl and Michael Shulman. “A type theory for synthetic ∞-categories”. In: *Higher Structures* 1.1 (2017), pp. 147–224. URL: https://arxiv.org/abs/1705.07442.

[SAG19] Jonathan Sterling, Carlo Angiuli, and Daniel Gratzer. “Cubical Syntax for Reflection-Free Extensional Equality”. In: *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*. Ed. by Herman Geuvers. Vol. 131. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 31:1–31:25. ISBN: 978-3-95977-107-8. DOI: 10.4230/LIPIcs.FSCD.2019.31.

[Sat17] Christian Sattler. *The Equivalence Extension Property and Model Structures*. Preprint. 2017. arXiv: 1704.06911 [math.CT].

[See84] R. A. G. Seely. “Locally cartesian closed categories and type theory”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 95.1 (1984), pp. 33–48. DOI: 10.1017/S0305004100061284.

[SH18] Jonathan Sterling and Robert Harper. “Guarded Computational Type Theory”. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS ’18. Oxford, United Kingdom: ACM, 2018, pp. 879–888. ISBN: 978-1-4503-5583-4. DOI: 10.1145/3209108.3209153.

[Shu15] Michael Shulman. “Univalence for inverse diagrams and homotopy canonicity”. In: *Mathematical Structures in Computer Science* 25.5 (2015), 1203––1277. DOI: 10.1017/S0960129514000565.

[Shu19] Michael Shulman. *All* $(\infty, 1)$*-toposes have strict univalent universes*. Preprint. 2019. arXiv: 1904.07004 [math.AT].

[Smi88] Jan M. Smith. “The Independence of Peano’s Fourth Axiom from Martin-Löf’s Type Theory Without Universes”. In: *The Journal of Symbolic Logic* 53.3 (1988), pp. 840–845. ISSN: 00224812. DOI: 10.2307/2274575.

[Smi89] Scott Fraser Smith. “Partial Objects in Type Theory”. PhD thesis. Cornell University, 1989. URL: https://ecommons.cornell.edu/handle/1813/6778.

[Soj16] Kristina Sojakova. “The Equivalence of the Torus and the Product of Two Circles in Homotopy Type Theory”. In: *ACM Transactions on Computational Logic* 17.4 (2016), 29:1–29:19. ISSN: 1529-3785. DOI: 10.1145/2992783.

[Str06] Thomas Streicher. “Identity Types vs. Weak ω-Groupoids: Some Ideas and Problems”. Talk given in Uppsala at the meeting on “Identity Types: Topological and Categorical Structure”. 2006. URL: http://www.mathematik.tu-darmstadt.de/~streicher/TALKS/uppsala.pdf.gz.

[Str93] Thomas Streicher. “Investigations Into Intensional Type Theory”. Habilitation thesis. Ludwig-Maximilians-Universität München, 1993. URL: https://www2.mathematik.tu-darmstadt.de/~streicher/HabilStreicher.pdf.

[Swa18a] Andrew W. Swan. *Identity Types in Algebraic Model Structures and Cubical Sets*. Preprint. 2018. arXiv: 1808.00915 [math.LO].

[Swa18b] Andrew W. Swan. *Separating Path and Identity Types in Presheaf Models of Univalent Type Theory*. Preprint. 2018. arXiv: 1808.00920 [math.LO].

[Tai67] W. W. Tait. “Intensional interpretations of functionals of finite type I”. In: *Journal of Symbolic Logic* 32.2 (1967), pp. 198–212. DOI: 10.2307/2271658.

[TD88] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction, Volume I*. Vol. 121. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1988. ISBN: 9780444702661.

[TTS18] Nicolas Tabareau, Éric Tanter, and Matthieu Sozeau. “Equivalences for Free: Univalent Parametricity for Effective Transport”. In: *Proceedings of the ACM on Programming Languages* 2.ICFP (2018), 92:1–92:29. ISSN: 2475-1421. DOI: 10.1145/3236787.

[Tur85] D. A. Turner. “Miranda: A non-strict functional language with polymorphic types”. In: *Functional Programming Languages and Computer Architecture (FPCA 1985)*. Ed. by Jean-Pierre Jouannaud. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 1–16. ISBN: 978-3-540-39677-2.

[UF13] The Univalent Foundations Program, Institute for Advanced Study. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Self-published, 2013. URL: https://homotopytypetheory.org/book/.

[VAG+] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. *UniMath — a computer-checked library of univalent mathematics*. URL: https://github.com/UniMath/UniMath.

[VMA19] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. “Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types”. In: *Proceedings of the ACM on Programming Languages* 3.ICFP (2019), 87:1–87:29. ISSN: 2475-1421. DOI: 10.1145/3341691.

[Voe10a] Vladimir Voevodsky. *The equivalence axiom and univalent models of type theory*. Notes from a talk at Carnegie Mellon University. 2010. URL: http://www.math.ias.edu/vladimir/files/CMU_talk.pdf.

[Voe10b] Vladimir Voevodsky. "Univalent Foundations Project". A modified version of an NSF grant application. 2010. URL: http://www.math.ias.edu/vladimir/files/univalent_foundations_project.pdf.

[Voe13] Vladimir Voevodsky. "A simple type system with two identity types". Talk at Andre Joyal's 70th birthday conference. (Slides available at https://www.math.ias.edu/vladimir/sites/math.ias.edu.vladimir/files/HTS_slides.pdf). 2013. URL: https://www.math.ias.edu/vladimir/sites/math.ias.edu.vladimir/files/HTS.pdf.

[Voe14] Vladimir Voevodsky. "Foundations of Mathematics: their past, present and future". Paul Bernays Lectures 2014 given at ETH Zurich. 2014. URL: http://www.math.ias.edu/vladimir/Lectures.

[Voe15] Vladimir Voevodsky. *Martin-Lof identity types in the C-systems defined by a universe category*. Preprint. 2015. arXiv: 1505.06446 [math.CT].

[Voe16] Vladimir Voevodsky. "Multiple Concepts of Equality in the New Foundations of Mathematics". Talk at Foundations of Mathematics: Univalent Foundations and Set Theory in Bielefeld, Germany. 2016. URL: http://www.math.ias.edu/vladimir/Lectures.

[War08] Michael A. Warren. "Homotopy theoretic aspects of constructive type theory". PhD thesis. Carnegie Mellon University, 2008. URL: http://mawarren.net/papers/phd.pdf.