The Inverse Method for Intuitionistic Linear Logic (The Propositional Fragment)

Kaustuv Chaudhuri

September 24, 2003

CMU-CS-03-140

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present a forward sequent calculus for intuitionistic propositional linear logic ($\otimes$, $\mathbf{1}$, $\&$, $\top$, $\multimap$, $\oplus$, $\mathbf{0}$, $!$) and a corresponding inverse-method search strategy. Our approach centres around resource management, inspired by similar approaches for backward-directed calculi such as top-down linear logic programming. Surprisingly, the resource management problems for the forward direction turn out to have a different character to those of the backward direction, arising for different connectives. Our approach identifies conditions for which we may relax linearity to allowing (implicit) weakening. We characterize two such classes of affine behaviour – as a form of *weak* sequent designed to handle $\top$-weakening lazily, and as affine contexts to control the multiplicative unit $\mathbf{1}$ using a general matching framework.

Partially supported by NSF Grant CCR-9988281 "*Logical and Meta-Logical Frameworks*".

Keywords: linear logic, inverse method, resource management

1 Introduction

In this paper, we examine the feasibility of efficient forward reasoning for intuitionistic linear logic [10, 4], concentrating primarily on the propositional fragment. In forward reasoning, search for the proof of a *goal sequent* begins with *initial sequents*, and uses inference rules to construct new sequents (using some strategy), with the goal of eventually discovering a proof of the goal. Thus, it directly contrasts with backward or *goal-directed* reasoning, which applies rules in the backward direction to iteratively refine a collection of goals, until all goals become initial (axiomatic). However, the kind of forward reasoning in this paper—the inverse method [19, 25]—also merits the description "goal-directed" because it restricts all rule applications to subformulas of the goal sequent, using a strong subformula property of the sequent calculus. Linear logic disallows arbitrary contraction and weakening; thus the number of occurrences of formulas plays a critical role in the proof theory, thereby allowing encodings of theories with precise counting semantics. A linear hypothesis must have *exactly one use* in a proof, which lends a view of a linear hypotheses as *resources*, and proofs as *consumers* of such resources. This has led to various applications in reasoning about state; for example, concurrent computation [1], or games [17, 2]. Intuitionistic linear logic [4, 8] imposes a strict separation of plural resources from the singular conclusion, elevating linear implication ($\multimap$) to the status of a logical connective independent of other connectives.

The novelty in automated reasoning in linear logic lies in handling resources efficiently – the *resource management problem*. We can trace the origin of this problem to the lack of structural weakening and contraction; indeed, without these rules linear logic with additives and exponentials becomes undecidable, even for the propositional case [18]. We can recognise the following classes of resource management problems, which we explain in greater detail in section 2.2.

Multiplicative Non-determinism, which arises from multiplicative rules with more than one premiss, for example for $\otimes R$:

$$\frac{\Delta \Longrightarrow A \qquad \Delta' \Longrightarrow B}{\Delta, \Delta' \Longrightarrow A \otimes B}$$

Absent weakening, in the backward direction such rules must infer a division (into Δ and Δ' above) of the linear resources of the conclusion to distribute into the premisses. Note that this kind of non-determinism does not exist in a forward reading, where we simply conjoin the resources of the premisses to construct the conclusion.

Structural Non-determinism, which occurs for unrestricted resources, and the linear resources in the rules for the additive units, such as $\top$:

$$\frac{}{\Delta \Longrightarrow \top}$$

For these rules, the conclusion sequent contains resources that don't occur structurally in the (possibly non-existent) premisses. Thus, a forward reading of these rules has to invent this context using extra-logical means, a futile approach in general because of the lack of a decision procedure. Fortunately, a clean solution exists for this problem, which we explain in section 3.

Interestingly, the rules for the additive units present significant problems in the backward direction also. In the common input-output interpretation of (backward) proof search [15, 7, 12], the additive unit $\top$ can "consume" an arbitrary number of resources in its branch of the proof. Thus, proving $\top \otimes A$ may require backtracking if search proceeds down the $\top$ branch first without determining the number of resources needed for the other branch. (For a more complete discussion, see [7].)

The problem of structural non-determinism thus exists in both forward and backward reasoning, but the nature— *invention* of unknown resources in the forward direction, and *allocation/garbage-collection* of resources in the backward direction—differs sufficiently that we cannot immediately adapt resource management approaches for the latter to the former.

Unknown Use Non-determinism, which to our knowledge has remained unidentified before this paper, occurs because of the multiplicative unit **1**. In the forward direction, the left rule for **1** applies for any sequent; without contraction, iterations of this rule can generate an arbitrary number of trivially different sequents. Furthermore, in tandem with binary rules such as $\& L$ that don't require both operands structurally in the premiss, we can iterate a two-rule sequence to generate arbitrarily many copies of the resources like $A \mathbin{\&} \mathbf{1}$. We call this "unknown use" to show our uncertainty about the exact number of such resources that contribute to the final proof. Unlike the previous two cases, we don't solve this resource management problem completely in this paper; indeed, by undecidability, such a solution cannot exist.

Other non-deterministic choices do exist during proof search, but they don't share the peculiar nature of resource management problems, and certainly occur for ordinary (non-linear) logic also; for example, *disjunctive non-determinism* for connectives with multiple introduction rules (on the left or right); *conjunctive non-determinism* for multi-premiss rules, where the order of exploration affects search in significant ways;[1] and various other possibilities. Because of the standard nature of these problems, we refer readers to the Handbook article on the inverse method [9].

The rest of this paper has the following organisation – in section 2 we introduce the backward calculus for linear logic, which forms a basis for the rest of the paper. We will show every one of our systems sound and complete with respect to this calculus, though the notions of completeness will get increasingly complex. Following a closer analysis of the non-determinism in this calculus from the viewpoint of forward reasoning (section 2.2), we will develop a sequence of calculi to tackle the various resource management problems. In section 3 we present a first forward calculus designed to handle the affine resource non-determinism of the affine units $\top$ and **0**. Subsequently, in section 4, we will examine the issue of unknown use non-determinism, and present a forward calculus with an explicitly identified *matching judgement*, designed to handle implicit weakening rules. The resulting sequent calculus will concretely expose all resource management problems in the proof theory. In the final section (5), we will describe an inverse method search procedure using this forward calculus.

Related Work. Resource management in backward reasoning has a relatively long history given the age of linear and sub-structural logics, with the earliest identification of this issue in proof search dating back to the work of Harland and Pym in 1991 [11]. Much subsequent work has examined resource management in the settings of backward proof search and logic programming for the (linear) uniform fragment [14, 15, 13], most recently by Cervesato *et al.* [7], and Harland and Pym [12]. The weakening annotation introduced in this paper bears a strong resemblance to a similar notation in [7], although the interpretation differs considerably because of the different nature of forward search.

To the best of our knowledge, resource management in the forward direction has not received any satisfactory treatment in the literature. The oldest work on forward-reasoning in linear logic, due to Mints [20], discusses a kind of resolution calculus for linear logic; however, his resolution calculus differs significantly from the usual notion of resolution because of the inclusion of *axiomatic clauses* of the form $\Gamma, \top$. The context Γ in these clauses remains unspecified, so for an implementation it becomes necessary to restrict the use of such axioms, specifically by discovering permutations in the resolutions steps that allow pushing these axioms downwards.

[1] In the forward direction, conjunctive non-determinism arises from saturation-based (*i.e.*, fair) search, a necessity to ensure completeness.

The resolution calculus of Mints suffers from an additional problem, arising from the inclusion of explicit weakening rules for the classical "why not" modality, ?. Tammet [24] has performed a fuller examination of the allowable permutations, together with more efficient treatment of weakening and exponentials, but both Mints and Tammet describe what we now understand as resource management problems in terms of search strategies. In other words, their calculi lack the explicit examination of resource management issues in the proof theory, in the style of Cervesato *et al.* [7] or Harland and Pym [12]. We have not encountered any other investigations of forward reasoning in linear logic in our literature survey.

2 Backward Sequent Calculus

We begin with a brief introduction to backward sequent calculi in the style of Gentzen for propositional linear logic. Propositions, called *formulas*, consist of atomic formulas, or formulas joined by connectives from one of the following classes – multiplicative, additive or exponential. For the multiplicative connectives, we have conjunction ($\otimes$, $\mathbf{1}$) and linear implication ($\multimap$); for the additives, conjunction ($\&$, $\top$) and disjunction ($\oplus$, $\mathbf{0}$); and for the exponentials the modal operator (!). We write formulas using capital letters A, B, $\ldots$, reserving the letters C and P as far as possible to stand for "conclusion" and "atomic formulas" respectively.

We represent facts about formulas using sequents, using the terminology introduced by Gentzen. Each sequent consists of hypotheses on the left, and conclusions on the right, separated by a *sequent arrow* $\Longrightarrow$. Because of our intuitionistic setting, we disallow sequents with more than one conclusion. We divide the hypotheses into zones separated by semi-colons (**;**), following the notation invented by Andreoli [3]. Inside a zone, we delineate individual hypotheses with commas (,), and allow free exchange of hypotheses *within* a given zone.

In this section, we consider sequents with two zones of hypotheses.[2] The first one, the *unrestricted* zone, consists of hypotheses that can have arbitrarily many uses in a proof, including none at all; we write this zone schematically as Γ. The other zone consists of *linear* hypotheses that must have *exactly one use* in any proof; we write this zone as Δ. Thus, sequents have the following schematic form:

$$\Gamma \,\mathbf{;}\, \Delta \Longrightarrow C$$

Rules for inferring sequents determine a *sequent calculus*. Every inference rule consists of a single *conclusion* sequent, and a varying number of *premisses* which may include sequents or other relations on formulas. Semantically, we shall read these rules backwards – from the conclusion to the premisses – by connecting them using "if" in the meta-language; *i.e.*, the conclusion holds if the premisses hold. Inference rules fall into one of two possible classes — *Judgemental Rules*, which together with the admissible cut rules (theorem 2) determine the meta-logical properties of the logic; and *Logical Rules* that define the nature of logical connectives. Each logical rule determines the meaning of a *single* connective in the goal sequent, either on the left (*left rule*) or the right (*right rule*) of the sequent arrow. Every logical rule, when read in the backward direction, evidently proceeds by analysis of the topmost connective of a *principal formula*. (We will refer to the other formulas among the assumptions or the conclusion as *side formulas*.)

For our calculus, we identify two judgemental rules. The first, called the *rule of initiality* (or just *init*), states that we may conclude a formulas A if we have a *single* linear hypothesis A. We call a sequent *initial* if we can infer it using init. The other judgemental rule transfers a copy of any unrestricted resource into the zone of linear resources, when reading the rule from the conclusion to the premiss. The copied resource continues in the unrestricted context because

[2] We shall use the terms "zone" and "context" interchangeably.

Judgemental Rules

$$\frac{}{\Gamma\,;A \Longrightarrow A}\ \text{init} \qquad \frac{\Gamma, A\,;\Delta, A \Longrightarrow C}{\Gamma, A\,;\Delta \Longrightarrow C}\ \text{copy}$$

Multiplicative Rules

$$\frac{\Gamma\,;\Delta, A, B \Longrightarrow C}{\Gamma\,;\Delta, A \otimes B \Longrightarrow C}\ \otimes L \qquad \frac{\Gamma\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Delta' \Longrightarrow B}{\Gamma\,;\Delta, \Delta' \Longrightarrow A \otimes B}\ \otimes R$$

$$\frac{\Gamma\,;\Delta \Longrightarrow C}{\Gamma\,;\Delta, \mathbf{1} \Longrightarrow C}\ \mathbf{1}L \qquad \frac{}{\Gamma\,;\cdot \Longrightarrow \mathbf{1}}\ \mathbf{1}R$$

$$\frac{\Gamma\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Delta', B \Longrightarrow C}{\Gamma\,;\Delta, \Delta', A \multimap B \Longrightarrow C}\ \multimap L \qquad \frac{\Gamma\,;\Delta, A \Longrightarrow B}{\Gamma\,;\Delta \Longrightarrow A \multimap B}\ \multimap R$$

Additive Rules

$$\frac{\Gamma\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Delta, A \,\&\, B \Longrightarrow C}\ \& L_1 \qquad \frac{\Gamma\,;\Delta, B \Longrightarrow C}{\Gamma\,;\Delta, A \,\&\, B \Longrightarrow C}\ \& L_2 \qquad \frac{\Gamma\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Delta \Longrightarrow B}{\Gamma\,;\Delta \Longrightarrow A \,\&\, B}\ \& R$$

$$\frac{\Gamma\,;\Delta, A \Longrightarrow C \qquad \Gamma\,;\Delta, B \Longrightarrow C}{\Gamma\,;\Delta, A \oplus B \Longrightarrow C}\ \oplus L \qquad \frac{\Gamma\,;\Delta \Longrightarrow A}{\Gamma\,;\Delta \Longrightarrow A \oplus B}\ \oplus R_1 \qquad \frac{\Gamma\,;\Delta \Longrightarrow B}{\Gamma\,;\Delta \Longrightarrow A \oplus B}\ \oplus R_2$$

$$\frac{}{\Gamma\,;\Delta, \mathbf{0} \Longrightarrow C}\ \mathbf{0}L \qquad \frac{}{\Gamma\,;\Delta \Longrightarrow \top}\ \top R$$

Exponential Rules

$$\frac{\Gamma, A\,;\Delta \Longrightarrow C}{\Gamma\,;\Delta, !A \Longrightarrow C}\ !L \qquad \frac{\Gamma\,;\cdot \Longrightarrow A}{\Gamma\,;\cdot \Longrightarrow !A}\ !R$$

Figure 1: Rules for the backward sequent calculus

we may need further copies of it later. The number of applications of this rule for any given unrestricted resource defines the number of uses of that resource in the proof of the goal.

We further classify logical rules into multiplicative, additive, and exponential rules, corresponding to the kind of top-level connective they describe. Multiplicative rules involve a distribution of the linear context of the conclusion sequent into the linear contexts of the premisses; additive rules require identical side formulas in the premisses and the conclusion; and exponential rules describe the modal nature of !. Figure 1 lists all the rules.

With the intended backward interpretation of rules, we view logical rules as composing a derivation tree, where each internal node either analyses one principal connective, or copies a formula from the unrestricted (global) assumptions, and initial sequents constitute the leaves. Some structural properties of the unrestricted context follow by simple induction.

Theorem 1 (Structural Properties).

1. *(weakening) If* $\Gamma\,;\Delta \Longrightarrow C$, *then* $\Gamma, A\,;\Delta \Longrightarrow C$.
2. *(contraction) If* $\Gamma, A, A\,;\Delta \Longrightarrow C$, *then* $\Gamma, A; \Delta \Longrightarrow C$.

The cut-free sequent calculus as presented admits structural cut rules. We can prove such a cut-admissibility theorem using a simple nested structural induction.

Property 2 (Admissibility of Cut).

1. *If* $\Gamma\,;\Delta \Longrightarrow A$ *and* $\Gamma\,;\Delta', A \Longrightarrow C$*, then* $\Gamma\,;\Delta, \Delta' \Longrightarrow C$.
2. *If* $\Gamma\,;\cdot \Longrightarrow A$ *and* $\Gamma, A\,;\Delta \Longrightarrow C$*, then* $\Gamma\,;\Delta' \Longrightarrow C$.

We omit the fairly standard proof, but see [8, 22].

2.1 Backwards sequent calculi for linear logic – a brief history

A brief note on the genealogy of this presentation of linear logic. We trace the idea of dividing the hypotheses into an unrestricted and a linear zone back to Andreoli [3] for what he called a "dyadic system" for (classical) linear logic. This idea has seen considerable use since then: Hodas and Miller in the setting of logic programming in the uniform fragment [15], Benton *et al.* for linear term calculi [6]; more recently by Barber and Plotkin for the system DILL [4], Polakow and Pfenning for ordered logic [22], and Howe in the setting of focused (backward) proof search for linear logic [16]. (The last of these unfortunately does not address the problem of focused proof-search in the two-zone setting, but rather uses it only to establish soundness and completeness of a one-zone focusing system, with explicit dereliction and promotion rules for modal contexts of the form $!\Gamma$.)

In an interesting alternate formulation of the separation of linear and unrestricted hypotheses, Benton [5] (who attributes his essential approach to unpublished work of Bart Jacobs in 1993), has described a system called LNL logic ("linear/non-linear") that consists of two separate universes of "unrestricted" and "linear" objects, with transitions F and G between the two arising as categorically adjoint functors. Each universe contains a separate term calculus, and the functors allow arbitrary combinations of objects as necessary. We don't use LNL as the basis for this paper, despite the existence of a reasonably clean cut-free calculus for the multiplicative-exponential fragment, because a full description of additive connectives remains open for LNL.

In a recent work, Chang *et al.* [8] give a backward sequent calculus for a conservative extension of intuitionistic linear logic with a novel interpretation of the classical ? operator as a *monad of possibility*. In this calculus it becomes possible to interpret the classical linear logic, classical affine logic (*i.e.*, CLL + arbitrary weakening), and the mysterious MIX rules (introduced by Girard [10]), using uniform parametric translations. We don't extend our system to include this possibility monad, though we don't believe such an extension presents significant problems.[3]

2.2 Non-determinism in the linear sequent calculus

Before proceeding to the forward sequent calculus, we examine the sources of non-determinism in the backward sequent calculus. This examination will make the case for the forward calculus in the next section, and the subsequent inverse method in section 5.

Multi-premiss rules like $\otimes R$ cause the largest amount of resource non-determinism in the backward direction.

$$\frac{\Gamma\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Delta' \Longrightarrow B}{\Gamma\,;\Delta, \Delta' \Longrightarrow A \otimes B}\;\otimes R$$

Knowing just the total linear context Δ, Δ', we face an exponential number of possibilities for the composition of Δ and Δ'. As mentioned in the introduction, We call this form of non-determinism *multiplicative resource non-determinism*. In the domain of top down linear logic programming — refining goals by applying inference rules in the *backward* direction until they become initial (eg. *Lolli* [14] or *Lygon* [26]) — approaches to combating this kind of non-determinism fall into two broad kinds.

[3]Not, at least, without focused derivations.

The first kind commit to an input-output interpretation of hypotheses. For the $\otimes R$ rule for example, proof search proceeds eagerly along the first premiss until it reaches the initial sequents with some unconsumed resources. These unconsumed resources then form the linear context for the second branch of the derivation tree corresponding to the second premiss. Proof search therefore becomes completely deterministic, though not free of complications. For example, when attempting to prove $\top \otimes A$, the first branch can consume an arbitrary number of resources; thus an unprincipled implementation of the input-output idea will continue to involve a potentially exponential number of backtracking operations. As a possible answer to such complications, Cervesato *et al.* [7] refine the sequent judgement with boolean "strictness" flags and add a context of *strict* resources, which adequately solves the resource management problems for the uniform fragment of linear logic.

Approaches of the second kind perform general search with constraint solving. For example, in [12], boolean flags mark uses of resources, with inference rules guarded by constraints on these boolean flags. Particular proof strategies then correspond to particular solutions for these constraint problems. In fact, we may view the first kind of approach as a kind of solution to the constraint problem, where the boolean constraints encode the input-output interpretation. Without detailing such constraint-based resource management systems, we refer to the work of Harland, Pym and Winikoff, now almost a decade old [11, 13].

Surprisingly, multiplicative resource non-determinism does not occur at all in the forward reading of such inference rules, where we start with the sequents involving linear contexts Δ and Δ', and conclude a sequent involving Δ, Δ'. We certainly don't have to select among exponentially many choices in forward reasoning; lest we give the impression that forward reasoning suffers from no resource non-determinism, we quickly point out that different problems arise in this direction; we focus on these issues for the remainder of this paper.

As mentioned before, the additive units $\top$ and $\mathbf{0}$ bring about complications in the simple input-output model of backwards search, but these very connectives present problems of a different nature for forward search.

$$\frac{}{\Gamma\,;\Delta \Longrightarrow \top}\ \top R \qquad \frac{}{\Gamma\,;\Delta, \mathbf{0} \Longrightarrow C}\ \mathbf{0}L$$

The arbitrary linear contexts Δ (and side formula C) do not occur in the (non-existent) premisses. In the forward direction, we face the seemingly impossible task of inventing these contexts from nothing! (A similar situation arises for the unrestricted context in axiomatic rules, including init.)

The solution to the problem of inventing these unknown formulas lies in a delayed generation of these formulas, an approach also used by Tammet [24] to describe resolution for classical linear logic. Cut-free sequent calculi enjoy a subformula property (see section 5), so we have to create only the portion of these contexts whose composition we can infer from other premisses. The present paper exposes this *lazy generation* of resources, both unrestricted and linear, directly in the proof-theory of the forward sequent calculus.

As a second major source of unknown use non-determinism, we tackle the problem of $\mathbf{1}L$ and $\&L$ rules working in concert. Non-determinism of this form is a neglected topic, primarily because it doesn't arise in backward-reasoning systems.

$$\frac{\Gamma\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Delta, A \mathbin{\&} \mathbf{1} \Longrightarrow C} \qquad \frac{\dfrac{\Gamma\,;\Delta \Longrightarrow C}{\Gamma\,;\Delta, \mathbf{1} \Longrightarrow C}\ *}{\Gamma\,;\Delta, A \mathbin{\&} \mathbf{1} \Longrightarrow C}$$

As clearly shown by these rules, $A \mathbin{\&} \mathbf{1}$ encodes an "at-most one use" interpretation for the formula A. The first of these rules describes the "one use" case, and the second the "zero use" case. We can consider such an encoding as an idiom for recovering *affine logic* in the exact setting of linear logic. For forward reasoning, we lose control about when and how often to introduce

$A \mathbin{\&} \mathbf{1}$ (assuming it occurs as a negative subformula of the eventual goal) using the second of the above rules. In other words, we cannot determine the *multiplicity* of the resource $A \mathbin{\&} \mathbf{1}$ in any intermediate sequent of the proof, and sequents can therefore grow indefinitely during search. We approach this resource management problem by constraining the $\mathbf{1}L$ rule ($*$ above); for more detail, see section 4.

3 Forward reasoning and the additive units

For expository purposes, we build our system of forward reasoning in stages. For the first stage, we show how to eliminate structural non-determinism by removing the need to "invent" formulas. Because of the significantly different nature of the resulting calculus, we use $\longrightarrow$ as the sequent arrow.

We interpret the unrestricted zone as a *strict* context with implicit contraction. It will contain only those resources that actually participate in a particular proof of the sequent. For example, the init rule does not use any unrestricted hypotheses, so we remove the unrestricted context Γ entirely from the conclusion to get:

$$\frac{}{\cdot \,;\, A \longrightarrow A}\ \text{init}$$

As a result of this strict interpretation, multi-premiss rules will have different unrestricted contexts in the premisses and the conclusion. For these rules, we union (*i.e.*, factor duplicated hypotheses into a single hypothesis) the unrestricted zones in the premisses to form the unrestricted zone in the conclusion.

For the linear zone, we implicitly allow a kind of weakening for the linear context in specific cases. We cannot of course allow weakening of all linear contexts, so we identify the *weakness* of a sequent by means of a Boolean flag w — sequents have the shape $\Gamma \,;\, \Delta \longrightarrow^{w} \gamma$, where γ stands schematically for $\cdot$ or a formula C; we shall abuse notation slightly and write $C \supseteq \gamma$ to mean γ is either $\cdot$ or C. We define the precise nature of weakness by means of a correspondence with the backward sequent arrow of the previous section.

$$\Gamma \,;\, \Delta \longrightarrow^{0} C \qquad \text{corresponds to} \qquad \Gamma' \,;\, \Delta \Longrightarrow C \text{ for any } \Gamma' \supseteq \Gamma \tag{lin}$$

$$\Gamma \,;\, \Delta \longrightarrow^{1} \gamma \qquad \text{corresponds to} \qquad \Gamma' \,;\, \Delta' \Longrightarrow C \text{ for any } \Delta' \supseteq \Delta, \Gamma' \supseteq \Gamma \text{ and } C \supseteq \gamma \tag{weak}$$

We call 1-annotated sequents *weak sequents*. As a global consistency condition for the logic, we shall ensure that $\cdot \,;\, \cdot \longrightarrow^{1} \cdot$ remains unprovable. The judgemental rules require an update. Initial sequents clearly cannot satisfy (weak) because of the requirement of a *single* linear hypothesis; moreover, initial sequents don't determine any members of the unrestricted context. Therefore, we give initial sequents the annotation 0, and leave the unrestricted context empty.

$$\frac{}{\cdot \,;\, A \longrightarrow^{0} A}\ \text{init}$$

For the copy rule, the nature of the annotation plays no significant role, so we merely propagate it from the premiss to the conclusion. Furthermore, because in the premiss we already have the formula A in the linear zone, we don't include it in the unrestricted zone also. If a single formula gets copied twice, we contract ($\cup$) the unrestricted context to avoid generating more than one copy of the formula.

$$\frac{\Gamma \,;\, \Delta, A \longrightarrow^{w} \gamma}{\Gamma \cup \{A\} \,;\, \Delta \longrightarrow^{w} \gamma}\ \text{copy}$$

For the logical rules, we once again observe three major categories. For the multiplicative connectives, the weakness of any premiss suffices to produce a weak conclusion, so we obtain a disjunction of the weakening annotations. For example, for $\otimes R$ we write

$$\frac{\Gamma\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Delta' \longrightarrow^{w_2} B}{\Gamma \cup \Gamma'\,;\Delta, \Delta' \longrightarrow^{w_1 \vee w_2} A \otimes B}\ \otimes R$$

The formulas A and B must exist in the conclusions of the premisses; allowing an empty conclusion for any of the premisses makes the conclusion sequent an explicitly weaker form of one premiss, and therefore will generate no new knowledge.

We perform a further optimisation directly in the logic. Simply applying the correspondence (weak) to generate the forward rules corresponding to $\multimap R$ produces the following pair:

$$\frac{\Gamma\,;\Delta, A \longrightarrow^{1} B}{\Gamma\,;\Delta \longrightarrow^{1} A \multimap B}\ \multimap R \qquad \frac{\Gamma\,;\Delta \longrightarrow^{1} B}{\Gamma\,;\Delta \longrightarrow^{1} A \multimap B}\ \multimap R'$$

The second of these utilises the implicit weakening in (weak) by not requiring A in the linear context of the premiss. However, if A does indeed occur in Δ for any reason, then we have a choice of which rule to apply. The conclusion of applying $\multimap R$ turns out to be stronger than with $\multimap R'$ in the sense of (weak), *i.e.*, the linear context in the latter contains an extra A. To prevent this case, we restrict the application of $\multimap R'$ with a side-condition of *negative existence*.

$$\frac{\Gamma\,;\Delta \longrightarrow^{1} B \qquad A \notin \Delta}{\Gamma\,;\Delta \longrightarrow^{1} A \multimap B}\ \multimap R'$$

We don't treat such negative existence conditions as facts in the same way as sequents – for example, in an implementation, we never enter them into a fact database for use in subsumption checks.

For the additive rules, we notice that the (implicit) weakenability of weak sequents manifests in the forward direction as a *weak equality* for the linear contexts, which we write $\Delta/w_1 = \Delta'/w_2$, with the following definition

$$\begin{array}{lll} \Delta/0 = \Delta'/0 & \text{if} & \Delta = \Delta' \\ \Delta/1 = \Delta'/0 & \text{if} & \Delta \subseteq \Delta' \\ \Delta/1 = \Delta'/1 & \text{always} & \end{array}$$

This gives us the following form of the $\&R$ rule:

$$\frac{\Gamma\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Delta' \longrightarrow^{w_2} B \qquad \Delta/w_1 = \Delta'/w_2}{\Gamma \cup \Gamma'\,;\Delta \cup \Delta' \longrightarrow^{w_1 \wedge w_2} A \,\&\, B}\ \&R$$

where we reuse $\cup$ for the linear contexts to mean a multiplicity-respecting union of elements, *i.e.*, if A occurs m times in Δ and n times in Δ', then it occurs $\max(m, n)$ times in $\Delta \cup \Delta'$.[4] Dual to the multiplicative connectives, the conclusion of additive rules receive a conjunction of the annotations of the premisses. For $\top$ (the unit of $\&$), we use the annotation 1 (the unit of $\wedge$). The annotations propagate for the single-premiss rules, with similar justification as for the copy rule.

The right rule for ! requires a special restriction for soundness: we cannot allow an annotation of 1 in the conclusion. For simple counterexamples, neither of the following derivations is

[4] We reuse the same symbol $\cup$ because unioning for the linear context doesn't differ operationally from that of the unrestricted context. The "set" interpretation for the unrestricted context arises out of contraction inherent in the copy rule.

Judgemental rules

$$\frac{}{\cdot\,;A \longrightarrow^{0} A}\;\text{init} \qquad \frac{\Gamma\,;\Delta, A \longrightarrow^{w} \gamma}{\Gamma \cup \{A\}\,;\Delta \longrightarrow^{w} \gamma}\;\text{copy}$$

Multiplicative connectives

$$\frac{\Gamma\,;\Delta, A \longrightarrow^{1} \gamma \qquad B \notin \Delta}{\Gamma\,;\Delta, A \otimes B \longrightarrow^{1} \gamma}\;\otimes L_1 \qquad \frac{\Gamma\,;\Delta, B \longrightarrow^{1} \gamma \qquad A \notin \Delta}{\Gamma\,;\Delta, A \otimes B \longrightarrow^{1} \gamma}\;\otimes L_2$$

$$\frac{\Gamma\,;\Delta, A, B \longrightarrow^{w} \gamma}{\Gamma\,;\Delta, A \otimes B \longrightarrow^{w} \gamma}\;\otimes L \qquad \frac{\Gamma\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Delta' \longrightarrow^{w_2} B}{\Gamma \cup \Gamma'\,;\Delta, \Delta' \longrightarrow^{w_1 \vee w_2} A \otimes B}\;\otimes R$$

$$\frac{\Gamma\,;\Delta \longrightarrow^{0} C}{\Gamma\,;\Delta, \mathbf{1} \longrightarrow^{0} C}\;\mathbf{1}L \qquad \frac{}{\cdot\,;\cdot \longrightarrow^{0} \mathbf{1}}\;\mathbf{1}R$$

$$\frac{\Gamma\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Delta', B \longrightarrow^{w_2} \gamma}{\Gamma \cup \Gamma'\,;\Delta, \Delta', A \multimap B \longrightarrow^{w_1 \vee w_2} \gamma}\;\multimap L$$

$$\frac{\Gamma\,;\Delta, A \longrightarrow^{w} \gamma \qquad C \supseteq \gamma}{\Gamma\,;\Delta \longrightarrow^{w} A \multimap C}\;\multimap R \qquad \frac{\Gamma\,;\Delta \longrightarrow^{1} C \qquad A \notin \Delta}{\Gamma\,;\Delta \longrightarrow^{1} A \multimap C}\;\multimap R'$$

Additive connectives

$$\frac{\Gamma\,;\Delta, A \longrightarrow^{w} \gamma}{\Gamma\,;\Delta, A \,\&\, B \longrightarrow^{w} \gamma}\;\& L_1 \qquad \frac{\Gamma\,;\Delta, B \longrightarrow^{w} \gamma}{\Gamma\,;\Delta, A \,\&\, B \longrightarrow^{w} \gamma}\;\& L_2$$

$$\frac{\Gamma\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Delta' \longrightarrow^{w_2} B \qquad \Delta/w_1 = \Delta'/w_2}{\Gamma \cup \Gamma'\,;\Delta \cup \Delta' \longrightarrow^{w_1 \wedge w_2} A \,\&\, B}\;\& R \qquad \frac{}{\cdot\,;\cdot \longrightarrow^{1} \top}\;\top R$$

$$\frac{\Gamma\,;\Delta, A \longrightarrow^{w_1} \gamma \qquad \Gamma'\,;\Delta', B \longrightarrow^{w_2} \gamma' \qquad \Delta/w_1 = \Delta'/w_2}{\Gamma \cup \Gamma'\,;\Delta \cup \Delta', A \oplus B \longrightarrow^{w_1 \wedge w_2} \gamma \cup \gamma'}\;\oplus L$$

$$\frac{\Gamma\,;\Delta \longrightarrow^{w} A}{\Gamma\,;\Delta \longrightarrow^{w} A \oplus B}\;\oplus R_1 \qquad \frac{\Gamma\,;\Delta \longrightarrow^{w} B}{\Gamma\,;\Delta \longrightarrow^{w} A \oplus B}\;\oplus R_2 \qquad \frac{}{\cdot\,;\mathbf{0} \longrightarrow^{1} \cdot}\;\mathbf{0}L$$

Exponential modality

$$\frac{\Gamma, A\,;\Delta \longrightarrow^{w} \gamma}{\Gamma\,;\Delta, !A \longrightarrow^{w} \gamma}\;!L \qquad \frac{\Gamma\,;\Delta \longrightarrow^{1} \gamma \qquad A \notin \Gamma}{\Gamma\,;\Delta, !A \longrightarrow^{1} \gamma}\;!L' \qquad \frac{\Gamma\,;\cdot \longrightarrow^{w} A}{\Gamma\,;\cdot \longrightarrow^{0} !A}\;!R$$

Figure 2: Rules for a forward sequent calculus

sound with respect to (lin) and (weak):

$$\frac{\cdot\,;\cdot \longrightarrow^{0} \mathbf{1}}{\cdot\,;\cdot \longrightarrow^{1} !\mathbf{1}} \qquad \frac{\cdot\,;\cdot \longrightarrow^{1} \top}{\cdot\,;\cdot \longrightarrow^{1} !\top}$$

Figure 2 lists the complete collection of rules for the forward calculus. Except for the initial rule, rules in the forward direction never introduce any unaccounted resources into the conclusion. As a direct result, the conclusion of every rule contains new formulas, making search knowledge-monotonic. The collection of initial sequents forms the input of a forward search procedure, so these require careful selection. We discuss this in more detail in section 5 when we develop the inverse method search procedure.

Soundness and completeness. We now formalise the informally stated correspondences in (lin) and (weak). Weak sequents in the soundness theorem require a somewhat unusual form in order to facilitate the structural induction.

Theorem 3 (Soundness).
1. *If* $\Gamma\,;\Delta \longrightarrow^0 C$*, then* $\Gamma\,;\Delta \Longrightarrow C$.
2. *If* $\Gamma\,;\Delta \longrightarrow^1 \gamma$*, then for any* $\Delta' \supseteq \Delta$ *and* $C \supseteq \gamma$, $\Gamma\,;\Delta' \Longrightarrow C$.

Proof. By induction on the derivation $\mathcal{F}_1 :: \Gamma\,;\Delta \longrightarrow^0 C$ or $\mathcal{F}_2 :: \Gamma\,;\Delta \longrightarrow^1 \gamma$. Since every rule in the forward direction has more restrictions than the corresponding rule in the backward direction, the induction proceeds in a parallel fashion for all rules. We illustrate with just one such rule, $\&R$. Suppose we have:

$$\mathcal{F} \;=\; \frac{\Gamma\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Delta' \longrightarrow^{w_2} B \qquad \Delta/w_1 = \Delta'/w_2}{\Gamma \cup \Gamma'\,;\Delta \cup \Delta' \longrightarrow^{w_1 \wedge w_2} A \,\&\, B}\;\&R$$

If $w_1 \wedge w_2 = 1$, then by part (2) on the premisses, we know that:

$$\begin{array}{c}\text{for any } \Delta_1 \supseteq \Delta,\;\; \Gamma\,;\Delta_1 \Longrightarrow A\\ \text{for any } \Delta_2 \supseteq \Delta',\;\; \Gamma'\,;\Delta_2 \Longrightarrow B\end{array}$$

Given $\Delta'' \supseteq \Delta \cup \Delta'$, instantiate both Δ_1 and Δ_2 with Δ'' and use $\&R$ and weakening for the unrestricted context in the backward calculus to obtain $\Gamma\cup\Gamma';\Delta'' \Longrightarrow A\&B$, which satisfies part (2). On the other hand, if $w_1 \wedge w_2 = 0$, then we have two possibilities. If both $w_1 = 0$ and $w_2 = 0$, then the rule in the forward direction becomes structurally isomorphic to that in the backward direction if we weaken both unrestricted contexts to $\Gamma \cup \Gamma'$, so part (1) holds immediately. If $w_1 = 0$ and $w_2 = 1$, then by part (2) on the first premiss, and part (1) on the second, we know that:

$$\begin{array}{c}\Gamma\,;\Delta \Longrightarrow A\\ \text{for any } \Delta_2 \supseteq \Delta',\;\; \Gamma'\,;\Delta_2 \Longrightarrow B\end{array}$$

By the matching criterion $\Delta/0 = \Delta'/1$ we know $\Delta' \subseteq \Delta$, so $\Delta \cup \Delta' = \Delta$. Thus, we instantiate Δ_2 with Δ, and then use $\&R$ and weakening for the unrestricted context in the backward calculus to get $\Gamma \cup \Gamma'\,;\Delta \Longrightarrow A \,\&\, B$, which satisfies part (1). This last case demonstrates the need for the strengthened induction hypotheses in part (2). □

Dually, for the completeness theorem, we account for the case that the forward calculus proves a stronger form of the goal sequent.

Theorem 4 (Completeness). *If* $\Gamma\,;\Delta \Longrightarrow C$*, then for some* $\Gamma' \subseteq \Gamma$,
1. *either* $\Gamma';\Delta \longrightarrow^0 C$,
2. *or* $\Gamma'\,;\Delta' \longrightarrow^1 \gamma$ *for some* $\Delta' \subseteq \Delta$ *and* $\gamma \subseteq C$.

Proof. By structural induction on the derivation $\mathcal{B}$ of $\Gamma\,;\Delta \Longrightarrow C$. For every logical rule of the backward calculus, the conclusion has a (not necessarily strictly) weaker form than the corresponding rule(s) in the forward calculus, so the induction proceeds straightforwardly. For the judgemental rules, init has a trivial verification. The copy rule presents the only interesting case.

$$\mathcal{B} \;=\; \frac{\begin{array}{c}\mathcal{B}'\\ \Gamma \cup \{A\}\,;\Delta, A \Longrightarrow C\end{array}}{\Gamma \cup \{A\}\,;\Delta \Longrightarrow C}\;\text{copy}$$

If we know that $\Gamma \cup \{A\}\,;\Delta, A \longrightarrow^0 C$, then by the copy rule in the forward direction, we have $\Gamma \cup \{A\}\,;\Delta \longrightarrow^0 C$, which satisfies case (1). On the other hand, if for some $\Gamma' \subseteq \Gamma$, $\Delta' \subseteq (\Delta, A)$ and $\gamma \subseteq C$, we know that $\Gamma'\,;\Delta' \longrightarrow^1 \gamma$, then we have two cases:

- $A \notin \Delta'$, which case trivially satisfies this theorem because $\Delta' \subseteq \Delta$ and $\Gamma' \subseteq \Gamma \cup \{A\}$.
- $A \in \Delta'$, in which case by the copy rule in the forward direction, $\Gamma', A\,;\Delta' \backslash A \longrightarrow^1 \gamma$. We then just note that $\Gamma' \cup \{A\} \subseteq \Gamma \cup \{A\}$. □

Structural Properties. The lazier nature of resources in the forward direction necessitates an update to the structural meta-theorems about the logic, particularly the cut principles. The forward direction aims to disallow weakening in all forms, including as admissible structural rules. The question of contraction never arises in the forward direction because the conclusion sequents inferred by all rules have implicit contraction. An implementation of this calculus in a theorem prover would either contract *eagerly* after every rule application, or use *lazy contraction*, the latter of which presents an interesting area for logical treatment.

Theorem 5 (Cut).
1. *If* $\Gamma\,;\Delta \longrightarrow^{w_1} A$ *and* $\Gamma'\,;\Delta', A \longrightarrow^{w_2} \gamma$, *then* $\Gamma \cup \Gamma'\,;\Delta, \Delta' \longrightarrow^{w_1 \vee w_2} \gamma$.
2. *If* $\Gamma\,;\cdot \longrightarrow^{w} A$ *and* $\Gamma', A\,;\Delta \longrightarrow^{w'} \gamma$, *then* $\Gamma \cup \Gamma'\,;\Delta \longrightarrow^{w'} \gamma$.

Proof. By lexicographic structural induction on the two given derivations. All the cases of the proof follow a straightforward pattern, similar to proofs of theorem 2, for example in [8]. □

The forward calculus as presented suffers from no structural non-determinism, but $\mathbf{1}L$ does present unknown-use nondeterminism as described earlier. In the next section we shall argue for a removal of the $\mathbf{1}L$ rule by making its applications implicit in the calculus.

4 Affine resources

In the absence of negative **1** in the logic, the forward calculus of the previous section suffices to remove all resource non-determinism. With the addition of **1**, particularly negative occurrences, we have the problem of affine non-determinism, as mentioned in section 2.2, which arises from the interaction of **1** with other connectives. For most connectives, **1** has only a *unitary* function, where an equivalent proposition can be found which doesn't require (that particular instance of) **1**. The full list of such equivalences is as follows:[5]

$$A \otimes \mathbf{1} \equiv \mathbf{1} \equiv \mathbf{1} \otimes A \qquad \mathbf{1} \multimap A \equiv A \qquad \mathbf{1} \,\&\, \mathbf{1} \equiv \mathbf{1} \qquad \mathbf{1} \oplus \mathbf{1} \equiv \mathbf{1} \qquad !\mathbf{1} \equiv \mathbf{1}$$

For the rest of this paper we assume a logic in **1**-normal form (**1** nf), which we define as that fragment without unitary uses of **1**. This simpler fragment allows an examination of the occurrences of **1** actually relevant to resource management.

An interesting class of propositions have the form $A \& \mathbf{1}$ or $\mathbf{1} \& A$; as a resource, $A \& \mathbf{1}$ provides a choice of either using A linearly in the proof, or not using A at all, *i.e.*, it encodes an *at-most one use* or *affine* interpretation. Indeed, such propositions allow us to recover affine logic in the exact setting of linear logic, by translating affine implications $A \to B$ into $A \,\&\, \mathbf{1} \multimap B$. There is another, more popular embedding of affine logic into linear logic that translates $A \to B$ into $A \multimap B \otimes \top$. The difference between the two encodings manifests as a choice between a *local* and a *global* translation — translating into $A \,\&\, \mathbf{1} \multimap B$ doesn't destroy the linear nature of resources, but $A \multimap B \otimes \top$ makes every resource affine because of the presence of positive $\otimes\top$. Yet, and despite the fact that positive $\top$ complicates backwards search, encodings in logic programming

[5] In the presence of quantifiers, we have some additional equivalences: $\exists x.\mathbf{1} \equiv \mathbf{1}$ and $\forall x.\mathbf{1} \equiv \mathbf{1}$

languages like Lolli prefer the second encoding. Rather than repeat this approach and disallow negative **1**, for the rest of this section, we examine the nature of resource non-determinism caused by negative **1**.

4.1 Characterising non-unitary uses of 1

First, consider the effect of removing $\mathbf{1}L$ entirely from the logic. In the $\mathbf{1}$nf fragment, only the following instances of **1** remain: $A \multimap \mathbf{1}$, $A \,\&\, \mathbf{1}$, $\mathbf{1} \,\&\, A$, $A \oplus \mathbf{1}$, $\mathbf{1} \oplus A$ and the formula **1** itself. On the right, the corresponding rules are fully deterministic. On the left, all of these forms – except **1** itself – have the following specialized rules:

$$\frac{\Gamma\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Delta' \Longrightarrow C}{\Gamma\,;\Delta,\Delta', A \multimap \mathbf{1} \Longrightarrow C}\ {\multimap}\mathbf{1}L$$

$$\frac{\Gamma\,;\Delta \Longrightarrow C}{\Gamma\,;\Delta, A \,\&\, \mathbf{1} \Longrightarrow C}\ \&\mathbf{1}L_1 \qquad \frac{\Gamma\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Delta, A \,\&\, \mathbf{1} \Longrightarrow C}\ \&\mathbf{1}L_2$$

$$\frac{\Gamma\,;\Delta \Longrightarrow C}{\Gamma\,;\Delta, \mathbf{1} \,\&\, A \Longrightarrow C}\ \mathbf{1}\&L_1 \qquad \frac{\Gamma\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Delta, \mathbf{1} \,\&\, A \Longrightarrow C}\ \mathbf{1}\&L_2$$

$$\frac{\Gamma\,;\Delta, A \Longrightarrow C \qquad \Gamma\,;\Delta \Longrightarrow C}{\Gamma\,;\Delta, A \oplus \mathbf{1} \Longrightarrow C}\ \oplus\mathbf{1}L \qquad \frac{\Gamma\,;\Delta \Longrightarrow C \qquad \Gamma\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Delta, \mathbf{1} \oplus A \Longrightarrow C}\ \mathbf{1}\oplus L$$

We have described the situation with $\&\mathbf{1}$ and $\mathbf{1}\&$ before and clearly visible above in the pair of rules $\mathbf{1}\&L_1$ and $\&\mathbf{1}L_1$, formulas of the form $A \,\&\, \mathbf{1}$ define an affine interpretation for the resource A. We examine this case in detail in the next section. For $\mathbf{1} \oplus L$ and $\oplus\mathbf{1}L$, the premisses appear to give the formula A a meaning of optional use – we can prove the conclusion C both in the presence and absence of A. In fact, one might view this kind of optional use (one *and* zero times) as the external version of the affine case (at-most one use); thus, one might imagine a substructural logic where external options are internalised using locally sound and complete introduction/elimination rules.[6] Fortunately, the treatment of the affine case in the next section provides a satisfactory answer for the optional case also.

For the ${\multimap}\mathbf{1}L$ rule, we don't have a complete answer. We can certainly construct examples with uncontrolled iteration of this rule, as follows.

$$\cfrac{\Gamma\,;\Delta \Longrightarrow A \qquad \cfrac{\Gamma\,;\Delta \Longrightarrow A \qquad \cfrac{\Gamma\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Delta', \mathbf{1} \Longrightarrow C}{\Gamma\,;\Delta, \Delta', A \multimap \mathbf{1} \Longrightarrow C}}{\Gamma\,;\Delta, \Delta, \Delta', A \multimap \mathbf{1}, A \multimap \mathbf{1} \Longrightarrow C}}{\vdots}$$

We leave a treatment of this and other sources of unknown use non-determinism to future work, but note that that no complete solution can exist because of the undecidability of multiplicative-additive-exponential linear logic. On the other hand, categorizing and solving other kinds of unknown use non-determinism can give decision procedures for larger fragments. These investigations will depend on the need for the increased expressivity; for example, by showing how a negative $A \multimap \mathbf{1}$ gives a more natural encoding than other possibilities.[7]

In the next section we give first a backward and then a forward calculus to handle the affine case. One particular note – we remove all hypotheses **1** in the ultimate goal sequent. Thus, we never need to use the $\mathbf{1}L$ rule at all, so we just discard it. We can easily add these extra **1**s to the goal sequent if needed after search completes.

[6] To the best of our knowledge, no treatment of such a *logic of optional use* exists in the literature.

[7] Such formulas don't exist in the uniform fragment! [7]

4.2 Affine zones for the backward calculus

To handle the affine resources, we insert a new *affine zone* Ψ among the hypotheses of sequents, giving the following shape for sequents: $\Gamma\,;\Psi\,;\Delta \Longrightarrow C$. We view this affine zone as a multiset of formulas, just like the linear zone, but with an additional structurally admissible rule of weakening (theorem 6).

For the judgemental rules, we have a rule of *affine dereliction* to turn an affine hypothesis into a linear hypothesis. This corresponds to committing to an actual use of the affine resource.

$$\frac{\Gamma\,;\Psi\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Psi, A\,;\Delta \Longrightarrow C}\ \text{aff-dl}$$

On the other hand, affine resources can remain unused because we allow any number of them to "escape" through initial and other axiomatic sequents:

$$\frac{}{\Gamma\,;\Psi\,;A \Longrightarrow A}\ \text{init} \qquad \frac{}{\Gamma\,;\Psi\,;\cdot \Longrightarrow \mathbf{1}}\ \mathbf{1}R \qquad \frac{}{\Gamma\,;\Psi\,;\Delta \Longrightarrow \top}\ \top R \qquad \frac{}{\Gamma\,;\Psi\,;\Delta, \mathbf{0} \Longrightarrow C}\ \mathbf{0}L$$

We use the operators $\mathbf{1} \mathbin{\&} -$ (and $- \mathbin{\&} \mathbf{1}$) to internalise affine use. However, unlike the internalisation of unrestricted use as !, we restrict these to left-formulas (*i.e.*, as negative formulas in the eventual goal), and use the usual right rules for $\&$ and $\mathbf{1}$ to infer $\mathbf{1} \mathbin{\&} A$ on the right. Figure 3 lists all the rules. No rules require $\mathbf{1}$ as a resource, but we have (derived) rules for the situations where $\mathbf{1}$ occurs as an operand of the principal connective. To enforce an absence of $\mathbf{1}$ among the hypotheses, we add some side-conditions to $\&L$.

This presentation of a resource-management motivated three zoned logic bears a strong resemblance to a similar system of Cervesato *et al.* [7] for the domain of (backward-reasoning) linear logic programming in the uniform fragment. The primary difference lies in the interpreation of the new zone – *strict* in [7] versus affine in this paper. The design of their three-zoned system derives its primary motivation from the nature of $\&$ and $\top$, with the strict contexts designed to handle the additive nature of $\&$. In a similar sense in which strict contexts arise for a systematic approach to resource management in backward search, we claim that affine contexts arise naturally in the setting of forward search.

Structural properties. We obtain an easily shown admissible structural weakening theorem for the affine context, in addition to the straightforward extension of the structural properties for the unrestricted context in theorem 1 to the three-zoned setting.

Theorem 6 (New Structural Properties).
(Ψ-weakening) If $\Gamma\,;\Psi\,;\Delta \Longrightarrow C$ then $\Gamma\,;\Psi, A\,;\Delta \Longrightarrow C$ for any $A \neq \mathbf{1}$. □

Cut also has new cases to handle cutting affine resources, and a special case that corresponds to having $\mathbf{1}$ as a hypothesis in the two-zoned calculus.[8]

Theorem 7 (Cut). *For $A \neq \mathbf{1}$,*

1. *If $\Gamma\,;\Psi\,;\Delta \Longrightarrow \mathbf{1}$ and $\Gamma\,;\Psi'\,;\Delta' \Longrightarrow C$, then $\Gamma\,;\Psi, \Psi'\,;\Delta, \Delta' \Longrightarrow C$.*
2. *If $\Gamma\,;\Psi\,;\Delta \Longrightarrow A$ and $\Gamma\,;\Psi'\,;\Delta', A \Longrightarrow C$, then $\Gamma\,;\Psi, \Psi'\,;\Delta, \Delta' \Longrightarrow C$.*
3. *If $\Gamma\,;\Psi\,;\cdot \Longrightarrow A$ and $\Gamma\,;\Psi', A\,;\Delta \Longrightarrow C$, then $\Gamma\,;\Psi, \Psi'\,;\Delta \Longrightarrow C$.*
4. *If $\Gamma\,;\cdot\,;\cdot \Longrightarrow A$ and $\Gamma, A\,;\Psi\,;\Delta \Longrightarrow C$, then $\Gamma\,;\Psi\,;\Delta \Longrightarrow C$.*

[8] Incidentally, this latter case resembles Girard's MIX rule for classical two-sided sequent calculi:

$$\frac{\Gamma \Longrightarrow \Delta \qquad \Gamma' \Longrightarrow \Delta'}{\Gamma, \Gamma' \Longrightarrow \Delta, \Delta'}\ \text{MIX}$$

The logical meaning of MIX has recently been investigated by Chang *et al.*. [8].

Judgemental Rules

$$\frac{}{\Gamma\,;\Psi\,;A \Longrightarrow A}\ \text{init} \qquad \frac{\Gamma, A\,;\Psi\,;\Delta, A \Longrightarrow C}{\Gamma, A\,;\Psi\,;\Delta \Longrightarrow C}\ \text{copy} \qquad \frac{\Gamma\,;\Psi\,;\Delta, A \Longrightarrow C}{\Gamma\,;\Psi, A\,;\Delta \Longrightarrow C}\ \text{aff-dl}$$

Multiplicative Rules

$$\frac{\Gamma\,;\Psi\,;\Delta, A, B \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, A \otimes B \Longrightarrow C}\ \otimes L \qquad \frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Psi'\,;\Delta' \Longrightarrow B}{\Gamma\,;\Psi, \Psi'\,;\Delta, \Delta' \Longrightarrow A \otimes B}\ \otimes R$$

$$\frac{}{\Gamma\,;\Psi\,;\cdot \Longrightarrow \mathbf{1}}\ \mathbf{1}R$$

$$\frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Psi'\,;\Delta, B \Longrightarrow C}{\Gamma\,;\Psi, \Psi'\,;\Delta, \Delta', A \multimap B \Longrightarrow C}\ \multimap L$$

$$\frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Psi'\,;\Delta \Longrightarrow C}{\Gamma\,;\Psi, \Psi'\,;\Delta, \Delta', A \multimap \mathbf{1} \Longrightarrow C}\ \mathbf{1}\!\multimap\! L \qquad \frac{\Gamma\,;\Psi\,;\Delta, A \Longrightarrow B}{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \multimap B}\ \multimap R$$

Additive Rules

$$\frac{\Gamma\,;\Psi\,;\Delta, A \Longrightarrow C \qquad B \neq \mathbf{1}}{\Gamma\,;\Psi\,;\Delta, A \,\&\, B \Longrightarrow C}\ \& L_1 \qquad \frac{\Gamma\,;\Psi, A\,;\Delta \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, A \,\&\, \mathbf{1} \Longrightarrow C}\ \mathbf{1}\& L$$

$$\frac{\Gamma\,;\Psi\,;\Delta, B \Longrightarrow C \qquad A \neq \mathbf{1}}{\Gamma\,;\Psi\,;\Delta, A \,\&\, B \Longrightarrow C}\ \& L_2 \qquad \frac{\Gamma\,;\Psi, B\,;\Delta \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, \mathbf{1} \,\&\, B \Longrightarrow C}\ \&\mathbf{1} L$$

$$\frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \qquad \Gamma\,;\Psi\,;\Delta \Longrightarrow B}{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \,\&\, B}\ \& R \qquad \frac{}{\Gamma\,;\Psi\,;\Delta \Longrightarrow \top}\ \top R$$

$$\frac{\Gamma\,;\Psi\,;\Delta, A \Longrightarrow C \qquad \Gamma\,;\Psi\,;\Delta, B \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, A \oplus B \Longrightarrow C}\ \oplus L \qquad \frac{}{\Gamma\,;\Psi\,;\Delta, \mathbf{0} \Longrightarrow C}\ \mathbf{0}L$$

$$\frac{\Gamma\,;\Psi\,;\Delta, A \Longrightarrow C \qquad \Gamma\,;\Psi\,;\Delta \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, A \oplus \mathbf{1} \Longrightarrow C}\ \oplus\mathbf{1}L \qquad \frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow C \qquad \Gamma\,;\Psi\,;\Delta, B \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, \mathbf{1} \oplus B \Longrightarrow C}\ \mathbf{1}\oplus L$$

$$\frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow A}{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \oplus B}\ \oplus R_1 \qquad \frac{\Gamma\,;\Psi\,;\Delta \Longrightarrow B}{\Gamma\,;\Psi\,;\Delta \Longrightarrow A \oplus B}\ \oplus R_2$$

Exponential Rules

$$\frac{\Gamma, A\,;\Psi\,;\Delta \Longrightarrow C}{\Gamma\,;\Psi\,;\Delta, !A \Longrightarrow C}\ !L \qquad \frac{\Gamma\,;\cdot\,;\cdot \Longrightarrow A}{\Gamma\,;\Psi\,;\cdot \Longrightarrow !A}\ !R$$

Figure 3: Backward sequent calculus extended with affine contexts

Proof. By a lexicographic structural induction; e may appeal to the induction hypothesis for a smaller cut formula, or else if the cut formula is the same then the cases in the order presented, or else if one of the derivations becomes smaller. The proof in detail isn't particularly surprising; see [22, 8] for similar proofs. ☐

Correctness. In order to show soundness, we employ a useful shorthand, $\Psi \mathbin{\&} \mathbf{1}$, to stand for a context consisting of every proposition A in Ψ replaced with $A \mathbin{\&} \mathbf{1}$ or $\mathbf{1} \mathbin{\&} A$, as appropriate. We obtain a succinct soundness theorem.

Theorem 8 (Soundness). *If* $\Gamma\,;\Psi\,;\Delta \Longrightarrow C$*, then* $\Gamma\,;\Psi \mathbin{\&} \mathbf{1}, \Delta \Longrightarrow C$*.*

Proof. By induction on the structure of the derivation $\mathcal{D}$ of $\Gamma\,;\Psi\,;\Delta \Longrightarrow C$. All cases except init and $\mathbf{1}R$ have trivial verifications. For these two rules, we first chain an alternating sequence of $\mathbf{1}L$ and $\&L_i$ to account for all linear hypotheses in $\Psi \mathbin{\&} \mathbf{1}$, and then use init or $\mathbf{1}R$ in the two-zoned system, respectively. ☐

We also obtain a strong completeness theorem, schematic for affine contexts.

Theorem 9 (Completeness). *If* $\Gamma\,;\Delta \Longrightarrow C$*, then* $\Gamma\,;\Psi\,;\Delta \Longrightarrow C$ *for any* Ψ*.*

Proof. By straightforward structural induction on the derivation of $\Gamma\,;\Delta \Longrightarrow C$. ☐

This section has served primarily a motivational purpose; we now turn our attention to our original goal of controlling affine non-determinism in forward reasoning.

4.3 Affine contexts in the forward calculus

Like before with the unrestricted contexts, in the forward direction we create only that subset of the affine context that we can infer from other premisses and the conclusion, with the sole difference that in order to maintain the affine interpretation, we treat the affine context multiplicatively, Rules for formulas with $\mathbf{1}$ as an operand require particular attention; for example, consider the following tempting possibilities for $A \mathbin{\&} \mathbf{1}$:

$$\frac{\Gamma\,;\Psi, A\,;\Delta \longrightarrow^{w} \gamma}{\Gamma\,;\Psi\,;\Delta, A \mathbin{\&} \mathbf{1} \longrightarrow^{w} \gamma}\ \&\mathbf{1}L \qquad \frac{\Gamma\,;\Psi\,;\Delta \longrightarrow^{w} \gamma \qquad A \notin \Psi}{\Gamma\,;\Psi\,;\Delta, A \mathbin{\&} \mathbf{1} \longrightarrow^{w} \gamma}\ \&\mathbf{1}L'$$

The $\&\mathbf{1}L'$ rule lacks any structural control on the number of occurrences of $A \mathbin{\&} \mathbf{1}$. We have already seen this problem before in the presence of the $\mathbf{1}L$ rule, removing which makes the iterative nature of this rule obvious. We attack this problem by treating this second instance as a kind of weakening; thus, we use the second of the above rules only after we have more information about the multiplicity of $A \mathbin{\&} \mathbf{1}$.

When do we learn anything about the multiplicity of a formula? Certainly, we can never infer the exact multiplicity of any given formula by just looking at the final goal sequent – this would make the fragment decidable, and we already know that linear logic in the presence of additive connectives is undecidable. However, we do know that the multiplicity of linear $A \mathbin{\&} \mathbf{1}$ exceeds the multiplicity of A in the affine context; this suffices to control the iteration of $\&\mathbf{1}L'$ as follows – remove this rule entirely from consideration during search, and assume for every other rule with a weak premiss that the formula $A \mathbin{\&} \mathbf{1}$ exists implicitly in the linear context.

Of course, in the proof theory it becomes tedious to modify every logical rule with the tests and side conditions corresponding to these implicitly present affine resources, so for this paper we introduce a layer of abstraction between the inference rule and the matching conditions that enable the rule. This gives us an interesting *logic of matching conditions*. Conceptually, matching conditions in the forward direction generalize the notion of *occurrence* in a context, written exactly like adjunctions (Γ, A) for historical reasons. This notation makes perfect sense in backward reasoning, because the contexts, ambiently or explicitly, serve as parameters for the search procedure. In contrast, because information flows in the opposite direction in forward reasoning, inference rules construct the contexts of the conclusion from those of the premisses, treating contexts as localized (first-class) objects.[9] As a matching condition, adjunction describes only the rather simple condition of occurrence.

In order to obtain a more complex and process-oriented view of matching, we define a new judgement on zoned contexts $\Gamma\,;\Psi\,;\Delta$ (written Υ):

$$\Upsilon \vdash \Upsilon' + \Delta$$

which we read "Υ *admits the adjunction* Υ', Δ." This judgement takes Υ and Δ as input, and produces the output Υ' if it succeeds. The rules for this judgement proceed purely in the bottom-up direction, with the output Υ' read off from the completed derivation. The key property maintained by this matching judgement is the matching lemma (11); roughly, it describes an allowable and evident weakening.

The simplest rule for this judgement merely admits the trivial adjunction.

$$\frac{}{\Upsilon \vdash \Upsilon + \cdot}\ \vdash_{=}$$

The remaining rules fall into three categories for the three different zones. For the linear zone:

$$\frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta'}{\Gamma\,;\Psi\,;\Delta, A \vdash \Upsilon + \Delta', A}\ \vdash_{\text{linear}}$$

For the affine zone:

$$\frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta'}{\Gamma\,;\Psi, A\,;\Delta \vdash \Upsilon + \Delta', A \mathbin{\&} \mathbf{1}}\ \vdash_{\&\mathbf{1}} \qquad \frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta'}{\Gamma\,;\Psi, A\,;\Delta \vdash \Upsilon + \Delta', \mathbf{1} \mathbin{\&} A}\ \vdash_{\mathbf{1}\&}$$

$$\frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta' \qquad A \notin \Psi}{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta', A \mathbin{\&} \mathbf{1}}\ \vdash'_{\&\mathbf{1}} \qquad \frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta' \qquad A \notin \Psi}{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta', \mathbf{1} \mathbin{\&} A}\ \vdash'_{\mathbf{1}\&}$$

For the unrestricted zone:

$$\frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta'}{\Gamma, A\,;\Psi\,;\Delta \vdash \Upsilon + \Delta', !A}\ \vdash_{!} \qquad \frac{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta' \qquad A \notin \Gamma}{\Gamma\,;\Psi\,;\Delta \vdash \Upsilon + \Delta', !A}\ \vdash'_{!}$$

In section 3 (figure 2) we included additional conditions of non-existence for the multiplicative rules $\otimes L_1$, $\otimes L_2$ and $\multimap R'$. Negative existence shows up as a failure of the matching condition; concretely, we write $\Upsilon \not\vdash \Delta$ if for no Υ' can we show $\Upsilon \vdash \Upsilon' + \Delta$. Armed with this matching judgement, we reconstruct the forward calculus of section 3 using affine contexts and other insights of section 4; figure 4 lists the rules. In every rule of figure 2 requiring a particular form for the contexts in the premisses, we use our matching judgement in place of special contexts for the premisses. Additionally, the matching judgement obviates the left rules $\mathbf{1}\&L$, $\&\mathbf{1}L$ and $!L$, so we simply omit them.

Correctness. As expected, the comparitively complex nature of these rules makes soundness and completeness non-trivial properties. In fact, even simple statements of correspondence, like (lin) and (weak) before, seem difficult to obtain. For a manageable description, we have to invoke the matching judgement.

[9] We find this phenomenon in an even stronger form when we add quantifiers and relax all equalities to unifiability – existential variables in backward search are treated *globally*, affecting otherwise disjoint branches in the derivation tree, and requiring undo operations for backtracking. Forward reasoning localizes these variables, giving a much simpler view of unification.

Judgemental rules

$$\frac{}{\cdot\,;\cdot\,;A \longrightarrow^0 A}\ \text{init}$$

$$\frac{\Upsilon \longrightarrow^w \gamma \qquad \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A}{\Gamma\,;\Psi, A\,;\Delta \longrightarrow^w \gamma}\ \text{aff-dl} \qquad \frac{\Upsilon \longrightarrow^w \gamma \qquad \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A}{\Gamma, A\,;\Psi\,;\Delta \longrightarrow^w \gamma}\ \text{copy}$$

Multiplicative connectives

$$\frac{\Upsilon \longrightarrow^w \gamma \qquad \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A, B}{\Gamma\,;\Psi\,;\Delta, A \otimes B \longrightarrow^w \gamma}\ \otimes L$$

$$\frac{\Upsilon \longrightarrow^1 \gamma \qquad \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A \qquad \Upsilon \not\vdash B}{\Gamma\,;\Psi\,;\Delta, A \otimes B \longrightarrow^1 \gamma}\ \otimes L_1$$

$$\frac{\Upsilon \longrightarrow^1 \gamma \qquad \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + B \qquad \Upsilon \not\vdash A}{\Gamma\,;\Psi\,;\Delta, A \otimes B \longrightarrow^1 \gamma}\ \otimes L_2$$

$$\frac{\Gamma\,;\Psi\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Psi'\,;\Delta' \longrightarrow^{w_2} B}{\Gamma \cup \Gamma'\,;\Psi, \Psi'\,;\Delta, \Delta' \longrightarrow^{w_1 \vee w_2} A \otimes B}\ \otimes R \qquad \frac{}{\cdot\,;\cdot\,;\cdot \longrightarrow^0 \mathbf{1}}\ \mathbf{1}R$$

$$\frac{\Gamma\,;\Psi\,;\Delta \longrightarrow^{w_1} A \qquad \Upsilon \longrightarrow^{w_2} \gamma \qquad \Upsilon \vdash (\Gamma'\,;\Psi'\,;\Delta') + B}{\Gamma \cup \Gamma'\,;\Psi, \Psi'\,;\Delta, \Delta', A \multimap B \longrightarrow^{w_1 \vee w_2} \gamma}\ \multimap L$$

$$\frac{\Upsilon \longrightarrow^w \gamma \qquad \Upsilon \vdash \Upsilon' + A \qquad C \supseteq \gamma}{\Upsilon' \longrightarrow^w A \multimap C}\ \multimap R \qquad \frac{\Upsilon \longrightarrow^1 C \qquad \Upsilon \not\vdash A}{\Upsilon \longrightarrow^1 A \multimap C}\ \multimap R'$$

Additive Connectives

$$\frac{\begin{array}{c}\Upsilon \longrightarrow^w \gamma \\ \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A \qquad B \neq \mathbf{1}\end{array}}{\Gamma\,;\Psi\,;\Delta, A \,\&\, B \longrightarrow^w \gamma}\ \& L_1 \qquad \frac{\begin{array}{c}\Upsilon \longrightarrow^w \gamma \\ \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + B \qquad A \neq \mathbf{1}\end{array}}{\Gamma\,;\Psi\,;\Delta, A \,\&\, B \longrightarrow^w \gamma}\ \& L_2$$

$$\frac{\Gamma\,;\Psi\,;\Delta \longrightarrow^{w_1} A \qquad \Gamma'\,;\Psi'\,;\Delta' \longrightarrow^{w_2} B \qquad \Delta/w_1 = \Delta'/w_2}{\Gamma \cup \Gamma'\,;\Psi \cup \Psi'\,;\Delta \cup \Delta' \longrightarrow^{w_1 \wedge w_2} A \,\&\, B}\ \& R \qquad \frac{}{\cdot\,;\cdot\,;\cdot \longrightarrow^1 \top}\ \top R$$

$$\frac{\begin{array}{l}\Upsilon \longrightarrow^{w_1} \gamma \\ \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A \qquad \Gamma'\,;\Psi'\,;\Delta' \longrightarrow^{w_2} \gamma' \qquad \Delta/w_1 = \Delta'/w_2\end{array}}{\Gamma \cup \Gamma'\,;\Psi \cup \Psi'\,;\Delta \cup \Delta', A \oplus \mathbf{1} \longrightarrow^{w_1 \wedge w_2} \gamma \cup \gamma'}\ \oplus\mathbf{1}L$$

$$\frac{\begin{array}{c}\Upsilon' \longrightarrow^{w_2} \gamma' \\ \Gamma\,;\Psi\,;\Delta \longrightarrow^{w_1} \gamma \qquad \Upsilon \vdash (\Gamma'\,;\Psi'\,;\Delta') + B \qquad \Delta/w_1 = \Delta'/w_2\end{array}}{\Gamma \cup \Gamma'\,;\Psi \cup \Psi'\,;\Delta \cup \Delta', \mathbf{1} \oplus B \longrightarrow^{w_1 \wedge w_2} \gamma \cup \gamma'}\ \mathbf{1}\oplus L$$

$$\frac{\begin{array}{c}\Upsilon \longrightarrow^{w_1} \gamma \qquad\qquad \Upsilon' \longrightarrow^{w_2} \gamma' \\ \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + A \qquad \Upsilon' \vdash (\Gamma'\,;\Psi'\,;\Delta') + B \qquad \Delta/w_1 = \Delta'/w_2\end{array}}{\Gamma \cup \Gamma'\,;\Psi \cup \Psi'\,;\Delta \cup \Delta', A \oplus B \longrightarrow^{w_1 \wedge w_2} \gamma \cup \gamma'}\ \oplus L$$

$$\frac{\Upsilon \longrightarrow^w A}{\Upsilon \longrightarrow^w A \oplus B}\ \oplus R_1 \qquad \frac{\Upsilon \longrightarrow^w B}{\Upsilon \longrightarrow^w A \oplus B}\ \oplus R_2 \qquad \frac{}{\cdot\,;\cdot\,;\mathbf{0} \longrightarrow^1 \cdot}\ \mathbf{0}L$$

Exponential rules

$$\frac{\Upsilon \longrightarrow^w A \qquad \Upsilon \vdash (\Gamma\,;\cdot\,;\cdot) + \cdot}{\Gamma\,;\cdot\,;\cdot \longrightarrow^0 !A}\ !R$$

Figure 4: Forward sequent calculus with affine contexts and matching judgement

$$\begin{array}{lll} \Gamma\,;\Psi\,;\Delta \longrightarrow^{0} C & \text{corresponds to} & \Gamma''\,;\Psi''\,;\Delta' \Longrightarrow C \\ & \text{for any } \Gamma' \supseteq \Gamma,\ \Psi' \supseteq \Psi, \text{ and } \Delta' \supseteq \Delta & \\ & \text{such that } (\Gamma'\,;\Psi'\,;\Delta') \vdash (\Gamma''\,;\Psi''\,;\Delta) + (\Delta' \backslash \Delta) & \text{(lin')} \end{array}$$

$$\begin{array}{lll} \Gamma\,;\Psi\,;\Delta \longrightarrow^{1} \gamma & \text{corresponds to} & \Gamma'\,;\Psi'\,;\Delta' \Longrightarrow C \\ & \text{for any } \Gamma' \supseteq \Gamma,\ \Psi' \supseteq \Psi,\ \Delta' \supseteq \Delta, \text{ and } C \supseteq \gamma & \text{(weak')} \end{array}$$

To start with, we need to establish some properties of the matching judgement.

Lemma 10 (Bounding). *If* $(\Gamma\,;\Psi\,;\Delta) \vdash (\Gamma'\,;\Psi'\,;\Delta') + \Delta''$, *then:*

1. $\Gamma' \subseteq \Gamma$, $\Psi' \subseteq \Psi$ *and* $\Delta' \subseteq \Delta$; *and*
2. $!(\Gamma \backslash \Gamma'), (\Psi \backslash \Psi') \mathbin{\&} \mathbf{1}, (\Delta \backslash \Delta') \subseteq \Delta''$.

Proof. Structural induction on the derivation of $(\Gamma\,;\Psi\,;\Delta) \vdash (\Gamma'\,;\Psi'\,;\Delta') + \Delta''$. □

Additionally, we require a *matching lemma* that drives the completeness theorem.

Lemma 11 (Matching). *If* $\Upsilon \Longrightarrow C$ *and* $\Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + \Delta'$ *then* $\Gamma\,;\Psi\,;\Delta, \Delta' \Longrightarrow C$.

Proof. Structural induction on the derivation of $\mathcal{M} :: \Upsilon \vdash (\Gamma\,;\Psi\,;\Delta) + \Delta'$. We illustrate with a pair of cases.

(i) The last rule of $\mathcal{M}$ is $\vdash_{\text{linear}}$, *i.e.*.

$$\frac{\Gamma\,;\Psi\,;\Delta \vdash (\Gamma'\,;\Psi'\,;\Delta') + \Delta''}{\Gamma\,;\Psi\,;\Delta, A \vdash (\Gamma'\,;\Psi'\,;\Delta') + \Delta'', A}$$

$\Gamma\,;\Psi\,;\Delta, A \Longrightarrow C$	hypothesis
$\Gamma\,;\Psi\,;\Delta \Longrightarrow A \multimap C$	$\multimap R$
$\Gamma'\,;\Psi'\,;\Delta', \Delta'' \Longrightarrow A \multimap C$	ind. hyp.
$\Gamma'\,;\Psi'\,;\Delta', \Delta'', A \Longrightarrow C$	$\multimap R$ inversion

(ii) The last rule of $\mathcal{M}$ is $\vdash_{\&\mathbf{1}}$, *i.e.*,

$$\frac{\Gamma\,;\Psi\,;\Delta \vdash (\Gamma'\,;\Psi'\,;\Delta') + \Delta''}{\Gamma\,;\Psi, A\,;\Delta \vdash (\Gamma'\,;\Psi'\,;\Delta') + \Delta'', A \mathbin{\&} \mathbf{1}}$$

$\Gamma\,;\Psi, A\,;\Delta, A \Longrightarrow C$	hypothesis
$\Gamma\,;\Psi\,;\Delta, A \mathbin{\&} \mathbf{1} \Longrightarrow C$	$\&\mathbf{1}L$
$\Gamma\,;\Psi\,;\Delta \Longrightarrow A \mathbin{\&} \mathbf{1} \multimap C$	$\multimap R$
$\Gamma'\,;\Psi'\,;\Delta', \Delta'' \Longrightarrow A \mathbin{\&} \mathbf{1} \multimap C$	ind. hyp.
$\Gamma'\,;\Psi'\,;\Delta', \Delta'', A \mathbin{\&} \mathbf{1} \Longrightarrow C$	$\multimap R$ inversion

The other cases follow similarly. For the matching rules for the unrestricted context, we appeal to theorem 1 (extended for the affine zone). □

With these lemmas, we may now prove soundness and completeness of the forward calculus with respect to the backward calculus. Although the soundness theorem doesn't differ much from before, the completeness theorem has a somewhat unusual form, depending on the matching judgement. Nevertheless, we can prove these theorems purely by structural induction on the derivations. The difficulty in these theorems lies not in the inductions themselves, which follow straightforwardly, but rather in the choice of sufficiently strong induction hypotheses that make the inductions valid.

Theorem 12 (Soundness).
1. *If* $\Upsilon \longrightarrow^0 C$ *then* $\Upsilon \Longrightarrow C$.
2. *If* $\Gamma \,;\, \Psi \,;\, \Delta \longrightarrow^1 \gamma$ *then* $\Gamma \,;\, \Psi \,;\, \Delta' \Longrightarrow C$ *for any* $\Delta' \supseteq \Delta$ *and* $C \supseteq \gamma$.

Proof. By structural induction on the derivation of $\Upsilon \longrightarrow^w \gamma$, similar to the proof of theorem 3, but using the matching and bounding lemmas as required. We omit the easy details. □

Theorem 13 (Completeness). *If* $\Gamma \,;\, \Psi \,;\, \Delta \Longrightarrow C$, *then for some* $\Gamma' \subseteq \Gamma$, $\Psi' \subseteq \Psi$, *and* $\gamma \subseteq C$ *such that the following match holds*

$$(\Gamma' \,;\, \Psi' \,;\, \Delta) \vdash (\Gamma'' \,;\, \Psi'' \,;\, \Delta'') + \Delta'''$$

one of the following hold:
1. *either* $\Gamma'' \,;\, \Psi'' \,;\, \Delta'', \Delta''' \longrightarrow^0 C$;
2. *or* $\Gamma'' \,;\, \Psi'' \,;\, \Delta', \Delta''' \longrightarrow^1 \gamma$ *for some* $\Delta' \subseteq \Delta''$.

Proof. By structural induction on the derivation $\mathcal{D}$ of $\Gamma \,;\, \Psi \,;\, \Delta \Longrightarrow C$, using the bounding and matching lemmas. We have the following characteristic cases for the last rule in $\mathcal{D}$:
1. init, copy, aff-dl, $\mathbf{1}R$, $\top R$ or $!R$; these cases follow immediately because the rules in the forward and backward direction differ structurally only in the presence of the matching derivation, for which we invoke the matching lemma.
2. $\mathbf{1}\&L$ or $\&\mathbf{1}L$; for example

$$\mathcal{D} \quad = \quad \cfrac{\begin{matrix}\mathcal{D}' \\ \Gamma \,;\, \Psi, A \,;\, \Delta \Longrightarrow C\end{matrix}}{\Gamma \,;\, \Psi \,;\, \Delta, A \mathbin{\&} \mathbf{1} \Longrightarrow C}\ \mathbf{1}\&L$$

   Invoking the bounding lemma (case 1), assume given $\Gamma_1 \cup \Gamma_2 \subseteq \Gamma$ and $\Psi_1, \Psi_2 \subseteq (\Psi, A)$. Then, we have
   1. if $\Gamma_1 \,;\, \Psi_1 \,;\, \Delta, !\Gamma_2, \Psi_2 \mathbin{\&} \mathbf{1} \longrightarrow^0 C$, then
      1. if $\mathbf{1} \mathbin{\&} A \in \Psi_2 \mathbin{\&} \mathbf{1}$, then we satisfy case (1);
      2. otherwise, $\Psi_1, \Psi_2 \subseteq \Psi$ and we satisfy case (1).
   3. otherwise, $\Gamma_1 \,;\, \Psi_1 \,;\, \Delta', !\Gamma_2, \Psi_2 \mathbin{\&} \mathbf{1} \longrightarrow^0 C$ for some $\Delta' \subseteq \Delta$; the above argument still applies, except now we satisfy case (2) instead of (1).
4. Other rules require a similar but simpler enumeration of possibilities. □

Lest the completeness theorem give the impression that matching as a judgement makes forward reasoning unusably complex, we restate the correspondences (lin′) and (weak′) in simpler terms using the bounding lemma.

$$\Gamma \,;\, \Psi \,;\, \Delta \longrightarrow^0 C \qquad \text{corresponds to} \qquad \Gamma' \,;\, \Psi' \,;\, \Delta' \Longrightarrow C$$

for any $\Gamma' \supseteq \Gamma$, $\Psi' \supseteq \Psi$, and for $\Delta' \supseteq \Delta$ where every element of $\Delta' \backslash \Delta$ has one of the forms $A \mathbin{\&} \mathbf{1}$, $\mathbf{1} \mathbin{\&} A$, or $!A$; and

$$\Gamma \,;\, \Psi \,;\, \Delta \longrightarrow^1 \gamma \qquad \text{corresponds to} \qquad \Gamma' \,;\, \Psi' \,;\, \Delta' \Longrightarrow C$$

for any $\Gamma' \supseteq \Gamma$, $\Psi' \supseteq \Psi$, $\Delta' \supseteq \Delta$, and $C \supseteq \gamma$.

These correspondences finally enable a usable model of affine non-determinism in the presence of negative **1** in the logic. In the next and final section of this paper, we discuss the implementation of this calculus in the inverse method.

5 The inverse method

For an actual implementation we must investigate a focusing version of the forward calculus, but we can already see many features of the inverse method by working with the calculus without focusing. Historically, the inverse method for classical (non-linear) logic owes its development to Maslov [19]. Subsequently, Voronkov [25], Mints [21], and more recently Tammet [23, 24] have adapted it for non-classical and intuitionistic logics, though not for linear logic. Mints [20] has investigated resolution calculi for classical linear logic, but his methods don't have an immediate application to the inverse method. We describe the key issues for the inverse method for linear logic in this section, and refer interested readers to the handbook article [9] on the inverse method for a more complete reference.

Subformula property. This key technical property makes the inverse method possible. Stated simply, in cut-free sequent calculus proofs, we need to consider only sequents composed of subformulas of the goal sequent. To illustrate, assuming we have sequents containing A and B, then we never consider a rule to infer a sequent about $A \,\&\, B$ from these sequents, unless $A \,\&\, B$ occurs as a subformula of the goal sequent. Formally, we present this property property in terms of a *subformula relation* for propositions. To describe the subformula relation in its strongest form, we decorate subformulas with certain marks:

1. *Sign* (also known as *polarity*), which we write as a superscript $^+$ or $^-$ (or possibly both). The operands of all binary connectives inherit the sign of the formula, with the exception of $A \multimap B$, for which A receives the opposite sign. Formulas to the right of the sequent arrow recive the positive sign, and those on the left the negative sign. Thus, these signs indicate the side of the sequent arrow where the formula occurs as a principal formula.
2. *Weight*, which we write as a subscript $_!$, and schematically as $_\mathrm{w}$. Top-level formulas in the unrestricted context, and operands of ! receive this decoration, but the subformulas do not inherit the decoration. These signs, therefore, determine whether the formula is allowed to occur in the unrestricted context, and thus serves as a guide for the copy rule.
3. *Affineness*, which we indicate as a subscript $_\&$, and schematically as $_\mathrm{a}$. Similar to the weight decoration, only top-level formulas in the affine context, and the operand A in $A \,\&\, \mathbf{1}$ and $\mathbf{1} \,\&\, A$, receive this mark, and subformulas don't inherit it.

The *decorated subformula relation* $\leq$ describes a relation between decorated formulas, generated freely from the following rules. (We omit the trivial rules for propositional constants.)

$$
\begin{array}{lll}
A^{\pm} \leq (A * B)^{\pm}_{\mathrm{wa}} & B^{\pm} \leq (A * B)^{\pm}_{\mathrm{wa}} & \ldots * \in \{\otimes, \&, \oplus\} \\
A^{\mp} \leq (A \multimap B)^{\pm}_{\mathrm{wa}} & B^{\pm} \leq (A \multimap B)^{\pm}_{\mathrm{wa}} & \\
A^{\pm}_{!} \leq (!A)^{\pm}_{\mathrm{wa}} & A^{\pm}_{\&} \leq (A \,\&\, \mathbf{1})^{\pm}_{\mathrm{wa}} & A^{\pm}_{\&} \leq (\mathbf{1} \,\&\, A)^{\pm}_{\mathrm{wa}}
\end{array}
$$

Using this relation, we may state the subformula property for the forward calculus in the strongest form as follows:

Theorem 14 (Subformula Property). *Any sequent appearing in a proof of $\Gamma^-_! \,;\, \Psi^-_\& \,;\, \Delta^- \Longrightarrow C^+$ must have the form:*

$$A^-_{1!}, A^-_{2!}, \ldots \,;\, B^-_{1\&}, B^-_{2\&}, \ldots \,;\, D^-_1, D^-_2, \ldots \Longrightarrow C'^+$$

where every formula $A^-_!$, $B^-_\&$, D^- and C'^+ relates to some formula in $\Gamma^-_! \cup \Psi^-_\& \cup \Delta^- \cup \{C^+\}$ by the subformula relation $\leq$.

Proof. Straightforward structural induction on the derivation of $\Gamma^-_! \,;\, \Psi^-_\& \,;\, \Delta^- \Longrightarrow C^+$. ☐

By theorem 13, sequents in the forward calculus contain a subset of formulas in the backward calculus. Thus,

Corollary 15 (Subformula Property for the Forward Calculus). *In proving* $\Gamma_!^- ; \Psi_\&^- ; \Delta^- \longrightarrow C^+$ *we need only consider suequents consisting of subformulas in the sense of Thm.14.*

Proof. Contrapositively, given a proof of $\Gamma_!^- ; \Psi_\&^- ; \Delta^- \longrightarrow C^+$, if it refers to sequents consisting of non-subformulas of the goal, use theorem 13 to create a corresponding proof in the backward calculus, then note a violation of theorem 14. ☐

Labelling and specialized rules. The subformula property gives us the core of the inverse method procedure. We start with intial sequents of the form $\cdot;\cdot;A^- \longrightarrow^0 A^+$, where A occurs as both a positive and a negative subformula of the goal sequent. Since we need some way to refer to subformulas of the goal sequent, we label label all subformulas with new fresh (propsitional) labels, which we write using the propositonal variable L.

We also specialize all rules to these labels by means of a pre-processing stage before entering into the main search procedure. We don't maintain general rules for conjunction, disjunction *etc.*, but instead have a version of every rule for every label, with the label taking the role of the principal formula. We can even perform further optimizations on these rules, for example, pre-computing all compositions of invertible rules (which necessarily have bounded depth), because the subformula property guarantees that we require no additional rules to prove any (provable) goal sequent. Search then proceeds by a saturation based strategy, applying all possible rules to the fringe of a database of facts.

Subsumption. Because of the conjunctive non-determinism in the forward direction, arising from the saturation-based search, it becomes critical to detect redundancies using a process of sequent subsumption. The only complications in our linear setting lie in handling the linear context for weak sequents, for which we allow subsumption of sequents with weaker contexts even though we don't have an admissible structural theorem for weakening the linear context of weak sequents. Nevertheless, we don't lose completeness because we can always use the stronger sequent for any purpose the weaker sequent might serve; indeed, we justify the negative-existence conditions for $\multimap R'$, for example, with exactly this reason.

In the presence of quantifiers, the order of contraction, which takes the form of a sequence of unification steps (known as *factoring*), and subsumption becomes a critical issue. Eager factoring, *i.e.,* before any subsumption tests, might end up as useless work if the sequents don't lead to a proof. This problem exists already for the unrestricted case, and for the linear case the only complications occur in additive rules that perform multiset unions. we shall examine the exact nature of factoring for these cases in a future work; fortunately, this problem doesn't exist in the propositional fragment in this paper.

Search procedure Finally, a brief summary of the search procedure:

1. Label all subformulas of the goal sequent, and decorate using signs, weights and affinities.
2. Determine all initial sequents for atomic formulas with both signs.
3. Specialize all left rules for negative subformulas, all right rules for positive subformulas, instances of the copy rule for heavy subformulas, and instances of aff-dl for affine subformulas.
4. Starting from the initial sequents, apply the inference rules in a fair way (saturation search), adding new facts to a database used for subsumption checks. As an optimization, after applying all possible rules for a given sequent, mark the sequent as "old", and never consider it for generating new facts again. Thus, the unmarked sequents form the active *fringe* of the database.
5. Stop when we match the goal sequent, using the conditions of the completeness theorem (theorem 13). Otherwise, if no rules apply, abort the search procedure.

6 Conclusion

We have presented a forward sequent calculus for the propositional linear logic. Our calculus has the following properties from the perspective of resource management:

- No undetermined resources. We identify sequents with weakenable linear contexts, and introduce such resources implicitly.
- Controlled affine resources. All rules have *structural* resource introductions, controlled by tight matching criteria.

Our framework is sufficiently general that it admits some ready extensions. As remarked earlier, extending the calculus to first order connectives requires relaxing equalities to unification. While it is relatively straightforward for the logical rules, the negative existence conditions in the matching rules require special consideration.

In order to complete a practical implemention of an inverse method theorem prover that uses this forward calculus, we require two important theoretical extensions – (1) a version of the sequent calculus that incorporates focused derivations in the sense of Andreoli [3, 16], and (2) an efficient indexing mechanism. Focused derivations impose strict controls on rule application, allowing the creation of big-step derived inference rules, and thereby cuts down on the number of new sequents. We shall examine the interactions between inversion, focusing, and matching in a future work.

Another key issue occurs with the rule $\multimap \mathbf{1} L$ that allows a kind of uncontrolled application. While somewhat rarer, this scenario no doubt fits the definition of a resource management problem, since it allows the linear context to grow uncontrolledly. We see two possibilities for handling this – one would attempt to re-apply our guiding maxim of "control through laziness", and derive a further refinement of the matching judgement for these cases. A more promising approach would use a possibility monad [8] to handle $A \multimap \mathbf{1}$ by translating their uses into Girard's MIX rule, and then using an interpretatino of MIX in the possibility monad.

Acknowledgements. We thank Frank Pfenning for many discussions on all aspects of linear logic. We also thank Brigitte Pientka and Jason Reed for useful discussions.

References

[1] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111(1&2):3–57, 1993.

[2] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 15(2):543–574, 1994.

[3] Jean Marc Andreoli. Logic programming with focussing proofs in linear logic. *Journal of Logic and Computation*, 2, 1992.

[4] Andrew Barber and Gordon Plotkin. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, University of Edinburgh, 1996.

[5] Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. Technical Report 352, University of Cambridge Computer Laboratory, 1994. [65 page version].

[6] Nick Benton, Gavin Bierman, Valeria de Paiva, and Martin Hyland. A term calculus for intuitionistic linear logic. In M. Bezem and G. F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications (TLCA)*, volume 664, pages 75–90, Utrecht, The Netherlands, March 1993. Springer-Verlag.

[7] Iliano Cervesato, Joshua S. Hodas, and Frank Pfenning. Efficient resource management for linear logic proof search. In R. Dyckhoff, H. Herre, and P. Schroeder-Heister, editors, *Proceedings of the 5th International Workshop on Extensions of Logic Programming*, pages 67–81, Leipzig, Germany, March 1996. Springer-Verlag LNAI 1050.

[8] Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131, Carnegie Mellon University, 2003.

[9] Anatoli Degtyarev and Andrei Voronkov. *Handbook of Automated Reasoning*, chapter The Inverse Method, pages 179–272. Number ISBN 0-262-18223-8. MIT Press, September 2008.

[10] Jyean-Yves. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[11] James Harland and David Pym. The uniform proof-theoretic foundations of linear logic programming. In V. Saraswat and K. Ueda, editors, *Proceedings of the International Logic Programming Symposium*, pages 034–318, San Diego, California, October 1991.

[12] James Harland and David J. Pym. Resource-distribution via boolean constraints. In W. McCune, editor, *Proceedings of CADE-14*, pages 222–236, Townsville, Australia, July 1997. Springer-Verlag LNAI 1249.

[13] James Harland and Philip Winikoff. Deterministic resource management for the linear logic programming Lygon. Technical Report TR 94/23, Melbourne University, Department of Computer Science, 1994.

[14] Joshua S. Hodas. Lolli: an extension of λProlog with linear logic context management. In Dale Miller, editor, *Proceedings of the 1992 workshop on the λProlog programming language*, Philadelphia, 1992.

[15] Joshua S. Hodas and Dale Miller. Logic programming in a fragment of linear logic. *Journal of Information and Computation*, 110(2):327–365, 1994.

[16] Jakob M. Howe. *Proof search issues in some non-classical logics*. PhD thesis, University of St. Andrews, September 1999.

[17] Yves Lafont and Thomas Streicher. Games semantics for linear logic. In *Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science*, Amsterdam, The Netherlands, July 1991. IEEE Computer Society Press, Los Amitos, California.

[18] Patrick D. Lincoln, John C. Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56:239–311, 1992.

[19] S. Maslov. The inverse method of establishing deducibility in the classical predicate calculus. *Soviet Mathematical Doklady*, 5:1420–1424, 1964.

[20] Grigori Mints. Resolution calculus for the first order linear logic. *Journal of Logic, Language and Information*, 2(1):59–83, 1993.

[21] Grigori Mints. Resolution strategies for the intuitionistic logic. In *Constraint Programming*, NATO ASI Series F, pages 289–311. Springer-Verlag, 1994.

[22] Jeff Polakow and Frank Pfenning. Relating natural deducion and sequent calculus for intuitionistic non-commutative linear logic. In Andrew Scedrov and Achim Jung, editors, *Proceedings of the 15th Conference on Mathematical Foundations of Programming Semantics*, volume 20 of *Electronic Notes in Theoretical Computer Science*, New Orleans, Louisiana, April 1999.

[23] Tanel Tammet. A resolution theorem prover for intuitionistic logic. In M. McRobbie and J. Slaney, editors, *Proceedings of CADE-13*, pages 2–16, New Brunswick, New Jersey, 1996. Springer-Verlag LNCS 1104.

[24] Tanel Tammet. Resolution, inverse method and the sequent calculus. In *Proceedings of the 5th Kurt Gödel Colloquim on Computational Logic and Proof Theory (KGC'97)*, pages 65–83, Vienna, Austria, 1997. Springer-Verlag LNCS 1289.

[25] Andrei Voronkov. Theorem proving in non-standard logics based on the inverse method. In D. Kapur, editor, *Proceedings of the CADE-11*, pages 648–662, Saratoga Springs, New York, 1992. Springer-Verlag LNCS 607.

[26] Michael Winikoff and James Harland. Implementing the linear logic programming language Lygon. In *Proceedings of the International Logic Programming Symposium (ILPS)*, pages 66–80, December 1995.