Formal Verification of the Winning Strategies of Pursuit-Evasion Games

Weihan Li

CMU-CS-22-122
August 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213

Thesis Committee:
André Platzer (Chair)
Stefan Mitsch

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Keywords: Formal Verification, Differential Game Logic, Pursuit-Evasion-Games

Abstract

A pursuit-evasion game is an adversarial hybrid game that combines both continuous and discrete dynamics that is widely used to model robotics tasks in the literature. We model the game rules formally, present a formal verification approach for the winning strategies and prove the design correctness of the proposed algorithms. To accomplish this, we use Differential Game Logic (dGL) to implement the proofs with the KeYmaera X theorem prover, which rigorously proves the safety of the model and the correctness of the winning strategies. The games we consider have two different models of motion: discrete dynamics and continuous dynamics. In particular, we focus on two types of games: Cops-and-Robbers games, which are placed on discrete graphs with movements by stepping the graph edges and Lion-and-Man games, which are played on continuous planes with continuous movement. We set up the model in dGL, identify variants and invariants to reason about winning strategies for different types of game regions and discuss pursuer/evader winning conditions. We separately considered game regions with certain properties, i.e. different families of graphs and planes with selected properties, and conducted proofs that serve for the most general cases.

Acknowledgments

First I would like to thank my advisor, Professor André Platzer, who guided me through finishing this project. I am fortunate to work with him, from starting this research work until finishing the thesis throughout my undergraduate years to my master's study.

I also want to thank Dr. Stefan Mitsch who serves on my thesis committee. He has also gave me significant support including technical support on using KeYmaera X, ideas and suggestions of important decisions in the research and so on. I also thank Dr. Rose Bohrer, who has also served as my advisor in my undergraduate study, and has provided me important guidance on this piece of work.

Along the way, I am very grateful for the help and advices from multiple CMU professors. Without them, I wouldn't be able to accomplish what I achieved today. I also want to thank my family and friends who has provided me unconditional support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

A Pursuit-Evasion Game is a type of mathematical game that is widely discussed in the literature. It provides a general theoretical model for search problems in robotics, and it has wide applications including search-and-rescue, collision-avoidance, path-finding, etc. These problems aim to find efficient strategies for the agents to perform different tasks in cases that involve two parties. The objective of the pursuer is to catch the evader, and the objective of the evader is to escape. Therefore, the game is two-player, zero-sum game, and has a Nash Equilibrium. Research on this problem focuses on finding and proving the existence of winning strategies in mathematical ways, and discussing the computational complexity in Computer Science. This type of game combines discrete dynamics and continuous dynamics, which can be naturally defined using hybrid games. Hybrid games have two adversarial players, zero-sum with discrete or continuous dynamics, i.e. the players compete against each other to win the game.

Formal verification approaches are commonly applied to hybrid games, since they provide strong safety guarantee for the agents' control algorithms. The goal of such approaches coincide with our goal of designing smart and safe winning strategies in pursuit-evasion games, and through proving the safety and correctness of the strategies, we also construct explicit strategies that can be directly applied to robotics systems. In this paper, we use Differential Game Logic (dGL) [14], a proof calculus designed specifically to represent hybrid games and verify game strategies. Modeling different types of game regions in dGL contributes a way of characterizing families of game regions, and proving dGL formulas provides safety and correctness guarantee of players' winning strategies. Therefore, hybrid games and dGL perfectly capture the nature of studying pursuit-evasion games.

In this paper, we consider two distinct variants of pursuit-evasion games in particular: "Cops and Robbers" games and "Lion and Man" games. The former represent games played on discrete graphs, and the latter represent games played on continuous planes. For each game, this paper contributes dGL models that characterize the rules and dynamics, definitions and categorizations for several families of game regions, mathematical proofs and dGL proofs for the corresponding winning strategies. We verified a selection of the proofs using the theorem prover KeYmaera X [6], which is found in Appendix B, and the rest using paper-based proofs. KeYmaera X proofs unrolls the structure of proofs clearer and ensures safety and correctness throughout the proof. Paper-based proofs can make use of mathematical definitions and theorems which supports more complicated proofs.

This work is structured as follows: Chapter 2 provides a summarization of background work. Chapter 3 provides game formations and relevant mathematical definitions. Chapter 4 discusses Cops and Robbers game and Chapter 5 focuses on Lion and Man game. Chapter 6 provides a summary of the work and potential future work.

Chapter 2

Background Work

2.1 Pursuit-Evasion Game

A pursuit-evasion game is a family of mathematical games that contains two adversarial parties: in which one is the pursuer and the other is the evader. The pursuer aims to track down the evader and the evader escapes away from the pursuer. The game is usually called discrete pursuit-evasion if the game is restricted on a graph; and is called continuous pursuit-evasion if the game is played on a geometric plane.

The concept of Cops and Robbers game proposed by Nowakowski and Winkler in 1983 [12] formalizes the graph variant. Aigner [1] generalizes the case to multiple cops and robbers. An extension of this work focuses on the cop number of a graph: i.e. the minimum number of cops required to win the game. Meyniel's conjecture by Frankl [4] states that for graphs of size n, $O(\sqrt{n})$ cops suffice to win, and the conjecture still remains open. The decision problem of the cop-number of a graph is EXPTIME-complete [9]. In this work, we mainly focus on 1-cop scenarios.

Continuous pursuit-evasion game extends discrete games to the continuous space with continuous dynamics. A particular variant, "Lion and Man" game was first proposed by Rado in 1925. It is a geometric version of "Cops and Robbers" game. In this game, both players are allowed to move by a maximum time range in each round. Research has focused on the capture time for different types of planes. If the game is played simultaneously in continuous time, the man can always escape. If the players move in turns, a single lion has a winning strategy in a simply-connected polygon [16], and three lions is sufficient to catch the man in any polygon [8, 2].

Less relevant to this work are the visibility problem [7] and random graphs [15]. A summary is in [3].

2.2 Differential Game Logic

Differential Game Logic (dGL) [14] is a proof calculus primarily used to prove the existence and correctness of game strategies of hybrid games. dGL is an extension of Differential Dynamic Logic (dL) [13], which is the logic that describes dynamics of hybrid systems, by adding the dual operator d to describe the adversarial dynamics of both parties.

In dGL, we usually call the two players as *Angel* and *Demon*. Hybrid games are adversarial, sequential, and perfect-information. We recall the following syntax and semantics, with illustration of proof rules through an example.

2.2.1 Syntax

Definition 2.2.1. (Hybrid Games) [14] dGL is built upon the following grammar:

$$\alpha, \beta ::= x := e \mid ?Q \mid x' = f(x)\&Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d.$$

In order, the syntax enables the ability of assignment, test, the system of differential equations, choice, sequence, repetition and dual. These operators are Angel's operators, with Demon's operators are defined using the dual operator d. We sometimes abbreviate as follows:

Table 2.1: Demon's Operations

Original	Abbreviation
$(x := e)^d$	$x := e$
$?Q^d$	None
$(x' = f(x)\&Q)^d$	None
$(\alpha^d \cup \beta^d)^d$	$(\alpha \cap \beta)$
$(\alpha; \beta)^d$	$\alpha^d; \beta^d$
$((\alpha^d)^*)^d$	$\alpha^\times$

Definition 2.2.2. (dGL Formulas) [14]

$$\phi := \theta_1 \sim \theta_2 \mid \forall x \phi \mid \exists x \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \to \phi_2 \mid \phi_1 \leftrightarrow \phi_2,$$

where θ_1, θ_2 are real arithmetic terms and $\sim$ stands for real arithmetic operators ($>, <, \geq, \leq, =, \neq,$). α is a hybrid game, x is a variable.

2.2.2 Semantics

The intuitive understanding of hybrid games semantics is summarized in Table 2.2. In dGL, common formulas are as defined similarly as First Order Logic. $[\alpha]\phi$ means that

Demon has a winning strategy of game α to make ϕ true, and $\langle\alpha\rangle\phi$ means that Angel has a winning strategy of game α to make ϕ true.

The most important note here is the usage of dual operator d. We consider Angel and Demon to be the two players playing against each other in the hybrid game. The dual operator demonstrates which player is in the move, as explained in Table 2.1.

To formally define the semantics, we use the notation $[\![\phi]\!]$ which is the set of states in which ϕ is true. For hybrid game α, we define $\varsigma_\alpha(X)$ as the set of states such that Angel can play game α to reach winning state X; $\delta_\alpha(X)$ as the set of states such that Demon can play game α to reach winning state X.

Table 2.2: Hybrid Game Semantics [14]

Syntax	Semantics
$x := e$	Assign the value of e to the variable x, leaving all other variables unchanged
$?Q$	Test if Q is true, continue running; else terminate
$x' = f(x) \& Q$	Follow the system of differential equation $x' = f(x)$ for a certain amount of time when Q holds true
$\alpha \cup \beta$	Run hybrid game α or β (Angel's choice)
$\alpha \cap \beta$	Run hybrid game α or β (Demon's choice)
$\alpha; \beta$	Sequentially run β after α
α^*	Run α repeatedly for any ≥ 0 amount of iterations (Angel's choice)
$\alpha^\times$	Run α repeatedly for any ≥ 0 amount of iterations (Demon's choice)
α^d	Changes player to run α

The semantics are defined inductively as follows, see [14] for details.

Definition 2.2.3. (Semantics) [14] Let $\mathcal{J}$ be the set of states.

1. $[\![e \geq f]\!] = \{\omega \in \mathcal{J} : \omega[\![e]\!] \geq \omega[\![f]\!]\}$, where $\omega[\![e]\!]$ is the value of e interpreted in state ω.
2. $[\![\neg P]\!] = ([\![P]\!])^C$, which is the set of states where P is false.
3. $[\![P \wedge Q]\!] = [\![P]\!] \cap [\![Q]\!]$. This means that $P \wedge Q$ is true if and only if P is true and Q is true.
4. $[\![\exists x P]\!] = \{\omega \in \mathcal{J} : \exists r \in \mathbb{R}, \omega_x^r \in [\![P]\!]\}$ where ω_x^r stands for the state that replaces variable x with value r.

5. $[\![\langle\alpha\rangle P]\!] = \varsigma_\alpha(P)$. P is in Angel's winning region after playing hybrid game α.

6. $[\![[\alpha]P]\!] = \delta_\alpha(P)$. P is in Demon's winning region after playing hybrid game α.

Note that the semantics for Demon's moves can be inferred using α^d, therefore we do not further elaborate here. Now we define hybrid game semantics inductively.

1. $\varsigma_{x:=e}(X) = \{\omega \in \mathcal{J} : \omega_X^{\omega[e]} \in X\}$. A winning state after assignment is exactly the winning states that assign x to e.

2. $\varsigma_{x'=f(x)\&Q}(X) = \{\phi(0) \in \mathcal{J} : \phi(r) \in X$ for some solution $\phi : [0, r] \to \mathcal{J}$ of any duration $r \in \mathbb{R}$ s.t. $\phi \vdash x' = f(x) \wedge Q\}$. Angel wins the differential equation $x' = f(x)\&Q$ if $\phi(0)$, $\phi(r) \in X$ and Q is maintained through non-negative time duration r.

3. $\varsigma_{?Q}(X) = [\![Q]\!] \cap X$. Angel wins for the states that satisfy both Q and X.

4. $\varsigma_{\alpha\cup\beta}(X) = \varsigma_\alpha(X) \cup \varsigma_\beta(X)$. Angel wins for the states that satisfy X from running α or β.

5. $\varsigma_{\alpha;\beta}(X) = \varsigma_\alpha(\varsigma_\beta(X))$. Angel wins for the states that satisfy X from running α then β.

6. $\varsigma_{\alpha^*}(X) = \bigcap\{Z \in \mathcal{J} : X \cup \varsigma_\alpha(Z) \subseteq Z\}$. Angel wins for the minimum intersection of states that contain X and running α once still maintain in this set.

7. $\varsigma_{\alpha^d}(X) = (\varsigma_\alpha(X^C))^C$. Angel wins if Angel loses the winning state X^C.

Now we have all the tools required to understand an intuitive example in "Cops and Robbers" game. We will let Cop be Angel and Robber be Demon.

Example 2.2.4. The following formula describes a naive scenario in the game that if the robber chooses a vertex first, then the cop c can always choose a vertex that is the same as the robber r.

$$\langle\{r := *; ?(r \in V)\}^d; \{c := *; ?(c \in V)\};\rangle(c = r)$$

We can decompose the formula as follows: $(c = r)$ is the winning condition. $\langle\{r := *; ?(r \in V)\}^d; \{c := *; ?(c \in V)\};\rangle$ is the game which the cop has a winning strategy. Then we decompose the game into $\langle\{r := *; ?(r \in V)^d\}\rangle$ and $\langle\{c := *; ?(c \in V)\}\rangle$. The former describes the robber's move and the rule to follow; the latter describes the cop's move and the rule to follow. In the robber's rule, it first randomly assigns a value to variable r, but it has to pass the test $r \in V$, otherwise it fails the game. This interprets the idea that restricts the robber to select a legal vertex from the set of V. The similar rule is defined for the cop. Since the cop moves after the robber, the winning strategy for the cop is to select the exact same position the robber chooses.

Now we give another example to explain the use of ODE and repetition in dGL.

Example 2.2.5. The following formula describes a hybrid game where in each round, r selects a random real number, then c selects an ODE to run. The goal of the game is $c = r$.

$$\langle\{\{r := *;\}^d; \{c' = 1 \cup c' = -1\}\}^*\rangle(c = r)$$

2.2.3 Axioms

In this paper, our proofs are built upon dGL axioms using sequent calculus. The proof structure forms like this:

Definition 2.2.6. A **sequent** is formed like $\Gamma \vdash \Delta$, where Γ stands for the set of assumptions, and Δ stands for the set of assertions to prove. The conclusion $\Gamma \vdash \Delta$ says whenever every assumption holds, at least one of the assertions hold.

A **proof rule** is formed like

$$\frac{\Gamma_1 \vdash \Delta_1, ..., \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta},$$

which expresses the case that if all premises $\Gamma_i \vdash \Delta_i$ are valid, then the conclusion $\Gamma \vdash \Delta$ must be valid.

We can build up dGL axioms using the semantics definitions.

$[\cdot]$ Determinacy Axiom:	$[\alpha]P \leftrightarrow \neg\langle\alpha\rangle(\neg P)$
$\langle:=\rangle$ Assignment Axiom:	$\langle x := \theta\rangle P(x) \leftrightarrow P(\theta)$
$\langle\cup\rangle$ Choice Axiom:	$\langle\alpha \cup \beta\rangle P \leftrightarrow \langle\alpha\rangle P \wedge \langle\beta\rangle P$
$\langle;\rangle$ Composition Axiom:	$\langle a; b\rangle P \leftrightarrow \langle a\rangle\langle b\rangle P$
$\langle*\rangle$ Iteration Axiom:	$\langle a^*\rangle P \leftrightarrow P \vee \langle a\rangle\langle a^*\rangle P$
$\langle'\rangle$ Solution Axiom:	$\langle x' = f(x)\rangle p(x) \leftrightarrow \exists t \geq 0 \langle x := y(t)\rangle p(x)$
$\langle?\rangle$ Test Axiom:	$\langle ?Q\rangle P \leftrightarrow Q \wedge P$
$\langle d\rangle$ Duality Axiom:	$\langle\alpha^d\rangle P \leftrightarrow \langle\neg\alpha\rangle\neg P.$

Table 2.3: Differential Game Logic Axioms [14]

Using the equivalence relations in the axiom above, we can infer a large proportion of dL sequent calculus proof rules. For example, we can deduce the following proof rules for $[\cup]$:

$$\frac{\Gamma \vdash [\alpha]P \quad \Gamma \vdash [\beta]P}{\Gamma \vdash [\alpha \cup \beta]P} \cup R \qquad \frac{[\alpha]P \wedge [\beta]P \vdash \Delta}{[\alpha \cup \beta]P \vdash \Delta} \cup L$$

A full list of proof rules we used can be found in Appendix A.

We further make remark on two rules we rely on heavily in providing the constructive game strategy here:

$$\frac{\Gamma \vdash J, \Delta \quad J \vdash P \quad J \vdash [\alpha]J}{\Gamma \vdash [\alpha^*]P, \Delta} \; loop$$

$$\frac{\Gamma \vdash \exists d. J(d) \quad d \leq 0, J(d) \vdash P \quad d > 0, J(d) \vdash \langle \alpha \rangle J(d-1)}{\Gamma \vdash \langle \alpha^* \rangle P, \Delta}\ con$$

Note that J in loop rule stands for the **loop invariant** of the hybrid game. It serves to unwrap the box program using an induction-like rule, since for $[\alpha^*]P$ to hold, we need all runs of hybrid game α reach a winning state where P is true. Therefore, if J holds initially, J implies P and J is maintained by one round of the game, we can combine the premises and reach the conclusion.

J in con rule stands for the **loop variant** of the hybrid game. It describes progress and serves to unwrap the diamond program using an induction-like rule, since for $\langle \alpha^* \rangle P$ to hold, there exists a certain number of rounds of the game that reaches a winning state where P holds. However, this number of rounds is often hard to calculate, and for large runs of the game, this rule substitutes the rule to unwrap one round at a time. Therefore, we are looking for a function J that depends on x such that J decreases by 1 in each round. We separate the premises to case `init`, `step` and `post`.

- `init`: In this case we show that $J(x)$ holds using the original hypothesis Γ.
- `step`: In this case we show that given $J(x)$ in the premise, after one run of α, $J(x-1)$ holds. We can interpret $J(x)$ as $x = d$ where d is equal to some value that changes in the game, so that after one round of game, d decreases by 1.
- `post`: In this case we show that after $J(x)$ has reached a minimum value, Angel can now break the loop and finish the game. Therefore we need to prove that for $x \leq 0$, $J(x) \vdash P$ at the end.

As we define the value d that decreases monotonically in the game, we need to make sure that d is guaranteed to decrease by some constant value. This is required to prove that d eventually reaches 0. We use the constant 1 without loss of generality in our proofs, as we can multiply a constant to get 1. In some cases, the value is not decreasing linearly but even quicker. This is also provable as long as we guarantee that over a game round, d decreases by at least 1 by defining $J(x) := x \geq d$.

Example 2.2.7. We can use a sequence of proof rules to formally prove the formula in Example 2.2.5:

$$\langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r).$$

The proof follows a typical sequent logic proof structure: the final conclusion formula starts at the bottom. The proof rules are labeled on the right of the horizontal bar of each step. The top line is where we finish the proof, which can be shown by arithmetic.

$$\vdash c < r \lor c \geq r \qquad \textit{cont.}$$

$$\frac{}{\vdash \langle c' = 1 \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r) \lor \langle c' = -1 \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ cut$$

$$\frac{}{\vdash \langle \{c' = 1 \cup c' = -1\} \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ \langle \cup \rangle$$

$$\frac{}{\vdash [r := *] \langle \{c' = 1 \cup c' = -1\} \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ [:=]$$

$$\frac{}{\vdash \langle \{r := *\}^d; \rangle \langle \{c' = 1 \cup c' = -1\} \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ \langle d \rangle$$

$$\frac{}{\vdash \langle \{r := *\}^d; \{c' = 1 \cup c' = -1\} \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ \langle ; \rangle$$

$$\frac{}{\vdash (c = r) \lor \langle \{r := *\}^d; \{c' = 1 \cup c' = -1\} \rangle \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ \lor R$$

$$\frac{}{\vdash \langle \{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^* \rangle (c = r)}\ \langle * \rangle$$

cont.:

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{*}{c<r\land t=r-c\vdash (t\geq 0\land \forall c\geq 0\rightarrow c_1=t+c(c_1=r))}}{c<r\vdash \exists t(t\geq 0\land \forall c\geq 0\rightarrow c_1=t+c(c_1=r))}\,\exists R}{c<r\vdash \langle c'=1\rangle(c=r)}\,\texttt{solve}}{c<r\vdash \langle c'=1\rangle(c=r)\lor\langle c'=-1\rangle(c=r)}\,\lor R
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{*}{c\geq r\land t=c-r\vdash (t\geq 0\land \forall c\geq 0\rightarrow c_1=t+c(c_1=r))}}{c\geq r\vdash \exists t(t\geq 0\land \forall c\geq 0\rightarrow c_1=-t+c(c_1=r))}\,\exists R}{c\geq r\vdash \langle c'=-1\rangle(c=r)}\,\texttt{solve}}{c\geq r\vdash \langle c'=1\rangle(c=r)\lor\langle c'=-1\rangle(c=r)}\,\lor R
}{c<r\lor c\geq r\vdash \langle c'=1\rangle\langle\{\{r:=*\}^d;\{c'=1\cup c'=-1\}\}^*\rangle(c=r)\lor\langle c'=-1\rangle\langle\{\{r:=*\}^d;\{c'=1\cup c'=-1\}\}^*\rangle(c=r)}\,\lor L
$$

2.2.4 KeYmaera X

KeYmaera X[1][6] is an automatic theorem prover designed to prove dGL theorem's correctness. It builds upon a trusted core of axioms, uniform substitution and propositional dL sequent calculus. A more detailed introduction is given in [6]. In KeYmaera X, Bellerophron Tactic Language [5] is a programming language for automatic proof constructions and proof search operations of the KeYmaera X core. We also represent our KeYmaera X proofs in this manner that can restore the corresponding dL sequent calculus.

In defining KeYmaera X models, we need to define a list of **Definitions** of constants, predicates and formulas; a list of **ProgramVariables**, and a **Problem** that gives the actual sequent to prove. The following table summarizes some common notations in KeYmaera X syntax. Note that we will not distinguish dGL syntax and KeYmaera X syntax in this work, as most of the notations are self-explanatory. ASCII syntax is described in [11], and definitions are described in [10].

HP	KeYmaera X
$x := e$	`x:=e;`
$x := *$	`x:=*;`
$x' = f(x)\&Q$	`{x'=f(x)&Q}`
$a \cup b$	`a ++ b`
a^*	`{a*}`
a^d	`a^@`
$a \cap b$	`a -- b`

Table 2.4: KeYmaera X syntax of dGL

Example 2.2.8. The Bellerophron code in Appendix B, `dGL-Example.kyx` represent the sequent proof in Example 2.2.7 of the formula

$$\langle\{\{r := *\}^d; \{c' = 1 \cup c' = -1\}\}^*\rangle(c = r).$$

[1]https://keymaerax.org/

Chapter 3

Preliminaries

In this chapter, we formally define the game variant we consider in the following proofs.

The pursuit-evasion game variant we use is always perfect information, which means that all information available to the reader is presented to the players, including all history moves, the playground, the current positions, etc. The game play is always sequential, i.e. the players make moves round by round. The pursuer is also called the cop (lion), and the evader is called the robber (man).

3.1 "Cops and Robbers" on Discrete Graphs

In Chapter 4, we will consider only nonempty simply undirected and connected graphs. Given graph G, we say $V(G)$ to be the set of vertices labeled using $\{1, 2, ..., n\}$ where $n = |G|$ is the number of vertices. We say $E(G)$ to be the set of edges, where each edge is denoted by a pair of vertices of the form e.g. $\{2, 3\}$. For simplicity, we sometimes write vertex $v \in V(G)$ and edge $e \in E(G)$. For a vertex $v \in V(G)$, we say the neighbors of v to be the set of vertices that share an edge with v, and denote this as $N(v)$. We abbreviate the combined set $\{v\} \cup N(v)$ as $N[v]$.

The classical version consists of a cop C and a robber R. The game is played on a finite graph with the players making alternative moves, starting with C. First C chooses a vertex to start then R. Then in each round, each player makes a move to another vertex that neighbors its previous position. The goal of C is to catch R, i.e. to be on the same vertex as R in the same round; the goal of R is to escape from C, i.e. cannot be caught in any round. For the initial selection part, if the cop chooses position after the robber instead, the cop can select the same position as the robber. For the actual game rounds, since the game is sequential, the actual ordering does not matter and each player can base his strategy upon the previous player.

We only consider games that have n cops, but always 1 robber. If there are more than one cop in the game, the cops share the same information set and act collaboratively.

This allows us to always regard the set of cops as the pursuer, and the robber as the evader. For multi-player versions, the cops move one by one in each round, followed by the robber's move.

Example 3.1.1. We provide a "Cops and Robbers" Game example here to illustrate the game.

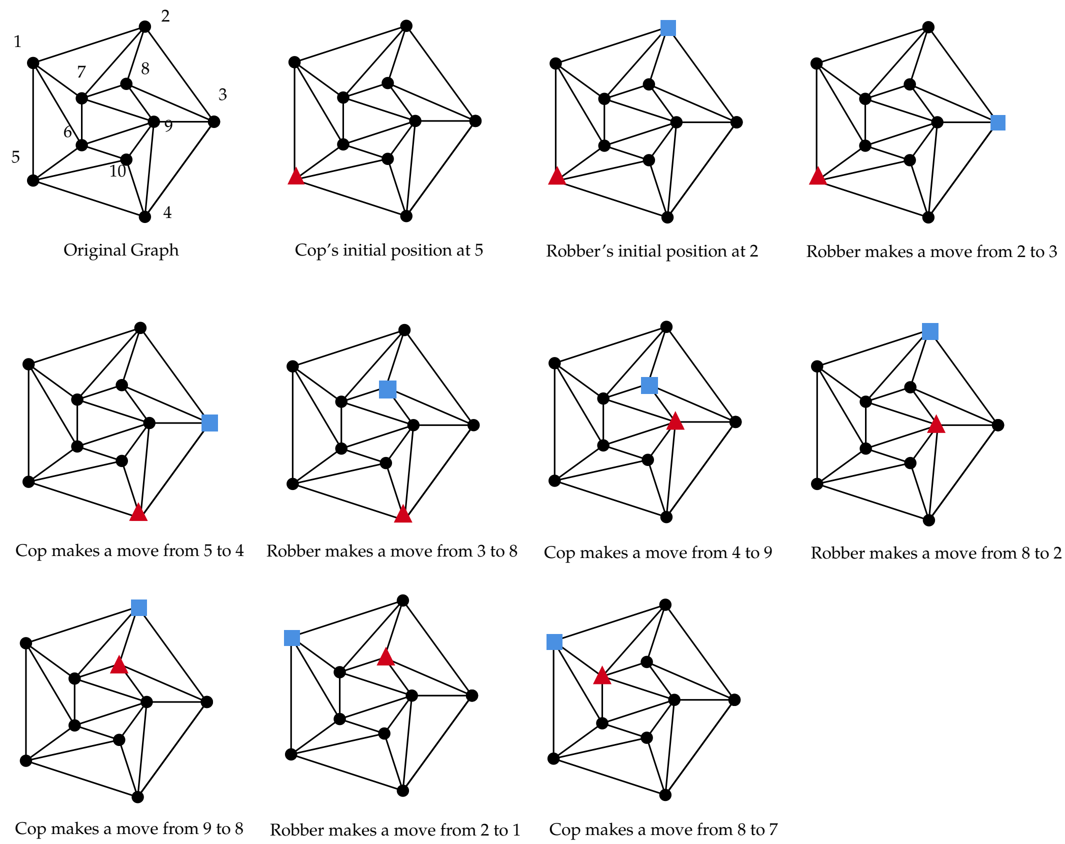

Figure 3.1: The blue square represents the robber; the red triangle represents the cop. The figures in order describes the "Cops and Robbers" game dynamics.

We can informally see that this graph in Figure 3.1 is robber-win. This is because the robber's strategy forms a cycle and can keep a distance of at least 1 with the cop. This will be formally proven using Theorem 4.3.9, since this graph is not dismantlable.

3.2 "Lion and Man" on Continuous Planes

In Chapter 5, we will consider Euclidean planes and straight-line movements. The positions of the cop and the robber are in Cartesian coordinates such that the cop's position is denoted (x_C, y_C) and the robber's position is denoted (x_R, y_R). The straight line movements simplify the movements of the players that allow us to define simple ODEs. The players have constant speeds that may or may not differ in the game.

The game is played on a geometric plane. The general setup remains the same as "Cops and Robbers" game. The cop first selects an initial position, then the robber selects an initial position. In each round, the robber first selects a direction of movement, and then the cop selects a direction. Then we allow the players to move simultaneously by a time range of at most 1, controlled by the cop. The winning condition of the game is for the cop to get arbitrarily close to the robber, i.e. in a constant-distance ball.

This particular order of play is by careful selection. For initialization, we let the cop move first, otherwise the cop wins by always selecting the robber's initial position. But in each round we let the robber move first, so that the cop has a chance to react to the robber's move.

This type of dynamics is different from the traditional "Lion and Man" game, in which the players move within a unit ball sequentially. Instead, we allow the players to act simultaneously, while their moves are restricted by straight lines. Details including the advantages and disadvantages are discussed in Chapter 5.

Example 3.2.1. We provide a "Lion and Man" Game example here to illustrate the game.

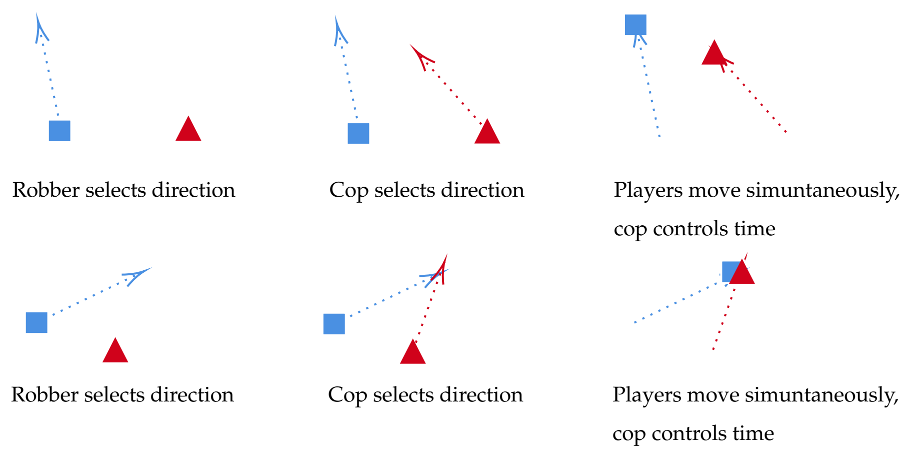

Figure 3.2: The blue square represents the robber; the red triangle represents the cop. The figures in order describes the "Lion and Man" game dynamics.

Chapter 4

Pursuit-Evasion Games on Discrete Graphs

In this section, we provide a detailed discussion of Cops-and-Robbers game, as described in Section 3, by considering 4 problems on different graph families and different numbers of cops.

4.1 Model Setup

For each discrete graph $G = (V, E)$, let $V = [n]$ be the set of vertices. Correspondingly, for each model, we define the following functions:

- `vertex`: `Real` $\rightarrow$ `Bool` s.t. $\texttt{vertex}(x) = x \in V$.
- `edge`: `Real`$\times$`Real` $\rightarrow$ `Bool` s.t. $\texttt{edge}(x, y) = \texttt{vertex}(x) \wedge \texttt{vertex}(y) \wedge (x, y) \in E$.

We first define the initial selection as

$$\texttt{initialize} \equiv \{c := *; ?\texttt{vertex}(c);\}\{r := *; ?\texttt{vertex}(r);\}^d$$

And a round of the game as

$$\texttt{step} \equiv \{co := c; c := *; ?\texttt{edge}(c, co);\}; \{?(c \neq r); ro := r; r := *; ?\texttt{edge}(r, ro);\}^d$$

The complete model of the game for a cop-win graph is as follows:

$$\langle\{\texttt{initialize}\}\{\texttt{step}\}^*\rangle(c = r) \tag{4.1}$$

This model generalizes the theorem for cop-win graphs of traditional cop-vs-robber with 1 cop and 1 robber. The diamond syntax is used here to describe that there exists a winning strategy for the cop in a finite number of rounds. The cop first selects a position and then the robber does. We use tests to guarantee that both members are responsible for taking a legal position, otherwise they lose the game. In each round, the cop first selects a new position and then the robber selects a new position. We also use an additional test $c \neq r$ to avoid the situation that even if the cop catches the robber in his round, the robber then moves away. If the cop catches the robber, which is the case that $c = r$, then the robber fails the test and loses the game immediately.

Notice that for the formal proofs, we use the simplified model by eliminating the initial moves, and instead build them into the precondition, where c_0 and r_0 are constants that satisfy $c_0, r_0 \in V$.

$$c = c_0 \wedge r = r_0 \vdash \langle \{\texttt{step}\}^* \rangle (c = r) \tag{4.2}$$

On the other hand, we may define the general theorem for a robber-win graph:

$$[\{\texttt{initialize}\}\{\texttt{step}\}^*](c \neq r) \tag{4.3}$$

Notice that this is the dual of 4.1, which implies that a finite graph must be either cop-win or robber-win.

4.1.1 A Concrete Example

We consider a star graph of 4 vertices as in Figure 4.1. We further define the following predicates:

- $\texttt{vertex}(x) = (x = 1 \vee x = 2 \vee x = 3 \vee x = 4)$
- $\texttt{edge}(x, y) = (x = 2 \wedge (y = 1 \vee y = 3 \vee y = 4)) \vee (y = 2 \wedge (x = 1 \vee x = 3 \vee x = 4))$

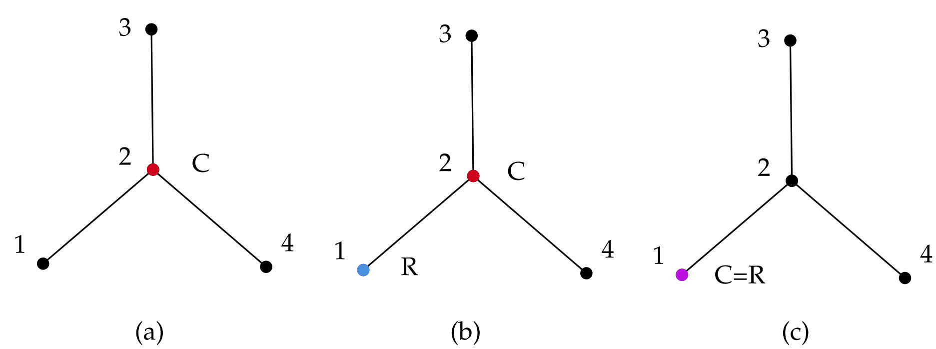

Figure 4.1: A star graph with leaves 1,3,4 connected to center node 2; (a) The cop chooses initial position 2; (b) The robber chooses initial position 1; (c) The cop moves in the first round from 2 to 1 and catches the robber.

We can now prove Theorem 4.1 by first letting $c = 2$, it passes the test. Then we expand the possible positions of the robber and discuss by cases. Suppose $r = 1$, then we move into diamond star rule and expand by one step. Now we store the old position of the cop as $c_o = 2$ and select $c = 1$. Now the edge $cc_o = \{1, 2\}$ passes the edge test, but $c = r$, so the cop wins.

This proof sketch shows the analysis we make when we have a concrete graph example: first we define the functions that characterize the graph; then we expand the definitions and discuss the cop's strategy by cases.

4.2 Abstraction

We see from the previous example in Section 4.1 that defining `vertex` and `edge` functions only serves for a specific graph. This kind of definition also results in brute-force expanding the diamond star rule by enumerating each specific move. It is natural to consider the more general case as in graph families. In discrete graphs, we can make this generalization by abstracting the size of the graph as an integer n. Although the nature of KeYmaera X variables does not allow it to directly define integer types, we can define natural numbers by proving lemmas of its group properties.

Definition 4.2.1. (Natural Numbers) $n \in \mathbb{N}^+$ if and only if `nat(n)`,

$$\texttt{nat(n)} \equiv \langle\{?(n > 1); n := n - 1; \}^*\rangle(n = 1)$$

Note that throughout this paper, we use the definition of natural numbers such that the minimum natural number is 1. We delete the consideration of 0 for simplicity here, and since the graphs we consider are finite non-empty graphs, we don't need 0 case by construction.

Lemma 4.2.2. (Plus) For all $m, n \in \mathbb{N}$, $m + n \in \mathbb{N}^+$.

Lemma 4.2.3. (Multiplication) For all $m, n \in \mathbb{N}^+$, $m \times n \in \mathbb{N}^+$.

Lemma 4.2.4. (Inequality) For all $n \in \mathbb{N}^+$, $c \in \mathbb{R}$, $n \geq c \to n + 1 > c$.

Although the lemmas above do not cover all common mathematical properties of natural numbers, these three are the ones we rely on to show winning strategies in the section below. To prove these lemmas, we need to first verify natural numbers induction.

Lemma 4.2.5. (Induction) Given arbitrary predicate p, we have

$$p(1) \to \Big(\forall x\big(nat(x) \to p(x) \to p(x+1)\big)\Big) \to \Big(\forall y\big(nat(y) \to p(y)\big)\Big)$$

The formula is obviously valid since it's the representation of mathematical induction in dGL.

Lemma 4.2.5 provides sufficient tools to formalize the proof of the mathematical properties of natural numbers. The formal proofs for Lemma 4.2.2, 4.2.3, 4.2.4 and 4.2.5

use KeYmaera X, and are found in the Appendix. Here we give a proof sketch of Lemma 4.2.4. We first formalize the lemma in dL language:

$$\forall m\Big(nat(m) \to \forall n\big(nat(n) \to nat(m+n)\big)\Big).$$

Define predicate $p(x) := nat(m+x)$. Then we need to show the base case $nat(m) \to nat(m+1)$ and the induction step $nat(m+x) \to nat(m+x+1)$. Both cases can be proven simply by expanding the definition and unrolling $\langle * \rangle$ once.

4.3 Proofs of Winning Strategies

Now we can formally prove the theorems for some common graph families of arbitrary size. By considering a certain type of graph, we could define specification in `edge` definition and the corresponding cop's strategy. In this section, we separately considered the family of cycles, trees, dismantlable graphs and grids.

4.3.1 Cycles

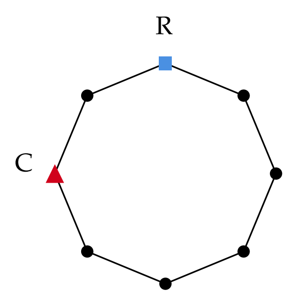

Figure 4.2: A cycle graph example. The blue rectangle denotes the robber; the red triangle denotes the cop. The invariant in Equation 4.4 describes that the distance between the players is kept constant.

Definition 4.3.1. A **cycle** graph is a graph that consists of a single cycle. This can be formally defined by

- $\texttt{vertex}(x) = \texttt{nat}(x) \wedge x \leq n$.
- $\texttt{edge}(x, y) = \texttt{mod}((x-y), n, 1) \vee \texttt{mod}((x-y), n, -1)$.

where `mod` is defined as

$$\texttt{mod}(x, n, r) \equiv (x = r \vee x + n = r).$$

Recall that in mathematics, the modulo operation mod is defined as

$$x \bmod y = r \text{ if and only if } x = cy + r \text{ for some } c \in \mathbb{N}.$$

This definition is a simplified version of mod, since the domain we work with is in $[n] = \{1, 2, ..., n\}$.

Theorem 4.3.2. The robber has a winning strategy on cycles of size n, when $n \geq 4$.

The selection of n here is because, when $n \leq 3$, the only possible cycle graph is a triangle. We can show by brute-forcing all possible moves that the cop has a winning strategy.

The formal model is defined as

$$\texttt{vertex}(c) \land \texttt{vertex}(r) \land \texttt{mod}(r - c, n, 2) \vdash [\texttt{step}^*](c \neq r).$$

Proof. The winning strategy of the robber is to always keep a constant distance from the cop. We formally define the invariant as

$$J \equiv \texttt{mod}(r - c, n, 2). \tag{4.4}$$

For arbitrary initial position c satisfying $\texttt{vertex}(c)$, the robber selects $r = (c + 2) \bmod n$. Now we expand $[^*]$ using `loop` rule.

We separately prove the following cases:

- `init`: To show $c \in V(G) \land r \in V(G) \land \texttt{mod}(r - c, n, 2) \vdash J$:
  By the way of initial selection, the claim is obvious.
- `step`: To show $J \vdash [\texttt{step}]J$:
  Since we are working from the robber's perspective, we define the robber's strategy and consider all opponent moves. Therefore, given J as premise, we will discuss cases on all possible cop moves: in this particular graph, the cop can either move to $c + 1 \bmod n$ or move to $c - 1 \bmod n$ following the game rules. This allows the robber to respond by case:
  - If the cop moves to $c + 1 \bmod n$, the robber moves to $r + 1 \bmod n$.
  - If the cop moves to $c - 1 \bmod n$, the robber moves to $r - 1 \bmod n$.

  In either case, the invariant J is kept by arithmetic.
- `post`: We also infer that $J \vdash (c \neq r)$ from arithmetic.

This proof is also done in KeYmaera X. □

4.3.2 Trees

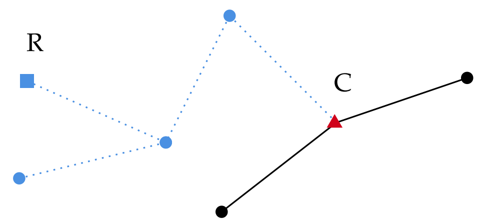

Figure 4.3: A tree graph example depicting the cop's strategy. The blue rectangle denotes the robber; the red triangle denotes the cop. The blue dotted area depicts $S(c, r)$. The cop's strategy shrinks the size of this region over rounds.

Definition 4.3.3. A **tree** graph is an undirected, acyclic and connected graph.

Theorem 4.3.4. The cop has a winning strategy on a tree graph.

Since trees are hard to represent in dGL, we proved an example case of this theorem, i.e. the line graph, in KeYmaera X. The general tree graph case is proved below using paper-based proof.

Define the following variable in a tree G:

- Let $d_G(r, c)$ be the distance between r and c in G. If r, c are not connected, define $d_G(r, c) = \infty$.
- Let $P(c, r)$ be the set of all vertices on the unique path from c to r, $p_i(c, r)$ be the ith vertex on this path, 0-indexed.
- Let $S(c, r) = \{v \in V(G), c \notin P(v, r)\}$, which describes the active region for r to move, without meeting the cop.

Theorem 4.3.5. When G is a tree, if c is on the path between a and b, then the path between a and b is exactly the path between a and c and the path between c and b. i.e.

$$a \in V(G) \land b \in V(G) \land c \in V(G) \land c \in P(a, b) \vdash P(a, b) = P(a, c) \cup P(c, b).$$

Lemma 4.3.6. If c moves to the next vertex on the path from c to r, then $S(c, r)$ must decrease by at least 1. $|S(c, r)| > |S(p_1(c, r), r)|$.

We define the formal model as following:

$$c = c_0 \land \texttt{vertex}(r) \vdash \langle \texttt{step}^* \rangle (c = r)$$

where c_0 is the medium vertex on the longest path in the tree, i.e. the center of the tree.

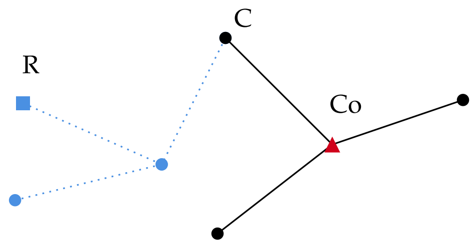

Figure 4.4: The cop's strategy following the state in Figure 4.3.2. The cop moves from C_0 to C. The blue dotted area dipicts $S(c, r)$.

Proof. Let the initial selection of the robber be r_0 that satisfies $\texttt{vertex}(r_0)$. We formally define the variant d as

$$J(d) \equiv d \geq |S(c, r)|.$$

This means that the active region of the robber decreases by at least 1 after a game round. The intuitive strategy of the cop is to move to the next position on the path from c to r, as shown in Figure 4.4. We separately prove the following cases:

- `init`: To show $c = c_0 \wedge \texttt{vertex}(r) \vdash \exists d. J(d)$:
  Select $d = |S(c_0, r_0)|$ and the conclusion follows by propositional logic.
- `step`: To show $d \geq 0, J(d) \vdash \langle \texttt{step} \rangle J(d)$:
  Given $J(d)$ as premise and $d \geq 0$ we will discuss all possible robber's moves:

  Since $d > 0$, there exists at least a legal move r' such that $r' \neq c$. Notice that by the property of the tree, $S(c, r) = S(c, r')$. Then the cop selects $c' = p_1(c, r)$.

  Applying Lemma 4.3.6, $|S(c, r)| \geq |S(p_1(c, r), r)| + 1$. Therefore $J(d)$ holds.
- `post`: To show $d \leq 0, J(d) \vdash (c = r)$. This means $d = 0$, i.e. the size of the active region of r is 0, so $c = r$.

□

4.3.3 General Graph

In this section, we aim to identify winning strategies for all undirected, discrete graphs. First we will introduce a type of graph, the dismantlable graph. It can be dismantled by repeatedly removing a corner vertex. This type of graphs creates a dismantling order that helps us to define the cop-winning strategies. An example is shown in Figure 4.5.

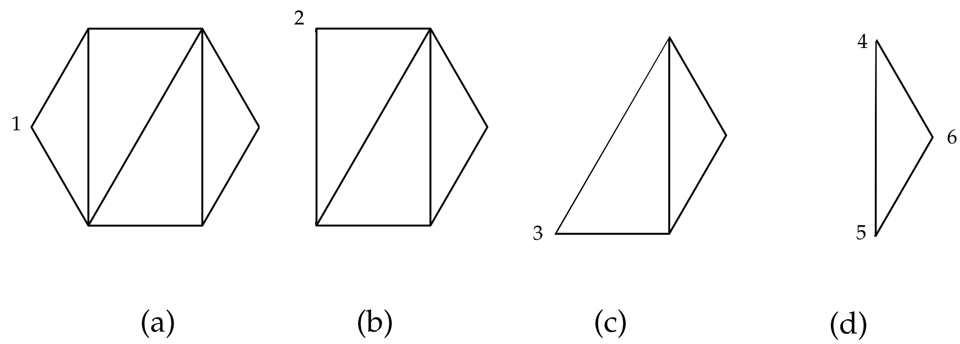

Figure 4.5: An dismantlable graph example. The labels denote the dismantling order by removing corners of the graph. (a) is the original graph; (b) is the remaining graph after removing 1; (c) is the remaining graph after removing 2; (d) is the remaining graph after removing 3, and the final 4,5,6 are all corner vertices.

Definition 4.3.7. For a vertex v, let $N(v)$ be the set of neighbors of v and $N[v] = N(v) \cup \{v\}$. A **corner** of a graph G is a vertex v such that $N[v] \subseteq N[u]$ for some $u \in N(v)$.

Definition 4.3.8. A graph is **dismantlable** if it has a dismantling-order, which is an ordering of the vertices from 1 to n, such that

For all $i \in [n]$, vertex i is a corner of the vertex-induced subgraph $\{i, i+1, ..., n\}$ of G.

A **dismantling order** is an elimination ordering for chordal planar graphs. i.e. we can always find a corner vertex and delete it, and repeat this process until there's no vertex left in the graph.

Theorem 4.3.9. The cop has a winning strategy on a graph G of n vertices if and only if it is dismantlable.

Proof. Let the dismantling order for G be $f : G \to [n]$ where $f(v) = i$ is the order of v. Let the vertex set be labeled as in the dismantling order. Since the graph is guaranteed to contain a corner vertex, we will let G_i be the graph $G_i = G \cap \{i, ..., n\}$.

There is a mapping $f_i : G_i \to G_{i+1}$ for all the vertices such that

$$f_i(v) = \begin{cases} u & \text{if v is the corner vertex s.t. } N[v] \subseteq N[u] \\ v & \text{otherwise} \end{cases}$$

Define $F_i : G_1 \to G_i = f_i \cdot f_{i-1} ... \cdot f_1$. At each step i, if robber is on u, the cop must be on $F_i(u)$; next step the robber moves to $v \in N(u)$, the cop moves to $F_{i+1}(v)$.

Now we can use this rule to construct a dGL proof. We define the formal model as

$$c = n \wedge r \in [n] \vdash \langle \texttt{step}^* \rangle (c = r)$$

Let the variant be

$$J(d) \equiv F_d(r) = c.$$

- `init`: The cop will select the initial position n, so that by definition, $F_n(r) = n$ for all $r \in [n]$.
- `step`: To show $F_d(r) = c \vdash \langle \texttt{step} \rangle F_{d-1}(r) = c$.

  By definition, the new position $r_1 \in N(r)$. Since $F_d(r) = c$, then $f_d(F_{d-1}(r)) = c$. Then $N[F_{d-1}(r)] \subseteq N[c]$ in G_d. Since $r_1 \in N(r)$, $F_{d-1}(r_1) \in N[F_{d-1}(r)]$ in G_{d-1}. So $F_{d-1}(r_1) \in N[c]$. Now we can let the cop's new position be $F_{d-1}(r_1)$ that satisfies the test, and thus proves the conclusion.
- `post`: Since F_1 is the identity function, then $F_1(r) = c$ implies $r = c$.

□

4.3.4 Grids

Theorem 4.3.9 provides a general strategy to identify 1-cop-win graphs and constructive winning strategies. Here we extend the discussion to multiple cop games. This game setting adds a level of difficulty in the level of proofs. By Theorem 4.3.2, grid graphs are obviously robber-win for 1 cop, because they contain at least a cycle of size 4. Instead, we claim that grid graphs are 2-cop-win. An example grid graph is shown in Figure 4.6.

Definition 4.3.10. A **grid** graph is a graph forms a rectangular tilling. It can be formally defined as

- $\texttt{vertex}(x, y) = \texttt{nat}(x) \wedge \texttt{nat}(y) \wedge x \geq 0 \wedge x \leq n \wedge y \geq 0 \wedge y \leq n$.
- $\texttt{edge}(x_1, y_1, x_2, y_2) = \texttt{vertex}(x_1, y_1) \wedge \texttt{vertex}(x_2, y_2) \wedge ((|x_1 - x_2| = 1 \wedge y_1 = y_2) \vee (|y_1 - y_2| = 1 \wedge x_1 = x_2))$

Here we modified the model setup from Section 4.1, since we now model the game in 2D coordinates and we will need separate variables to represent the cops. Let (x_1, y_1) denote the position of Cop C_1, (x_2, y_2) denote the position of Cop C_2 and (x_R, y_R) denote the position of the Robber R. We define the coordinates of the game to be $(1, 1)$ to (n, n).

Theorem 4.3.11. 2 cops that share all information have a winning strategy in a grid graph of size $n \times n$.

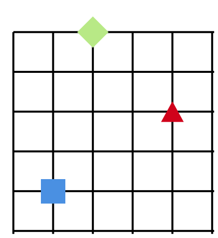

Figure 4.6: A grid graph example of size 5×5. The green diamond denotes C_1; the red triangle denotes C_2; the blue rectangle denotes R.

To formalize the model, we define the following hybrid program:

$$\texttt{move1} \equiv \{x_{CO1} := x_{C1}; y_{CO1} := y_{C1}; x_{C1} := *; y_{C1} := *; ?\texttt{edge}(x_{C1}, y_{C1}, x_{CO1}, y_{CO1})\}$$
$$\texttt{move2} \equiv \{x_{CO2} := x_{C2}; y_{CO2} := y_{C2}; x_{C2} := *; y_{C2} := *; ?\texttt{edge}(x_{C2}, y_{C2}, x_{CO2}, y_{CO2})\}$$
$$\texttt{moveR} \equiv \{x_{R1} := x_R; y_{R1} := y_R; x_R := *; y_R := *; ?\texttt{edge}(x_R, y_R, x_{R1}, y_{R1})\}$$

We use the shorthand $\texttt{same}(x_1, y_1, x_2, y_2) := x_1 = x_2 \wedge y_1 = y_2$ to verify that the coordinates (x_1, y_1), (x_2, y_2) are the same.

Then we define the game round:

$$\texttt{step} \equiv \{\texttt{moveR};\}^d\{?\neg(\texttt{same}(x_{C1}, y_{C1}, x_R, y_R) \vee \texttt{same}(x_{C2}, y_{C2}, x_R, y_R)); \texttt{move1}; \texttt{move2};\}$$

Now we can define the formal model:

$$x_{C1} = n \wedge y_{C1} = n \wedge x_{C2} = n \wedge y_{C2} = n$$
$$\vdash \langle \texttt{step}^* \rangle (\texttt{same}(x_{C1}, y_{C1}, x_R, y_R) \vee \texttt{same}(x_{C2}, y_{C2}, x_R, y_R))$$

Definition 4.3.12. Let $\texttt{manhattan}(x, y) = x + y$ denote the **Manhattan Distance** between (x, y) and the origin $(0, 0)$.

Since the domain we work with is for $x, y \in [n]$, we can simplify the absolute values here.

Proof. The variant is defined as

$$\begin{aligned} J(d) \equiv & \neg(\texttt{same}(x_{C1}, y_{C1}, x_R, y_R) \vee \texttt{same}(x_{C2}, y_{C2}, x_R, y_R)) \rightarrow \\ & (x_R \le x_{C1} \wedge x_R \le x_{C2} \wedge y_R \le y_{C1} \wedge y_R \le y_{C2} \wedge \\ & d = \texttt{manhattan}(x_{C1}, y_{C1}) + \texttt{manhattan}(x_{C2}, y_{C2}) - \texttt{manhattan}(x_R, y_R)) \end{aligned}$$

- `init`: Select initial position $(x_{C1}, y_{C1}) = (x_{C2}, y_{C2}) = (n, n)$. Then $J(d)$ can be proven automatically.
- `step`: To show $d > 0, J(d) \vdash \langle \texttt{step} \rangle J(d)$, we first separately consider the case where the hypothesis part in J is false. Then the cops' strategy can exactly copy the move of the robber, which makes $J(d)$ holds.

  When the hypothesis part is true, we will discuss by case of initial positions:

  – Let the robber be at (x_R, y_R). Let the new position the robber selects to be (x_{R1}, y_{R1}). Note that the distance $(x_{R1} - x_R)^2 + (y_{R1} - y_R)^2 \le 1$, and $-1 \le x_{R1} + y_{R1} - x_R - y_R \le 1$.

    Define the first cop's strategy to be, given the new position (x_{R1}, y_{R1}):

    Let the new position for C_1 to be $\begin{cases} (\mathrm{x}_{C1}, y_{C1} - 1) & \text{if } \mathrm{x}_{R1} = x_{C1}, y_{R1} - y_{C1} \ge 0 \\ (\mathrm{x}_{C1} - 1, y_{C1}) & \text{if } \mathrm{x}_{R1} < x_{C1} \\ (\mathrm{x}_{C1} + 1, y_{C1}) & \text{if } \mathrm{x}_{R1} > x_{C1} \end{cases}$

Define the second cop's strategy to be, given the new position (x_{R1}, y_{R1}):

Let the new position for C_2 to be $\begin{cases} (\mathrm{x}_{C2} - 1, y_{C2}) & \text{if } \mathrm{y}_{R2} = y_{C2}, x_{R2} - x_{C2} \geq 0 \\ (\mathrm{x}_{C2}, y_{C2} - 1) & \text{if } \mathrm{y}_{R2} < y_{C2} \\ (\mathrm{x}_{C2}, y_{C2} + 1) & \text{if } \mathrm{y}_{R2} > y_{C2} \end{cases}$

We first show that the invariant $x_{C1} \geq x_{R1} \wedge y_{C1} \geq y_{R1}$ holds: If $x_{R1} = x_{C1}$, then we must have the case $y_{R1} - y_{C1} > 0$, since otherwise the cop has caught the robber. Then it's safe to decrease y_{R1} by 1. In other cases, the invariant holds for simple geometry. A similar analysis can be done for $C2$ case.

Then we reason about the invariant. Notice that $\texttt{manhattan}(x, y)$ changes by absolute value at most one. We argue that only the following cases in Table 4.1 (illustrated in Figure 4.7) are possible:

Table 4.1: The possible cases for the cop's strategy

$\Delta\texttt{manhattan}(x_R, y_R)$	$\Delta\texttt{manhattan}(x_{C1}, y_{C1})$	$\Delta\texttt{manhattan}(x_{C2}, y_{C2})$
-1	-1	-1
+1	+1	-1
+1	-1	+1
+1	-1	-1

We present pictures for these cases as Figure 4.7.

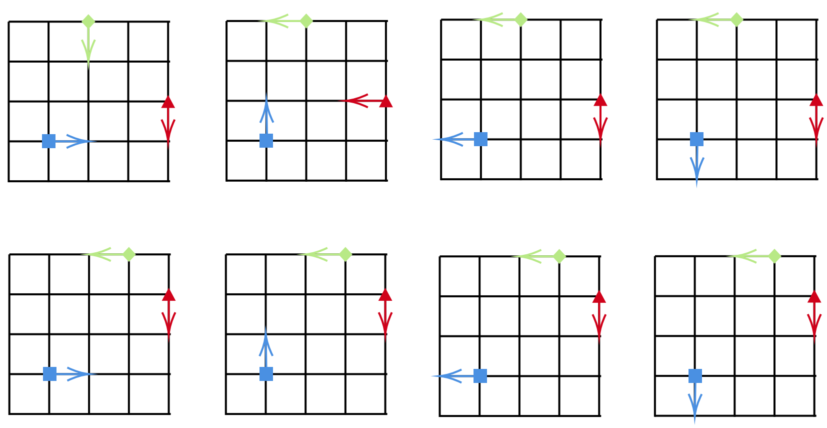

Figure 4.7: Cases to illustrate the Cops' strategy: The green diamond denotes C_1; the red triangle denotes C_2; the blue rectangle denotes R. The arrows denote the movement through a game round. The positions are relative.

An intuitive way to understand this strategy is that C_1 moves in order to reach $x_{C1} = x_R$ first, when this is achieved then move on the y-axis; C_2 moves in order to reach $y_{C2} = x_R$ first, when this is achieved then move on the x-axis.

- `post`: To show $d \leq 0, J(d) \vdash (\texttt{same}(x_{C1}, y_{C1}, x_R, y_R) \vee \texttt{same}(x_{C2}, y_{C2}, x_R, y_R))$, use $\rightarrow L$ to separately consider the case where the hypothesis part in J. If the hypothesis is true, the conclusion also follows. Otherwise we reach the case that all $C1, C2, R$ are on the same coordinate, therefore the conclusion is also true.

$\square$

Chapter 5

Pursuit-Evasion Games on Continuous Planes

In this section, we provide a detailed discussion of "Lion and Man" game, as described in Section 3. Instead of moving along edges, the players now move inside a continuous 2D plane with straight-line, continuous and simultaneous movements. We have transformed from discrete dynamics to continuous dynamics, which is a more generalized scenerio and a better simulation of real physics. We consider problems in the following subsections including different properties of game regions, and discuss the effect they make on the winning party and the winning strategies.

We also make note of the definition of legal game regions more carefully using illustrations in Figure 5.1.

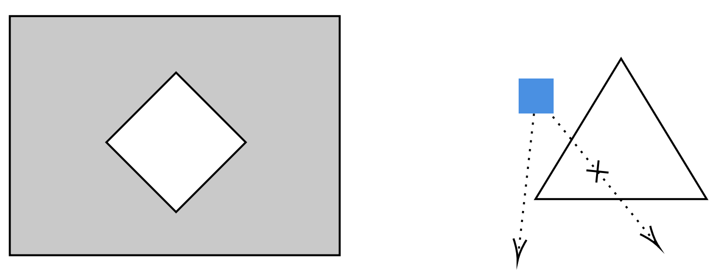

Figure 5.1: Example of obstacles. The white region on the left picture is considered illegal, the grey region and the black boundaries are considered legal; the dotted lines denote the possible moves for the blue rectangle given the triangle obstacle: the one without the cross is legal and the other one is illegal.

5.1 Model Setup

Definition 5.1.1. We use the following predicates to denote the winning condition:

- $\texttt{inBall}(x_C, y_C, x_R, y_R, r) \equiv (x_C - x_R)^2 + (y_C - y_R)^2 \leq r^2$ where r is a constant.
- $\texttt{inRegion}(x, y)$: $\texttt{Real}^2 \to \texttt{Bool}$: describes whether the current location (x, y) is in the game region.

We define the position of cop to be (x_C, y_C) and the position of the robber to be (x_R, y_R). We also define velocity v_C and v_R correspondingly. Unlike the previous section where the players move along graph edges at the same speed, here we allow them to move freely inside the legal game region. For the simplicity of modeling, we restrict the players to move in straight lines. To this end, we use vectors to define the direction of movement for the players: (x_{C1}, y_{C1}) and (x_{R1}, y_{R1}). Then we define the following helper predicates:

We first define the actual game moves using the following hybrid program:

$$\texttt{ctrl} \equiv \{\{x_{R1} := *; y_{R1} := *; ?x_{R1}^2 + y_{R1}^2 = 1;\}^d \{x_{C1} := *; y_{C1} := *; ?x_{C1}^2 + y_{C1}^2 = 1;\}$$

$$\begin{aligned} \texttt{plant} \equiv & \{x_C' = v_C \cdot x_{C1}, y_C' = v_C \cdot y_{C1}, x_R' = v_R \cdot x_{R1}, y_R' = v_R \cdot y_{R1}, t' = 1 \\ & \&\big(t \leq 1 \wedge \texttt{inRegion}(x_C, y_C)\big)\} \end{aligned}$$

And the predicate cop_win as

$$\big(\texttt{inRegion}(x_R, y_R) \to \texttt{inBall}(x_C, y_C, x_R, y_R, r)\big)$$

Here ctrl is the round when the robber selects direction first, and then the cop selects direction. Both players need to ensure that this direction vector is a unit vector. plant is the actual movement of the players. The players move in the direction defined in ctrl, with a predefined velocity constant. cop_win states the final winning condition for the cop. Therefore the complete model of the game for a cop-win region is as follows:

$$(x_C, y_C) = (x_{C0}, y_{C0}) \vdash \langle\{\texttt{ctrl}; t := 0; \texttt{plant}\}^*\rangle \texttt{cop\_win} \tag{5.1}$$

Similarly, the complete model of the game for a robber-win region is as follows:

$$(x_R, y_R) = (x_{R0}, y_{R0}) \vdash [\{\texttt{ctrl}; t := 0; \texttt{plant}\}^*](\neg \texttt{cop\_win}) \tag{5.2}$$

where $x_{C0}, y_{C0}, x_{R0}, y_{R0}$ are constants that state the initial positions.

There are several details in this selection of dynamics: First, we select the domain constraint with a restriction of time. This is easier to model and can generalize to different game regions, and we can control the termination of a game round.

Second we let the cop to be in charge of time. Note that we also add a restriction of the maximum time, otherwise the cop could force the robber to move until he reaches out of the legal region. However, if we let the robber to be in charge of time, the robber can always select time duration 0 to avoid getting caught. Assume that the model instead restricts the robber to move at least 1 second, which is interchangeable with the model that restricts the cop to move at most 1 second. This also explains why we add the constraint of not moving out of boundary inside the ODE, but only for the cop. It doesn't make sense to also restrict the robber here since it's not the cop's job to do it. Instead, the constraint for the robber is added in the cop's winning condition. This makes sure that the robber will be responsible for not to be out of region in the game move, because otherwise the cop would stop immediately and the robber loses the game.

Third, we let the robber select direction first. This is a selection that actually favors the cop, since it can determine its strategy based on the robber's direction. This is because we primarily work with cop-winning cases as they are often more involved.

In the end, we define the winning condition to be within a constant range of distance, rather than letting the cop be at the same position as the robber. This is because, if we have the distance between the cop and the robber to be less than 1, sometimes we need the cop to move less than 1 second to avoid going past the robber. So the purpose is to simplify proofs. However, this does not affect the result. If we can reduce the variant to reach this condition, it can also reduce to the case where the cop goes to the exact same position as the robber.

5.2 Faster Cop

5.2.1 Unbounded Plane

We first discuss the scenario where the cop is faster than the robber, and consider the unbounded plane. The intuition is that the cop can catch the robber since the cop is faster, and here we formally prove the winning strategy. The example plane is shown in Figure 5.2.

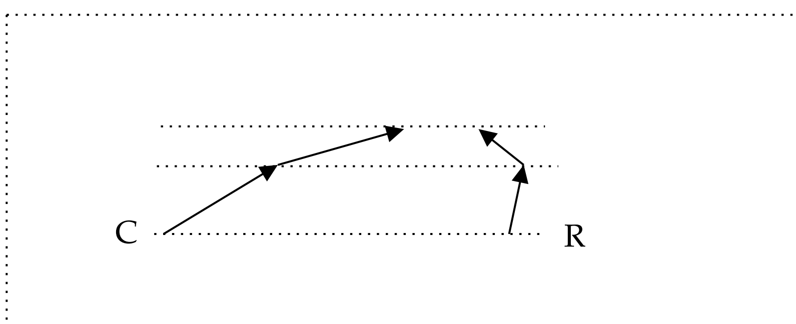

Figure 5.2: An unbounded plane example. Whenever the robber moves, the cop moves according to the invariant $y_C = y_R$.

Theorem 5.2.1. If the game region is unbounded Euclidean plane with speed $v_C > v_R > 0$, then the cop has a winning strategy.

Without loss of generality, the model assumes the cop is at $(-1, 0)$ and the robber is at $(1, 0)$ in the initial state. The cop moves in a constant velocity v_C and the robber moves in a constant velocity v_R where $v_R = 1, v_C = 2$. We will not restrict a closed plane in the model, because the cop is guaranteed to catch the robber even if in open region.

Proof. The proof structure relies on `con` rule and the following variant J:

$$J(d) \equiv d \geq x_R - x_C - 1 \land y_C = y_R \land x_R \geq x_C.$$

Intuition: In each round, we will select the direction of the cop based on the direction of the robber that satisfies the invariant $y_C = y_R$, which is calculated using $y_{C1} = \frac{y_{R1}}{2}$, $x_{C1} = \sqrt{1 - \frac{y_{R1}}{2}^2}$. The $x-$axis distance of x_C is guaranteed to approach x_R by at least 1 in each round, so in the end we will have `inBall`$(x_C, y_C, x_R, y_R, 1)$.

This lower bound attains equality when C and R are moving in the same direction, where $\Delta(d) = v_C - v_R$.
Let $H = v_C \cdot y_{C1} = v_R \cdot y_{R1}$, we know

$$\Delta(d) \leq v_R \cdot x_{R1} - v_C \cdot x_{C1}.$$

Let $d_1 = v_R \cdot x_{R1}, d_2 = v_C \cdot x_{C1}$. So we have $d_1 - d_2 = (d_1^2 - d_2^2)/(d_1 + d_2)$. Note that $d_1^2 - d_2^2 = v_C^2 - H^2 - v_R^2 + H^2$, and $d_1 + d_2 = \sqrt{v_C^2 - H^2} + \sqrt{v_R^2 - H^2}$ where H is the movement in y-axis. Obviously $d_1 + d_2$ reaches maximum if $H = 0$, where $d_1 - d_2$ reaches minimum. Therefore, the variant $x_R - x_C$ is guaranteed to decrease by at least 1.

- `init`: To show $C = (-1, 0), R = (1, 0) \vdash \exists d. J(d)$, we first note that the variant is satisfied by selecting $d = x_R - x_C - 1$. Then the condition $y_C = y_R = 0$ and $x_R \geq x_C$ are met by arithmetic.
- `step`: To show $d > 0, J(d) \vdash \langle \texttt{step} \rangle J(d-1)$, since $d > 0$, $x_R - x_C > 1$. Let (x_{R1}, y_{R1}) be the direction of the robber, then we select (x_{C1}, y_{C1}) according to the strategy stated above. This guarantees that after a certain amount of time, the invariant $y_C = y_R$ holds.

  Now we consider the following cases separately:

  - If $x_R + x_{R1} < x_C + x_{C1}$, then the players cannot safely move by time amount of 1. Let t satisfy $x_R + tx_{R1} = x_C + tx_{C1}$, which also satisfies $0 \leq t < 1$, and ODE moves by t. Now the final condition satisfies $x_R = x_C$. Since initially we have $x_R - x_C > 1$, d has decreased by at least 1.
  - If $x_R + x_{R1} \geq x_C + x_{C1}$, then the players can safely move time duration of 1 without violating the invariant. Therefore the final condition satisfies $x_R \geq x_C$. And by math we reasoned before, d has decreased by at least 1.
- `post`: To show $d \leq 0, J(d) \vdash$ `inBall`$(x_C, y_C, x_R, y_R, 1)$, obviously when $y_C = y_R$, $x_R - x_C \leq 1$, the final condition is satisfied.

This proof is also done in KeYmaera X. □

5.2.2 Unbounded Plane with Line Obstacle

Now we discuss the case if we add a line obstacle to the unbounded plane. An example game region is shown in Figure 5.3.

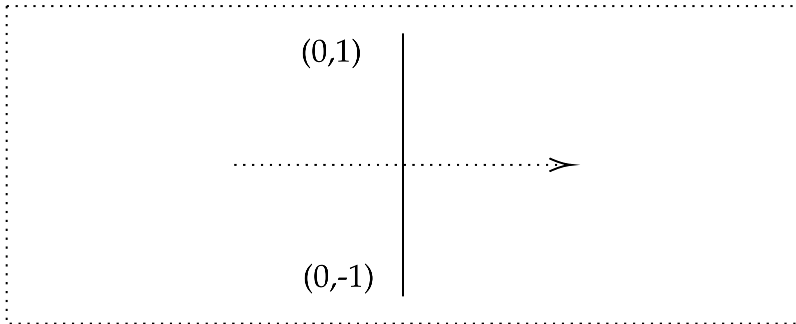

Figure 5.3: An example game region of a line obstacle at $(0,-1)$ to $(0,1)$.

Theorem 5.2.2. If the game region is an unbounded Euclidean plane containing 1 line-segment obstacle and $v_C > v_R > 0$, the cop has a winning strategy.

In our proof, we specify our field to be Figure 5.3 to ease computation complexity. We can always define the coordinates for the open planes with a line obstacle to look like Figure 5.3. The optimal strategy is to follow the most time-wise sufficient path to the position of the robber in the previous round. We will define this formally as the time-optimal path.

Definition 5.2.3. We define the **Euclidean Distance** between (x_1, y_1) and (x_2, y_2) in the plane with obstacles by $euc(x_1, y_1, x_2, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Definition 5.2.4. We define the **time-optimal path** between (x_1, y_1) and (x_2, y_2) in the plane with obstacles by the shortest path one can navigate from (x_1, y_1) to (x_2, y_2) without trespassing the obstacle. So

$$\begin{aligned} top(x_C, y_C, x_R, y_R) = \min\big(&euc(x_C, y_C, 0, 1) + euc(0, 1, x_R, y_R), \\ &euc(x_C, y_C, 0, -1) + euc(0, -1, x_R, y_R)\big) \end{aligned}$$

Theorem 5.2.5. (Axiom) When the field consists of only line segment obstacles, the time-optimal paths are also line segments.

Definition 5.2.6. We define the function $dist : \mathbb{R}^4 \to \mathbb{R}$ of the time-optimal path between (x_C, y_C) and (x_R, y_R) to be

$$dist(x_C, y_C, x_R, y_R) = \begin{cases} top(x_C, y_C, x_R, y_R) & \text{if the line segment connecting } (x_C, y_C) \\ & \text{and } (x_R, y_R) \text{ intersects with the obstacle} \\ euc(x_C, y_C, x_R, y_R) & \text{otherwise} \end{cases}$$

To formalize our proof, an additional definition to describe the point at which a line crossed the y-axis is required:

Definition 5.2.7. $\texttt{cut}(x_C, y_C, x_R, y_R)$ is the **intercept** of the line crossing (x_C, y_C) and (x_R, y_R) with the y-axis.

$$\texttt{cut}(x_C, y_C, x_R, y_R) \equiv y_C - \frac{y_C - y_R}{x_C - x_R} \cdot x_C$$

The legal region of this game is defined with the following predicate:

$$\texttt{inRegion}(x, y) \equiv x_C \neq 0 \lor y_C \geq 1 \lor y_C \leq 1.$$

Now the formal model to prove is

$$v_C = 2 \land v_R = 1 \land x_C = -2 \land y_C = 0 \land x_R = 2 \land y_R = 0 \vdash$$
$$\langle\{\texttt{ctrl}; t := 0; \texttt{plant}\}^*\rangle\big(\texttt{inRegion}(x_R, y_R) \to \texttt{inBall}(x_C, y_C, x_R, y_R, 2)\big)$$

Now we show the formal proof for Theorem 5.2.2.

Proof. Since $\langle\alpha^*\rangle P \leftrightarrow \big\langle\langle\alpha^*\rangle^*\big\rangle P$, we can rewrite it as

$$v_C = 2 \land v_R = 1 \land x_C = -2 \land y_C = 0 \land x_R = 2 \land y_R = 0 \vdash$$
$$\big\langle\{\langle\{\texttt{ctrl}; t := 0; \texttt{plant}\}^*\rangle\}^*\big\rangle\big(\texttt{inRegion}(x_R, y_R) \to \texttt{inBall}(x_C, y_C, x_R, y_R, 2)\big)$$

This change of modeling is required in the cop's strategy, as we will see later that the cop sometimes needs to move two rounds in order to meet the variant requirement.

For the outermost loop, we define loop variant $J(d) \equiv$

$$(\texttt{inRegion}(x_C, y_C) \land (\texttt{inRegion}(x_R, y_R) \to d - 2 \geq \texttt{dist}(x_C, y_C, x_R, y_R) \lor \texttt{dist}(x_C, y_C, x_R, y_R) \leq 2))$$

We define the following predicates:

$$\texttt{unsafe\_robber} \equiv \neg\texttt{inRegion}(x_R, y_R) \equiv (x_R = 0 \land y_R < 1 \land y_R > -1)$$
$$\texttt{euc\_cop} \equiv (x_C \cdot x_R \geq 0 \lor \mathit{cut}(x_C, y_C, x_R, y_R) \geq 1 \lor \mathit{cut}(x_C, y_C, x_R, y_R) \leq -1)$$
$$\texttt{top\_cop} \equiv (x_C \cdot x_R < 0 \land (\mathit{cut}(x_C, y_C, x_R, y_R) > -1 \land \mathit{cut}(x_C, y_C, x_R, y_R) < 1))$$

Now we can expand the variant to cases:

$$J(d) \equiv \texttt{inRegion}(x_C, y_C) \land \Big(\texttt{inRegion}(x_R, y_R) \to$$
$$\Big(\big(\texttt{euc\_cop} \to (d - 2 \geq \mathit{euc}(x_C, y_C, x_R, y_R) \lor \mathit{euc}(x_C, y_C, x_R, y_R) \leq 2)\big)\land$$
$$\big(\texttt{top\_cop} \to (d - 2 \geq \mathit{top}(x_C, y_C, x_R, y_R) \lor \mathit{top}(x_C, y_C, x_R, y_R) \leq 2))\big)\Big)\Big)$$

This intuitively describes that the time-optimal distance between the cop and the robber decreases by at least constant speed.

Now we can use `con` rule to separate into cases:

- `init`: Proven via arithmetic.
- `step`: Proven in Lemma 5.2.9.
- `post`: Proven via arithmetic. Note that when $d \leq 0$, $\texttt{inBall}(x_C, y_C, x_R, y_R, 2)$ is satisfied.

□

To show the `step` case, we first assume $d > 2$ by renaming the variable d to $d + 2$ to start with.

Lemma 5.2.8. For all $x_R, y_R, x_C, y_C \in \mathbb{R}$, $top(x_R, y_R, x_C, y_C) \geq euc(x_R, y_R, x_C, y_C)$.

This is obvious because the Euclidean distance is the shortest distance in 2D space.

The following lemma expands the formula that is needed to prove the `step` case in Theorem 5.2.2 by cases, either the robber is not in legal region, the cop can or cannot reach the robber in a straight line without passing the obstacle.

Lemma 5.2.9. $\forall d \geq dist(x_R, y_R, x_C, y_C)$ and $d > 2$, we have

$$\begin{aligned} & d > 2 \wedge \Big(\texttt{unsafe\_robber} \vee \big((\texttt{euc\_cop} \rightarrow d \geq euc(x_C, y_C, x_R, y_R)) \wedge \\ & \qquad\qquad\qquad (\texttt{top\_cop} \rightarrow d \geq top(x_C, y_C, x_R, y_R))\big)\Big) \\ & \vdash \langle \{\{\texttt{ctrl;}t := 0;\}\{\texttt{plant}\}\}^* \rangle \\ & \qquad \Big(\texttt{unsafe\_robber} \vee \big((\texttt{euc\_cop} \rightarrow d - 1 \geq euc(x_C, y_C, x_R, y_R)) \wedge \\ & \qquad\qquad\qquad (\texttt{top\_cop} \rightarrow d - 1 \geq top(x_C, y_C, x_R, y_R))\big)\Big) \end{aligned}$$

Proof. We will first break down the proof into 3 cases according to the premise (`unsafe_robber`, `euc_cop`, `top_cop`) using `orL`, note that in each case we also have the assumption that $d > 2$:

Case 1 `unsafe_robber`: The cop then selects $t = 0$ and thus reaches conclusion `unsafe_robber`.

Case 2 `euc_cop`: The assumption states that $d > 2 \wedge d \geq euc(x_C, y_C, x_R, y_R)$.

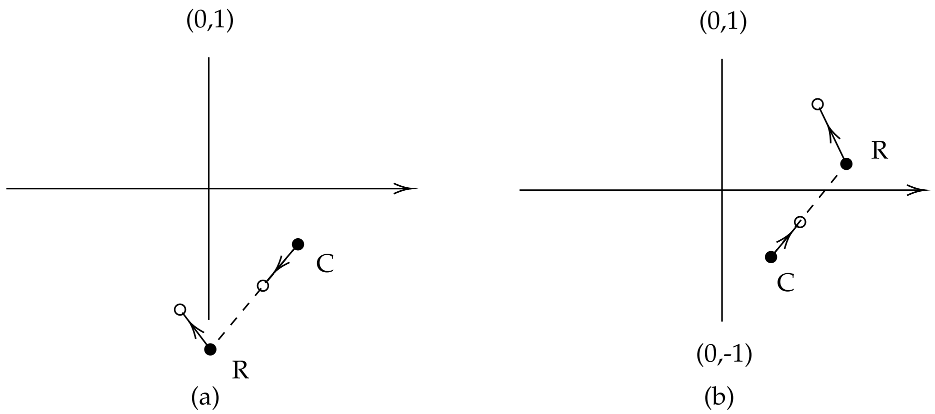

Figure 5.4: (a) illustrates an example of `euc_cop` condition, after one game round becomes `top_cop` (**Subcase 2.2**); (b) illustrates an example of `euc_cop` condition, after one game round becomes `euc_cop` (**Subcase 2.3**).

We will store the initial positions: $x_{C0} = x_C, y_{C0} = y_C, x_{R0} = x_R, y_{R0} = y_R$, and select (x_{C1}, y_{C1}) to move in the direction to (x_{R0}, y_{R0}).

$$x_{C1} = \frac{x_{R0} - x_{C0}}{\sqrt{(x_{R0} - x_{C0})^2 + (y_{R0} - y_{C0})^2}}, y_{C1} = \frac{y_{R0} - y_{C0}}{\sqrt{(x_{R0} - x_{C0})^2 + (y_{R0} - y_{C0})^2}}$$

The cop selects $t = 1$. We first show that the domain constraint holds:

$$\forall r(0 \le r \le 1) \rightarrow r \le 1 \land \neg(x_{C0} + v_C \cdot x_{C1} = 0 \land y_{C0} + v_C \cdot y_{C1} < 1 \land y_{C0} + v_C \cdot y_{C1} > -1)$$

This can be proven by breaking down into each subcase and by arithmetic. Then we show the post condition holds: This is equivalent to showing

$$\begin{aligned}
&d \ge 2 \land d \ge euc(x_C, y_C, x_R, y_R) \land euc(x_C, y_C, x_R, y_R) \ge 2 \rightarrow \\
&\langle \texttt{ctrl}; t := 0; \texttt{plant} \rangle \big(\texttt{unsafe_robber} \lor (\texttt{euc_cop} \rightarrow d - 1 \ge euc(x_C, y_C, x_R, y_R)) \lor \\
&(\texttt{top_cop} \rightarrow d - 1 \ge top(x_C, y_C, x_R, y_R))\big)
\end{aligned}$$

We can now discuss the proof by casing on the ending positions of **Case 2** for the players:

Subcase 2.1: Ends in `unsafe_robber`. The robber loses the game, the cop can choose $t = 1$ and exact post condition as needed.

Subcase 2.2: Ends in `euc_cop`. Intuition: The cop moves towards the robber, and the robber moves distance 1, as illustrated in Figure 5.4(a). The distance is still $euc(x_R, y_R, x_C, y_C)$. So

$$\begin{aligned}
euc(x_R, y_R, x_C, y_C) &\le euc(x_{R0}, y_{R0}, x_C, y_C) + euc(x_R, y_R, x_{R0}, y_{R0}) \\
&= euc(x_{R0}, y_{R0}, x_C, y_C) + 1 \\
&= euc(x_{R0}, y_{R0}, x_{C0}, y_{C0}) - 1
\end{aligned}$$

This can be proven by expanding the definitions and arithmetic.

Subcase 2.3: Ends in `top_cop`. Intuition: The cop moves towards the robber, and the robber moves distance 1, as illustrated in Figure 5.4(b). Without loss of generality, we suppose the distance now becomes $top(x_R, y_R, x_C, y_C) = euc(x_R, y_R, 0, 1) + euc(x_C, y_C, 0, 1)$. So

$$\begin{aligned} top(x_R, y_R, x_C, y_C) &= euc(x_R, y_R, 0, 1) + euc(x_C, y_C, 0, 1) \\ &\leq euc(x_{R0}, y_{R0}, x_C, y_C) + euc(x_R, y_R, x_{R0}, y_{R0}) \\ &= euc(x_{R0}, y_{R0}, x_{C0}, y_{C0}) - 1 \end{aligned}$$

This can be proven by expanding the definitions and arithmetic.

Now this concludes the proof for **Case 2**.

Case 3 `top_cop`: The assumption states that $d > 2 \wedge d \geq top(x_C, y_C, x_R, y_R)$.

As compared to **Case 2** where $d > 2 \wedge$ `euc_cop`, we may need to unroll the repetition operator twice using $\langle * \rangle$ in this case, i.e. change direction twice.

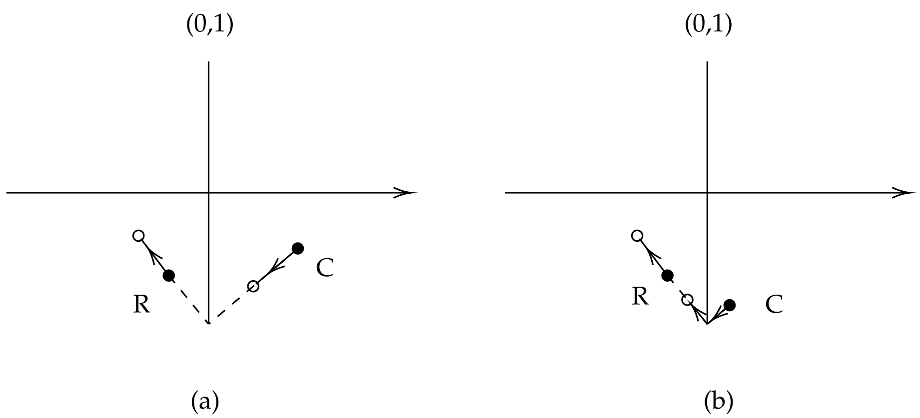

Figure 5.5: (a) illustrates an example of `top_cop` condition, after one game round becomes `top_cop` (**Subcase 3.2**); (b) illustrates an example of `top_cop` condition, after one game round becomes `euc_cop` (**Subcase 3.3**).

We will store the initial positions: $x_{C0} = x_C, y_{C0} = y_C, x_{R0} = x_R, y_{R0} = y_R$, and we first show that the domain constraint holds:

$$\forall r(0 \leq r \leq 1) \rightarrow r \leq 1 \wedge \neg(x_{C0} + v_C \cdot x_{C1} = 0 \wedge y_{C0} + v_C \cdot y_{C1} < 1 \wedge y_{C0} + v_C \cdot y_{C1} > -1)$$

This can be proven via expanding definitions and arithmetic. Then we show the post condition hold by sub-casing on the end positions for the players:

Subcase 3.1: Ends in `unsafe_robber`. Exact post condition as needed.

Subcase 3.2: Ends in state euc_cop**.** As illustrated in Figure 5.5(a), the cop will need to change direction twice. Therefore, we expand the repetition twice and get

$$d \geq 2 \wedge d \geq top(x_C, y_C, x_R, y_R) \wedge top(x_C, y_C, x_R, y_R) \geq 2 \vdash$$
$$\texttt{cop\_win} \vee \langle \texttt{ctrl}; t := 0; \texttt{plant} \rangle \texttt{cop\_win} \vee$$
$$\langle \texttt{ctrl}; t := 0; \texttt{plant} \rangle \langle \texttt{ctrl}; t := 0; \texttt{plant} \rangle \texttt{cop\_win}$$

Note that the only case possible is that $euc(x_{C0}, y_{C0}, 0, 1) \leq 2$ or $euc(x_{C0}, y_{C0}, 0, -1) \leq 2$. Now the cop will first move in the direction towards the closest endpoint of the line obstacle, and then move in the direction towards the new position of the robber:

- First select $t = \frac{\sqrt{(x_{C0})^2 + (1 - y_{C0})^2)}}{2}$ and

  $$x_{C1} = \frac{0 - x_{C0}}{\sqrt{(x_{C0})^2 + (1 - y_{C0})^2)}}, y_{C1} = \frac{1 - x_{C0}}{\sqrt{(x_{C0})^2 + (1 - y_{C0})^2)}}$$

  This selection ensures that after the move, the cop is at $x_C = 0, y_C = 1$. Then select $t' = 1 - t$, so that the cop moves the maximum amount towards the new position of the robber and

  $$x_{C1} = \frac{x_{R0}}{\sqrt{(x_{R0})^2 + (1 - y_{R0})^2)}}, y_{C1} = \frac{x_{R0} - 1}{\sqrt{(x_{R0})^2 + (1 - y_{R0})^2)}}$$

  So

  $$\begin{aligned} euc(x_R, y_R, x_C, y_C) &\leq euc(x_{R0}, y_{R0}, x_C, y_C) + euc(x_R, y_R, x_{R0}, y_{R0}) \\ &\leq top(x_{R0}, y_{R0}, x_{C0}, y_{C0}) - euc(x_{C0}, y_{C0}, 0, 1) - euc(0, 1, x_C, y_C) + 1 \\ &\leq top(x_{R0}, y_{R0}, x_{C0}, y_{C0}) - 1 \end{aligned}$$

- We can prove the similar case for $(0, -1)$.

Subcase 3.3: Ends in state top_cop**.** As illustrated in Figure 5.5(b), this is the case where $euc(x_{C0}, y_{C0}, 0, 1) \geq 2$ and $euc(x_{C0}, y_{C0}, 0, -1) \geq 2$. Now the cop selects the direction to move towards the closest endpoint of the line obstacle. So select $t = 1$ and

- $$x_{C1} = \frac{0 - x_{C0}}{\sqrt{(x_{C0})^2 + (1 - y_{C0})^2)}}, y_{C1} = \frac{1 - x_{C0}}{\sqrt{(x_{C0})^2 + (1 - y_{C0})^2)}}$$

  So

  $$\begin{aligned} top(x_R, y_R, x_C, y_C) &= top(x_{R0}, y_{R0}, x_{C0}, y_{C0}) - euc(x_R, y_R, x_{R0}, y_{R0}) \\ &= top(x_{R0}, y_{R0}, x_{C0}, y_{C0}) - 1 \end{aligned}$$

- We can prove the similar case for $(0, -1)$.

This concludes the proof of **Case 3**. □

5.3 Equal Speed Cop

Now we consider the case when the cop's speed is the same as the robber, slightly restricting the cop's ability to move comparing to the previous section. We consider only the case where the players have positive velocity and ignore the case where the velocity is equal to 0, since then no player moves in the game and is always robber-win.

5.3.1 Unbounded Plane (with Obstacle)

We first consider the unbounded plane. The intuitive conjecture is that the robber wins the game, since the plane is continuous and the cop has no way to keep the robber inside a closed region. Indeed, the robber's strategy follows from a similar approach as the cycle graph case in Section 4. An example game region is illustrated in Figure 5.6.

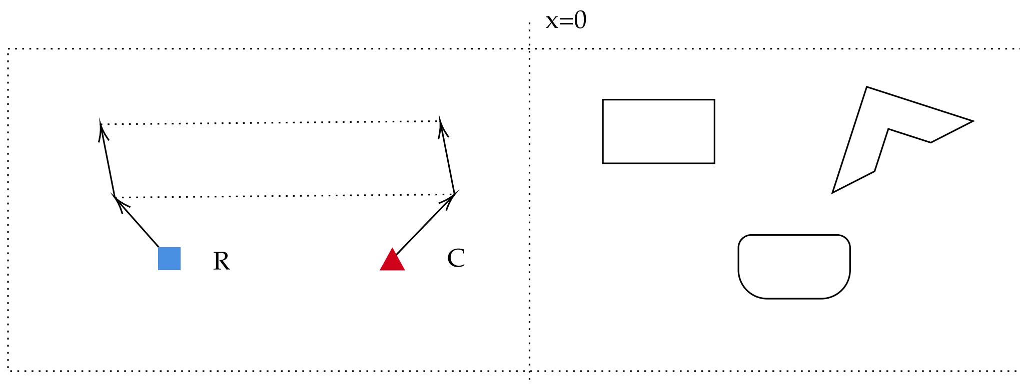

Figure 5.6: An example graph of the unbounded plane with obstacles as illustrated in the picture. The blue rectangle denotes the robber; the red triangle denotes the cop. The robber's strategy is depicted as above, that it is always moving away from the obstacle in the x-axis.

Theorem 5.3.1. If the legal game region is the Euclidean Plane with finitely many obstacles, and the speed $v_C = v_R > 0$, then the robber has a winning strategy to win the game.

The formal model to prove is

$$x_R \leq 0 \land x_C \leq 0 \land x_C - x_R \geq 2 \vdash [\{\texttt{ctrl}; t := 0; \texttt{plant}\}^*](\neg \texttt{cop\_win})$$

Proof. We will define $\texttt{inRegion}(x, y) = x \leq 0$. Note that this is a safe definition that generalizes all cases of different obstacles, by selecting the horizontal axis that all obstacles are at the right of this axis as $x = 0$. We can also define the position of the robber to be $x_R = \min(x_C - 2, -2)$, $y_R = x_R$. Note that for the cop to be in the safe region, $x_C \leq 0$ in our model, but this suffices to define the strategy for generalized cases. Now we consider the invariant

$$J \equiv abs(x_C - x_R) \geq 2.$$

- $\texttt{init}$: To show $x_R = \min(x_C - 2, -2) \wedge \texttt{inRegion}(x_R, y_R) \vdash J$. This is true by simple arithmetic.
- $\texttt{step}$: To show $J \vdash [\texttt{step}]J$: we formalize the following strategy for the robber: $(x_{R1}, y_{R1}) = (-1, 0)$. Then given $x_R \leq 0$, after positive $t \leq 1$ duration, we must have maintained $x_R \leq 0$. Since x_R is decreasing, J is maintained after $\texttt{step}$.
- $\texttt{post}$: Since $abs(x_C - x_R) \geq 2$, obviously $(x_C - x_R)^2 + (y_C - y_R)^2 \geq 1$.

This proof is also done in KeYmaera X. □

5.3.2 Bounded Region (No Obstacle)

Now it is natural to consider bounded regions. We first start with simple game regions with no obstacles. We consider an arbitrary convex, closed and bounded region that contains the origin $(0, 0)$ in the 2D-plane. Note that this requirement is without loss of generality, since we can redefine the coordinate system so that it contains the origin. An example game region is illustrated in Figure 5.7.

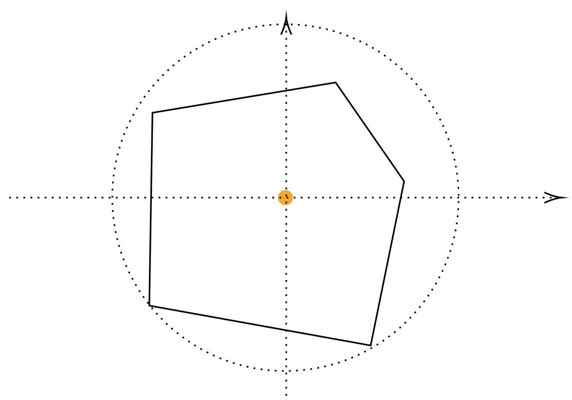

Figure 5.7: An example of a convex, closed and bounded region (the polygon region). The dotted circle represents the maximum radius required to bound this region.

Definition 5.3.2. A bounded convex region satisfies the following properties:

There exists $R > 1$ such that

$$\texttt{isBounded} \equiv \forall x, \forall y, (\texttt{inRegion}(x, y) \rightarrow \texttt{NormToOrigin}(x, y) \leq R^2))$$

and

$$\begin{aligned}\texttt{isConvex} \equiv & \forall (x_1, y_1), \forall (x_2, y_2), \big(\texttt{inRegion}(x_1, y_1) \wedge \texttt{inRegion}(x_2, y_2) \rightarrow \\ & \forall t, (0 \leq t \leq 1 \rightarrow \texttt{inRegion}(tx_1 + (1-t)x_2, ty_1 + (1-t)y_2)\big)\end{aligned}$$

where we also define the following helper functions:

$$\texttt{inRegion}(x, y) \equiv x^2 + y^2 \leq R^2.$$
$$\texttt{NormToOrigin}(x, y) \equiv x^2 + y^2.$$

Definition 5.3.3. Let d_C be the distance from the cop to the origin, d_R be the distance from the robber to the origin. i.e.

$$d_C = \sqrt{x_C^2 + y_C^2}, d_R = \sqrt{x_R^2 + y_R^2}.$$

Theorem 5.3.4. If the game region is an arbitrary convex, bounded region, and the speed $v_C = v_R > 0$, then the cop has a winning strategy, as formalized below:

$$\texttt{isBounded} \land \texttt{isConvex} \land \texttt{inRegion}(x_R, y_R) \land x_C = y_C = 0 \land v_C = v_R = 1$$
$$\vdash \langle\{\{\texttt{ctrl;}t := 0; \texttt{plant;}\}^*\}\rangle \texttt{cop\_win}$$

The main ingredient for proving Theorem 5.3.4 is the variant $J(d)$ below:

$$J(d) \equiv \texttt{isBounded} \land \texttt{isConvex} \land \big(\texttt{inRegion}(x_R, y_R) \to (\texttt{inRegion}(x_C, y_C)$$
$$\land x_R \cdot y_C = x_C \cdot y_R \land d_C^2 \leq d_R^2 \land (x_R - x_C)^2 + (y_R - y_C)^2 \leq d_R^2 \land d \geq R^2 - d_C^2 - 1)\big).$$

This variant argues that the cop and the robber are always on the same radius and the cop is always closer to the origin than the robber, and the value d_C^2 is increasing at at least constant speed.

Lemma 5.3.5. The variant d_C^2 increase by at least 1 after one round. We prove this using the following model:

$$d > 0, J(d) \vdash \langle \texttt{ctrl;}t := 0; \texttt{plant}\rangle J(d-1)$$

Proof. The geometric idea behind the proof is illustrated in Figure 5.8. Both balls denote the unit ball available for the player to move. So if the robber moves from R to R_0, the cop moves from C to C_0, which maintains the invariant part and is the furthest possible point to reach R_0.

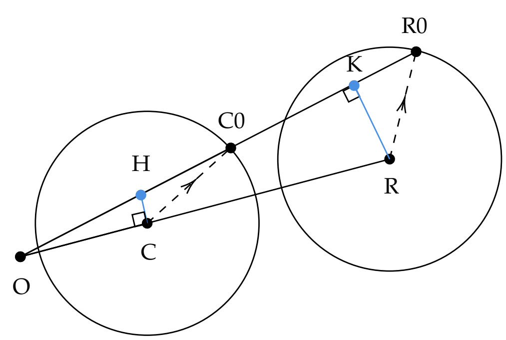

Figure 5.8: This figure shows the movement of C and R in a round. O is the origin. When the robber moves from R to R_0, the cop moves from C to C_0.

We now formalize this idea. The circle centred at R denotes the range of R's movement and the circle centred at C denotes the range of C's movement. The line connecting O, C_0, R_0 is the radius that guarantees $x_{C0} \cdot y_C = x_C \cdot y_{C0}$. For shorthand we say the unit vector $\vec{C_1} = \overrightarrow{CC_0}$, $\vec{R_1} = \overrightarrow{RR_0}$. We can see from the picture that there is guaranteed to have ≥ 1 solution for this $C0$. This can be solved using the following equation:

$$(x - x_C)^2 + (kx - y_C)^2 = 1.$$

with the definition of the following parameters:

$$k = \frac{y_{R0}}{x_{R0}}, A = 1 + k^2, B = -2x_C - 2ky_C, C = x_C^2 + y_C^2 - 1, D = B^2 - 4AC.$$

So there are two solutions for $C0$:

$$x_{C0} = \frac{-B \pm \sqrt{D}}{2A}, y_{C0} = k\frac{-B \pm \sqrt{D}}{2A}$$

Let H be a point on $OR0$ such that CH is perpendicular to OC. K to be the vertical point on $OR0$ such that RK is perpendicular to $OR0$. Geometry shows that $CH/RK = OH/OR = OC/OK$. Therefore we can write

$$(x_H - x_C) \cdot x_C + (kx_H - y_C) \cdot y_C = 0,$$

and we can solve $x_H = \frac{x_C^2 + y_C^2}{x_C + k \cdot y_C}$. By geometry, $RR_0 = 1$ and $RK \leq RR_0$, $RK \leq 1$. By assumption $CR \geq 2$, $CH < RK \leq 1 = CC_0$.

By geometry, $(x_{C1}, y_{C1}) \cdot (x_C, y_C) \geq 0$. This implies $|OC_0|^2 \geq |OC|^2 + |CC0|^2 = |OC|^2 + 1$. □

We redefine the coordinate system so that the players always start on the positive $x-$axis at the start of the round.

Lemma 5.3.6. When $d_R - d_C \geq 2$, the formula $d > 0, J(d) \vdash \langle \texttt{ctrl}; t := 0; \texttt{plant} \rangle J(d-1)$ is true.

Proof. After applying dGL axioms $\langle ; \rangle$, $\langle := \rangle$, $\langle ? \rangle$, and solving ODE, this is equivalent to showing

$$\begin{aligned} x_R > x_C \geq 0, y_R = y_C = 0, d_R - d_C \geq 2, y_{R1} > 0, x_{R1}^2 + y_{R1}^2 = 1, \\ x_{R0} = x_R + x_{R1}, y_{R0} = y_R + y_{R1}, d \geq d_C^2, x_C \cdot y_R = x_R \cdot y_C, d_C^2 \leq d_R^2 \\ \vdash \exists x_{C1}, y_{C1}(x_{C1}^2 + y_{C1}^2 = 1 \wedge x_{C0} = x_{C1} + x_C \wedge y_{C0} = y_{C1} + y_C \\ \wedge x_{C0} \cdot y_{R0} = x_{R0} \cdot y_{C0} \wedge d_{C0}^2 \leq d_{R0}^2 \wedge d \geq d_{C0}^2 + 1). \end{aligned}$$

Notice that since $d_R - d_C \geq 2$, $x_R \geq 2 \neq 0$. Now we will select $x_{C1} = \frac{-B+\sqrt{D}}{2A} - x_C$, $y_{C1} = kx_{C1} - y_C$, which is the solution that satisfies the variant by Lemma 5.3.5. Therefore, we need to break down the conclusion to prove into parts using `andR`.

- To show $d \geq d_{C0}^2 + 1$, we use Lemma 5.3.5.
- To show $x_{C0} \cdot y_{R0} = x_{R0} \cdot y_{C0}$, the conclusion follows by simple arithmetic.
- To show $d_C^2 \leq d_R^2$: Since the players are only allowed to move within a unit ball of the original position, $d_R^2 \geq (x_R - 1)^2$, $d_C^2 \leq (x_C + 1)^2$. So $d_C^2 \geq d_R^2$.
- To show $(x_R - x_C)^2 + (y_R - y_C)^2 \leq d_R^2$: We can expand this equation and plug in the definition of d_R, which is equivalent to showing $x_C + x_{C1} > 0, x_R + x_{R1} > 0$. Since $x_{C1} > 0$ by selection, and $x_R \geq 2$, $x_{R1} \geq -1$ by assumptions and game rules, this is proven.

□

Now we are able to show the complete model.

Proof. We use `con` rule and prove the following cases:

- `init`: To show

  $$\texttt{isBounded} \wedge \texttt{isConvex} \wedge \texttt{inRegion}(x_R, y_R) \wedge x_C = y_C = 0 \wedge v_C = v_R = 1 \vdash \exists d. J(d),$$

  the invariant part follows from propositional logic; if we plug in $x_C = y_C = 0$ and select $d = R^2 - 1$, the variants also follow by arithmetic.
- `step`: To show $d > 0, J(d) \vdash \langle \texttt{step} \rangle J(d-1)$, we consider the following cases:
  - If $d_R - d_C > 2$, if $\texttt{inRegion}(x_R, y_R)$, then $\texttt{inRegion}(x_C, y_C)$ by convexity. Then we apply Lemma 5.3.6.
  - If $d_R - d_C \leq 2$, cop selects $t = 0$ and the robber cannot pass the test.
- `post`: To show $d \leq 0, J(d) \vdash \texttt{inRegion}(x_R, y_R) \rightarrow \texttt{inBall}(x_C, y_C, x_R, y_R, r)$, note that the conclusion holds if $\texttt{inRegion}(x_R, y_R)$ is false. When $\texttt{inRegion}(x_R, y_R)$ is true, we have $R^2 - d_C^2 \leq 1$. Since $d_R^2 \leq R^2$, the conclusion holds.

This proof is also done in KeYmaera X. □

5.3.3 Bounded Region (with 1 Obstacle)

Now we consider the case of bounded region as in the previous section, but add obstacles in the area. We will not restrict the settings, but instead discuss sufficient and complete conditions for cop-win and robber-win cases.

Before diving into the problem, we start by stating a lemma that follows from the previous section. The intuition of this lemma is for the cop to force the robber into an obstacle-free region and eventually catch the robber by forcing the robber into a shrinking region over time. We can use this lemma to regulate the players' strategies that follows in a bounded region with 1 obstacle, as illustrated in Figure 5.9.

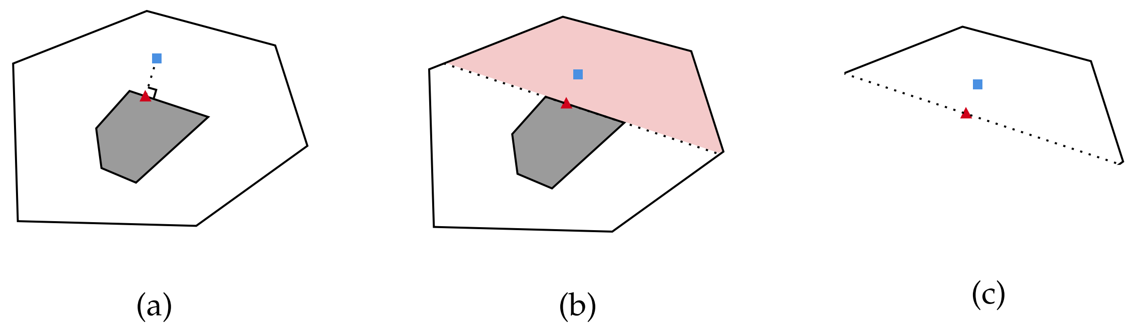

Figure 5.9: This describes how the lemma connects to a 1 obstacle region. The blue rectangle is the robber and the red triangle is the cop. (a) describes the condition inside the original graph; (b) cuts the graph into two parts; (c) describes the final cop-winning condition we need to apply Lemma 5.3.8.

Definition 5.3.7. We define the **projection** p from region G to H, where $H \subseteq G$, such that for all $(x_h, y_h) \in H$, if $(x_g, y_g) \in G$, $dist(x_g, y_g, x_h, y_h) \geq dist(x_g, y_g, p(x_g, y_g))$. Note that this is only a well-defined function when there's only one unique such $p(x_g, y_g)$ for all $(x_g, y_g) \in G$.

Lemma 5.3.8. Let R be a curve in the plane, H be the line segment connecting the endpoints of the curve. If the region enclosed by R and H is convex, we let this region be G. If there exists a projection $p : G \to H$, the robber $(x_R, y_R) \in G$ and the cop satisfies $(x_C, y_C) = p(x_R, y_R)$, then the cop can catch the robber in finite time.

The formal model to show is

$$\texttt{isConvex} \wedge \texttt{isBounded} \wedge x_C = x_R = 0 \wedge y_C \leq y_R \vdash \langle\{\texttt{ctrl}; t := 0; \texttt{plant}\}^*\rangle \texttt{cop\_win}$$

Proof. By assumption, this region satisfies `isConvex` which is defined the same as in the previous section, and `isBounded`, which is refined as

$$\texttt{isBounded} \equiv \forall x, \Big(\exists R, \big(\forall y, (\texttt{inRegion}(x, y) \wedge y \geq h \rightarrow x^2 + y^2 \leq R)\big)\Big),$$

where h satisfies the following:

We will define the coordinate system such that H lies on the horizontal line $y = h$, and H is contained in the line segment from $(-b, h)$ to (b, h) for some positive b. The curve R is above $y = h$. The initial position of the cop is at $(0, h)$ and $x_C = x_R = 0$. Let (x_{C0}, y_{C0}) and (x_{R0}, y_{R0}) be the coordinates for the new position of the cop and the robber correspondingly, b is defined as the minimum radius that contains the edge H. By rewriting these conditions into formulas, h is defined through the following equations:

$$\begin{aligned} x_{C0}^2 + y_{C0}^2 &\geq h^2 + b^2 \\ x_{C0} \cdot y_{R0} &= x_{R0} \cdot y_{C0} \\ x_{C0} &= x_{C1} \\ y_{C0} &= h + y_{C1} \\ x_{C1}^2 + y_{C1}^2 &\leq 1 \end{aligned}$$

Notice that this h satisfies the requirement that starting from the initial position $C = (0, h)$ and $R = (0, y_R)$, the new position of the cop using the strategy in Lemma 5.3.8, satisfies $x_{C0}^2 + y_{C0}^2 \geq h^2 + b^2$, as illustrated in Figure 5.10.

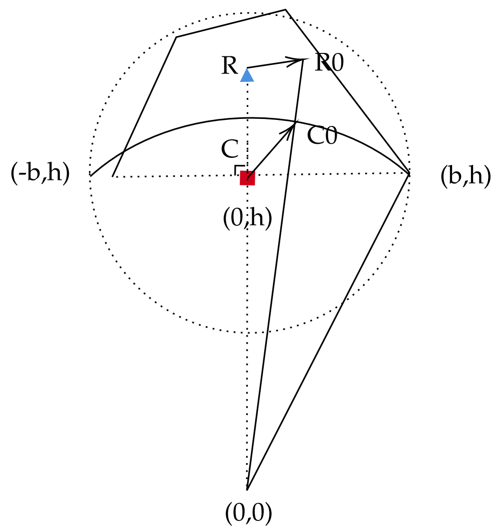

Figure 5.10: Illustration of convex region G in the coordinate system and the cop's strategy. The cop moves from C to C_0 if the robber moves from R to R_0.

To find the minimum h required, (5.1) implies that $h_{min} = \frac{b^2-1}{2y_{C1}}$. Therefore we consider the case that minimum y_{C1} is reached, when RR_0 is perpendicular OR_0. So we have the additional equation:

$$\frac{1}{y_R^2} = \frac{x_{C1}^2}{x_{C1}^2 + (h + y_{C1})^2}$$

Combining the equations above, we obtain

$$y_R^2 \cdot y_{C1}^4 + (3 - 2b^2 - y_R^2) \cdot y_{C1}^2 + \frac{1}{4}(b^2 - 1)^2 = 0$$

Solving this equality gives us $y_{C1} \geq \frac{\sqrt{-3+2b^2+y_R^2}}{\sqrt{2}y_R}$.

Therefore we select $h = \frac{b^2-1}{2y_{C1}^*}$ where $y_{C1}^* = \frac{\sqrt{-3+2b^2+y_R^2}}{\sqrt{2}y_R}$ which satisfies the previous equations.

Under this coordinate system, projection p is defined as $p(x_g, y_g) = (\tilde{x_g}, h)$ where $\tilde{x_g}$ is x_g if $(\tilde{x_g}, h) \in G$, and the closest corner on H otherwise.

Let the variant be

$$J(d) \equiv y_C \cdot x_R = x_C \cdot y_R \wedge x_R^2 \geq x_C^2 \wedge x_R \cdot x_C \geq 0 \wedge d \geq r^2 - (x_C^2 + y_C^2) \wedge x_C^2 + y_C^2 \geq b^2 + h^2$$

This is the variant as defined in Lemma 5.3.6 with the addition of the last condition $x_C^2 + y_C^2 \geq b^2 + h^2$, to guarantee that the robber never escapes the unbounded region G.

First consider the case that $dist(x_C, y_C, x_R, y_R) \leq 2$, then the final condition is already met.

Then it suffices to assume $dist(x_C, y_C, x_R, y_R) > 2$: We will first apply $\langle * \rangle$ rule, i.e.

$$dist(x_C, y_C, x_R, y_R) > 2 \land x_C = x_R = 0 \land y_C \leq y_R \vdash \langle \texttt{ctrl;}t := 0; \texttt{plant;}\rangle \langle \{\texttt{ctrl;}t := 0; \texttt{plant;}\}^* \rangle \big(\texttt{inRegion}(x_R, y_R) \rightarrow \texttt{inBall}(x_C, y_C, x_R, y_R, 2)\big)$$

and show

$$dist(x_C, y_C, x_R, y_R) > 2 \land x_C = x_R = 0 \land y_C \leq y_R \vdash \langle \texttt{ctrl;}t := 0; \texttt{plant;}\rangle (\exists d. J(d)).$$

Since $dist(x_C, y_C, x_R, y_R) > 2$, then we can select $t = 1$. We select (x_{C1}, y_{C1}) using the strategy as in Figure 5.10: select (x_{C1}, y_{C1}) that satisfies $x_C \cdot y_R = x_R \cdot y_C$ and has the closest distance to (x_R, y_R). By selection of h, $x_C^2 + y_C^2 \geq b^2 + h^2$ hold.

This guarantees that $x_R^2 + y_R^2 \geq b^2 + h^2$, i.e. R is always inside the guarded region of C.

Now use the strategy as in Lemma 5.3.6, we can show that

$$\texttt{isConvex} \land \texttt{isBounded} \land \exists d. J(d) \vdash \langle \texttt{step}^* \rangle \big(\texttt{inRegion}(x_R, y_R) \rightarrow \texttt{inBall}(x_C, y_C, x_R, y_R, 2)\big),$$

so that the rest of the conclusion is proven. □

Now to answer the problem we raised at the beginning of this subsection, it suffices to find cases such that the cop reaches the projection of the robber on the boundary of the obstacle. Figure 5.9 shows a mapping from this situation to applying Lemma 5.3.8.

We first consider a counterexample where the cop fails to catch the robber:

Conjecture 5.3.9. If the game region is convex, bounded and contains a single regular polygon as an obstacle, the game is robber-win.

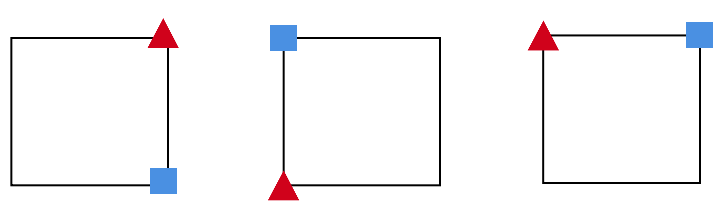

Figure 5.11: In the case where the obstacle is a square, the red triangle represents the cop and the blue rectangle represents the robber. Then the robber can move one edge away from the cop in each round.

The intuition of this conjecture is shown via Figure 5.11. It is clear that if the cop starts from a corner of the obstacle, the robber can select a position such that the distance between the cop and the robber never decreases. However, if the cop selects a random position in the game space, then it's unclear whether the robber has a fixed strategy to enter this case.

Now we consider the problem from another angle.

Conjecture 5.3.10. If the game region is convex, bounded and contains a singular irregular polygon as an obstacle, the game is cop-win.

Now we claim that if both the cop and the robber greedily chase around the obstacle, the cop can eventually catch the robber. This happens because the cop's strategy guarantees to stay on the boundary, while the robber doesn't. Therefore, the total distance the cop runs is strictly positive and less than the robber. However, a formal proof requires defining $J(d)$ to be a strictly decreasing variant such that over a constant time $t > 0$, $J(d)$ is guaranteed to decrease by $\geq c$ where c is a positive constant. For some cases, this constant can be found through brute-force calculations, but it varies for different cases and it's hard to find a lower bound.

Note that this modification allows both the robber and the cop to move freely within a timestep of 1. The mathematical reasoning stay the same, as we can still view each move of the agents to be inside the ball of size 1. This essentially switches the concurrent behavior of the system to turn-based system. The reason for this change is that in the concurrent setting, the robber is allowed to change direction in real time as the cop changes direction, and in the case of obstacles, the robber has incentive to do so, resulting in the case that the robber always wins.

Chapter 6

Conclusion

In this work, we discussed different types of pursuit-evasion games, and used dGL to create models for game rules, and presented formal verification approach for winning strategies. In both discrete and continuous dynamics, the essence is to find correct invariants and variants required to show the proofs. In Chapter 4, we considered "Cops and Robbers" game played on discrete graphs. We investigated graph families starting from cycles, trees, to a general proof of dismantlable graphs, which is the sufficient and complete condition for 1-cop games. We also extended the proofs from concrete examples to abstract properties by using group properties of natural numbers. In addition, we also considered a proof on grids for two-cop games. This modification requires a change in modeling, but cannot extend to different cop numbers. In Chapter 5, we considered "Lion and Man" game played on continuous planes. We investigated several planes casing on the boundary condition, the cop's speed and the obstacle condition. We also discussed the differences among different selections of the game dynamics, and selected the final one that balances between the difficulty of performing formal proofs and whether it simulates reality well.

We compared the difference between dGL proofs and traditional mathematical proofs, and conclude that the advantage of dGL, as shown in continuous games, represents simultaneous moves easily and the use of ODE and domain constraint well explains the game dynamics while keeping the agents in the safe region. This significantly differs from traditional game settings that discusses turn-based games. We can see that turn-based games favors the cop, since similar cop-winning strategies cannot be adapted to the simultaneous version, while simultaneous version better simulates real-life scenario. However, challenge arises when it comes to defining complicated math definitions properly, and makes the abstraction of proofs difficult using the current set of tools. For future work, it would be valuable to combine other types of tools with dGL to perform formal verification, including program simulations and mathematical ways to potentially simplify the game modeling.

Appendix A

Differential Game Logic Proof Rules

Propositional Logic Proof Rules

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \wedge Q, \Delta} \wedge R \qquad \frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \vee Q, \Delta} \vee R \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma \vdash \neg P, \Delta} \neg R$$

$$\frac{\Gamma, P, Q \vdash \Delta}{\Gamma, P \wedge Q \vdash \Delta} \wedge L \qquad \frac{\Gamma, P \vdash \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \vee Q \vdash \Delta} \vee L \qquad \frac{}{\Gamma, \neg P \vdash \Delta} \neg L$$

$$\frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \rightarrow Q, \Delta} \rightarrow R \qquad \frac{}{\Gamma, P \vdash P, \Delta} id \qquad \frac{}{\Gamma, false \vdash \Delta} FL$$

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \rightarrow Q \vdash \Delta} \rightarrow L \qquad \frac{\Gamma \vdash P \quad P \vdash \Delta}{\Gamma \vdash \Delta} cut \qquad \frac{}{\Gamma \vdash true, \Delta} TR$$

dGL Axioms

$[\alpha]P \leftrightarrow \neg\langle\alpha\rangle(\neg P)$

$\langle{:=}\rangle$: $\langle x := \theta\rangle P(x) \leftrightarrow P(\theta)$ — $[{:=}]$: $[x := \theta]P(x) \leftrightarrow P(\theta)$

$\langle\cup\rangle$: $\langle\alpha \cup \beta\rangle P \leftrightarrow \langle\alpha\rangle P \wedge \langle\beta\rangle P$ — $[\cup]$: $[\alpha \cup \beta]P \leftrightarrow [\alpha]P \vee [\beta]P$

$\langle ; \rangle$: $\langle\alpha; \beta\rangle P \leftrightarrow \langle\alpha\rangle\langle\beta\rangle P$ — $[;]$: $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$

$\langle * \rangle$: $\langle\alpha^*\rangle P \leftrightarrow P \vee \langle\alpha\rangle\langle\alpha^*\rangle P$ — $[*]$: $[\alpha^*]P \leftrightarrow P \wedge [\alpha][\alpha^*]P$

$\langle ' \rangle$: $\langle x' = f(x)\rangle p(x) \leftrightarrow \exists t \geq 0 \langle x := y(t)\rangle p(x)$

$[']$: $[x' = f(x)]p(x) \leftrightarrow \forall t \geq 0 [x := y(t)]p(x)$

$\langle ? \rangle$: $\langle ?Q\rangle P \leftrightarrow Q \wedge P$ — $[?]$: $[?Q]P \leftrightarrow Q \implies P$

$\langle d \rangle$: $\langle\alpha^d\rangle P \leftrightarrow \neg\langle\alpha\rangle\neg P$ — $[d]$: $[\alpha^d]P \leftrightarrow \neg[\alpha]\neg P$

$\langle := * \rangle$: $\langle x := *\rangle P \leftrightarrow \exists x, P$ — $[:= *]$: $[x := *]P \leftrightarrow \forall x, P$

dGL Proof Rules

$$\texttt{loop}: \frac{\Gamma \vdash J \quad J \vdash [\alpha]J \quad J \vdash P}{\Gamma \vdash [\alpha^*]P, \Delta}$$

$$\texttt{con}: \frac{\Gamma \vdash \exists d J(d) \quad d > 0, J(d) \vdash \langle\alpha\rangle J(d-1) \quad d \leq 0, J(d) \vdash P}{\Gamma \vdash \langle\alpha^*\rangle P, \Delta}$$

$$\texttt{discreteGhost}: \frac{\Gamma \vdash [x := e]P, \Delta}{\Gamma \vdash P, \Delta}.$$

Appendix B

KeYmaera X Tactics

A list of proofs is listed at
`https://github.com/ReedyHarbour/Pursuit-Evasion-Game-Tactics.git`

References

[1] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.

[2] Deepak Bhadauria, Kyle Klein, Volkan Isler, and Subhash Suri. Capturing an evader in polygonal environments with obstacles: The full visibility case. *The International Journal of Robotics Research*, 31:1176–1189, 09 2012.

[3] Anthony Bonato and Richard J. Nowakowski. The game of cops and robbers on graphs. 2011.

[4] Peter Frankl. Cops and robbers in graphs with large girth and cayley graphs. *Discret. Appl. Math.*, 17:301–305, 1987.

[5] Nathan Fulton, Stefan Mitsch, Brandon Bohrer, and André Platzer. Bellerophon: Tactical theorem proving for hybrid systems. In Mauricio Ayala-Rincón and César A. Muñoz, editors, *ITP*, volume 10499 of *LNCS*, pages 207–224. Springer, 2017.

[6] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In Amy P. Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 527–538. Springer, 2015.

[7] Volkan Isler and Nikhil Karnad. The role of information in the cop-robber game. *Theoretical Computer Science*, 399:179–190, 06 2008.

[8] Nikhil Karnad and Volkan Isler. Lion and man game in the presence of a circular obstacle. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5045–5050. IEEE, 2009.

[9] William B. Kinnersley. Cops and robbers is exptime-complete. *Journal of Combinatorial Theory, Series B*, 111:201–220, 2015.

[10] Stefan Mitsch. Implicit and explicit proof management in keymaera X. In José Proença and Andrei Paskevich, editors, *Proceedings of the 6th Workshop on Formal Integrated Development Environment, F-IDE@NFM 2021, Held online, 24-25th May 2021*, volume 338 of *EPTCS*, pages 53–67, 2021.

[11] Stefan Mitsch, Andrew Sogokon, Yong Kiam Tan, Xiangyu Jin, Bohua Zhan, Shuling Wang, and Naijun Zhan. ARCH-COMP19 category report: Hybrid systems theorem proving. EasyChair.

[12] Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2):235–239, 1983.

[13] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008.

[14] André Platzer. Differential game logic. *ACM Trans. Comput. Log.*, 17(1):1:1–1:51, 2015.

[15] Frédéric Simard, Josée Desharnais, and François Laviolette. General cops and robbers games with randomness. *CoRR*, abs/2004.11503, 2020.

[16] Sampath Kannan Volkan Isler and Sanjeev Khanna. Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, 21(5):875–884, 2005.