



University of Padova

DEPARTMENT OF MATHEMATICS "TULLIO LEVI-CIVITA"

MASTER DEGREE IN COMPUTER SCIENCE

Some decidability questions in abstract program semantics



Supervisor

Prof. Paolo Baldan

Co. Supervisor

Prof. Francesco Ranzato

Candidate

Luca Zaninotto

ACADEMIC YEAR 2023-2024

Abstract

This thesis explores program verification through abstract interpretation in the context of computability theory. Abstract Interpretation is a program analysis technique, based on approximating the semantics of programs over so-called *abstract domains*, usually represented as complete lattices, whose elements represent program properties. These approximations rely on some abstract operators, which usually include fixpoint iterations. Traditionally, to ensure convergence of such iterations, and therefore the termination of the analyzer, the literature relied on two important operators: the *widening* and the *narrowing* operators, first defined in [CC77]: the first one to compute an upper bound on some chain in the complete lattice, and the second one to recover some additional information from the program and refine the upper bound provided by the widening. This approach brings however a loss of precision: the inferred invariant is not the one provided by the abstract semantics, but an overapproximation of it. The thesis argues that for non relational abstract domains better results can be obtained. In particular, for the *interval* domain, where each variable is associated with an interval of integers the abstract semantics is computed in an exact way. Instead, for the *non-relational collecting* domain where each variable is mapped to a (possibly non-convex) subset of \mathbb{Z} over which it can vary, we just show that it is possible to decide the termination of the abstract interpreter and, when this is the case, exactly compute the abstract semantics.

Contents

| | |
|---|-----------|
| Introduction | 1 |
| 1 Background | 5 |
| 1.1 Recursion theory | 5 |
| 1.2 Order theory | 6 |
| 1.3 Abstract Interpretation | 7 |
| 1.3.1 General concepts | 7 |
| 1.3.2 Fixpoint approximations | 11 |
| 2 Framework | 13 |
| 2.1 The Imp language | 13 |
| 2.2 Semantics | 13 |
| 2.2.1 Syntactic sugar | 17 |
| 2.2.2 Small step semantics | 17 |
| 2.3 Transition system | 18 |
| 2.4 Functions in Imp | 22 |
| 2.5 Deciding invariant finiteness | 25 |
| 3 Abstract domains | 31 |
| 3.1 Abstract inductive semantics | 31 |
| 3.2 Non relational collecting | 35 |
| 3.2.1 Properties | 37 |
| 3.3 Interval domain | 39 |
| 3.3.1 Variable-wise lifting | 40 |
| 3.3.2 Properties | 42 |
| 4 Program bounds and analysis termination | 43 |
| 4.1 Program bounds | 44 |
| 4.2 Bounding interval analysis | 45 |
| 4.3 Computing interval semantics | 51 |
| 4.4 Bounded non-relational collecting semantics | 57 |
| 4.5 Computing non-relational collecting semantics | 59 |
| 5 Conclusions | 67 |
| A Additional proofs | 69 |
| A.1 Lemma 4.7 proof | 69 |
| A.2 Lemma 4.18 proof | 74 |
| A.3 Lemma 4.20 proof | 79 |

Introduction

Because of its widespread adoption software has become a crucial aspect of everyone’s life for all sorts of tasks, from the more mundane ones – like sending text messages or view online content – to the most crucial ones. Banking, aviation, space industry, car controls are only a small example of important everyday tasks that software runs in the modern era. Such tasks demand requirements of safety and reliability which are difficult to pair with the growing complexity and size of contemporary software. Errors can be expensive both in monetary and in human lives terms, hence preventing them becomes more and more valuable as well as detecting them early.

Notable examples of such bugs are Meltdown and Spectre [Lip+18; Koc+19]. Those vulnerabilities exploited an hardware related bug in floating-point division to access data outside the bounds imposed to a program by the operating system, resulting in the theft of arbitrary data, meaning a malicious actor could access – for example – passwords stored locally or data of other customers in a cloud environment. Another notable example is the first internet worm, which allowed the deployer to run arbitrary code on a significant portion of the computers on the internet at the time [Spa89; See89; Orm03; Eis+89]. The last example is set on 4 June 1996, when the Ariane 501 satellite launch failed catastrophically 40 seconds after initiation of the flight sequence, incurring a direct cost of approximately 370 million US dollars [Dow97]. To assess the causes of the incident, the automated analysis of the Ariane code [Lac+98] was done using a static analyzer based on Abstract Interpretation [Le 97].

Software verification is therefore a crucial task, which cannot be accomplished using testing practices alone: testing in fact can be used to show the *presence* of bugs (if a test fails the bug occur), but they do not offer any *mathematical guarantee* of their absence. The latter can be obtained through *formal methods*, i.e., by mathematical proving the correctness of a program with respect to some *specification*.

Formal methods. Despite the progress done to bring the usefulness of formal methods to everyone (e.g. with [OHe19; Dis+19] or with the Grand Challenge of software verification [JOW06; HM08; Woo06]) their use is still restricted to specific niches of developers. This is due to some problems with the technique itself. Firstly, the problem is intrinsic in the theory of computation. Consider the following program in a pseudo-C language

```
1  int* p = NULL;
2  arbitrary_function();
3  *p = 0;
```

If control reaches Line 3 the program will crash (as we are trying to access address 0x0). Hence we have to prove that `arbitrary_function()` does not halt. Unfortunately Turing [Tur21] shows that this problem is undecidable. Moreover, Rice’s Theorem [Ric53] expands on this stating that *all non-trivial semantics properties* of programs are undecidable. The consequence is that we cannot have an *universal verifier*, i.e., a verification tool that proves or disproves the correctness of every program with respect to some specification. However we do not need to solve such a general problem. We as humans tend to use patterns and structures, even to write our logic. The outcome is that we work with just a small subset of all possible programs, and therefore to work in practice our analyzers have to trace the correctness of just that small subset of programs.

Moreover the result of our analysis does not have to be the most precise description of the outcome of the program we are analyzing. We need a tool that can state that our program satisfies a property (an *invariant*) which is *sound* to the real property of the program, i.e., a property which is *less precise* than the real one. Notable example of such analyzers are well known and available today on the internet. For example Astreè [Cou+05] and Mopsa [MOM23] are two sound analyzers of C and python code, which can infer program properties and catch bugs ahead of time. They both use a technique called *abstract interpretation* which (roughly) involves interpreting a given program by mapping variables to an abstract representation of some properties we are interested in.

Abstract interpretation. Since universal program verification is a fundamentally undecidable problem, the best we can do is to consider non-universal verification, at the cost of getting non-conclusive answers. In this work we focus on one of the major technique for software analysis that can be used to implement a sound verifier: abstract interpretation. To best introduce the technique we start from an example. Consider the following fragment of pseudo-C code:

```

1  int x = 0;
2  for(int i = 0; i < 5; i++){
3      int val = rand();
4      if (val > 0.5) x += 2;
5      else x -= 2;
6  }
7  printf("%d", x);

```

Code 1: Incrementing or decrementing randomly

Each execution of the snippet could result in a different value of x being printed on screen. From a mathematical point of view, before entering the loop the value of x is fixed, it can only be 0. Abstracting the execution means *abstracting* the values the variables can assume. For this example variables can assume *interval* values, e.g., $x \in [0, 0]$ at the beginning of the interval. Assuming `rand()` returns a `float` value in $[0, 1]$, at each iteration either x is incremented or decremented by 2. Hence, after the first iteration $x \in [-2, +2]$, after the second $x \in [-4, +4]$, and so on. The loop could carry on forever, however, because of the `for` guard $i < 5$ we reach a stall at $x \in [-10, +10]$. Therefore at the end of the loop, what our analysis can infer is that $x \in [-10, +10]$.

The analysis is sound: the most precise property of the value of x would be being in the set $S = \{-10, -8, -6, -4, -2, 0, +2, +4, +6, +8, +10\}$, as for each iteration the value of x can increment or decrement by 2, and our result $[-10, +10]$ is a *superset* of S .

Abstract interpretation, introduced by Radhia and Patrick Cousot in [CC77; CC79] is therefore a framework that generalizes this idea by providing tools to compute efficiently general abstractions. As a result, a sound verifier can be obtained by comparing the resulting abstraction with the program specification: if the latter satisfies the former then also the program does and therefore it is correct, otherwise nothing can be said about the program, as the abstraction is an *over-approximation*.

With the example we already introduced the idea of the framework of abstract interpretation:

- We start from a *concrete semantics*, describing the meaning of program commands in a *computational domain*;
- We define an *abstract domain*, which models some properties of interest of the concrete computation and leaves out the details (in our example, the *interval* domain);
- We induce an *abstract semantics*, based on the concrete semantics and our abstract domain, which allows to *abstractly execute* our program on the abstract domain in order to compute the program properties modeled by the abstract domain;
- The result of our abstract execution is the final property of the program.

Generally the abstract execution of our program (i.e. the abstract interpretation) involves *fixpoint computations* on algebraic structures called *lattices*. Lattices are sets equipped with a notion of *order* between elements, while fixpoint computations usually involve computing a minimal element in a *chain* of lattice elements such that the computation does not proceed further. In other words, at which point the guard of a loop is not satisfied anymore and which guarantees can we infer for the program *after* the loop? As an example, consider the snippet of Code 1. We argued that our computation of the abstract value of the variable x could start from $[0, 0]$ and proceed with $[-2, +2], [-4, +4], \dots$. In this case we find a fixpoint with $[-10, +10]$, since the guard of the loop is not respected anymore (because of the variable i) and therefore “executing” the loop again would result in the same initial value $[-10, +10]$, hence a *fixpoint* is reached.

However this is not always the case. Consider instead the following snippet

```

1  int x = 0;
2  while(true) {
3      x += 1;
4  }
```

Code 2: Program with a divergent loop

Obviously the concrete computation does not halt, but looking at the chain of iterands for our program for the variable x we could infer the chain

$$[0, 0], [+1, +1], \dots, [+k, +k], \dots \quad (\dagger)$$

and therefore (intuitively) at the end of the loop we might want to say that our variable has a value in the range $[0, +\infty]$, which is sound to the *real* property of non-termination (i.e., $x \in \emptyset$, x has *no* final value). The procedure we used to infer that the variable x diverges, i.e., its invariant at the end of the loop is $[0, +\infty]$, comes from intuition, and is not an automatic procedure. What a machine could do in fact is exploring the chain we are building in (\dagger) element by element, and if at some point of the chain an element is repeated it can infer a fixpoint. What happens with this naive method is that whenever an *infinite* ascending (or descending) chain is involved the termination of the analysis is not guaranteed.

To infer properties while dealing with infinite chains the standard approach is to use *widening* operators (from [CC92], usually denoted as ∇) to infer the *divergence* of a variable after some round of increments. Such technique however, while still providing a sound analysis and ensuring termination, limits a lot the precision of the analysis itself. For example, we could widen our analysis after the second step with a naive widening $[0, 0] \nabla [+1, +1] = [0, +\infty]$ and infer that our variable after the end of the loop has a value between 0 and $+\infty$ while ensuring that our analysis halts in a finite time. It might happen however that the widening loses more information than needed to ensure termination. Consider in fact the following snippet

```

1  int x = 0;
2  while(x < 10) {
3      x += 1;
4  }
```

Code 3: Program with a convergent loop

The initial steps of the analysis would still be $x \in [0, 0]$ and $x \in [1, 1]$, which widen become again $[0, 0] \nabla [1, 1] = [0, +\infty]$, which again is a sound analysis but is very imprecise. Simply by looking at the program we can infer that the cycle converges and at the end of the loop $x \in [0, 10]$. To recover simple information as this from the program [CC92] also introduce a *narrowing* operator, denoted as Δ . This combination however still has some limitation, as it is heavily relying on the information directly embedded in the program. This opens a question, which is the general problem faced in the thesis:

Problem. Is it possible to have a precise analysis of abstract properties, while ensuring the termination of the analyzer?

The thesis starts by formalising some expected results about the undecidability of the collecting semantics for a simple imperative, Turing complete language. The language include a minimal

set of constructs that make it Turing powerful: linear assignments, interval based guards, non-deterministic choice (which jointly with guards allows us to encode conditionals) and a fixpoint operation (which can be used to encode while loops). Roughly speaking, what is shown is that precise invariants cannot be computed. Actually, it is not even possible to decide the finiteness of invariants, as the halting problem, which is notoriously undecidable, reduces to the finiteness of program invariants.

We try to provide some answers to the question above, by focusing on so-called non-relational abstract semantics, i.e., abstract semantics which disregard the dependencies between the different variables in a program and provide, for each single variable, an abstraction of the set of possible values. It must be observed that if in the abstract domain of interest all ascending chains are finite, the problem trivialises as all fixpoint computations will converge in a finite amount of steps. Thus they can be conducted in a precise way, without the need of any additional theory. The thesis will start by focusing on the domain of intervals, where as observed above one can have infinite ascending chains and thus the computation of the exact abstract semantics is a challenging task. Actually, in principle, one might even think that this is not possible because, in absence of a result proving the converse, establishing whether the abstract analyser terminates (which reduces to establish whether ordinary fixpoint iteration through Kleene iteration does terminate) might be undecidable.

Actually, some results showing that, under some restrictions, interval analysis can be computed exactly were already available. In [SW05] the authors identify a class of interval equations for which the least solution can be computed precisely in polynomial time and the result is later generalised in [Gaw+09] which provides an algorithm that also deals with the arbitrary multiplication of intervals.

In this thesis we take a radically different approach. We show that a preliminary analysis of the program allows one to identify some bounds such that, intuitively, when variables are beyond the bounds the behaviour of the program stabilises. This is the basic tool which allows us to determine whether fixpoint computations will diverge, thus providing a way of establishing the exact result of the computation. Technically, this is obtained by restricting the abstract domain to intervals which are bounded or infinite. Then we prove that the interval semantics computed over this subdomain coincide with the semantics computed on the full domain. This, together with the fact that the bounded domain has no infinite ascending chain allows us to conclude.

The generality of the approach suggests the possibility of extending the result to other non-relational semantics. We then focus on what we call the non-relational collecting semantics, the most precise non-relational semantics, which maps each variable to a set of possible values, forgetting only the relation between variables. The fact that invariants are no longer convex sets, makes the problem sensibly more difficult. We do not succeed in showing that the abstract semantics is computable, but we manage to prove a partial result which, intuitively, says that the termination of the abstract interpreter is decidable or, equivalently, that all invariants generated during the analysis of the program are finite. In case the program has a single while loop, this corresponds exactly to the finiteness of the invariants. This partial result makes us confident about the possibility of proving that also the non-relational collecting semantics is computable in an exact way.

Outline The following document consists of 5 chapters. Chapter 1 provides the necessary background and fixes the necessary notation that will later be used in the following chapters: from recursion and order theory to talk about program termination and undecidability to abstract interpretation to prove some properties of our analysis. Chapter 2 introduces the framework of the thesis: the `Imp` language and its constraints, its *concrete* semantics and its properties: namely undecidability of some properties of programs written in the language. Chapter 3 proves the properties of interval and non-relational collecting analysis on the `Imp` language, while Chapter 4 proves that, similarly to previous work, it is possible to bound the domains we previously introduced to remove infinite ascending chains, hence ensuring analysis termination. Finally Chapter 5 presents the thesis results and proposes a direction for future work on the topic.

CHAPTER 1

Background

The following chapter aims to provide context, notation and the important external references for the work that will follow on. We start with Section 1.1 where we introduce some notation and the important aspect of recursion theory needed to understand the following chapters. Later, in Section 1.2 we explore order theory and set the notation that we will use in the rest of the thesis to talk about this topic. Finally Section 1.3 introduces the notation and concepts we need to properly talk about abstract interpretation in later chapters.

1.1 Recursion theory

This first section aims to provide background and terminology for the parts in recursion theory that will follow. More in detail, we will take some notation from [Cut80] and introduce some notation based on the same book. We start with functions: total and partial functions are essential to recursion theory:

Definition 1.1 (Total and partial functions). Let X, Y be two sets. We denote by

$$X \rightarrow Y$$

the set of all total functions from X to Y . And by

$$X \hookrightarrow Y$$

the set of all partial functions from X to Y .

Partial functions are actually functions from a subset $S \subseteq X$ which is called the *natural domain* of f .

Definition 1.2 (Domain of partial functions). Let $f : X \hookrightarrow Y$. We write $f(x) \downarrow$ to indicate that f is defined on x , and $f(x) \uparrow$ to indicate that f is undefined on x . The domain of f is

$$\text{dom}(f) = \{x \in X \mid f(x) \downarrow\}$$

We then need (mostly in Section 2.4) to talk about partial recursive functions and their properties. We therefore define partial recursive and total recursive functions as follows:

Notation 1.3 (partial and total recursive functions). By $\mathbb{N}^k \xrightarrow{r} \mathbb{N}$ we denote the set of partial recursive functions on natural numbers, while by $\mathbb{N} \xrightarrow{r} \mathbb{N}$ we denote the set of total recursive functions on natural numbers.

We also need to talk about decidable properties and decidable sets. We therefore introduce the notion of recursive and recursively enumerable sets.

Definition 1.4 (Recursively enumerable and recursive sets). A set $A \subseteq \mathbb{N}^k$ is *recursively enumerable* (r.e. or semi-decidable) if $A = \text{dom}(f)$ for some $f \in \mathbb{N}^k \xrightarrow{\tau} \mathbb{N}$.

A set $A \subseteq \mathbb{N}$ is a recursive set if both A and its complement $\bar{A} = \mathbb{N} \setminus A$ are semi-decidable, i.e., there exists some $f \in \mathbb{N} \xrightarrow{\tau} \mathbb{N}$ s.t.

$$f = \lambda n. (n \in A)?1 : 0$$

1.2 Order theory

Within Theoretical Computer Science, especially in the field of semantics, partial orders hold significant importance. They are extensively employed in Abstract Interpretation, as highlighted in [Min18], serving different levels of the theory to model core notions. These notions include the idea of approximation, where certain analysis results may be less precise than others, creating a partial order where some results are incomparable. Moreover, partial orders are fundamental in conveying the concept of soundness: an analysis is deemed sound if its result is an over-approximation of the actual behavior. These mathematical notions, essential for discussions surrounding the Abstract Interpretation formalism, primarily involve order and lattice theory.

Definition 1.5 (Partially ordered set). Let X be a non-empty set, $\sqsubseteq \subseteq X \times X$ be a reflexive, anti-symmetric and transitive relation on that set, i.e., $\forall x, y, z \in X$:

1. $x \sqsubseteq x$ (reflexivity)
2. $x \sqsubseteq y \wedge y \sqsubseteq x \implies x = y$ (antisymmetry)
3. $x \sqsubseteq y \wedge y \sqsubseteq z \implies x \sqsubseteq z$ (transitivity)

Then the tuple $\langle X, \sqsubseteq \rangle$ is a *partially ordered set* (POSet).

Definition 1.6 (Least upper bound). Let $\langle X, \sqsubseteq \rangle$ be a POSet and let $Z \subseteq X$. We say that $\bar{z} \in Z$ is an *upper bound* of Z if $\forall z \in Z \ z \sqsubseteq \bar{z}$. It is the *least upper bound* of Z if $\forall z'$ upper bounds of Z , $\bar{z} \sqsubseteq z'$.

Definition 1.7 (Greatest lower bound). Let $\langle X, \sqsubseteq \rangle$ be a POSet and let $Z \subseteq X$. We say that $\bar{z} \in Z$ is a *lower bound* of Z if $\forall z \in Z \ \bar{z} \sqsubseteq z$. It is the *greatest lower bound* of Z if $\forall z'$ upper bounds of Z , $z' \sqsubseteq \bar{z}$.

Usually then we are talking about least and greatest lower bound the host set is often implicit, and we therefore simply write $\text{lub}(Z)$ and $\text{glb}(Z)$. In abstract interpretation we often rely on special kinds of POSets, where the existence of the greatest lower bound and the least upper bound is ensured for each subset of the original POSet. These sets are called complete lattices

Definition 1.8 (Complete lattice). A POSet $\langle X, \sqsubseteq \rangle$ is called a *complete lattice* if

$$\forall Y \subseteq X \quad \exists \bigcup Y \wedge \exists \bigcap Y$$

Complete lattices are a subset of the class of chain complete partial ordered sets. These kinds of partial orders are defined using the concept of chains:

Definition 1.9 (Chain). Let $\langle D, \sqsubseteq \rangle$ be a partially ordered set. Then $Y \subseteq D$ is a chain if for any $y_1, y_2 \in Y$ it holds that

$$y_1 \sqsubseteq y_2 \vee y_2 \sqsubseteq y_1$$

Definition 1.10 (CCPOs). A chain complete partially ordered set (ccpo) is a poset $\langle D, \sqsubseteq \rangle$ such that every chain of D has a least upper bound.

The last building block we will use in the following chapters is the Kleene-Knaster-Tarski theorem. This theorem is a fundamental result in order theory and provides a powerful tool for analyzing and establishing the existence of fixed points in complete lattices. To state it we need to first link functions and order theory with some definitions

Definition 1.11 (Monotone functions). Let $\langle D, \sqsubseteq \rangle$ and $\langle D', \sqsubseteq' \rangle$ be complete lattices. The total function $f : D \rightarrow D'$ is *monotone* if

$$d_1 \sqsubseteq d_2 \implies f(d_1) \sqsubseteq' f(d_2)$$

Monotonicity however does not preserve upper bounds, just orders. In particular if we take a chain $Y \subseteq D$ of some ccpo $\langle D, \sqsubseteq \rangle$ and some monotone function $f : D \rightarrow D$, in general $\sqcup \{f(d) \mid d \in Y\} \sqsubseteq f(\sqcup Y)$, but not $\sqcup \{f(d) \mid d \in Y\} = f(\sqcup Y)$. Therefore we introduce the concept of continuity, functions that preserve both order and upper bounds

Definition 1.12 (Continuous functions). Let $\langle X, \sqsubseteq \rangle$ and $\langle X', \sqsubseteq' \rangle$ be ccpos. The total function $f : D \rightarrow D'$ is *continuous* if

- f is monotone;
- $\sqcup' \{f(d) \mid d \in D\} = f(\sqcup X)$

Continuous functions over ccpos are important for the Kleene fixed-point theorem, usually attributed to Tarski from [Tar55], which is also called Kleene iteration. It gives us an iteration strategy to find the least fixpoint of a function over a ccpo, provided that the function is continuous.

Theorem 1.13 (Kleene fixed-point). Let $f : D \rightarrow D$ be a continuous function over a chain complete partial order $\langle D, \sqsubseteq \rangle$ with the least element \perp . Then

$$\text{lfp}(f) = \bigsqcup \{f^n(\perp) \mid n \in \mathbb{N}\}$$

where

- $f^0 = \text{id}$
- $f^{n+1} = f \circ f^n \quad \forall n \in \mathbb{N}$

is the least fix point of f .

1.3 Abstract Interpretation

Abstract interpretation is among the most well known methods of static analysis of programs. First introduced by Patrick and Radhia Cousot in [CC77; CC79], it consists in a sound-by-construction method to infer program properties given a model of their behavior. The general idea is that we can approximate the semantics of a program with monotonic functions over ordered sets (usually *complete lattices*). To do so we usually first introduce *Abstract Domains* that capture some essential aspects of program execution while ignoring the details of the computation, which would make the analysis computationally infeasible. This analysis however carries the issue of completeness, which is closely related to the issue of choosing the best abstract domain to decide program correctness without raising false alarms. Achieving completeness in analysis is often desirable, but it can lead to the problem of undecidability. This means that even though we strive for the most accurate analysis of a program, such as through its interpreter, we can't guarantee that the process will always terminate. The technique per-se is a concept which is around since the '70s, hence an extensive amount of literature has been produced. For a brief history of the technique, see [GR22].

The main source of this chapter comes from the notes on abstract interpretation in [Min18].

1.3.1 General concepts

Abstract interpretation heavily relies on order theory, which we introduced in Section 1.2, and builds on top of it. The core idea is that we use an *abstract domain* as an approximation of the *concrete domain*, in such a way that abstract computations are *sound* with respect to the concrete ones. The minimal structure that we will require both in the abstract and the concrete domains is a partial order that models the amount of information each instruction carries with respect to the program execution. Thus, the concrete domain is a partially ordered set $\langle C, \leq \rangle$ (e.g. integers powersets) and the abstract domain is another partially ordered set $\langle A, \sqsubseteq \rangle$ (e.g. intervals). The minimal amount of connection between these two worlds is a *concretization* functions

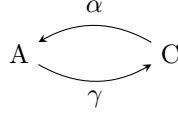


Figure 1.1: Galois connection between an abstract domain A and a concrete domain C

Definition 1.14 (Concretization). A concretization function $\gamma : \langle A, \sqsubseteq \rangle \rightarrow \langle C, \leq \rangle$ is a *monotonic* function from an *abstract* partially ordered set $\langle A, \sqsubseteq \rangle$ to a *concrete* partially ordered set $\langle C, \leq \rangle$

Trough concretization we have a first notion of *soundness*

Definition 1.15 (Soundness). Given an *abstract* domain A and a *concrete* domain C , we call $a \in A$ a *sound* abstraction of $c \in C$ iff $c \leq \gamma(a)$.

While monotonic concretizations are sufficient to reason about soundness, more structure is useful to design a sound and *accurate* analyzer. The standard abstract interpretation framework from [CC77] also assumes the existence of some monotonic *abstraction function* $\alpha : \langle C, \leq \rangle \rightarrow \langle A, \sqsubseteq \rangle$, such that $\langle \alpha, C, A, \gamma \rangle$ forms a Galois connection:

Definition 1.16 (Galois connection). Given two partially ordered sets $\langle C, \leq \rangle, \langle A, \sqsubseteq \rangle$, the tuple $\langle \alpha, C, A, \gamma \rangle$ is a Galois connection if

- A, C are complete lattices;
- $\alpha : \langle C, \leq \rangle \rightarrow \langle A, \sqsubseteq \rangle$ and $\gamma : \langle A, \sqsubseteq \rangle \rightarrow \langle C, \leq \rangle$ are monotonic;
- for all $a \in A, c \in C$,

$$c \leq \gamma(a) \iff \alpha(c) \sqsubseteq a. \quad (1.1)$$

We denote $\langle \alpha, C, A, \gamma \rangle$ as $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$.

Usually though we use an alternative characterization of Galois Connections:

Theorem 1.17. $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$ is a Galois connection iff the function pair $\langle \alpha, \gamma \rangle$ satisfies all the following properties:

- (1) α, γ are monotonic;
- (2) $\forall c \in C \quad c \leq \gamma(\alpha(c))$ i.e., $\gamma \circ \alpha$ is extensive;
- (3) $\forall a \in A \quad \alpha(\gamma(a)) \sqsubseteq a$, i.e., $\alpha \circ \gamma$ is reductive.

Proof. Assume that $\langle \alpha, \gamma \rangle$ satisfies (1.1), then we want to prove that the properties of Theorem 1.17 hold.

- (1) Applying (1.1) with $a \triangleq \alpha(c)$ we get

$$c \leq \gamma(\alpha(c)) \quad (1.2)$$

i.e., $\gamma \circ \alpha$ is extensive, which is our first thesis.

- (2) Applying (1.1) with $c \triangleq \gamma(a)$ we get

$$\alpha(\gamma(a)) \sqsubseteq a \quad (1.3)$$

i.e., $\alpha \circ \gamma$ is reductive, which is our second thesis.

- (3) By (1) $\forall c, c' \in C$ it holds that $c \leq c' \implies c \leq \gamma(\alpha(c'))$. Hence, we can apply again (1.1) with $a \triangleq \alpha(c')$ and get that $\alpha(c) \sqsubseteq \alpha(c')$, i.e., α is monotonic.

- (4) By (2) $\forall a, a' \in C$ it holds that $a \sqsubseteq a' \implies \alpha(\gamma(a)) \sqsubseteq a$. Hence, we can apply (1.1) with $c \triangleq \gamma(a)$ and get that $\gamma(a) \leq \gamma(a')$, i.e., γ is monotonic.

Assume conversely that the four properties hold. Then we want to prove that (1.1) holds.

- (\implies) First assume that $c \leq \gamma(a)$. Then $\alpha(c) \sqsubseteq \alpha(\gamma(a))$ by monotonicity of α and $\alpha(\gamma(a)) \sqsubseteq a$ by reductivity, hence $\alpha(c) \sqsubseteq a$.
- (\impliedby) Likewise, assume that $\alpha(c) \sqsubseteq a$. Then $\gamma(\alpha(c)) \leq \gamma(a)$ by monotonicity of γ and $c \leq \gamma(\alpha(c))$ by extensivity, hence $c \leq \gamma(a)$.

□

Galois connections carry with them some well known properties, that are useful to state best correct approximations (bca) and to prove the soundness by construction of the analyzer:

Theorem 1.18 (Galois connection properties). *Given a Galois connection $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$ we have:*

1. $\gamma \circ \alpha \circ \gamma = \gamma$ and $\alpha \circ \gamma \circ \alpha = \alpha$;
2. $\alpha \circ \gamma$ and $\gamma \circ \alpha$ are idempotent;
3. $\forall c \in C \ \alpha(c) = \sqcap \{a \mid c \leq \gamma(a)\}$;
4. $\forall a \in A \ \gamma(a) = \vee \{c \mid \alpha(c) \sqsubseteq a\}$;
5. α maps concrete lubs to abstract lubs:

$$\forall X \subseteq C \quad \exists \vee X \implies \alpha(\vee X) = \sqcup \{\alpha(x) \mid x \in X\}$$

6. γ maps abstract glbs to concrete glbs:

$$\forall X \subseteq A \quad \exists \sqcap X \implies \gamma(\sqcap X) = \wedge \{\gamma(x) \mid x \in X\}$$

Theorem 1.18 states two important properties of Galois connections: *soundness* and *optimality*. Recall that $c \leq \gamma(a)$ means by Definition 1.15 that a is a *sound* approximation of c . Then, given $c \in C$ recall that $c \leq \gamma(\alpha(c))$, which means that $\alpha(c)$ is a sound abstraction of c . Finally, Property 3 states that $\alpha(c)$ is the best (i.e., smallest) sound abstraction of c , i.e., the *optimal* abstraction.

Additionally, from the theorem we can derive the following corollary:

Corollary 1.19 (Best abstraction). *If we have a Galois connection $\langle \alpha, C, A, \gamma \rangle$, then $\forall c \in C, \alpha(c)$ is the best abstraction of c , i.e., the smallest abstract element which is a sound abstraction of c .*

In general we saw that for a Galois connection $\gamma \circ \alpha$ is idempotent, and generally not the identity function, as abstracting looses precision. Concretizing however, should not loose precision, so we could expect $\alpha \circ \gamma$ to be the identity function. When this is the case, we have a *Galois insertion*:

Definition 1.20 (Galois Insertion). A Galois connection $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$ is a *Galois insertion* if one of the following, equivalent properties hold:

1. α is surjective: $\forall a \in A \ \exists c \in C \mid \alpha(c) = a$;
2. γ is injective: $\forall a, a' \in A \ \gamma(a) = \gamma(a') \implies a = a'$;
3. $\alpha \circ \gamma$ is the identity function.

We denote Galois insertions as $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$.

Next, we can show that given a Galois connection and by doing a *point-wise lifting* of the values in the concrete and abstract domain, we get another Galois connection. This will later be useful in Chapter 3 to lift proofs from a domain to its point-wise lifting.

Theorem 1.21. *Given a Galois connection $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$ we can derive a new Galois connection with the point wise lifting, i.e. $\langle S \rightarrow C, \dot{\leq} \rangle \xleftrightarrow[\dot{\alpha}]{\dot{\gamma}} \langle S \rightarrow A, \dot{\sqsubseteq} \rangle$ where*

$$\begin{aligned} f \dot{\leq} f' &\triangleq \forall s \in S \quad f(s) \leq f'(s) \\ f \dot{\sqsubseteq} f' &\triangleq \forall s \in S \quad f(s) \sqsubseteq f'(s) \\ \dot{\alpha}(f) &\triangleq \lambda s \in S. \alpha(f(s)) \\ \dot{\gamma}(f) &\triangleq \lambda s \in S. \gamma(f(s)) \end{aligned}$$

and S is an arbitrary set.

Proof.

$$\begin{aligned} \dot{\alpha}(c) \dot{\sqsubseteq} a &\iff \forall x \in S \quad \alpha(c)(x) \sqsubseteq a(x) && \text{by definition} \\ &\iff \forall x \in S \quad c(x) \leq a(x) && \text{by (1.1)} \\ &\iff c \dot{\leq} \dot{\gamma}(a) && \text{by definition} \end{aligned}$$

□

from this, we can derive the following

Theorem 1.22. *if $\langle C, \leq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq \rangle$ is a Galois insertion, then its point-wise lifting is again a Galois insertion:*

$$\langle S \rightarrow C, \dot{\leq} \rangle \xleftrightarrow[\dot{\alpha}]{\dot{\gamma}} \langle S \rightarrow A, \dot{\sqsubseteq} \rangle$$

Proof. By Th. 1.21 it holds that $\langle S \rightarrow C, \dot{\leq} \rangle \xleftrightarrow[\dot{\alpha}]{\dot{\gamma}} \langle S \rightarrow A, \dot{\sqsubseteq} \rangle$. What we have to prove is that $\dot{\alpha} \circ \dot{\gamma} = \text{id}$ knowing that $\alpha \circ \gamma = \text{id}$.

$$\begin{aligned} (\dot{\alpha} \circ \dot{\gamma})(f) &= \dot{\alpha}(\lambda s \in S. \gamma(f(s))) && \text{by definition} \\ &= \lambda s \in S. (\alpha \circ \gamma)(f(s)) && \text{by definition} \\ &= \lambda s \in S. \text{id}(f(s)) && \text{by hypothesis} \\ &= f && \text{by definition} \end{aligned}$$

hence $\dot{\alpha} \circ \dot{\gamma} = \text{id}$, which means $\langle S \rightarrow C, \dot{\leq} \rangle \xleftrightarrow[\dot{\alpha}]{\dot{\gamma}} \langle S \rightarrow A, \dot{\sqsubseteq} \rangle$. □

In this context we also need a way of ensuring abstract operations are sound (and occasionally) exact. Even by only using the concretization map and no Galois connection, the notion of sound and exact abstraction carries naturally from domain elements to domain operators:

Definition 1.23 (Sound and exact operator abstraction). Let $\gamma : \langle A, \sqsubseteq \rangle \rightarrow \langle C, \leq \rangle$ be a concretization map from an abstract domain $\langle A, \sqsubseteq \rangle$ to a concrete domain $\langle C, \leq \rangle$, $f : C \rightarrow C$ be a concrete operator and $g : A \rightarrow A$ an abstract operator.

1. g is a *sound abstraction* of f if $\forall a \in A \quad f(\gamma(a)) \leq \gamma(g(a))$;
2. g is an *exact abstraction* of f if $f \circ \gamma = \gamma \circ g$.

Notice that an exact abstraction is always sound. Another remarkable thing of Galois connections, is that along with this notion we can introduce the notion of *Best correct approximation*:

Definition 1.24 (Best correct approximation (bca)). Given a Galois connection $\langle \alpha, C, A, \gamma \rangle$ and a concrete operator $f : C \rightarrow C$, the *best abstraction* of f is given by $\alpha \circ f \circ \gamma$.

It is imperative to prioritize the modular and composable nature of our abstractions. As elaborated in subsequent chapters, the semantics of a program typically emerge from the composition of atomic semantic functions drawn from a finite library representing fundamental language operations. This framework lends itself well to a modular abstraction scheme, where abstract operators are designed exclusively for this foundational set of operations. By adhering to the same principles governing concrete semantics, these abstract operators can be composed effectively. This is facilitated by the inherent composability and accuracy of sound abstractions:

Theorem 1.25 (Operator composition). *Let $f, f' : C \rightarrow C$ be concrete operators and $g, g' : A \rightarrow A$ abstract operators. The following properties hold:*

- (1) *if g, g' are sound abstractions of f and f' respectively and f is monotonic, then $g \circ g'$ is a sound abstraction of $f \circ f'$;*
- (2) *if g, g' are exact abstractions of f and f' respectively then $g \circ g'$ is an exact abstraction of $f \circ f'$.*

Proof. We proceed to prove the two properties in order:

- (1) g' is a sound abstraction of f' , hence

$$\begin{aligned} \forall a \in A \quad (f' \circ \gamma)(a) &\leq (\gamma \circ g')(a) \\ (f \circ f' \circ \gamma)(a) &\leq (f \circ \gamma \circ g')(a) && \text{by monotonicity of } f \\ (f \circ f' \circ \gamma)(a) &\leq (f \circ \gamma \circ g')(a) \leq (\gamma \circ g \circ g')(a) && \text{by monotonicity of } g \end{aligned}$$

- (2) since both $f \circ \gamma = \gamma \circ g$ and $f' \circ \gamma = \gamma \circ g'$ it holds that

$$f \circ f' \circ \gamma = f \circ \gamma \circ g' = \gamma \circ g \circ g'$$

□

1.3.2 Fixpoint approximations

Critical parts of program semantics are defined through the idea of least fixpoints $\text{lfp } f$ of some monotonic function or continuous operator $f : C \rightarrow C$ over the concrete domain $\langle C, \leq \rangle$. In order to abstract the computation of $\text{lfp } f$ in the abstract domain $\langle A, \sqsubseteq \rangle$ the natural idea is to start with a sound abstraction $g : A \rightarrow A$ of f . Then there are many ways of approximate the fixpoint computation through the use of g . We mention two, in order to see what its main problems are.

Kleene fixpoint. A first idea is to mimic the fixpoint computation with g instead of f . For instance by relying on the constructive definition of $\text{lfp } f$ as the limit of an iteration sequence from Kleene's Theorem 1.13:

$$\text{lfp}(f) = \sqcup \{f^i(\perp) \mid i \in \mathbb{N}\} \sqsubseteq \sqcup \{g^i(\perp) \mid i \in \mathbb{N}\} = \text{lfp}(g)$$

Tarski fixpoint. Tarski fixpoint is instead based on the Tarski characterization of the fixpoint:

$$\text{lfp}(f) = \sqcap \{x \in C \mid f(x) \sqsubseteq x\} \tag{1.4}$$

i.e., the least fixpoint of a monotonic function f over a complete lattice f is the greatest lower bound of the post-fixpoints. The observation is that any abstract post-fixpoint of a sound abstract represents, through γ , a concrete fixpoint (by Theorem 1.18). The theorem however does not state how to compute a post-fixpoint of g in the abstract, but is nonetheless useful as it allows us to provide a sound answer (a post-fixpoint) even with abstract fixpoints are difficult to compute (as for infinite ascending chains) or non-existent at all.

As mentioned before, in order to solve the convergence problem in abstract domains in [CC77] the authors proposed to use Tarski's characterization to provide post fixpoint of infinite ascending chains, introducing a binary operator: the *widening* operator ∇ .

Definition 1.26 (Widening operator). A binary operator $\nabla : A \times X \rightarrow A$ is a *widening operator* in an abstract domain $\langle A, \sqsubseteq \rangle$ if

1. it computes upper bounds:

$$\forall x, y \in A \quad x \sqsubseteq x \nabla y \wedge y \sqsubseteq x \nabla y;$$

2. it enforces convergence: for any sequence $\{y^i\}_{i \in \mathbb{N}}$ in A , the sequence $\{z^i\}_{i \in \mathbb{N}}$ computed as

$$\begin{aligned} z^0 &\triangleq y^0 \\ z^{i+1} &\triangleq z^i \triangle y^{i+1} \end{aligned}$$

stabilizes in finite time: $\exists k \geq 0 \mid x^{k+1} = x^k$.

With the widening operator we enforce the termination of a sound abstract analyzer for the computation of upper bounds:

Theorem 1.27. *Let f be a monotonic operator in a complete concrete lattice and g a sound abstraction of f . Then the following iteration*

$$\begin{aligned} x^0 &\triangleq \perp \\ x^{i+1} &\triangleq x^i \nabla g(x^i) \end{aligned}$$

converges in finite time, and its limit x is a sound abstraction of the least fixpoint $\text{lfp}(f)$, i.e., $\text{lfp}(f) \leq \gamma(x)$.

Proof. Convergence is ensured by Property 2 of Definition 1.26, while the soundness is because of Equation (1.4) given that, when encountering a stable value $x^{i+1} = x^i$, then $f(x^i) \sqsubseteq x^i \nabla f(x^i) = x^{i+1}$, i.e., x^i is an abstract post-fixpoint. \square

Where the first point states that \triangle refines its left argument while bringing a sound approximation, and the second point enforces termination.

CHAPTER 2

Framework

In order to talk about program properties we need a language to express such programs. We define the `Imp` language, made of regular commands and based on Kozen's Kleene algebra with tests, described in [Koz97].

2.1 The Imp language

We denote by \mathbb{Z} the set of integers with the usual order, extended with the least and greatest elements $-\infty$ and $+\infty$, s.t. $-\infty \leq z \leq +\infty$ for all $z \in \mathbb{Z}$. We also extend addition and subtraction by letting, for all $z \in \mathbb{Z}$ it holds that $+\infty + z = +\infty - z = +\infty$ and $-\infty + z = -\infty - z = -\infty$. We focus on the following non-deterministic language.

$$\begin{aligned} \text{Exp} \ni e &::= \mathbf{x} \in I \mid \mathbf{x} := k \mid \mathbf{x} := \mathbf{y} + k \\ \text{Imp}_s \ni D &::= e \mid D + D \mid D; D \\ \text{Imp} \ni C &::= D \mid C + C \mid C; C \mid C^* \mid \text{fix}(C) \end{aligned}$$

where $\mathbf{x}, \mathbf{y} \in \text{Var}$ a finite set of variables of interest, i.e., the variables appearing in the considered program, $I \in \mathbb{I}$ an *interval* (as defined in Definition 3.13), $a \in \mathbb{Z} \cup \{-\infty\}$, $b \in \mathbb{Z} \cup \{+\infty\}$, $a \leq b$ and $k \in \mathbb{Z}$ is any finite integer constant.

2.2 Semantics

In order to talk about program properties in our language, we first need to define its *semantics*. In the following section we introduce both a collecting semantics in order to reason about program *invariants* and a small step semantics, in order to reason about program *execution*.

Definition 2.1 (Semantics of Basic Expressions). Let *environments* be the maps from the set of variables to their numerical value: $\text{Env} \triangleq \{\rho \mid \rho : \text{Var} \rightarrow \mathbb{Z}\} \cup \{\perp\}$. For basic expressions $e \in \text{Exp}$ the *concrete semantics* $\llbracket \cdot \rrbracket : \text{Exp} \rightarrow \text{Env} \rightarrow \text{Env}$ is inductively defined by:

$$\begin{aligned} \llbracket e \rrbracket \perp &\triangleq \perp \\ \llbracket \mathbf{x} \in I \rrbracket \rho &\triangleq \begin{cases} \rho & \rho(\mathbf{x}) \in I \\ \perp & \text{otherwise} \end{cases} \\ \llbracket \mathbf{x} := k \rrbracket \rho &\triangleq \rho[\mathbf{x} \mapsto k] \\ \llbracket \mathbf{x} := \mathbf{y} + k \rrbracket \rho &\triangleq \rho[\mathbf{x} \mapsto \rho(\mathbf{y}) + k] \end{aligned}$$

The next building block is the concrete collecting semantics for the language, it associates each program in \mathbf{Imp} to a function which, given a set of initial environments X “collects” the set of final states produced by executing the program from X .

Definition 2.2 (Concrete collecting semantics). Let $\mathcal{C} \triangleq \langle \wp(\mathbf{Env}), \subseteq \rangle$ be a complete lattice called *concrete collecting domain*. The *concrete collecting semantics* for \mathbf{Imp} is given by the total function $\langle \cdot \rangle : \mathbf{Imp} \rightarrow \mathcal{C} \rightarrow \mathcal{C}$ which maps each program $C \in \mathbf{Imp}$ to a total function over the complete lattice \mathcal{C} , inductively defined as follows: given $X \in \mathcal{C}$

$$\begin{aligned} \langle e \rangle X &\triangleq \{ \langle e \rangle \rho \mid \rho \in X, \langle e \rangle \rho \neq \perp \} \\ \langle C_1 + C_2 \rangle X &\triangleq \langle C_1 \rangle X \cup \langle C_2 \rangle X \\ \langle C_1; C_2 \rangle X &\triangleq \langle C_2 \rangle (\langle C_1 \rangle X) \\ \langle C^* \rangle X &\triangleq \bigcup_{i \in \mathbb{N}} \langle C \rangle^i X \\ \langle \text{fix}(C) \rangle X &\triangleq \text{lfp}(\lambda Y \in \wp(\mathbf{Env}). (X \cup \langle C \rangle Y)) \end{aligned}$$

We observe that the semantics we described is additive:

Observation 2.3 (Additivity). Given $C \in \mathbf{Imp}$, $X, Y \in \mathcal{C}$,

$$\langle C \rangle (X \cup Y) = \langle C \rangle X \cup \langle C \rangle Y$$

Proof. We will prove it by induction on the program C . Let’s first explore the base cases.

Case (e). Therefore

$$\begin{aligned} \langle e \rangle (X \cup Y) &= \{ \langle e \rangle \rho \mid \rho \in X \cup Y, \langle e \rangle \rho \neq \perp \} \\ &= \{ \langle e \rangle \rho \mid \rho \in X \vee \rho \in Y, \langle e \rangle \rho \neq \perp \} \\ &= \{ \langle e \rangle \rho \mid \rho \in X, \langle e \rangle \rho \neq \perp \} \cup \{ \langle e \rangle \rho \mid \rho \in Y, \langle e \rangle \rho \neq \perp \} \\ &= \langle e \rangle X \cup \langle e \rangle Y \end{aligned}$$

Next we can explore the inductive cases.

Case $(C_1 + C_2)$. Therefore

$$\begin{aligned} \langle C_1 + C_2 \rangle (X \cup Y) &= \langle C_1 \rangle (X \cup Y) \cup \langle C_2 \rangle (X \cup Y) && \text{by definition} \\ &= \langle C_1 \rangle X \cup \langle C_1 \rangle Y \cup \langle C_2 \rangle X \cup \langle C_2 \rangle Y && \text{by inductive hypothesis} \\ &= \langle C_1 + C_2 \rangle X \cup \langle C_1 + C_2 \rangle Y \end{aligned}$$

Case $(C_1; C_2)$. Therefore

$$\begin{aligned} \langle C_1; C_2 \rangle (X \cup Y) &= \langle C_2 \rangle (\langle C_1 \rangle (X \cup Y)) && \text{by definition} \\ &= \langle C_2 \rangle (\langle C_1 \rangle X \cup \langle C_1 \rangle Y) && \text{by inductive hypothesis} \\ &= \langle C_2 \rangle (\langle C_1 \rangle X) \cup \langle C_2 \rangle (\langle C_1 \rangle Y) && \text{by inductive hypothesis} \end{aligned}$$

Case (C^*) . Therefore

$$\langle C^* \rangle (X \cup Y) = \bigcup_{i \in \mathbb{N}} \langle C \rangle^i (X \cup Y)$$

in order to use the inductive hypothesis we have to show that

$$\forall i \in \mathbb{N} \quad \langle C \rangle^i (X \cup Y) = \langle C \rangle^i X \cup \langle C \rangle^i Y$$

to do that, we work again by induction on i :

- the base case is $i = 0$ then $X \cup Y = X \cup Y$.
- For the inductive case we need to show that $i \implies i + 1$:

$$\begin{aligned}
\langle C \rangle^{i+1} (X \cup Y) &= \langle C \rangle (\langle C \rangle^i (X \cup Y)) \\
&= \langle C \rangle (\langle C \rangle^i X \cup \langle C \rangle^i Y) && \text{by induction hypothesis on } i \\
&= \langle C \rangle (\langle C \rangle^i X) \cup \langle C \rangle (\langle C \rangle^i Y) && \text{by induction hypothesis on } C \\
&= \langle C \rangle^{i+1} X \cup \langle C \rangle^{i+1} Y
\end{aligned}$$

Therefore we can use the inductive hypothesis internally and say

$$\begin{aligned}
\langle C^* \rangle (X \cup Y) &= \bigcup_{i \in \mathbb{N}} \langle C \rangle^i (X \cup Y) \\
&= \bigcup_{i \in \mathbb{N}} (\langle C \rangle^i X \cup \langle C \rangle^i Y) && \text{for the later statement} \\
&= \left(\bigcup_{i \in \mathbb{N}} \langle C \rangle^i X \right) \cup \left(\bigcup_{i \in \mathbb{N}} \langle C \rangle^i Y \right) \\
&= \langle C^* \rangle X \cup \langle C^* \rangle Y \quad \square
\end{aligned}$$

We can also observe that a program induces a monotone function in the concrete domain \mathcal{C} :

Lemma 2.4. *Given a program $C \in \text{Imp}$, the semantic function $\langle C \rangle : \mathcal{C} \rightarrow \mathcal{C}$ is monotone.*

Proof. We can prove this by induction on the program $C \in \text{Imp}$. Let $X, Y \in \mathcal{C}, X \subseteq Y$. We want to prove that $\langle C \rangle X \subseteq \langle C \rangle Y$.

Case (e). In this case

$$\begin{aligned}
\langle e \rangle X &= \{ \langle e \rangle \rho \mid \rho \in X, \langle e \rangle \rho \neq \perp \} \\
\langle e \rangle Y &= \{ \langle e \rangle \rho \mid \rho \in Y, \langle e \rangle \rho \neq \perp \}
\end{aligned}$$

$X \subseteq Y$ therefore $\rho \in X \implies \rho \in Y$ which also means that $\rho' \in \langle e \rangle X \implies \rho' \in \langle e \rangle Y$, therefore $\langle e \rangle X \subseteq \langle e \rangle Y$

Case $(C_1 + C_2)$. In this case we need to show that $\langle C_1 + C_2 \rangle X \subseteq \langle C_1 + C_2 \rangle Y$

$$\begin{aligned}
\langle C_1 + C_2 \rangle X &= \langle C_1 \rangle X \cup \langle C_2 \rangle X \\
\langle C_1 + C_2 \rangle Y &= \langle C_1 \rangle Y \cup \langle C_2 \rangle Y
\end{aligned}$$

by inductive hypothesis both $\langle C_1 \rangle X \subseteq \langle C_1 \rangle Y$ and $\langle C_2 \rangle X \subseteq \langle C_2 \rangle Y$ and therefore $\langle C_1 + C_2 \rangle X \subseteq \langle C_1 + C_2 \rangle Y$.

Case $(C_1; C_2)$. Therefore we need to show that $\langle C_1; C_2 \rangle X \subseteq \langle C_1; C_2 \rangle Y$

$$\begin{aligned}
\langle C_1; C_2 \rangle X &= \langle C_2 \rangle (\langle C_1 \rangle X) \\
\langle C_1; C_2 \rangle Y &= \langle C_2 \rangle (\langle C_1 \rangle Y)
\end{aligned}$$

By induction hypothesis $\langle C_1 \rangle X \subseteq \langle C_1 \rangle Y$, and by induction hypothesis again $\langle C_2 \rangle (\langle C_1 \rangle X) \subseteq \langle C_2 \rangle (\langle C_1 \rangle Y)$ which means $\langle C_1; C_2 \rangle X \subseteq \langle C_1; C_2 \rangle Y$.

Case (C^*) . Therefore we need to show that $\langle C^* \rangle X \subseteq \langle C^* \rangle Y$.

$$\begin{aligned}
\langle C^* \rangle X &= \bigcup_{i \in \mathbb{N}} \langle C \rangle^i X \\
\langle C^* \rangle Y &= \bigcup_{i \in \mathbb{N}} \langle C \rangle^i Y
\end{aligned}$$

what we need to prove is that

$$\forall j \in \mathbb{N} \quad \bigcup_{i=0}^j \langle C \rangle^i X \subseteq \bigcup_{i=0}^j \langle C \rangle^i Y$$

we can do this by induction on j :

- $j = 0$ therefore $X \subseteq Y$ which is true by hypothesis.
- Now we need to work on the inductive case $j \implies j + 1$. Notice that it holds that

$$\begin{aligned} \bigcup_{i=0}^{k+1} \langle C \rangle^i X &= X \cup \bigcup_{i=1}^{k+1} \langle C \rangle^i X && \text{by definition} \\ &= X \cup \langle C \rangle \left(\bigcup_{i=0}^k \langle C \rangle^i X \right) && \text{by additivity} \end{aligned}$$

and also for Y

$$\bigcup_{i=0}^{k+1} \langle C \rangle^i Y = Y \cup \langle C \rangle \left(\bigcup_{i=0}^k \langle C \rangle^i Y \right)$$

Also notice that

- (i) $X \subseteq Y$ by hypothesis;
- (ii) $\bigcup_{i=0}^k \langle C \rangle^i X \subseteq \bigcup_{i=0}^k \langle C \rangle^i Y$ by inductive hypothesis;
- (iii) $\langle C \rangle \left(\bigcup_{i=0}^k \langle C \rangle^i X \right) \subseteq \langle C \rangle \left(\bigcup_{i=0}^k \langle C \rangle^i Y \right)$ by additivity.

Therefore

$$\bigcup_{i=0}^{k+1} \langle C \rangle^i X = X \cup \langle C \rangle \left(\bigcup_{i=0}^k \langle C \rangle^i X \right) \subseteq Y \cup \langle C \rangle \left(\bigcup_{i=0}^k \langle C \rangle^i Y \right) = \bigcup_{i=0}^{k+1} \langle C \rangle^i Y$$

□

Proposition 2.5. *Kleene star $\langle C^* \rangle$ and the fixpoint $\text{fix}(C)$ share the same concrete semantics:*

$$\langle C^* \rangle = \langle \text{fix}(C) \rangle$$

Proof. To start, let $X \in \mathcal{C}$, $f = \lambda Y \in \mathcal{C}. (X \cup \langle C \rangle Y)$ and recall that $f^0 X = X$ and $f^{n+1} X = X \cup \langle C \rangle (f^n X)$.

$$\begin{aligned} \langle \text{fix}(C) \rangle X &= \text{lfp}(f) = \bigcup \{f^n \perp \mid n \in \mathbb{N}\} && \text{by fixpoint theorem (1.13)} \\ &= \bigcup_{i \in \mathbb{N}} (X \cup \langle C \rangle^i X) && \text{by definition} \\ &= \bigcup_{i \in \mathbb{N}} \langle C \rangle^i X \\ &= \langle C^* \rangle X \end{aligned}$$

□

This will not be the case for the abstract semantics (cf. Example 3.25), where the Kleene star can be more precise than the fixpoint semantics, but harder to compute and, as such, less suited for analysis. For the concrete semantics, however, since they are the same in the next proofs we only explore the case C^* since it captures also $\text{fix}(C)$. Since for a given program C and a set of initial states $X \in \mathcal{C}$ the collecting semantics $\langle C \rangle X$ expresses properties that hold at the end of the execution of C we will in the following chapters usually refer to $\langle C \rangle X$ as program *invariant*.

Notation 2.6 (Singleton shorthand). Sometimes we need to consider the semantics over the singleton set $\{\rho\}$. In these cases we will write $\langle C \rangle \rho$ instead of $\langle C \rangle \{\rho\}$.

2.2.1 Syntactic sugar

We define some syntactic sugar for the language. In the next chapters we will often use the syntactic sugar instead of its real equivalent for the sake of simplicity.

$$\begin{aligned}
x \in [a, b] &= x \in I && \text{with } I = [a, b] \\
x \leq k &= x \in [-\infty, k] \\
x > k &= x \in [k + 1, +\infty] \\
\text{true} &= x \in \top \\
\text{false} &= x \in \perp \\
x \in I_1 \vee x \in I_2 &= (x \in I_1) + (x \in I_2) \\
x \in I_1 \wedge x \in I_2 &= (x \in I_1); (x \in I_2) \\
\text{if } b \text{ then } C_1 \text{ else } C_2 &= (e; C_1) + (\neg e; C_2) \\
\text{while } b \text{ do } C &= \text{fix}(b; C); \neg b \\
x++ &= x := x + 1
\end{aligned}$$

2.2.2 Small step semantics

Now that we have defined the collecting semantics to express program properties, we need the small step semantics to talk about program execution. We start by defining *program states*: $\text{State} \triangleq \text{Imp} \times \text{Env}$ tuples of programs and program environments. With states we can define our small step semantics:

Definition 2.7 (Small step semantics). The small step transition relation for the language Imp $\rightarrow: \text{State} \times (\text{State} \cup \text{Env})$ is defined by the following rules:

$$\begin{aligned}
&\frac{\langle e \rangle \rho \neq \perp}{\langle e, \rho \rangle \rightarrow \langle e \rangle \rho} \text{ expr} \\
&\frac{}{\langle C_1 + C_2, \rho \rangle \rightarrow \langle C_1, \rho \rangle} \text{ sum}_1 \quad \frac{}{\langle C_1 + C_2, \rho \rangle \rightarrow \langle C_2, \rho \rangle} \text{ sum}_2 \\
&\frac{\langle C_1, \rho \rangle \rightarrow \langle C'_1, \rho' \rangle}{\langle C_1; C_2, \rho \rangle \rightarrow \langle C'_1; C_2, \rho' \rangle} \text{ comp}_1 \quad \frac{\langle C_1, \rho \rangle \rightarrow \rho'}{\langle C_1; C_2, \rho \rangle \rightarrow \langle C_2, \rho' \rangle} \text{ comp}_2 \\
&\frac{}{\langle C^*, \rho \rangle \rightarrow \langle C; C^*, \rho \rangle} \text{ star} \quad \frac{}{\langle C^*, \rho \rangle \rightarrow \rho} \text{ star}_{\text{fix}}
\end{aligned}$$

In the following chapters we will usually use the following notation to talk about program execution:

- \rightarrow^+ is the transitive closure of the relation \rightarrow ;
- \rightarrow^* is the reflexive and transitive closure of the relation \rightarrow .

With the following lemma we introduce a link between the small step semantics and the concrete collecting semantics: the invariant of a program is the collection of all the environments the program halts on when executing.

Lemma 2.8. For any $C \in \text{Imp}$, $X \in \wp(\text{Env})$

$$\langle C \rangle X = \{\rho' \in \text{Env} \mid \rho \in X, \langle C, \rho \rangle \rightarrow^* \rho'\}$$

where \rightarrow^* is the reflexive and transitive closure of the \rightarrow relation.

Proof. by induction on C :

Case (e). In this case it holds that $\langle e \rangle X = \{\langle e \rangle \rho \mid \rho \in X \wedge \langle e \rangle \rho \neq \perp\}$, $\forall \rho \in X. \langle e, \rho \rangle \rightarrow \langle e \rangle \rho$ if $\langle e \rangle \rho \neq \perp$, and because of the expr rule

$$\langle e \rangle X = \{\langle e \rangle \rho \mid \rho \in X \wedge \langle e \rangle \rho \neq \perp\} = \{\rho' \in \text{Env} \mid \rho \in X \langle e, \rho \rangle \rightarrow \rho'\}$$

Case ($C_1 + C_2$). In this case $\langle C_1 + C_2 \rangle X = \langle C_1 \rangle X \cup \langle C_2 \rangle X$, $\forall \rho \in X. \langle C_1 + C_2, \rho \rangle \rightarrow \langle C_1, \rho \rangle \vee \langle C_1 + C_2, \rho \rangle \rightarrow \langle C_2, \rho \rangle$ respectively according to rules sum_1 and sum_2 . By inductive hypothesis

$$\langle C_1 \rangle X = \{\rho' \in \text{Env} \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho'\} \quad \langle C_2 \rangle X = \{\rho' \in \text{Env} \mid \rho \in X, \langle C_2, \rho \rangle \rightarrow^* \rho'\}$$

Therefore

$$\begin{aligned} \langle C_1 + C_2 \rangle X &= \langle C_1 \rangle X \cup \langle C_2 \rangle X && \text{(by definition)} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \langle C_1, \rho \rangle \rightarrow^* \rho'\} \cup \{\rho' \in \text{Env} \mid \rho \in X, \langle C_2, \rho \rangle \rightarrow^* \rho'\} && \text{(by ind. hp)} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \langle C_1, \rho \rangle \rightarrow^* \rho' \vee \langle C_2, \rho \rangle \rightarrow^* \rho'\} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \langle C_1 + C_2, \rho \rangle \rightarrow^* \rho'\} \end{aligned}$$

Case ($C_1; C_2$). In this case $\langle C_1; C_2 \rangle X = \langle C_2 \rangle (\langle C_1 \rangle X)$. By inductive hp $\langle C_1 \rangle X = \{\rho' \in \text{Env} \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho'\} = Y$, by inductive hp again $\langle C_2 \rangle Y = \{\rho' \in \text{Env} \mid \rho \in Y, \langle C_2, \rho \rangle \rightarrow^* \rho'\}$. Therefore

$$\begin{aligned} \langle C_1; C_2 \rangle X &= \langle C_2 \rangle (\langle C_1 \rangle X) && \text{(by definition)} \\ &= \{\rho' \in \text{Env} \mid \rho'' \in \{\rho''' \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho'''\}, \langle C_2, \rho'' \rangle \rightarrow^* \rho'\} && \text{(by ind. hp)} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \langle C_1, \rho \rangle \rightarrow^* \rho'' \wedge \langle C_2, \rho'' \rangle \rightarrow^* \rho'\} && \text{(by composition lemma)} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \langle C_1; C_2, \rho \rangle \rightarrow^* \rho'\} \end{aligned}$$

Case (C^*). Then, in this case $\langle C^* \rangle X = \bigcup_{i \in \mathbb{N}} \langle C \rangle^i X$

$$\begin{aligned} \langle C^* \rangle X &= \bigcup_{i \in \mathbb{N}} \langle C \rangle^i X \\ &= \bigcup_{i \in \mathbb{N}} \{\rho' \in \text{Env} \mid \rho \in X. \langle C^i, \rho \rangle \rightarrow^* \rho'\} && \text{by inductive hypothesis} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \forall i \in \mathbb{N} \langle C^i, \rho \rangle \rightarrow^* \rho'\} \\ &= \{\rho' \in \text{Env} \mid \rho \in X. \langle C^*, \rho \rangle \rightarrow^* \rho'\} \end{aligned}$$

□

Note that $\langle C \rangle X = \emptyset \iff \nexists \rho' \in \text{Env}, \rho \in X \mid \langle C, \rho \rangle \rightarrow^* \rho'$, in other words the collecting semantics of some program C starting from some states $X \in \mathcal{C}$ is empty iff the program never halts on some state ρ' . Another observation is that due to non-determinism a program can halt on multiple final states, or have one branch of execution that halts on some final state, while the other never halts on any final state. Non-determinism implies that there are two different types of termination, intuitively a program can *always* halt or *partially* halt. We will better explore this concept in the next chapter.

2.3 Transition system

With the set of states **State**, the set of environments **Env** and the small step operational semantics \rightarrow we define a transition system, this will be useful to define universal and partial termination and to reason about program properties in the next chapters.

Definition 2.9 (Transition system). The transition system for the language Imp is

$$\text{TS} \triangleq \langle \text{State} \cup \text{Env}, \text{Env}, \rightarrow \rangle$$

where

- $\text{State} \cup \text{Env}$ is the set of configurations in the system;
- Env is the set of terminal states;
- \rightarrow is the small step semantics defined in Definition 2.7, which describes the transition relations in the system.

With the concept of derivation sequences we can define what we mean for *partial* and *universal* termination.

Definition 2.10 (Partial termination). Let $C \in \text{Imp}, \rho \in \text{Env}$. We say C *partially halts* on ρ when there is at least one derivation sequence of finite length in the transition system starting with $\langle C, \rho \rangle$ and ending in some state ρ' :

$$\langle C, \rho \rangle \downarrow \iff \exists k \in \mathbb{N} \mid \langle C, \rho \rangle \rightarrow^k \rho'.$$

Dually

$$\langle C, \rho \rangle \uparrow \iff \neg \langle C, \rho \rangle \downarrow$$

a program *always loops* if there is no finite derivation sequence in its transition system that leads to a final environment.

Definition 2.11 (Universal termination). Let $C \in \text{Imp}, \rho \in \text{Env}$. We say C *partially loops* on ρ when there is at least one derivation sequence of infinite length in the transition system starting from $\langle C, \rho \rangle$:

$$\langle C, \rho \rangle \uparrow \iff \forall k \in \mathbb{N} \langle C, \rho \rangle \rightarrow^k \langle C', \rho' \rangle \quad \text{for some } C' \in \text{Imp}, \rho' \in \text{Env}.$$

Dually

$$\langle C, \rho \rangle \downarrow \iff \neg \langle C, \rho \rangle \uparrow$$

a program *universally halts* on ρ iff there is no infinite derivation sequence starting from $\langle C, \rho \rangle$ in the transition systems.

Example 2.14 shows a program that partially halts, while Example 2.13 shows a program that always loops. Notice that the absence of infinite derivation sequences implies that $\text{TS}(\langle C, \rho \rangle)$ is finite. Example 2.14 shows a program that partially loops, while example 2.12 shows a program that universally halts.

Example 2.12. Consider the program

$$x := 0;$$

it universally halts, since $\forall \rho \in \text{Env}, \rho \neq \perp$

$$\langle x := 0, \rho \rangle \rightarrow \rho[x \mapsto 0]$$

according to the expr rule in definition 2.7. Therefore $\langle (x := 0), \rho \rangle \downarrow \forall \rho \in \text{Env} \setminus \{\perp\}$.

Example 2.13. Consider the program P

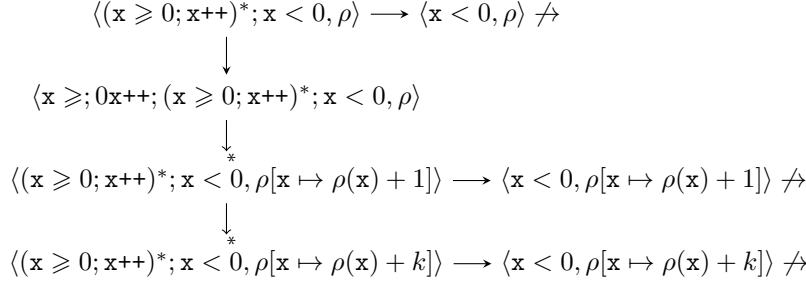
$$(x \geq 0; x++)^*; x < 0$$

The program never halts on $\forall \rho \in \text{Env}$ s.t. $\rho(x) \geq 0$. In fact in these cases it builds the transition system in figure 2.1, where the infinite derivation sequence

$$\langle (x \geq 0; x++)^*; x < 0, \rho \rangle \rightarrow^* \langle (x \geq 0; x++)^*; x < 0, \rho[x \mapsto \rho(x) + 1] \rangle \rightarrow^* \dots$$

$$\dots \rightarrow^* \langle (x \geq 0; x++)^*; x < 0, \rho[x \mapsto \rho(x) + k] \rangle \rightarrow^* \dots$$

is always present.

Figure 2.1: Transition system of $(x \geq 0; x++)^*; x < 0$

Example 2.14. Consider the program

$$(x++)^*$$

it partially halts $(\langle (x++)^*, \rho \rangle \downarrow)$, as according to the transition rule $\text{star}_{\text{fix}} \exists \rho \in \text{Env}$ s.t.

$$\frac{\rho \neq \perp}{\langle (x++)^*, \rho \rangle \rightarrow \rho} \text{star}_{\text{fix}}$$

But it also partially loops $(\langle (x++)^*, \rho \rangle \uparrow)$. In fact we can build the infinite derivation sequence

$$\langle (x++)^*, \rho[x \mapsto 0] \rangle \rightarrow^* \langle (x++)^*, \rho[x \mapsto 1] \rangle \rightarrow^* \langle (x++)^*, \rho[x \mapsto 2] \rangle \rightarrow^* \dots$$

Other useful lemmas in the system are the composition and decomposition lemma.

Lemma 2.15 (Decomposition lemma). *If $\langle C_1; C_2, \rho \rangle \rightarrow^k \rho''$, then there exists a state ρ' and a natural number k_1, k_2 s.t. $\langle C_1, \rho \rangle \rightarrow^{k_1} \rho'$ and $\langle C_2, \rho' \rangle \rightarrow^{k_2} \rho''$, where $k_1 + k_2 = k$*

Proof. The proof is on induction on $k \in \mathbb{N}$, i.e., by induction on the length of the derivation sequence.

Case ($k = 0$). Then

$$\langle C_1; C_2, \rho \rangle \rightarrow^0 \rho''$$

holds vacuously since $\langle C_1; C_2, \rho \rangle$ and ρ'' are different.

Case ($k \implies k + 1$). Then

$$\langle C_1; C_2, \rho \rangle \rightarrow^{k+1} \rho''$$

can be written as

$$\langle C_1; C_2, \rho \rangle \rightarrow \gamma \rightarrow^k \rho''$$

for some configuration γ . Now two cases apply, depending on the use of either comp_1 or comp_2 rules.

Case [comp_1]. Then $\gamma = \langle C'_1; C_2, \rho' \rangle$ and $\langle C_1; C_2, \rho \rangle \rightarrow \langle C'_1; C_2, \rho' \rangle$ because $\langle C_1, \rho \rangle \rightarrow \langle C'_1, \rho' \rangle$. Therefore we have

$$\langle C'_1; C_2, \rho' \rangle \rightarrow^k \rho''$$

Here we can use the induction hypothesis since the derivation sequence is shorter than the one we started with. Hence $\exists \rho''' \in \text{Env}$ and natural numbers k_1, k_2 s.t.

$$\langle C'_1; \rho' \rangle \rightarrow^{k_1} \rho''' \quad \wedge \quad \langle C_2, \rho''' \rangle \rightarrow^{k_2} \rho''$$

where $k_1 + k_2 = k$. Hence it holds that

$$\langle C_1, \rho \rangle \rightarrow^{k_1+1} \rho'''$$

and since $(k_1 + 1) + k_2 = k + 1$ it holds that

$$\langle C_1; C_2, \rho \rangle \rightarrow^{k+1} \rho''$$

which is our thesis.

Case [comp₂]. In this case $\gamma = \langle C_2, \rho' \rangle$ because $\langle C_1, \rho \rangle \rightarrow \rho'$ and it holds that

$$\langle C_1; C_2, \rho \rangle \rightarrow \langle C_2, \rho' \rangle \rightarrow^k \rho''$$

Hence our thesis follows by using the inductive hypothesis on $\langle C_2, \rho' \rangle$ and by choosing $k_1 = 1, k_2 = k$. \square

From the latter theorem follows its corollary, which abstracts the value of k .

Corollary 2.16. *If $\langle C_1; C_2, \rho \rangle \rightarrow^* \rho''$ then $\exists \rho'$ s.t. $\langle C_1, \rho \rangle \rightarrow^* \rho'$ and $\langle C_2, \rho' \rangle \rightarrow^* \rho''$.*

The second lemma states a similar but inverted property:

Lemma 2.17 (Composition lemma). *If $\langle C_1, \rho \rangle \rightarrow^k \rho'$ then $\langle C_1; C_2, \rho \rangle \rightarrow^k \langle C_2, \rho' \rangle$*

Proof. The proof works again by induction on the length k of the derivation sequence:

Case ($k = 0$). In this case the statement vacuously holds as $\langle C_1, \rho \rangle$ and ρ' are different.

Case ($k \implies k+1$). In this case we have to prove that if $\langle C_1, \rho \rangle \rightarrow^{k+1} \rho'$ then $\langle C_1; C_2, \rho \rangle \rightarrow^{k+1} \rho'$. To start we can notice that $\langle C_1, \rho \rangle \rightarrow^{k+1} \rho'$ means that we have 2 cases:

- (1) either $k = 0$, hence $\langle C_1, \rho \rangle \rightarrow \rho'$. But in this case we can use [comp₂] and deduce that

$$\frac{\langle C_1, \rho \rangle \rightarrow \rho'}{\langle C_1; C_2, \rho \rangle \rightarrow \langle C_2, \rho' \rangle}$$

- (2) Or $k > 0$, which means

$$\langle C_1, \rho \rangle \rightarrow \langle C'_1, \rho'' \rangle \rightarrow^k \rho'. \quad (2.1)$$

In this case we can use [comp₁] and notice that

$$\frac{\langle C_1, \rho \rangle \rightarrow \langle C'_1, \rho'' \rangle}{\langle C_1; C_2, \rho \rangle \rightarrow \langle C'_1; C_2, \rho'' \rangle}$$

Now, by induction on k in (2.1) we know that $\langle C'_1; C_2, \rho'' \rangle \rightarrow^k \rho'$, hence

$$\langle C_1; C_2, \rho \rangle \rightarrow \langle C'_1; C_2, \rho'' \rangle \rightarrow^k \rho'$$

which is our thesis. \square

Corollary 2.18. *If $\langle C_1, \rho \rangle \rightarrow^* \rho'$ then $\langle C_1; C_2, \rho \rangle \rightarrow^* \langle C_2, \rho' \rangle$.*

In order to better talk about the intermediate states in the execution of a program we also introduce the notion of reducts:

Definition 2.19 (Reducts). Let Imp^* denotes the set whose elements are statements in Imp . The reduction function $\text{red} : \text{Imp} \rightarrow \text{Imp}^*$ is recursively defined by the following clauses:

$$\begin{aligned} \text{red}(e) &\triangleq \{e\} \\ \text{red}(C_1 + C_2) &\triangleq \{C_1 + C_2\} \cup \text{red}(C_1) \cup \text{red}(C_2) \\ \text{red}(C_1; C_2) &\triangleq (\text{red}(C_1); C_2) \cup \text{red}(C_2) \\ \text{red}(C^*) &\triangleq \{C^*\} \cup (\text{red}(C); C^*) \end{aligned}$$

Where we overload the symbol $;$ with the operator $;$: $\text{Imp}^* \times \text{Imp} \rightarrow \text{Imp}^*$ defined by

$$\begin{aligned} \emptyset; C &\triangleq \emptyset \\ \{C_1, \dots, C_k\}; C &\triangleq \{C_1; C, \dots, C_k; C\} \end{aligned}$$

Notice that the set of reduction of any finite program $C \in \text{Imp}$ is finite.

2.4 Functions in Imp

Last section defined the language we are working with (the **Imp** language), its semantics and its transition system. Building upon those elements, we now present the first properties of the language. More in detail, in the following section we argue that the set of functions is at least a superset of the partially recursive functions described in [Cut80]. This way we can derive some properties from well known computability results, without proving them from scratch. We can do this by encoding partial recursive functions into Imp programs. We therefore start by better describing what we mean by partially recursive functions:

Definition 2.20 (Partially recursive functions). The class $\mathbb{N}^k \xrightarrow{r} \mathbb{N}$ of *partially recursive functions* is the least class of functions on the natural numbers which contains

- (a) the zero function:

$$\begin{aligned} z : \mathbb{N}^k &\rightarrow \mathbb{N} \\ (x_1, \dots, x_k) &\mapsto 0 \end{aligned}$$

- (b) the successor function

$$\begin{aligned} s : \mathbb{N} &\rightarrow \mathbb{N} \\ x_1 &\mapsto x_1 + 1 \end{aligned}$$

- (c) the projection function

$$\begin{aligned} U_i^k : \mathbb{N}^k &\rightarrow \mathbb{N} \\ (x_1, \dots, x_k) &\mapsto x_i \end{aligned}$$

and is closed under

- (1) composition: given a function $f : \mathbb{N}^k \xrightarrow{r} \mathbb{N}$ and functions $g_1, \dots, g_k : \mathbb{N}^n \xrightarrow{r} \mathbb{N}$ the *composition* $h : \mathbb{N}^n \xrightarrow{r} \mathbb{N}$ is defined by

$$h(\vec{x}) = \begin{cases} f(g_1(\vec{x}), \dots, g_k(\vec{x})) & \text{if } g_1(\vec{x}) \downarrow, \dots, g_k(\vec{x}) \downarrow \text{ and } f(g_1(\vec{x}), \dots, g_k(\vec{x})) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

- (2) primitive recursion: given $f : \mathbb{N}^k \xrightarrow{r} \mathbb{N}$ and $g : \mathbb{N}^{k+2} \xrightarrow{r} \mathbb{N}$ we define $h : \mathbb{N}^{k+1} \xrightarrow{r} \mathbb{N}$ by *primitive recursion* by

$$\begin{cases} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y + 1) &= g(\vec{x}, y, h(\vec{x}, y)) \end{cases}$$

- (3) minimalization: given $f : \mathbb{N}^{k+1} \xrightarrow{r} \mathbb{N}$, $h : \mathbb{N}^k \xrightarrow{r} \mathbb{N}$ defined through *unbounded minimalization* is

$$h(\vec{x}) = \mu y. f(\vec{x}, y) = \begin{cases} \text{least } z \text{ s.t.} & \begin{cases} f(\vec{x}, z) = 0 \\ f(\vec{x}, z) \downarrow \quad f(\vec{x}, z') \neq 0 \quad \forall z < z' \end{cases} \\ \uparrow & \text{otherwise} \end{cases}$$

We also need to define what it means providing (a_1, \dots, a_k) as input for an Imp program. We do this by special input states and variables: we can consider initial states $\rho = [\mathbf{x}_1 \mapsto a_1, \dots, \mathbf{x}_k \mapsto a_k]$ where each special variable \mathbf{x}_k maps to its initial value a_k , this way we can encode partial functions input into initial states for a program C. Observe that since we are interested in finite programs, it makes sense to consider only finite collections of free variables.

We also need to define what we mean by program output.

Notation 2.21 (Program output). Let $\text{Env} \ni \rho = [x_1 \mapsto a_1, \dots, x_n \mapsto a_n]$. We say

$$\begin{aligned} \langle C, \rho \rangle \Downarrow b &\iff \forall \rho' \mid \langle C, \rho \rangle \rightarrow^* \rho' \quad \rho'(y) = b \\ \langle C, \rho \rangle \downarrow b &\iff \exists \rho' \mid \langle C, \rho \rangle \rightarrow^* \rho' \quad \rho'(y) = b \end{aligned}$$

C universally (partially) halts on b whenever for every (for some) final state ρ $\rho(y) = b$. In other words we are using the special variable y as an output register.

Definition 2.22 (Imp computability). Let $f : \mathbb{N}^k \rightarrow \mathbb{N}$ be a function. We say that f is Imp computable if

$$\begin{aligned} \exists C \in \text{Imp} \mid \forall (a_1, \dots, a_k) \in \mathbb{N}^k \wedge b \in \mathbb{N} \\ \langle C, \rho \rangle \Downarrow b &\iff (a_1, \dots, a_k) \in \text{dom}(f) \wedge f(a_1, \dots, a_k) = b \end{aligned}$$

with $\rho = [x_1 \mapsto a_1, \dots, x_k \mapsto a_k]$.

We argue that the class of function computed by Imp is the same as the set of partially recursive functions $\mathbb{N} \xrightarrow{r} \mathbb{N}$ (as defined in [Cut80]). To do that we have to prove that the class of functions computed by the Imp language is a *rich*, i.e.

Definition 2.23 (Rich class). A class of functions \mathcal{A} is said to be rich if it includes (a), (b) and (c) and it is closed under (1), (2) and (3) from Definition 2.20.

Lemma 2.24 (Imp functions richness). *The class of Imp-computable function is rich.*

Proof. We proceed by proving that Imp has each and every one of the basic functions (zero, successor, projection).

- The zero function:

$$\begin{aligned} z : \mathbb{N}^k &\rightarrow \mathbb{N} \\ (x_1, \dots, x_k) &\mapsto 0 \end{aligned}$$

is Imp-computable:

$$z(a_1, \dots, a_k) \triangleq y := 0$$

- The successor function

$$\begin{aligned} s : \mathbb{N} &\rightarrow \mathbb{N} \\ x_1 &\mapsto x_1 + 1 \end{aligned}$$

is Imp-computable:

$$s(a_1) \triangleq y := x_1 + 1$$

- The projection function

$$\begin{aligned} U_i^k : \mathbb{N}^k &\rightarrow \mathbb{N} \\ (x_1, \dots, x_k) &\mapsto x_i \end{aligned}$$

is Imp-computable:

$$U_i^k(a_1, \dots, a_k) \triangleq y := x_i + 0$$

We then prove that it is closed under composition, primitive recursion and unbounded minimization.

Lemma 2.25. *let $f : \mathbb{N}^k \rightarrow \mathbb{N}$, $g_1, \dots, g_k : \mathbb{N}^n \rightarrow \mathbb{N}$ and consider the composition*

$$\begin{aligned} h : \mathbb{N}^k &\rightarrow \mathbb{N} \\ \vec{x} &\mapsto f(g_1(\vec{x}), \dots, g_k(\vec{x})) \end{aligned}$$

h is Imp-computable.

Proof. Since by hp $f, g_n \forall n \in [1, k]$ are computable, we consider their programs $F, G_n \forall n \in [1, k]$. Now consider the program

$$\begin{aligned} &G_1(\vec{x}); \\ &y_1 := y + 0; \\ &G_2(\vec{x}); \\ &y_2 := y + 0; \\ &\dots; \\ &G_k(\vec{x}); \\ &y_k := y + 0; \\ &F(y_1, y_2, \dots, y_k); \end{aligned}$$

Which is exactly h . Therefore Imp is closed under generalized composition. \square

Lemma 2.26. *Given $f : \mathbb{N}^k \rightarrow \mathbb{N}$ and $g : \mathbb{N}^{k+2} \rightarrow \mathbb{N}$ Imp computable, we argue that $h : \mathbb{N}^{k+1} \rightarrow \mathbb{N}$*

$$\begin{cases} h(\vec{x}, 0) = f(\vec{x}) \\ h(\vec{x}, y + 1) = g(\vec{x}, y, h(\vec{x}, y)) \end{cases}$$

defined through primitive recursion is Imp-computable.

Proof. We want a program to compute $h : \mathbb{N}^{k+1} \rightarrow \mathbb{N}$. By hypothesis we have programs F, G to compute respectively $f : \mathbb{N}^k \rightarrow \mathbb{N}$ and $g : \mathbb{N}^{k+2} \rightarrow \mathbb{N}$. Consider the program $H(\vec{x}, x_{k+1})$:

$$\begin{aligned} &s := 0; \\ &F(\vec{x}); \\ &(x_{k+1} \notin [0, 0]; G(\vec{x}, s, y); s := s + 1; x_{k+1} := x_{k+1} - 1)^*; \\ &x_{k+1} \in [0, 0]; \end{aligned}$$

which computes exactly h . Therefore Imp is closed under primitive recursion. \square

Lemma 2.27. *Let $f : \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ be a Imp-computable function. Then the function $h : \mathbb{N}^k \rightarrow \mathbb{N}$ defined through unbounded minimalization*

$$h(\vec{x}) = \mu y. f(\vec{x}, y) = \begin{cases} \text{least } z \text{ s.t.} & \begin{cases} f(\vec{x}, z) = 0 \\ f(\vec{x}, z) \downarrow & f(\vec{x}, z') \neq 0 \quad \forall z < z' \end{cases} \\ \uparrow & \text{otherwise} \end{cases} \quad (2.2)$$

is Imp-computable.

Proof. Let F be the program for the computable function f with arity $k + 1$, $\vec{x} = (x_1, x_2, \dots, x_k)$. Consider the program $H(\vec{x})$

$$\begin{aligned} &z := 0; \\ &F(\vec{x}, z); \\ &(y \notin [0, 0]; z := z + 1; F(\vec{x}, z))^*; \\ &y \in [0, 0]; \\ &y := z + 0; \end{aligned}$$

Which outputs the least z s.t. $F(\vec{x}, z) \downarrow 0$, and loops forever otherwise. Imp is therefore closed under bounded minimalization. \square

Since has the zero function, the successor function, the projections function and is closed under composition, primitive recursion and unbounded minimalization, the class of Imp-computable functions is rich. \square

Since it is rich and $\mathbb{N} \xrightarrow{r} \mathbb{N}$ is the least class of rich functions, $\mathbb{N} \xrightarrow{r} \mathbb{N} \subseteq \text{Imp}_f$ holds. Therefore we can say

$$f \in \mathbb{N}^k \xrightarrow{r} \mathbb{N} \implies \exists C \in \text{Imp} \mid \langle C, \rho \rangle \Downarrow b \iff f(a_1, \dots, a_k) \Downarrow b$$

with $\rho = [x_1 \mapsto a_1, \dots, x_k \mapsto a_k]$.

The final step is to recall the Rice theorem from [Ric53] and the definition of saturated sets:

Definition 2.28 (Saturated set). $A \subseteq \mathbb{N}$ is *saturated* (or *extensional*) is for all $x, y \in \mathbb{N}$

$$x \in A \wedge \varphi_x = \varphi_y \implies y \in A$$

In other words a set is saturated if it contains all the numbers that encode for programs that compute functions with some properties. Rice's theorem links extensional sets and decidability:

Theorem 2.29 (Rice's theorem). *Let $A \subseteq \mathbb{N}$, $A \neq \emptyset$, $A \neq \mathbb{N}$ be a saturated set. Then A is not recursive.*

Meaning that deciding whether a program is in some saturated set, i.e., the program has some extensional property, is impossible. From this we get a couple of facts that derive from well known computability results:

Corollary 2.30. $\langle C, \rho \rangle \Uparrow$ (i.e., $\langle C \rangle \rho = \emptyset$) is undecidable.

Proof. The set of functions $f \in \mathbb{N}^k \xrightarrow{r} \mathbb{N}$ s.t. $f(x) \Uparrow \forall x \in \mathbb{N}^k$ is not trivial and saturated, therefore it is not recursive by Rice's theorem [Ric53]. \square

Corollary 2.31. $\langle C, \rho \rangle \Downarrow$ is undecidable.

Proof. The set of functions $f \in \mathbb{N}^k \xrightarrow{r} \mathbb{N}$ s.t. $f(x) \Downarrow \forall x \in \mathbb{N}^k$ is not trivial and saturated, therefore it is not recursive by Rice's theorem [Ric53]. \square

2.5 Deciding invariant finiteness

In this section we argue that even the finiteness of the semantics of some program on some initial states is undecidable. We show that if we were able to establish whether $\langle C \rangle X$ is finite for some program $C \in \text{Imp}$ and some initial states $X \in \mathcal{C}$, we could decide whether $\langle C, \rho \rangle \Downarrow$ for all $\rho \in X$, which instead is known to be undecidable. The first step is showing that if we have a program where the $*$ operator does not appear, then the program can only produce a finite amount of finite derivation sequences.

Lemma 2.32. *If $D \in \text{Imp}_s$, and $X \in \wp(\text{Env})$ is finite, then*

- (i). $\langle D \rangle X$ is finite;
- (ii). $\forall \rho \in X \langle D, \rho \rangle \Downarrow$
- (iii). $|\text{TS}(\langle D, \rho \rangle)| < \infty$ for all $\rho \in X$.

where by $\text{TS}(\langle D, \rho \rangle)$ we mean the set of all derivation sequences starting from $\langle D, \rho \rangle$ in the transition system.

Proof. By induction on the program D :

Case ($D = e$). Let's prove each point in order.

- (i). $\langle e \rangle X = \{ \langle e \rangle \rho \mid \rho \in X, \langle e \rangle \rho \neq \perp \}$, which is finite, since X is finite;
- (ii). by expr rule $\forall \rho \in X$ either $\langle e, \rho \rangle \rightarrow \langle e \rangle \rho$ or $\langle e, \rho \rangle \not\rightarrow$. In both cases there are no infinite derivation sequences, and therefore $\text{TS}(\langle e, \rho \rangle) \Downarrow$;

(iii). Notice that $\forall \rho \in X$ either by the expr rule $\langle e, \rho \rangle \rightarrow \langle e \rangle \rho$ or $\langle e, \rho \rangle \not\rightarrow$ therefore

$$|\text{TS}(\langle e, \rho \rangle)| \leq |X| < \infty$$

Case ($D = D_1 + D_2$). In this case consider

- (i). $\langle D_1 + D_2 \rangle X = \langle D_1 \rangle X \cup \langle D_2 \rangle X$. By inductive hypothesis, both $\langle D_1 \rangle X, \langle D_2 \rangle X$ are finite, as they are sub expressions of D . Since the union of finite sets is finite, $\langle D_1 + D_2 \rangle X$ is finite;
- (ii). by inductive hypothesis again $\forall \rho \in X$ $\langle D_1, \rho \rangle \Downarrow$ and $\langle D_2, \rho \rangle \Downarrow$. By sum_1 rule $\langle D_1 + D_2, \rho \rangle \rightarrow \langle D_1, \rho \rangle$ and by sum_2 $\langle D_1 + D_2, \rho \rangle \rightarrow \langle D_2, \rho \rangle$. Therefore $\langle D_1 + D_2 \rangle \rho \Downarrow$.
- (iii). For the latter argument, since both $\langle D_1 \rangle \rho$ and $\langle D_2 \rangle \rho$ are finite and composed of finite derivation sequences $|\text{TS}(\langle D_1 + D_2, \rho \rangle)| < \infty$.

Case ($D = D_1; D_2$). We can notice the following:

- (i). $\langle D_1; D_2 \rangle X = \langle D_2 \rangle (\langle D_1 \rangle X)$. By inductive hypothesis $\langle D_1 \rangle X = Y$. By inductive hypothesis again $\langle D_2 \rangle Y$ is finite;
- (ii). by inductive hypothesis both $\forall \rho \in X$ $\langle D_1, \rho \rangle \Downarrow$ and $\forall \rho' \in Y$ $\langle D_2, \rho' \rangle \Downarrow$, therefore by composition lemma $\langle D_1; D_2, \rho \rangle \Downarrow$
- (iii). by inductive hypothesis both $|\text{TS}(\langle D_1, \rho \rangle)| < \infty$ and $|\text{TS}(\langle D_2, \rho' \rangle)| < \infty$ $\forall \rho \in X, \rho' \in \langle D_1 \rangle X$. For all derivation sequences starting from $\langle D_1, \rho \rangle$ where

$$\langle D_1, \rho \rangle \rightarrow^* \rho'$$

with $\rho' \in \langle D_1 \rangle X$ we can apply the composition lemma and state that

$$\langle D_1; D_2, \rho \rangle \rightarrow^* \langle D_2, \rho' \rangle \quad \forall \rho \in X$$

from there we can notice that since $|\langle D_2, \rho' \rangle| < \infty$ then $|\langle D_1; D_2, \rho' \rangle| < \infty$

□

In order to prove that finiteness is undecidable we need the following Lemma:

Lemma 2.33. *Let $D \in \text{Imp}_s$ and $\rho \in \text{Env}$. If*

$$\langle D \rangle^{k+1} \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho \quad \text{for some } k \in \mathbb{N} \quad (2.3)$$

then

$$\forall j \in \mathbb{N} \quad \langle D \rangle^{k+1+j} \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho \quad (2.4)$$

and therefore $\langle D^* \rangle \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho$

Proof. We can show (2.4) by induction on j :

Case ($j = 0$). In this case we want to show that $\langle D \rangle^{k+1} \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho$, which is true by hypothesis (2.3);

Case ($j \Rightarrow j + 1$). In this case we know that

$$\begin{aligned} \bigcup_{i=0}^k \langle D \rangle^i \rho &= \bigcup_{i=0}^{k+1} \langle D \rangle^i \rho && \text{since by (2.3) } \langle D \rangle^{k+1} \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho \\ &= \rho \cup \bigcup_{i=1}^{k+1} \langle D \rangle^i \rho \\ &= \rho \cup \langle D \rangle \left(\bigcup_{i=0}^k \langle D \rangle^i \rho \right) && \text{by additivity of } \langle D \rangle \end{aligned}$$

By inductive hypothesis

$$\langle D \rangle^{k+1+j} \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho$$

so, by monotonicity of $\langle D \rangle$

$$\langle D \rangle (\langle D \rangle^{k+1+j} \rho) \subseteq \langle D \rangle \left(\bigcup_{i=0}^k \langle D \rangle^i \rho \right)$$

and therefore

$$\langle D \rangle^{(k+1)+(j+1)} \rho \subseteq \left(\bigcup_{i=1}^{k+1} \langle D \rangle^i \rho \right) \subseteq \rho \cup \left(\bigcup_{i=1}^{k+1} \langle D \rangle^i \rho \right) = \bigcup_{i=0}^{k+1} \langle D \rangle^i \rho = \bigcup_{i=0}^k \langle D \rangle^i \rho$$

□

We also need to recall König's Lemma from [Kön26]:

Lemma 2.34 (König's Lemma). *Let T be a rooted tree with an infinite number of nodes, each with a finite number of children. Then T has a branch of infinite length.*

With Lemma 2.33 and Lemma 2.34 we can prove the following.

Lemma 2.35. *Given $D \in \text{Imp}_s$, and $\rho \in \text{Env}$, the predicate " $\langle D^* \rangle \rho$ is finite" is undecidable.*

Proof. We work by contradiction, showing that if we know whether $\langle C \rangle \rho$ is finite or infinite we can decide $\langle C, \rho \rangle \Downarrow$.

- Suppose that $\langle D^* \rangle \rho$ is infinite, then we can observe that because Lemma 2.33

$$\forall k \in \mathbb{N} \quad \langle D \rangle^{k+1} \rho \not\subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho \quad (2.5)$$

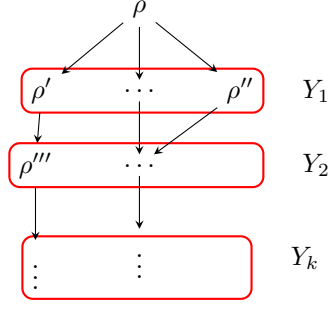
Otherwise $\langle D^* \rangle \rho \subseteq \bigcup_{i \in \mathbb{N}} \langle D \rangle^i \rho$ and we would contradict the hypothesis of $\langle D^* \rangle \rho$ being infinite. Therefore $\forall k \in \mathbb{N} \quad \langle D \rangle^{k+1} \rho \not\subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho$, otherwise $\langle D^* \rangle \rho \subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho$ which is impossible since the right term is a finite quantity. With this observation we build the tree $\langle \text{Env}, \rightarrow^D \rangle$, where $\rightarrow^D \subseteq \text{Env} \times \text{Env}$ and $\rho' \rightarrow^D \rho''$ if $\langle D, \rho' \rangle \rightarrow^* \rho''$. We define by the following rule the levels of the tree:

$$Y_0 = \{\rho\}$$

$$Y_{k+1} = (\langle D \rangle^{k+1} \rho) \setminus \left(\bigcup_{i=0}^k \langle D \rangle^i \rho \right)$$

Where Y_0 is the singleton set containing the root ρ and the k -th level is made of the environments in the Y_k set. Figure 2.2 shows a tree of \rightarrow^D relations and visualizes the levels Y_k . We can therefore make the following observations:

- The tree is rooted in $\rho \in Y_0$. In fact $\forall \rho' \in Y_1 \quad \rho \rightarrow^D \rho'$ by definition and $\forall \rho''' \in Y_{k+1} \exists \rho'' \in Y_k \mid \rho'' \rightarrow^D \rho'''$;
- since $\forall k \in \mathbb{N} \quad \langle D \rangle^{k+1} \rho \not\subseteq \bigcup_{i=0}^k \langle D \rangle^i \rho$ by (2.5), each level Y_k is non empty. Each level is also finite because of Lemma 2.32.(i). Therefore there is an infinite quantity of levels, where each node has a finite quantity of children.

Figure 2.2: Example of \rightarrow^D relations between elements of Env .

what is left to do is show that there is a derivation sequence from $\langle D^*, \rho \rangle$ of infinite length. We can do this by using König's Lemma 2.34 and deduce that there exists an infinite derivation sequence from ρ of \rightarrow^D relations

$$\rho \rightarrow^D \rho' \rightarrow^D \rho'' \rightarrow^D \dots$$

Where each element belongs to a different level Y_k , and therefore is different from every other environment appearing in the sequence. Observe that for all $\rho', \rho'' \in \text{Env}$ s.t. $\rho' \rightarrow^D \rho''$ since $\langle D, \rho' \rangle \rightarrow^* \rho''$ we can apply Corollary 2.18 of Lemma 2.17 and derive that $\langle D; D^*, \rho' \rangle \rightarrow^* \langle D^*, \rho'' \rangle$ and because of the star rule $\langle D^*, \rho' \rangle \rightarrow \langle D; D^*, \rho' \rangle$. We can therefore say that

$$\langle D^*, \rho' \rangle \rightarrow^* \langle D^*, \rho'' \rangle$$

Therefore, there exists an infinite derivation sequence

$$\langle D^*, \rho \rangle \rightarrow^* \langle D^*, \rho' \rangle \rightarrow^* \langle D^*, \rho'' \rangle \rightarrow^* \dots$$

which means $\langle D^*, \rho \rangle \uparrow$ and therefore $\langle D^*, \rho \rangle \Downarrow$ is false.

- if instead $\langle D^* \rangle \rho$ is finite, then we can reduce total termination to the presence of some cycle in one of the derivation sequences starting from $\langle D^*, \rho \rangle$. The statement we want to prove is the following:

if $\langle D^* \rangle \rho$ is finite, then $\langle D^*, \rho \rangle \Downarrow \iff$ no derivation sequence starting from $\langle D^*, \rho \rangle$ has cycles

- (\implies) In this case we want to prove that if $\langle D^* \rangle$ is finite and $\langle D, \rho \rangle \Downarrow$ then there are no cycles in any derivation sequence starting from $\langle D, \rho \rangle$. To do so we work by contradiction. Suppose there is some derivation sequence starting from $\langle D^*, \rho \rangle$ with some cycle

$$\langle D^*, \rho \rangle \rightarrow^* \langle D^*, \rho' \rangle \rightarrow^+ \langle D^*, \rho' \rangle \rightarrow^* \rho''$$

with $\rho'' \neq \rho, \rho'$, then we can notice that also the infinite derivation sequence

$$\langle D^*, \rho \rangle \rightarrow^* \langle D^*, \rho' \rangle \rightarrow^+ \langle D^*, \rho' \rangle \rightarrow^+ \langle D^*, \rho' \rangle \rightarrow^+ \dots$$

is part of the transition system for $\langle D, \rho \rangle$, and therefore $\langle D^*, \rho \rangle \Downarrow$ is false which is absurd.

- (\Leftarrow) In this case we want to prove that if $\langle D^* \rangle \rho$ is finite and there are no cycles in any derivation sequence starting from $\langle D, \rho \rangle$ then $\langle D, \rho \rangle \Downarrow$. We work again by contradiction. Suppose that we have an infinite derivation sequence starting from $\langle D^*, \rho \rangle$. It must be that $\forall i, j \in \mathbb{N} \ i \neq j, \rho_i \neq \rho_j$ with $\rho_0 = \rho$, otherwise there would be a cycle, which would contradict the hypothesis. Therefore the derivation sequence has the shape

$$\langle D^*, \rho \rangle \rightarrow^* \langle D^*, \rho_1 \rangle \rightarrow^* \langle D^*, \rho_2 \rangle \rightarrow^* \langle D^*, \rho_3 \rangle \rightarrow^* \dots$$

We can notice that for all $\rho' \in \{\rho, \rho_1, \dots\}$ and for the star_{fix} rule, $\langle D^*, \rho' \rangle \rightarrow \rho'$ and therefore $\rho' \in \langle D^* \rangle \rho$. This would mean that $\langle D^* \rangle \rho$ is infinite, which is absurd.

To conclude we can observe that there is a finite amount of reachable states from $\langle D^*, \rho \rangle$. Where by reachable we mean that there exists some derivation sequence ending up in that state.

We can notice that starting from any state $\langle D^*, \rho' \rangle$ with $\rho' \in \langle D^* \rangle \rho$ we have 2 possibilities:

- we either apply the star_{fix} rule, resulting in a finite derivation sequence

$$\langle D^*, \rho' \rangle \rightarrow \rho'$$

and therefore in a finite number of reached states;

- or we apply the star rule

$$\langle D^*, \rho' \rangle \rightarrow \langle D; D^*, \rho' \rangle$$

by lemma 2.32 we know that $\langle D, \rho' \rangle \Downarrow$ and $|\text{TS}(\langle D, \rho' \rangle)| < \infty$, therefore there is a finite number of environments ρ'' s.t. $\langle D, \rho' \rangle \rightarrow^* \rho''$. For each one of them we can use the composition lemma and observe that

$$\langle D; D^*, \rho' \rangle \rightarrow^* \langle D^*, \rho'' \rangle$$

Ending up in a state $\langle D^*, \rho'' \rangle$ where we can apply the same reasoning

Therefore starting from any state $\langle D^*, \rho' \rangle$ with $\rho' \in \langle D^* \rangle \rho$ (in particular ρ), we either terminate our derivation sequence or we end up in some state $\langle D^*, \rho' \rangle$ again, with $\rho' \in \langle D^* \rangle \rho$. Since there is a finite amount of states $\rho' \in \langle D^* \rangle \rho$, the number of reachable states from $\langle D^*, \rho \rangle$ is finite.

□

CHAPTER 3

Abstract domains

In the following chapters we present two domains that will play a relevant role in the sequel: the *interval* domain and the *non-relational* collecting domain. The two domains are in the class of *non-relational* domains, meaning that they do not represent the relation between variables. We are interested in these two domains as the properties that we will discuss in Chapter 4 will apply to these domains, with some restrictions that we will discuss later. In Section 3.3 we will talk about the interval domain, with its characterization and the domain properties in Section 3.3.2. Later, in Section 3.2 we will talk about the non-relational collecting abstraction by also showing some properties of the abstraction in Section 3.2.1.

3.1 Abstract inductive semantics

In order to talk about analysis over some abstract domain \mathbb{A} , we preliminarily introduce the definition of abstract semantics provided an abstract domain.

Definition 3.1 (Abstract inductive semantic). Given a domain \mathbb{A} and an abstract semantics $((\cdot))^{\mathbb{A}} : \text{Exp} \rightarrow \mathbb{A} \rightarrow \mathbb{A}$ for basic expressions, the *abstract inductive semantics* over \mathbb{A} is defined as the strict (i.e., preserving \perp) extension of the following function $\llbracket \cdot \rrbracket^{\mathbb{A}} : \text{Imp} \rightarrow \mathbb{A} \rightarrow \mathbb{A}$. For all $\eta \in \mathbb{A}$

$$\begin{aligned} \llbracket e \rrbracket^{\mathbb{A}} \eta &\triangleq ((e))^{\mathbb{A}} \eta \\ \llbracket C_1 + C_2 \rrbracket^{\mathbb{A}} \eta &\triangleq \llbracket C_1 \rrbracket^{\mathbb{A}} \eta \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}} \eta \\ \llbracket C_1 ; C_2 \rrbracket^{\mathbb{A}} \eta &\triangleq \llbracket C_2 \rrbracket^{\mathbb{A}} (\llbracket C_1 \rrbracket^{\mathbb{A}} \eta) \\ \llbracket C^* \rrbracket^{\mathbb{A}} \eta &\triangleq \bigsqcup_{i \in \mathbb{N}} (\llbracket C \rrbracket^{\mathbb{A}})^i (\eta) \\ \llbracket \text{fix}(C) \rrbracket^{\mathbb{A}} \eta &\triangleq \text{lfp}(\lambda \mu. (\eta \sqcup \llbracket C \rrbracket^{\mathbb{A}} \mu)) \end{aligned}$$

From this definition we can observe that soundness is preserved from the base cases, i.e., if we have two domains \mathbb{A} and \mathbb{A}^{\sharp} s.t. $\mathbb{A} \xleftrightarrow[\alpha]{\gamma} \mathbb{A}^{\sharp}$ according to some abstraction and concretization maps α and γ then the analysis over \mathbb{A}^{\sharp} is *sound* w.r.t. the analysis performed over \mathbb{A} , provided that the base cases are sound.

Theorem 3.2 (Abstraction soundness). *Let $C \in \text{Imp}$ and $\langle \mathbb{A}, \sqsubseteq \rangle, \langle \mathbb{A}^{\sharp}, \sqsubseteq^{\sharp} \rangle$ be two domains equipped with their partial order s.t. $\mathbb{A} \xleftrightarrow[\alpha]{\gamma} \mathbb{A}^{\sharp}$ for some abstraction and concretization maps α, γ such that*

$$\forall e \in \text{Exp} \quad ((e))^{\mathbb{A}} \circ \gamma \eta^{\sharp} \sqsubseteq (\gamma \circ ((e))^{\mathbb{A}^{\sharp}}) \eta^{\sharp}$$

i.e., the analysis over the base cases are sound. Then for all $\eta^{\sharp} \in \mathbb{A}^{\sharp}$:

$$(\llbracket C \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq (\gamma \circ \llbracket C \rrbracket^{\mathbb{A}^{\sharp}}) \eta^{\sharp}$$

i.e., the abstract analysis over \mathbb{A}^{\sharp} is sound w.r.t. the analysis over \mathbb{A} .

Proof. The proof will proceed again by induction on C .

Case (e). In this case by hypothesis it holds that

$$(\llbracket e \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} = (((e)) \circ \gamma) \eta^{\sharp} \sqsubseteq (\gamma \circ ((e))^{\mathbb{A}^{\sharp}}) \eta^{\sharp} = (\gamma \circ \llbracket e \rrbracket^{\mathbb{A}^{\sharp}}) \eta^{\sharp}$$

Which is exactly our thesis.

Case $(C_1 + C_2)$. In this case by inductive hypothesis we know that both the following hold:

$$(\llbracket C_1 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq \left(\gamma \circ \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp} \quad (3.1)$$

$$(\llbracket C_2 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq \left(\gamma \circ \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp} \quad (3.2)$$

and $\llbracket C_1 + C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} = \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp}$. What we have to prove is that

$$(\llbracket C_1 + C_2 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq \left(\gamma \circ \llbracket C_1 + C_2 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp}$$

or, equivalently

$$\begin{aligned} \llbracket C_1 + C_2 \rrbracket^{\mathbb{A}} (\gamma \eta^{\sharp}) &\sqsubseteq \gamma \left(\llbracket C_1 + C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \right) \\ \llbracket C_1 \rrbracket^{\mathbb{A}} (\gamma \eta^{\sharp}) \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}} (\gamma \eta^{\sharp}) &\sqsubseteq \gamma \left(\llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \right) \end{aligned}$$

Now we can notice that by Property 4 of Galois connections

$$\gamma \left(\llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \right) = \bigsqcup \left\{ \rho \in \mathbb{A} \mid \alpha(\rho) \sqsubseteq \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \right\} \quad (3.3)$$

Now

$$\begin{aligned} (\alpha \circ \llbracket C_1 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} &\sqsubseteq \left(\alpha \circ \gamma \circ \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp} && \text{by monotonicity of } \alpha \text{ in (3.1)} \\ &\sqsubseteq \left(\llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp} && \text{by reductivity of } \alpha \end{aligned}$$

and the same applies for (3.2). Hence because of (3.3) we can observe that

$$\begin{aligned} \llbracket C_1 \rrbracket^{\mathbb{A}} (\gamma \eta^{\sharp}) \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}} (\gamma \eta^{\sharp}) &\sqsubseteq \bigsqcup \left\{ \rho \in \mathbb{A} \mid \alpha(\rho) \sqsubseteq \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \right\} \\ &= \gamma \left(\llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \sqcup \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \eta^{\sharp} \right) \end{aligned}$$

which is our thesis.

Case $(C_1; C_2)$. In this case we have to prove that

$$(\llbracket C_1; C_2 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq \left(\gamma \circ \llbracket C_1; C_2 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp}$$

or equivalently

$$(\llbracket C_2 \rrbracket^{\mathbb{A}} \circ \llbracket C_1 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq \left(\gamma \circ \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \circ \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp}$$

Now we can notice that by inductive hypothesis $\llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}}$ and $\llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}}$ are sound abstractions of respectively $\llbracket C_1 \rrbracket^{\mathbb{A}}$ and $\llbracket C_2 \rrbracket^{\mathbb{A}}$, hence we have the hypothesis to apply Theorem 1.25 and deduce that

$$(\llbracket C_2 \rrbracket^{\mathbb{A}} \circ \llbracket C_1 \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq \left(\gamma \circ \llbracket C_2 \rrbracket^{\mathbb{A}^{\sharp}} \circ \llbracket C_1 \rrbracket^{\mathbb{A}^{\sharp}} \right) \eta^{\sharp}$$

which is our thesis.

Case $(\text{fix}(\mathbf{C}))$. In this case we have to prove that

$$(\llbracket \text{fix}(\mathbf{C}) \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} \sqsubseteq (\gamma \circ \llbracket \text{fix}(\mathbf{C}) \rrbracket^{\mathbb{A}^{\sharp}}) \eta^{\sharp}$$

we know by definition and Fixpoint Theorem 1.13 that

$$\begin{aligned} (\gamma \circ \llbracket \text{fix}(\mathbf{C}) \rrbracket^{\mathbb{A}^{\sharp}}) \eta^{\sharp} &= \gamma(\text{lfp}(\lambda \mu. \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} \mu)) = \gamma\left(\bigsqcup \left\{ (\lambda \mu. \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} \mu)^n \perp^{\sharp} \mid n \in \mathbb{N} \right\}\right) \\ (\llbracket \text{fix}(\mathbf{C}) \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} &= \text{lfp}(\lambda \mu. \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} \mu) = \bigsqcup \left\{ (\lambda \mu. \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} \mu)^n \perp \mid n \in \mathbb{N} \right\} \end{aligned}$$

We can now prove by induction on n that the two are in a \sqsubseteq relation. We do this by first proving that for all $k \in \mathbb{N}$

$$f^k \perp \sqsubseteq \gamma((f^{\sharp})^k \perp^{\sharp})$$

by induction on k .

Case $(n = 0)$. In this case

$$\begin{aligned} \gamma\left(\left(\lambda \mu. \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} \mu\right)^0 \perp^{\sharp}\right) &= \gamma(\perp^{\sharp}) \\ (\lambda \mu. \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} \mu)^0 \perp &= \perp \end{aligned}$$

hence $\perp \sqsubseteq \gamma(\perp^{\sharp})$ and our thesis holds.

Case $(n \implies n + 1)$. Let's call $f^{\sharp} = \lambda \mu. \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} \mu$ and $f = \lambda \mu. \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} \mu$. First we can observe that

$$\begin{aligned} f^{n+1} \perp &= f(f^n \perp) = \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} (f^n \perp) \\ f^{\sharp n+1} \perp^{\sharp} &= f^{\sharp}(f^{\sharp n} \perp^{\sharp}) = \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} (f^{\sharp n} \perp^{\sharp}) \end{aligned}$$

From this we can observe that

$$\begin{aligned} \gamma(f^{\sharp}(f^{\sharp n} \perp^{\sharp})) &= \gamma(\eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} (f^{\sharp n} \perp^{\sharp})) \\ &= \bigsqcup \left\{ c \in \mathbb{A} \mid \alpha(c) \sqsubseteq \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} (f^{\sharp n} \perp^{\sharp}) \right\} && \text{by Property 4 of Galois connections.} \\ &\sqsupseteq \gamma(\eta^{\sharp}) \sqcup \gamma(\llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} (f^{\sharp n} \perp^{\sharp})) && \text{by } (\alpha \circ \gamma) \text{ reductivity} \\ &\sqsupseteq \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} \gamma(f^{\sharp n} \perp^{\sharp}) && \text{by induction on } \mathbf{C} \\ &\sqsupseteq \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} (f^n \perp) && \text{by induction on } n \text{ and monotonicity of } \llbracket \cdot \rrbracket^{\mathbb{A}} \end{aligned}$$

which is our thesis.

Hence $f^n \perp \sqsubseteq \gamma(f^{\sharp n} \perp^{\sharp})$, for all $n \in \mathbb{N}$, therefore

$$\begin{aligned} (\llbracket \text{fix}(\mathbf{C}) \rrbracket^{\mathbb{A}} \circ \gamma) \eta^{\sharp} &= \bigsqcup \left\{ (\lambda \mu. \gamma(\eta^{\sharp}) \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}} \mu)^n \perp \mid n \in \mathbb{N} \right\} \\ &\sqsubseteq \gamma\left(\bigsqcup \left\{ (\lambda \mu. \eta^{\sharp} \sqcup \llbracket \mathbf{C} \rrbracket^{\mathbb{A}^{\sharp}} \mu)^n \perp^{\sharp} \mid n \in \mathbb{N} \right\}\right) \\ &= (\gamma \circ \llbracket \text{fix}(\mathbf{C}) \rrbracket^{\mathbb{A}^{\sharp}}) \eta^{\sharp} \end{aligned}$$

Which is our thesis. □

We can also observe that best correct approximations (from Definition 1.24) induce sound semantics:

Corollary 3.3 (bca induces soundness). *Let $\langle \mathbb{A}, \sqsubseteq \rangle, \langle \mathbb{A}^\sharp, \sqsubseteq^\sharp \rangle$ be two abstract domains equipped with their respective orders s.t. $\mathbb{A} \xrightarrow[\alpha]{\gamma} \mathbb{A}^\sharp$. If for base expressions $((\cdot))^{\mathbb{A}^\sharp} : \text{Exp} \rightarrow \mathbb{A}^\sharp \rightarrow \mathbb{A}^\sharp$ is defined as*

$$((e))^{\mathbb{A}^\sharp} \eta^\sharp \triangleq \alpha \left(((e))^{\mathbb{A}} \gamma(\eta^\sharp) \right)$$

Then

$$(\llbracket C \rrbracket^{\mathbb{A}} \circ \gamma) \eta^\sharp \sqsubseteq (\gamma \circ \llbracket C \rrbracket^{\mathbb{A}^\sharp}) \eta^\sharp$$

i.e., the abstract inductive semantics over \mathbb{A}^\sharp is sound w.r.t. the abstract inductive semantics over \mathbb{A} .

Proof. The proof follows from Theorem 3.2, where we have to prove that for base cases the two semantics are sound. In this case we know that $((e))^{\mathbb{A}^\sharp} \eta^\sharp = \alpha((e))^{\mathbb{A}} \gamma(\eta^\sharp)$ and we have to prove that

$$((e))^{\mathbb{A}} \circ \gamma \eta^\sharp \sqsubseteq (\gamma \circ ((e))^{\mathbb{A}^\sharp}) \eta^\sharp$$

equivalently, we can substitute and notice that we can prove that

$$((e))^{\mathbb{A}} (\gamma(\eta^\sharp)) \sqsubseteq \gamma(\alpha((e))^{\mathbb{A}^\sharp} \gamma(\eta^\sharp)).$$

Here we can call $\rho = ((e))^{\mathbb{A}} (\gamma(\eta^\sharp))$ and notice that by extensivity $\forall \rho \in \mathbb{A}$

$$\begin{aligned} \rho &\sqsubseteq (\gamma \circ \alpha) \rho \\ ((e))^{\mathbb{A}} (\gamma(\eta^\sharp)) &\sqsubseteq (\gamma \circ \alpha) \left(((e))^{\mathbb{A}} (\gamma(\eta^\sharp)) \right) \end{aligned}$$

which is our thesis. □

Lemma 3.4 (fix(C) is syntactic sugar). *For all $\eta \in \mathbb{A}$, $\llbracket \text{fix}(C) \rrbracket^{\mathbb{A}} \eta = \llbracket (\text{true} + C)^* \rrbracket^{\mathbb{A}} \eta$.*

Proof. First, let

$$\begin{aligned} f &= \lambda \mu. (\eta \sqcup \mu \sqcup \llbracket C \rrbracket^{\mathbb{A}} \mu) \\ g &= \lambda \mu. (\mu \sqcup \llbracket C \rrbracket^{\mathbb{A}} \mu) \end{aligned}$$

Let us first show by induction that

$$\forall i \geq 0 \quad f^{i+1}(\perp) = g^i(\eta) \tag{3.4}$$

Case ($i = 0$). $f(\perp) = \eta \sqcup \perp \sqcup \llbracket C \rrbracket^{\mathbb{A}} \perp = \eta = g^0(\eta)$.

Case ($i + 1$).

$$\begin{aligned} g^{i+1}(\eta) &= g(g^i(\eta)) \\ &= g(f^{i+1}(\perp)) && \text{By induction on } i \\ &= f^{i+1}(\perp) \sqcup \llbracket C \rrbracket^{\mathbb{A}} f^{i+1}(\perp) \\ &= \eta \sqcup f^{i+1}(\perp) \sqcup \llbracket C \rrbracket^{\mathbb{A}} f^{i+1}(\perp) && \text{Since } \eta \sqsubseteq f(\perp) \\ &= f(f^{i+1}(\perp)) \\ &= f^{i+2}(\perp) \end{aligned}$$

Let us also show that:

$$\text{lfp}(g) = \text{lfp}(f) \tag{3.5}$$

Observe that $\text{lfp}(g) = g(\text{lfp}(g)) = \eta \sqcup \llbracket C \rrbracket^{\mathbb{A}} (\text{lfp}(g))$, so we have that:

$$\eta \sqcup \text{lfp}(g) \sqcup \llbracket C \rrbracket^{\mathbb{A}} (\text{lfp}(g)) \sqsubseteq \text{lfp}(g)$$

As a consequence, $\text{lfp}(f) \sqsubseteq \text{lfp}(g)$ holds. The reverse inequality follows because, for all μ ,

$$g(\mu) = \eta \sqcup \llbracket \mathbb{C} \rrbracket^{\mathbb{A}} \mu \sqsubseteq \eta \sqcup \mu \sqcup \llbracket \mathbb{C} \rrbracket^{\mathbb{A}} \mu = f(\mu)$$

Then, we have that:

$$\begin{aligned} \llbracket \text{fix}(\mathbb{C}) \rrbracket^{\mathbb{A}} \eta &= \text{lfp}(\lambda \mu. (\eta \sqcup \llbracket \mathbb{C} \rrbracket^{\mathbb{A}} \mu)) \\ &= \text{lfp}(\lambda \mu. (\eta \sqcup \mu \sqcup \llbracket \mathbb{C} \rrbracket^{\mathbb{A}} \mu)) && \text{By (3.5)} \\ &= \bigsqcup_{i \in \mathbb{N}} f^i \perp && \text{By Knaster-Tarski Theorem} \\ &= \perp \sqcup \bigsqcup_{i \in \mathbb{N}} f^{i+1} \perp \\ &= \bigsqcup_{i \in \mathbb{N}} g^i \eta && \text{By (3.4)} \\ &= \llbracket (\text{true} + \mathbb{C})^* \rrbracket^{\mathbb{A}} \eta \end{aligned}$$

□

3.2 Non relational collecting

We now introduce the non-relational collecting domain. Conceptually, this domain associates each variable with a subset of \mathbb{Z} . The processes of abstraction and concretization, from and to the powerset of environments $\wp(\text{Env})$, involve environments where variables take values within their respective mappings in the abstract domain.

We preliminarily define the sum of a constant to a subset of \mathbb{Z} , the result will be another subset of \mathbb{Z} where all values have been increased by some $k \in \mathbb{Z}$.

Definition 3.5 (Set addition and subtraction). For a nonempty set $S \in \wp(\mathbb{Z})$ and $c \in \mathbb{N}$ define $S \pm c \triangleq \{x \pm c \mid x \in S\}$ (recall that $\pm\infty + c = \pm\infty - c = \pm\infty$).

We can therefore proceed by introducing the variable-wise lifting of the domain $\wp(\mathbb{Z})$, building the abstract domain \mathbb{C} :

Definition 3.6 (Abstract Non relational collecting domain). Let $\wp^*(\mathbb{Z}) = \wp(\mathbb{Z}) \setminus \{\emptyset\}$. The abstract domain \mathbb{C} for program analysis is the variable-wise lifting of $\wp(\mathbb{Z})$:

$$\mathbb{C} \triangleq (Var \rightarrow \wp^*(\mathbb{Z})) \cup \{\perp\}$$

In this domain, we define again abstraction and concretization maps, building a Galois connection with the concrete domain

Definition 3.7. We define the *concretization map* of abstract environments $\eta \in \mathbb{C}$, i.e., $\gamma : \mathbb{C} \rightarrow \wp(\text{Env})$ as follows

$$\begin{aligned} \gamma(\perp) &\triangleq \emptyset \\ \gamma(\eta) &\triangleq \{\rho \in \text{Env} \mid \forall \mathbf{x} \in Var \quad \rho(\mathbf{x}) \in \eta \mathbf{x}\} \end{aligned}$$

and the *abstraction map* of sets of concrete environments $X \in \wp(\text{Env})$, i.e., $\alpha : \wp(\text{Env}) \rightarrow \mathbb{C}$ as

$$\begin{aligned} \alpha(\emptyset) &\triangleq \perp \\ \alpha(X) &\triangleq \lambda \mathbf{x} . \{\rho(\mathbf{x}) \mid \rho \in X\} \end{aligned}$$

We can again define a notion of order for elements of \mathbb{C} based on the concretization map. Let $\eta, \vartheta \in \mathbb{C}$, then

$$\eta \dot{\sqsubseteq} \vartheta \text{ iff } \gamma(\eta) \subseteq \gamma(\vartheta)$$

Notice that because of the definition of the concretization map

$$\eta \dot{\subseteq} \vartheta \iff \forall \mathbf{x} \in \text{Var } \eta(\mathbf{x}) \dot{\subseteq} \vartheta(\mathbf{x})$$

we can notice that $\langle \mathbb{C}, \dot{\subseteq} \rangle$ is a complete lattice, as for every two elements $\eta, \vartheta \in \mathbb{C}$ there exists both $\eta \dot{\cup} \vartheta$ and $\eta \dot{\cap} \vartheta$ by characterizing least upper bounds and greatest lower bounds as the lifting of \sqcup and \sqcap operations. Let again $\eta, \vartheta \in \mathbb{C}$, then

$$\begin{aligned} \eta \dot{\cap} \vartheta &= \sigma & \text{if } \sigma(\mathbf{x}) &= \eta(\mathbf{x}) \cap \vartheta(\mathbf{x}) \quad \forall \mathbf{x} \in \text{Var} \\ \eta \dot{\cup} \vartheta &= \sigma & \text{if } \sigma(\mathbf{x}) &= \eta(\mathbf{x}) \cup \vartheta(\mathbf{x}) \quad \forall \mathbf{x} \in \text{Var} \end{aligned}$$

Again With these premises we can now define base operations on the non-relational collecting abstraction:

Definition 3.8 (Base expressions on non-relational collecting). Let $\eta \in \mathbb{C}$ and γ from Definition 3.14. The base expressions semantics $((\cdot))^{\mathbb{C}} : \text{Exp} \rightarrow \mathbb{C} \rightarrow \mathbb{C}$ is recursively defined as

$$\begin{aligned} ((\mathbf{x} \in I))^{\mathbb{C}} \eta &\triangleq \begin{cases} \eta[\mathbf{x} \mapsto \eta\mathbf{x} \cap \gamma(I)] & \text{if } \eta\mathbf{x} \cap \gamma(I) \neq \emptyset \\ \perp & \text{otherwise} \end{cases} \\ ((\mathbf{x} := k))^{\mathbb{C}} \eta &\triangleq \eta[\mathbf{x} \mapsto \{k\}] \\ ((\mathbf{x} := \mathbf{y} + k))^{\mathbb{C}} \eta &\triangleq \eta[\mathbf{x} \mapsto \eta\mathbf{y} + k] \end{aligned}$$

With these base operations, $[[\cdot]]^{\mathbb{C}}$ is defined accordingly to Definition 3.1.

Thanks to α and γ maps of Definition 3.7 there is a Galois connection between the concrete domain \mathcal{C} and the abstract domain \mathbb{C} .

Lemma 3.9 (\mathbb{C} abstracts \mathcal{C}). Let α, γ be the abstraction and concretization maps from Definition 3.7, then $\langle \alpha, \mathcal{C}, \mathbb{C}, \gamma \rangle$ is a Galois connection.

Proof. We use the characterization of Theorem 1.17, which means proving

- (i) α, γ are monotonic;
- (ii) $\gamma \circ \alpha$ is extensive;
- (iii) $\alpha \circ \gamma$ is reductive.

Lets start by proving (i). Let $\eta, \vartheta \in \mathbb{C}$ s.t. $\eta \dot{\subseteq} \vartheta$. This means that $\forall \mathbf{x} \in \text{Var } \eta\mathbf{x} \subseteq \vartheta\mathbf{x}$ and therefore $a \in \eta\mathbf{x} \implies a \in \vartheta\mathbf{x}$. By definition of γ this means that $\gamma(\eta) \subseteq \gamma(\vartheta)$. Let now $R, S \in \wp(\text{Env})$ s.t. $\rho \subseteq \sigma$, which means that $\rho \in R \implies \rho \in S$. By definition of α it follows that $\eta \in \alpha(R) \implies \eta \in \alpha(S)$ which means exactly $\alpha(R) \dot{\subseteq} \alpha(S)$ which is our thesis.

Let us now show (ii). What we want to prove is that $\forall R \in \wp(\text{Env})$ it holds that $R \subseteq \gamma(\alpha(R))$. If $R = \emptyset \implies \alpha(\emptyset) = \perp$ and therefore $\gamma(\perp) = \emptyset$ and therefore the statement holds. Otherwise $R \neq \emptyset \implies \alpha(R) = \lambda \mathbf{x}. \{\rho(\mathbf{x}) \mid \rho \in R\} = \eta$. Now, $\gamma(\eta) = \{\rho \in \text{Env} \mid \forall \mathbf{x} \in \text{Var } \rho(\mathbf{x}) \in \eta\mathbf{x}\}$, this means that $\sigma \in R \implies \sigma \in \gamma(\alpha(R))$, which is our thesis $R \subseteq \gamma(\alpha(R))$.

We are left to show (iii). Let $\eta \in \mathbb{C}$, we have 2 cases. Either $\eta = \perp$, which means $\alpha(\gamma(\perp)) = \alpha(\emptyset) = \perp \dot{\subseteq} \perp$, hence the statement holds. Otherwise $\gamma(\eta) = \{\rho \in \text{Env} \mid \forall \mathbf{x} \in \text{Var } \rho\mathbf{x} \in \eta\mathbf{x}\} = R$ and $\alpha(R) = \lambda \mathbf{x}. \{\rho\mathbf{x} \mid \rho \in R\} = \eta$, hence $\alpha(\gamma(\eta)) = \eta$, which implies our thesis $\alpha \circ \gamma$ is reductive, and therefore $\langle \alpha, \mathcal{C}, \mathbb{C}, \gamma \rangle$ is a Galois connection. \square

Notice that in particular $\alpha \circ \gamma = \text{id}$, and therefore there is a Galois insertion $\langle \mathcal{C}, \subseteq \rangle \xleftarrow[\alpha]{\gamma} \langle \mathbb{C}, \dot{\subseteq} \rangle$ between the concrete domain \mathcal{C} and the abstract domain \mathbb{C} .

Moreover the abstract semantics $\langle\langle \cdot \rangle\rangle$ is sound w.r.t. the concrete semantics $\langle \cdot \rangle$:

Lemma 3.10 (Non-relational collecting soundness). Let $\mathbf{C} \in \text{Imp}, \eta \in \mathbb{C}$

$$\langle \mathbf{C} \rangle \gamma(\eta) \subseteq \gamma(\langle\langle \mathbf{C} \rangle\rangle \eta)$$

i.e., Non-relational collecting inductive abstract semantics is sound w.r.t. concrete semantics.

Proof. Since there is a Galois connection between abstract inductive semantics (notice that $\langle \cdot \rangle = \llbracket \cdot \rrbracket^C$), We just need to show that the two semantics are sound on base operations.

Case $(x \in I)$. In this case we need to show that

$$\langle x \in I \rangle \gamma(\eta) \subseteq \gamma(\langle\langle x \in I \rangle\rangle \eta)$$

recall that

$$\begin{aligned} \langle x \in I \rangle \gamma(\eta) &= \{ \langle x \in I \rangle \rho \mid \rho \in \gamma(\eta) \ \langle x \in I \rangle \rho \neq \perp \} \\ \gamma(\langle\langle x \in I \rangle\rangle \eta) &= \gamma(\eta[x \mapsto \eta x \cap \gamma_{\mathbb{I}}(I)]) \end{aligned}$$

to conclude notice that

$$\begin{aligned} \gamma(\eta[x \mapsto \eta x \cap \gamma_{\mathbb{I}}(I)]) &= \{ \rho \in \mathbf{Env} \mid \forall x \in \mathbf{Var} \ \rho x \in \eta x \} \\ &\supseteq \{ \langle x \in I \rangle \rho \mid \rho \in \gamma(\eta) \ \langle x \in I \rangle \rho \neq \perp \} \end{aligned}$$

which is our thesis.

Case $(x := k)$. In this case we have to prove that

$$\langle x := k \rangle \gamma(\eta) \subseteq \gamma(\langle\langle x := k \rangle\rangle \eta)$$

to do so just notice that

$$\begin{aligned} \langle x := k \rangle \gamma(\eta) &= \{ \rho[x \mapsto k] \mid \rho \in \gamma(\eta) \} \\ &\subseteq \{ \rho \in \mathbf{Env} \mid \forall y \in \mathbf{Var} \ \rho y \in \eta[x \mapsto \{k\}]y \} \\ &= \gamma(\eta[x \mapsto \{k\}]) \end{aligned}$$

which is our thesis.

Case $(x := y + k)$. In this case similarly to the latter case we can just notice that

$$\begin{aligned} \langle x := y + k \rangle \gamma(\eta) &= \{ \rho[x \mapsto \rho y + k] \mid \rho \in \gamma(\eta) \} \\ &\subseteq \{ \rho \in \mathbf{Env} \mid \forall w \in \mathbf{Var} \ \rho w \in \eta[x \mapsto \eta y + k]w \} \\ &= \gamma(\eta[x \mapsto y + k]) \end{aligned}$$

Since we are sound on the base cases, by Theorem 3.2

$$\langle C \rangle \gamma(\eta) \subseteq \gamma(\langle\langle C \rangle\rangle \eta)$$

□

3.2.1 Properties

The non-relational collecting semantics is similar to the interval semantics we defined in Section 3.3, in the sense that both of them do not model relation between variables. Differing from interval analysis however, non-relational collecting semantics gains back additivity, which we lost with interval semantics. With additivity we can infer that $\text{fix}(C)$ and C^* have the same semantics (with Proposition 3.12).

Let's denote as $\langle\langle \cdot \rangle\rangle$ the abstract semantics over \mathbb{C} , i.e., $\langle\langle \cdot \rangle\rangle = \llbracket \cdot \rrbracket^C$.

Lemma 3.11 (Additivity). *Let $\eta, \vartheta \in \mathbb{C}$, $C \in \text{Imp}$ then*

$$\langle\langle C \rangle\rangle (\eta \dot{\cup} \vartheta) = (\langle\langle C \rangle\rangle \eta) \dot{\cup} (\langle\langle C \rangle\rangle \vartheta)$$

Proof. We can work by induction on C :

Case $(x \in S)$. Then

$$\begin{aligned} \langle\langle x \in S \rangle\rangle(\eta \dot{\cup} \vartheta) &= (\eta \dot{\cup} \vartheta)[x \mapsto (\eta \dot{\cup} \vartheta)x \cap S] \\ &= (\eta \dot{\cup} \vartheta)[x \mapsto (\eta x \cap S) \cup (\vartheta x \cap S)] \\ &= (\eta[x \mapsto (\eta x \cap S)]) \dot{\cup} (\vartheta[x \mapsto \vartheta x \cap S]) \\ &= \langle\langle x \in S \rangle\rangle \eta \dot{\cup} \langle\langle x \in S \rangle\rangle \vartheta \end{aligned}$$

Case $(x := k)$. Then

$$\begin{aligned} \langle\langle x := k \rangle\rangle(\eta \dot{\cup} \vartheta) &= (\eta \dot{\cup} \vartheta)[x \mapsto \{k\}] \\ &= (\eta[x \mapsto \{k\}]) \dot{\cup} (\vartheta[x \mapsto \{k\}]) \\ &= \langle\langle x := k \rangle\rangle \eta \dot{\cup} \langle\langle x := k \rangle\rangle \vartheta \end{aligned}$$

Case $(x := y + k)$. Then

$$\begin{aligned} \langle\langle x := y + k \rangle\rangle(\eta \dot{\cup} \vartheta) &= (\eta \dot{\cup} \vartheta)[x \mapsto y + k] \\ &= (\eta[x \mapsto y + k]) \dot{\cup} (\vartheta[x \mapsto y + k]) \\ &= \langle\langle x := y + k \rangle\rangle \eta \dot{\cup} \langle\langle x := y + k \rangle\rangle \vartheta \end{aligned}$$

Case $(C_1 + C_2)$. Then

$$\begin{aligned} \langle\langle C_1 + C_2 \rangle\rangle(\eta \dot{\cup} \sigma) &= \langle\langle C_1 \rangle\rangle(\eta \dot{\cup} \sigma) \dot{\cup} \langle\langle C_2 \rangle\rangle(\eta \dot{\cup} \sigma) && \text{by definition} \\ &= \langle\langle C_1 \rangle\rangle \eta \dot{\cup} \langle\langle C_1 \rangle\rangle \sigma \dot{\cup} \langle\langle C_2 \rangle\rangle \eta \dot{\cup} \langle\langle C_2 \rangle\rangle \sigma && \text{by inductive hypothesis} \\ &= \langle\langle C_1 + C_2 \rangle\rangle \eta \dot{\cup} \langle\langle C_1 + C_2 \rangle\rangle \sigma \end{aligned}$$

Case $(C_1; C_2)$. Then

$$\begin{aligned} \langle\langle C_1; C_2 \rangle\rangle(\eta \dot{\cup} \sigma) &= \langle\langle C_2 \rangle\rangle(\langle\langle C_1 \rangle\rangle(\eta \dot{\cup} \sigma)) \\ &= \langle\langle C_2 \rangle\rangle \left(\langle\langle C_1 \rangle\rangle \eta \dot{\cup} \langle\langle C_1 \rangle\rangle \sigma \right) && \text{by inductive hypothesis} \\ &= \langle\langle C_2 \rangle\rangle(\langle\langle C_1 \rangle\rangle \eta) \dot{\cup} \langle\langle C_2 \rangle\rangle(\langle\langle C_1 \rangle\rangle \sigma) && \text{by inductive hypothesis} \end{aligned}$$

Case (C^*) . Then

$$\langle\langle C^* \rangle\rangle(\eta \dot{\cup} \vartheta) = \bigcup_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i(\eta \dot{\cup} \vartheta)$$

What we have to show now is that $\forall i \in \mathbb{N} \langle\langle C \rangle\rangle^i(\eta \dot{\cup} \vartheta) = \langle\langle C \rangle\rangle^i \eta \dot{\cup} \langle\langle C \rangle\rangle^i \vartheta$. We can show this by induction on i :

Case $(i = 0)$. Then

$$\langle\langle C \rangle\rangle^0(\eta \dot{\cup} \vartheta) = \eta \dot{\cup} \vartheta = \langle\langle C \rangle\rangle^0 \eta \dot{\cup} \langle\langle C \rangle\rangle^0 \vartheta$$

and the statement holds.

Case $(i \implies i + 1)$. Notice that

$$\begin{aligned} \langle\langle C \rangle\rangle^{i+1}(\eta \dot{\cup} \vartheta) &= \langle\langle C \rangle\rangle \left(\langle\langle C \rangle\rangle^i(\eta \dot{\cup} \vartheta) \right) \\ &= \langle\langle C \rangle\rangle(\langle\langle C \rangle\rangle^i \eta \dot{\cup} \langle\langle C \rangle\rangle^i \vartheta) && \text{by inductive hypothesis} \\ &= \langle\langle C \rangle\rangle^{i+1} \eta \dot{\cup} \langle\langle C \rangle\rangle^{i+1} \vartheta && \text{by additivity} \end{aligned}$$

Therefore

$$\begin{aligned}
\langle\langle C^* \rangle\rangle(\eta \dot{\cup} \vartheta) &= \dot{\bigcup}_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i(\eta \dot{\cup} \vartheta) \\
&= \dot{\bigcup}_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i(\eta \dot{\cup} \vartheta) \\
&= \dot{\bigcup}_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i \eta \dot{\cup} \langle\langle C \rangle\rangle^i \vartheta \\
&= \left(\dot{\bigcup}_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i \eta \right) \dot{\cup} \left(\dot{\bigcup}_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i \vartheta \right) \\
&= \langle\langle C^* \rangle\rangle \eta \dot{\cup} \langle\langle C^* \rangle\rangle \vartheta
\end{aligned}$$

□

Proposition 3.12. $\text{fix}(C)$ and C^* semantics coincide:

$$\langle\langle \text{fix}(C) \rangle\rangle = \langle\langle C^* \rangle\rangle$$

Proof. Let $f = \lambda\mu.(\eta \dot{\cup} \langle\langle C \rangle\rangle\mu)$

$$\begin{aligned}
\langle\langle \text{fix}(C) \rangle\rangle \eta &= \text{lfp}(f) \\
&= \dot{\bigcup}_{i \in \mathbb{N}} \{f^n \perp \mid n \in \mathbb{N}\} && \text{by fixpoint Theorem 1.13} \\
&= \dot{\bigcup}_{i \in \mathbb{N}} \langle\langle C \rangle\rangle^i \eta \\
&= \langle\langle C^* \rangle\rangle \eta
\end{aligned}$$

□

3.3 Interval domain

Interval analysis are among the most well known standard abstract domains in abstract interpretation. They are generally studied as simple non-relational domains, as intervals are not able to capture the relation between variables occurring in the program. The following chapter aims at proving that interval analysis is decidable without a widening operator, i.e., despite the presence of infinite ascending chains the exact value of the analysis can be computed.

We first define what the set of intervals \mathbb{I} is and its abstraction and concretization map to the powerset of integers.

Definition 3.13 (Integer intervals). We call

$$\mathbb{I} \triangleq \{[a, b] \mid a \in \mathbb{Z} \cup \{-\infty\} \wedge b \in \mathbb{Z} \cup \{+\infty\} \wedge a \leq b\} \cup \{\perp\}$$

the set of integer intervals. In the rest of the thesis we will write \top instead of $[-\infty, +\infty]$

In order to later do the variable-wise lifting of the intervals domain and relate it to the concrete environment \mathcal{C} we need to define concretization and abstraction maps for the intervals domain

Definition 3.14. We define the *concretization map* $\gamma : \mathbb{I} \rightarrow \wp(\mathbb{Z})$ as

$$\begin{aligned}
\gamma([a, b]) &\triangleq \{x \in \mathbb{Z} \mid a \leq x \leq b\} \\
\gamma(\perp) &\triangleq \emptyset
\end{aligned}$$

And the *abstraction map* $\alpha : \wp(\mathbb{Z}) \rightarrow \mathbb{I}$ as

$$\alpha(S) \triangleq \begin{cases} \perp & \text{if } S = \emptyset \\ [\inf(S), \sup(S)] & \text{otherwise} \end{cases}$$

The next step is to define some order on \mathbb{I} . For this purpose we define a partial order \sqsubseteq based on the concretization map.

Definition 3.15 (Partial order on \mathbb{I}). Let $I, J \in \mathbb{I}$. Then

$$I \sqsubseteq J \iff \gamma(I) \subseteq \gamma(J)$$

Observe that $\langle \mathbb{I}, \sqsubseteq \rangle$ is a complete lattice. We next characterize least upper bound and greatest lower bound on the domain \mathbb{I} . Let $[a, b], [c, d] \in \mathbb{I}$

$$\begin{aligned} [a, b] \sqcup [c, d] &\triangleq [\min(a, c), \max(b, d)] \\ [a, b] \sqcap [c, d] &\triangleq \begin{cases} [\max(a, c), \min(b, d)] & \text{if } \min(b, d) < \max(a, c) \\ \perp & \text{otherwise} \end{cases} \end{aligned}$$

The generalization to infinite sets is obtained in the obvious way, by replacing \min and \max with \inf and \sup . The next building block is the definition of some more operations on intervals, namely the addition and subtraction of an integer constant:

Definition 3.16 (Interval addition and subtraction). For a nonempty interval $[a, b] \in \mathbb{I}$ and $c \in \mathbb{N}$ define $[a, b] \pm c \triangleq [a \pm c, b \pm c]$ (recall that conventionally $\pm\infty + c = \pm\infty - c = \pm\infty$).

Notice that γ and α maps are concretization and abstraction maps to and from $\wp(\mathbb{Z})$. We can therefore notice that there is a Galois connection between $\wp(\mathbb{Z})$ and \mathbb{I} .

Lemma 3.17 (\mathbb{I} abstracts $\wp(\mathbb{Z})$). Let α, γ be abstraction and concretization maps from Definition 3.14. Then $\langle \alpha, \wp(\mathbb{Z}), \mathbb{I}, \gamma \rangle$ is a Galois connection.

Proof. We once again rely on the characterization of Galois connections given by Theorem 1.17:

- (i) α and γ are monotonic: let $\iota, \kappa \in \dot{\mathbb{I}}$ s.t. $\iota \sqsubseteq \kappa \implies \gamma(\iota) \subseteq \gamma(\kappa)$ while if we let $R, S \in \mathbb{C}$ s.t. $R \subseteq S \implies \alpha(R) \sqsubseteq \alpha(S)$
- (ii) $\gamma \circ \alpha$ is extensive. Notice that $\forall R \in \mathbb{C}$ it holds that $R \sqsubseteq \{x \in \mathbb{Z} \mid \inf(R) \leq x \leq \sup(R)\} = \gamma(\alpha(R))$, which is our thesis.
- (iii) $\alpha \circ \gamma$ is reductive. Notice that $\forall \iota \in \dot{\mathbb{I}}$ it holds that $\alpha(\gamma(\iota)) = \iota$, which implies that it is reductive.

Therefore $\langle \alpha, \wp(\mathbb{Z}), \mathbb{I}, \gamma \rangle$ is a Galois connection. □

Moreover, since $\gamma \circ \alpha = \text{id}$ we can deduce that $\langle \wp(\mathbb{Z}), \dot{\sqsubseteq} \xrightarrow[\alpha]{\gamma} \langle \mathbb{I}, \sqsubseteq \rangle$ is a Galois insertion

3.3.1 Variable-wise lifting

We can therefore proceed to introduce the variable-wise lifting of the \mathbb{I} domain, building the abstract domain $\dot{\mathbb{I}}$.

Definition 3.18 (Abstract integer domain). Let $\mathbb{I}_* \triangleq \mathbb{I} \setminus \{\perp\}$. The abstract domain $\dot{\mathbb{I}}$ for program analysis is the variable-wise lifting of \mathbb{I} :

$$\dot{\mathbb{I}} \triangleq (\text{Var} \rightarrow \mathbb{I}_*) \cup \{\perp\}$$

In this domain, we define again abstraction and concretization maps, building a Galois connection with the concrete domain. We do so by overloading the α and γ functions, to refer also to the abstraction and concretization of abstract environments.

Definition 3.19 (Concretization and abstraction). We define the concretization map $\dot{\gamma} : \dot{\mathbb{I}} \rightarrow \mathbb{C}$ and the abstraction map $\dot{\alpha} : \mathbb{C} \rightarrow \dot{\mathbb{I}}$ as the point wise lifting of γ, α from Definition 3.14:

$$\begin{aligned} \dot{\gamma}(\eta) &\triangleq \lambda \mathbf{x}. \gamma(\eta \mathbf{x}) \\ \dot{\alpha}(\rho) &\triangleq \lambda \mathbf{x}. \alpha(\rho \mathbf{x}) \end{aligned}$$

for all $\eta \in \dot{\mathbb{I}}, \rho \in \mathbb{C}$.

We can again define a notion of order for elements of $\dot{\mathbb{I}}$ based on the concretization map. We do by overloading the \sqsubseteq notation. Let $\eta, \vartheta \in \dot{\mathbb{I}}$, then

$$\eta \sqsubseteq \vartheta \text{ iff } \dot{\gamma}(\eta) \dot{\subseteq} \dot{\gamma}(\vartheta)$$

Notice that because of $\dot{\gamma}$ definition

$$\eta \sqsubseteq \vartheta \iff \forall \mathbf{x} \in \text{Var} \quad \eta(\mathbf{x}) \sqsubseteq \vartheta(\mathbf{x})$$

i.e., two abstract environments are ordered if every variable's interval of the first environment is contained in the interval of the second abstract environment. Also, the least upper bounds and greatest lower bounds are obtained by lifting the \sqcup and \sqcap operations, i.e., let $\eta, \vartheta \in \dot{\mathbb{I}}$, then

$$\begin{aligned} \eta \sqcap \vartheta &= \sigma & \text{if } \sigma(\mathbf{x}) &= \eta(\mathbf{x}) \sqcap \vartheta(\mathbf{x}) \quad \forall \mathbf{x} \in \text{Var} \\ \eta \sqcup \vartheta &= \sigma & \text{if } \sigma(\mathbf{x}) &= \eta(\mathbf{x}) \sqcup \vartheta(\mathbf{x}) \quad \forall \mathbf{x} \in \text{Var} \end{aligned}$$

Again we can notice that $\langle \dot{\mathbb{I}}, \sqsubseteq \rangle$ is a complete lattice, as for every two elements $\eta, \vartheta \in \dot{\mathbb{I}}$ there exists both $\eta \sqcup \vartheta$ and $\eta \sqcap \vartheta$. With these premises we can define our abstract inductive semantics on intervals, by defining the base operations $((\cdot))^{\dot{\mathbb{I}}} : \text{Exp} \rightarrow \dot{\mathbb{I}} \rightarrow \dot{\mathbb{I}}$

Definition 3.20 (Base expressions on intervals). Let $\eta \in \dot{\mathbb{I}}$ then the base expressions semantics $((\cdot))^{\dot{\mathbb{I}}} : \text{Exp} \rightarrow \dot{\mathbb{I}} \rightarrow \dot{\mathbb{I}}$ is recursively defined as

$$\begin{aligned} ((\mathbf{x} \in I))^{\dot{\mathbb{I}}} \eta &\triangleq \begin{cases} \eta[\mathbf{x} \mapsto \eta \mathbf{x} \sqcap I] & \text{if } \eta \mathbf{x} \sqcap I \neq \perp \\ \perp & \text{otherwise} \end{cases} \\ ((\mathbf{x} := k))^{\dot{\mathbb{I}}} \eta &\triangleq \eta[\mathbf{x} \mapsto [k, k]] \\ ((\mathbf{x} := \mathbf{y} + k))^{\dot{\mathbb{I}}} \eta &\triangleq \eta[\mathbf{x} \mapsto \eta \mathbf{y} + k] \end{aligned}$$

With these base operations, $\llbracket \cdot \rrbracket^{\dot{\mathbb{I}}}$ is defined accordingly to Definition 3.1. The next point is to prove that the interval semantics $\llbracket \cdot \rrbracket^{\dot{\mathbb{I}}}$ is sound w.r.t. the concrete semantics $\langle \cdot \rangle$.

Observation 3.21. An alternative characterization of $((\cdot))^{\dot{\mathbb{I}}}$ views it as the b.c.a. of base expressions on \mathbb{C} :

$$((\mathbf{e}))^{\dot{\mathbb{I}}} = \dot{\alpha} \circ ((\mathbf{e}))^{\mathbb{C}} \circ \dot{\gamma}$$

Lemma 3.22 ($\dot{\mathbb{I}}$ abstracts \mathbb{C}). Let $\dot{\alpha}, \dot{\gamma}$ be the abstraction and concretization maps of Definition 3.19, then $\langle \dot{\alpha}, \mathbb{C}, \dot{\mathbb{I}}, \dot{\gamma} \rangle$ is a Galois connection.

Proof. By Lemma 3.17 $\langle \wp(\mathbb{Z}), \subseteq \rangle \xleftarrow[\alpha]{\gamma} \langle \mathbb{I}, \sqsubseteq \rangle$ and by Theorem 1.22 $\langle \mathbb{C}, \dot{\subseteq} \rangle \xleftarrow[\dot{\alpha}]{\dot{\gamma}} \langle \dot{\mathbb{I}}, \sqsubseteq \rangle$ \square

By latter Lemma and Observation 3.21 we can deduce the following

Lemma 3.23 (intervals domain abstracts non-relational collecting semantics).

$$\langle \langle \mathbb{C} \rangle \dot{\gamma}(\eta) \dot{\subseteq} \dot{\gamma}(\llbracket \mathbb{C} \rrbracket^{\dot{\mathbb{I}}} \eta)$$

Proof. This follows from the fact that base expressions on intervals are characterized by the best correct approximation of base expressions on \mathbb{C} and from Corollary 3.3 b.c.a. induces soundness. \square

Observation 3.24. Analysis on the intervals domain is sound to the concrete semantics. For all $\eta \in \dot{\mathbb{I}}$

$$\langle \mathbb{C} \rangle \gamma(\dot{\gamma}(\eta)) \subseteq \gamma(\langle \langle \mathbb{C} \rangle \dot{\gamma}(\eta) \rangle) \subseteq \gamma(\dot{\gamma}(\llbracket \mathbb{C} \rrbracket^{\dot{\mathbb{I}}} \eta))$$

Proof. The second inequality $\gamma(\langle \langle \mathbb{C} \rangle \dot{\gamma}(\eta) \rangle) \subseteq \gamma(\dot{\gamma}(\llbracket \mathbb{C} \rrbracket^{\dot{\mathbb{I}}} \eta))$ comes from Lemma 3.23 and monotonicity of γ , while the first one is from Lemma 3.10 \square

3.3.2 Properties

We can immediately see how in the abstract interval domain, the semantics of the Kleene star and the fixpoint operator is not the same. This intuitively happens because the Kleene star is the least upper bound of a chain of intervals, while the fix operator keeps iterating over least upper bounds.

Example 3.25. In the case exposed in Code 3.1, for instance, the following program P represents the difference between the Kleene Star and the Fix operator:

```

1  while x < 8 do
2    /* P1: */
3    if x = 2
4      x := x+6;
5    endif;
6    x := x-3;
7    if x <= 0
8      x:=0;
9    endif;
10  done;
11
```

Code 3.1: Program P denoting $\text{fix}(C)$ and C^* difference

starting with the finite interval $[3, 4]$ we get the following loop invariants depending on the way we model the overall program P :

$$\begin{aligned} \llbracket P_1^* \rrbracket^i &= \sqcup \{[3, 4], [0, 1], [0, 0], [0, 0], \dots\} = [0, 4] \\ \llbracket \text{fix}(P_1) \rrbracket^i &= \sqcup \{\perp, [3, 4], [0, 4], [0, 5], [0, 5], \dots\} = [0, 5] \end{aligned}$$

Both invariants are correct, because they over-approximate the most precise concrete invariant $\{0, 1, 3, 4\}$, however the Kleene invariant is strictly more precise than the Fix one.

Moreover notice that we lose the additivity property, which was instead part of the analysis in \mathbb{C} . Consider in fact the following example

Example 3.26. Let $\eta = [x \mapsto [1, 1]]$, $\vartheta = [x \mapsto [3, 3]]$ and the program $x \in [2, +\infty]$. We will show that

$$\llbracket x \in [2, +\infty] \rrbracket^i (\eta \sqcup \vartheta) \neq \llbracket x \in [2, +\infty] \rrbracket^i \eta \sqcup \llbracket x \in [2, +\infty] \rrbracket^i \vartheta$$

Notice in fact that

$$\begin{aligned} \llbracket x \in [2, +\infty] \rrbracket^i (\eta \sqcup \vartheta) &= \llbracket x \in [2, +\infty] \rrbracket^i ([x \mapsto [1, 3]]) \\ &= [x \mapsto [2, 3]] \end{aligned}$$

while

$$\begin{aligned} \llbracket x \in [2, +\infty] \rrbracket^i \eta &= \perp \\ \llbracket x \in [2, +\infty] \rrbracket^i \vartheta &= \vartheta \end{aligned}$$

and therefore $\perp \sqcup \vartheta = \vartheta \neq [x \mapsto [2, 3]]$, which was our thesis.

CHAPTER 4

Program bounds and analysis termination

In the last chapter we defined two abstract domains: the interval domain and the non-relational collecting domain. Both are non-relational domains, in the sense that they do not model the relation between variables, but only their possible individual values. Through the abstract inductive semantics of Definition 3.1 we defined a simple abstract interpreter for the `Imp` language, provided base expressions and an abstract domain \mathbb{A} . In Chapter 2 we proved that the `Imp` language is Turing-complete and therefore by Rice's theorem all non-trivial semantic properties of programs written in such language are not decidable. Exact invariants fall into this category, as they tie the initial state of a machine and a program to the output the program would return.

With this context, we can formalize the problem we will solve:

Problem 4.1 (Exact analysis computation). Given a program $C \in \text{Imp}$ and an abstract domain \mathbb{A} , for all $\eta \in \mathbb{A}$, compute

$$\llbracket C \rrbracket^{\mathbb{A}} \eta$$

In this chapter we argue that for the language `Imp` the abstract semantics is computable for the domain \mathbb{I} , while for the domain \mathbb{C} we will show that a further abstraction can be computed exactly. We will also show that the further abstraction coincides with the analysis over \mathbb{C} when the analysis terminates. Indeed, our proof shows that we can decide the termination of the analyser.

Observe that the exact computation of the abstract invariant provides, already for our simple language, a precision which is not obtainable with (basic) widening and narrowing. In the example in Code 4.1 if we consider the abstract domain \mathbb{I} , the semantics maps `x` and `y` to $[0, 2]$ and $[6, 8]$ respectively, while widening/narrowing to $[0, +\infty]$ and $[6, +\infty]$.

```
1  x:=0;
2  y:=0;
3  while (x<=5) do
4      if (y=0) then
5          y=y+1;
6      endif;
7      if (x=0) then
8          x:=y+7;
9      endif;
10 done;
```

Code 4.1: Code sample where analysis of $\text{fix}(C)$ is less precise than C^*

Of course, for the collecting semantics this is not the case. Already computing a finite upper bound for loop invariants when they are finite is impossible as this would allow to decide termination, as we have seen in Section 2.5.

The main idea, based on previous research is to *bound* the domain \mathbb{A} . Each program is associated to a bound, an ideal value above which for each variable we can safely assume that the program diverges. First, given a program, we associate the program with a *lower bound* and an *upper bound*. The rough idea is that, whenever a variable is beyond its bound, the behavior of the program with respect to that variable becomes stable.

4.1 Program bounds

We need to define the max and min function for intervals.

Definition 4.2. (max and min on intervals) Let $\mathcal{Z} \triangleq \mathbb{Z} \cup \{-\infty, +\infty\}$, then $\max : \mathbb{I} \rightarrow \mathcal{Z}$ is defined as

$$\begin{aligned}\max(\perp) &\triangleq -\infty \\ \max([a, b]) &\triangleq b\end{aligned}$$

while the min function $\min : \mathbb{I} \rightarrow \mathcal{Z}$ is defined as follows

$$\begin{aligned}\min(\perp) &\triangleq +\infty \\ \min([a, b]) &\triangleq a\end{aligned}$$

Notice in particular that since $\top = [-\infty, +\infty]$, $\max(\top) = +\infty$, $\min(\top) = -\infty$.

Definition 4.3 (Program bound). The *upper bound* associated with a command $C \in \text{Imp}$ is an integer number, denoted $(C)^b \in \mathbb{N}$, defined inductively as follows:

$$\begin{aligned}(\mathbf{x} \in I)^b &\triangleq \begin{cases} |\max(I)| & \text{if } \max(I) \in \mathbb{Z} \\ |\min(I)| & \text{if } \max(I) = +\infty \wedge \min(I) \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \\ (\mathbf{x} := k)^b &\triangleq |k| \\ (\mathbf{x} := \mathbf{y} + k)^b &\triangleq |k| \\ (C_1 + C_2)^b &\triangleq (C_1)^b + (C_2)^b \\ (C_1; C_2)^b &\triangleq (C_1)^b + (C_2)^b \\ (\text{fix}(C))^b &\triangleq (|\text{vars}(C)| + 1)(C)^b\end{aligned}$$

while the *lower bound* associated with a command $C \in \text{Imp}$ is again an integer number, denoted $(C)_b \in \mathbb{N}$, defined inductively as follows:

$$\begin{aligned}(\mathbf{x} \in I)_b &\triangleq \begin{cases} |\min(I)| & \text{if } \min(I) \in \mathbb{Z} \\ |\max(I)| & \text{if } \min(I) = -\infty \wedge \max(I) \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \\ (\mathbf{x} := k)_b &\triangleq |k| \\ (\mathbf{x} := \mathbf{y} + k)_b &\triangleq |k| \\ (C_1 + C_2)_b &\triangleq (C_1)_b + (C_2)_b \\ (C_1; C_2)_b &\triangleq (C_1)_b + (C_2)_b \\ (\text{fix}(C))_b &\triangleq (|\text{vars}(C)| + 1)(C)_b\end{aligned}$$

where $\text{vars}(C)$ denotes the set of variables occurring in C .

We can notice that the two definitions of the bound $(C)^b$ and $(C)_b$ coincide, except for the filtering instruction $\mathbf{x} \in S$.

4.2 Bounding interval analysis

The following section aims at proving that by bounding the interval domain to a subdomain with no infinite ascending chains, i.e., where every chain converges in finite time, we can still compute the most precise interval representation for each variable in our program. To do so, we first prove an easy graph-theoretic property which will later be helpful. Consider a finite directed and edge-weighted graph $\langle X, \rightarrow \rangle$ where $\rightarrow \subseteq X \times \mathbb{Z} \times X$ and $x \rightarrow_h x'$ denotes that $(x, h, x') \in \rightarrow$. Consider a finite path in $\langle X, \rightarrow \rangle$

$$p = x_0 \rightarrow_{h_0} x_1 \rightarrow_{h_1} x_2 \rightarrow_{h_2} \dots \rightarrow_{h_{\ell-1}} x_\ell$$

where:

- (i). $\ell \geq 1$
- (ii). the carrier size of p is $s(p) \triangleq |\{x_0, \dots, x_\ell\}|$
- (iii). the weight of p is $w(p) \triangleq \sum_{k=0}^{\ell-1} h_k$
- (iv). the length of p is $|p| \triangleq \ell$
- (v). given indices $0 \leq i < j \leq \ell$, $p_{i,j}$ denotes the subpath of p given by $x_i \rightarrow_{h_i} x_{i+1} \rightarrow_{h_{i+1}} \dots \rightarrow_{h_{j-1}} x_j$ whose length is $j - i$; $p_{i,j}$ is a cycle if $x_i = x_j$.

Lemma 4.4 (Positive cycles in weighted directed graphs). *Let p be a finite path*

$$p = x_0 \rightarrow_{h_0} x_1 \rightarrow_{h_1} x_2 \rightarrow_{h_2} \dots \rightarrow_{h_{\ell-1}} x_\ell$$

with $m \triangleq \max\{|h_j| \mid j \in \{0, \dots, \ell-1\}\} \in \mathbb{N}$ and $w(p) > (|X| - 1)m$. Then, p has a subpath which is a cycle having a strictly positive weight.

Proof. First note that $w(p) = \sum_{k=0}^{\ell-1} h_k > (|X| - 1)m$ implies that $|p| = \ell \geq |X|$. Then, we show our claim by induction on $|p| = \ell \geq |X|$.

($|p| = |X|$): Since the path p includes exactly $|X| + 1 = \ell + 1$ nodes, there exist indices $0 \leq i < j \leq \ell$ such that $x_i = x_j$, i.e., $p_{i,j}$ is a subpath of p which is a cycle. Moreover, since this cycle $p_{i,j}$ includes at least one edge, we have that

$$\begin{aligned} w(p_{i,j}) &= w(p) - (\sum_{k=0}^{i-1} h_k + \sum_{k=j}^{\ell-1} h_k) > && \text{as } w(p) > (|X| - 1)m \\ (|X| - 1)m - (\sum_{k=0}^{i-1} h_k + \sum_{k=j}^{\ell-1} h_k) &\geq && \text{as } \sum_{k=0}^{i-1} h_k + \sum_{k=j}^{\ell-1} h_k \leq (\ell - 1)m \\ &= (|X| - 1)m - (\ell - 1)m = && [\text{as } \ell = |X|] \\ &= (|X| - 1)m - (|X| - 1)m = 0 \end{aligned}$$

so that $w(p_{i,j}) > 0$ holds.

($|p| > |X|$): Since the path p includes at least $|X| + 2$ nodes, as in the base case, we have that p has a subpath which is a cycle. Then, we consider a cycle $p_{i,j}$ in p , for some indices $0 \leq i < j \leq \ell$, which is maximal, i.e., such that if $p_{i',j'}$ is a cycle in p , for some $0 \leq i' < j' \leq \ell$, then $p_{i,j}$ is not a proper subpath of $p_{i',j'}$.

If $w(p_{i,j}) > 0$ then we are done. Otherwise we have that $w(p_{i,j}) \leq 0$ and we consider the path p' obtained from p by stripping off the cycle $p_{i,j}$, i.e.,

$$p' \equiv \overbrace{x_0 \rightarrow_{h_0} x_1 \rightarrow_{h_1} \dots \rightarrow_{h_{i-1}} x_i}^{p'_{0,i}} = \overbrace{x_j \rightarrow_{h_{j+1}} \dots \rightarrow_{h_{\ell-1}} x_\ell}^{p'_{j+1,\ell}}$$

Since $|p'| < |p|$ and $w(p') = w(p) - w(p_{i,j}) \geq w(p) > (|X| - 1)m$, we can apply the inductive hypothesis on p' . We therefore derive that p' has a subpath q which is a cycle having strictly positive weight. This cycle q is either entirely in $p'_{0,i}$ or in $p'_{j+1,\ell}$, otherwise q would include the cycle $p_{i,j}$ thus contradicting the maximality of $p_{i,j}$. Hence, q is a cycle in the original path p having a strictly positive weight. \square

We will now prove a fundamental property of analysis over the lattice of intervals: if a variable in a program does not diverge, then the increment of that variable is bounded by the constants in the program. This property will later be useful to restrict the domain of intervals (which incorporates infinite ascending and descending chains) to one of its subsets that does not contain such chains.

Notation 4.5. For the following proof and whenever we will refer to the abstract semantics over intervals we will use the notation $\llbracket \cdot \rrbracket$ to refer to $\llbracket \cdot \rrbracket^{\mathbb{I}}$.

Lemma 4.6. *Let $C \in \text{Imp}$. For all $\eta \in \dot{\mathbb{I}}$ and $y \in \text{Var}$, if $\max(\llbracket C \rrbracket \eta y) \neq +\infty$ and $\max(\llbracket C \rrbracket \eta y) > (C)^b$ then there exist a variable $z \in \text{Var}$ and an integer $h \in \mathbb{Z}$ such that $|h| \leq (C)^b$ and the following two properties hold:*

$$(i) \max(\llbracket C \rrbracket \eta y) = \max(\eta z) + h;$$

$$(ii) \text{ for all } \eta' \in \dot{\mathbb{I}}, \text{ if } \eta' \sqsupseteq \eta \text{ then } \max(\llbracket C \rrbracket \eta' y) \geq \max(\eta' z) + h.$$

Proof. We preliminarily observe that we can safely assume $\eta \neq \perp$. In fact, if $\eta = \perp$ then $\llbracket C \rrbracket \perp = \perp$ and thus $\max(\llbracket C \rrbracket \eta y) = -\infty \leq (C)^b$, against the hypothesis $\max(\llbracket C \rrbracket \eta y) > (C)^b$. Moreover, when quantifying over η' such that $\eta' \sqsupseteq \eta$ in (ii), if $\max(\llbracket C \rrbracket \eta' y) = +\infty$ holds, then $\max(\llbracket C \rrbracket \eta' y) \geq \max(\eta' z) + h$ trivially holds, hence we will sometimes silently omit this case.

Case $(x \in I)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $+\infty \neq \max(\llbracket x \in I \rrbracket \eta y) > (x \in I)^b$. Clearly $\llbracket x \in I \rrbracket \eta \neq \perp$, otherwise we would get the contradiction $\max(\llbracket x \in I \rrbracket \eta y) = -\infty \leq (x \in I)^b$.

We distinguish two cases:

- If $y \neq x$, then for all $\eta' \in \dot{\mathbb{I}}$ such that $\eta \sqsubseteq \eta'$ it holds

$$\perp \neq \llbracket x \in I \rrbracket \eta' = \eta' [x \mapsto \eta x \sqcap I]$$

and thus

$$\max(\llbracket x \in I \rrbracket \eta' y) = \max(\eta' y) = \max(\eta' y) + 0$$

hence the thesis follows with $z = y$ and $h = 0$.

- If $y = x$ then

$$\max(\llbracket x \in I \rrbracket \eta y) = \max(\eta x \sqcap I)$$

Note that it cannot be $\max(I) \in \mathbb{Z}$. Otherwise, by Definition 4.3, $\max(\eta x \sqcap I) \leq \sup(I) = (x \in I)^b$, violating the assumption $\max(\llbracket x \in I \rrbracket \eta y) > (x \in I)^b$. Hence, $\max(I) = +\infty$ must hold and therefore $\max(\eta x \sqcap I) = \max(\eta(x)) = \max(\eta(x)) + 0$. It is immediate to check that the same holds for all $\eta' \sqsupseteq \eta$, i.e.,

$$\max(\eta x \sqcap I) = \max(\eta' x) + 0$$

and thus the thesis follows with $z = y = x$ and $h = 0$.

Case $(x := k)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max(\llbracket x := k \rrbracket \eta y) > (x := k)^b = |k|$.

Observe that it cannot be $x = y$. In fact, since $\llbracket x := k \rrbracket \eta = \eta [x \mapsto [k, k]]$, we would have $\llbracket x := k \rrbracket \eta y = \alpha_{\mathbb{I}}(\{k\}) = [k, k]$ and thus

$$\max(\llbracket x := k \rrbracket \eta y) = k \leq (x := k)^b$$

violating the assumption. Therefore, it must be $y \neq x$. Now, for all $\eta' \sqsupseteq \eta$, we have $\llbracket x := k \rrbracket \eta' y = \eta' y$ and thus

$$\max(\llbracket x := k \rrbracket \eta' y) = \max(\eta' y) = \max(\eta' y) + 0,$$

hence the thesis holds with $h = 0 \leq (x := k)^b$ and $z = y$.

Case $(x := w + k)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max(\llbracket x := w + k \rrbracket \eta y) > (x := w + k)^b = |k|$. Recall that $\llbracket x := w + k \rrbracket \eta = \eta[x \mapsto \eta w + k]$.

We distinguish two cases:

- If $y \neq x$, then for all $\eta' \sqsupseteq \eta$, we have $\llbracket x := w + k \rrbracket \eta' y = \eta' y$ and thus

$$\max(\llbracket x := w + k \rrbracket \eta' y) = \max(\eta' y)$$

hence the thesis follows with $h = 0 \leq (x := w + k)^b$ and $z = y$.

- If $x = y$ then for all $\eta' \sqsupseteq \eta$, we have $\llbracket x := w + k \rrbracket \eta' y = \eta' w + k$ and thus

$$\max(\llbracket x := w + k \rrbracket \eta' y) = \max(\eta' w) + k$$

hence, the thesis follows with $h = k$ (recall that $k \leq |k| = (x := w + k)^b$) and $z = w$.

Case $(C_1 + C_2)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max(\llbracket C_1 + C_2 \rrbracket \eta) > (C_1 + C_2)^b = (C_1)^b + (C_2)^b$. Recall that $\llbracket C_1 + C_2 \rrbracket \eta = \llbracket C_1 \rrbracket \eta \sqcup \llbracket C_2 \rrbracket \eta$. Hence, since $\max(\llbracket C_1 + C_2 \rrbracket \eta y) \neq +\infty$, we have that $\max(\llbracket C_1 \rrbracket \eta y) \neq \infty \neq \max(\llbracket C_2 \rrbracket \eta y)$. Moreover

$$\begin{aligned} \max(\llbracket C_1 + C_2 \rrbracket \eta y) &= \max(\llbracket C_1 \rrbracket \eta y \sqcup \llbracket C_2 \rrbracket \eta y) \\ &= \max\{\max(\llbracket C_1 \rrbracket \eta y), \max(\llbracket C_2 \rrbracket \eta y)\} \end{aligned}$$

Thus $\max(\llbracket C_1 + C_2 \rrbracket \eta y) = \max(\llbracket C_i \rrbracket \eta y)$ for some $i \in \{1, 2\}$. We can assume, without loss of generality, that the maximum is realized by the first component, i.e., $\max(\llbracket C_1 + C_2 \rrbracket \eta y) = \max(\llbracket C_1 \rrbracket \eta y) > (C_1 + C_2)^b$. Hence we can use the inductive hypothesis on C_1 and state that there exists $h \in \mathbb{Z}$ with $|h| \leq (C_1)^b$ and $z \in \text{Var}$ such that $\max(\llbracket C_1 \rrbracket \eta y) = \max(\eta z) + h$ and for all $\eta' \in \dot{\mathbb{I}}$, $\eta \sqsubseteq \eta'$,

$$\max(\llbracket C_1 \rrbracket \eta' y) \geq \max(\eta' z) + h$$

Therefore

$$\max(\llbracket C_1 + C_2 \rrbracket \eta y) = \max(\llbracket C_1 \rrbracket \eta y) = \max(\eta z) + h$$

and for all $\eta' \in \dot{\mathbb{I}}$, $\eta \sqsubseteq \eta'$,

$$\begin{aligned} \max(\llbracket C_1 + C_2 \rrbracket \eta' y) &= \max\{\max(\llbracket C_1 \rrbracket \eta' y), \max(\llbracket C_2 \rrbracket \eta' y)\} \\ &\geq \max(\llbracket C_1 \rrbracket \eta' y) \\ &\geq \max(\eta' z) + h \end{aligned}$$

with $|h| \leq (C_1)^b \leq (C_1 + C_2)^b$, as desired.

Case $(C_1; C_2)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max(\llbracket C_1; C_2 \rrbracket \eta y) > (C_1; C_2)^b = (C_1)^b + (C_2)^b$. Recall that $\llbracket C_1; C_2 \rrbracket \eta = \llbracket C_2 \rrbracket (\llbracket C_1 \rrbracket \eta)$. If we define

$$\llbracket C_1 \rrbracket \eta = \eta_1$$

since $\max(\llbracket C_2 \rrbracket \eta_1 y) \neq \infty$ and $\max(\llbracket C_2 \rrbracket \eta_1 y) > (C_1; C_2)^b \geq (C_2)^b$, by inductive hypothesis on C_2 , there are $|h_2| \leq (C_2)^b$ and $w \in \text{Var}$ such that $\max(\llbracket C_2 \rrbracket \eta_1 y) = \max(\eta_1 w) + h_2$ and for all $\eta'_1 \in \dot{\mathbb{I}}$ with $\eta_1 \sqsubseteq \eta'_1$

$$\max(\llbracket C_2 \rrbracket \eta'_1 y) \geq \max(\eta'_1 w) + h_2 \tag{4.1}$$

Now observe that $\max(\llbracket C_1 \rrbracket \eta w) = \max(\eta_1 w) > (C_1)^b$. Otherwise, if it were $\max(\eta_1 w) \leq (C_1)^b$ we would have

$$\max(\llbracket C_2 \rrbracket \eta_1 y) = \max(\eta_1 w) + h_2 \leq (C_1)^b + (C_2)^b = (C_1; C_2)^b,$$

violating the hypotheses. Moreover, $\max(\llbracket C_1 \rrbracket \eta \mathbf{w}) \neq +\infty$, otherwise we would have $\max(\llbracket C_2 \rrbracket \eta_1 \mathbf{y}) = \max(\eta_1 \mathbf{w}) + h_2 = +\infty$, contradicting the hypotheses. Therefore we can apply the inductive hypothesis also to C_1 and deduce that there are $|h_1| \leq (C_1)^b$ and $\mathbf{w}' \in Var$ such that $\max(\llbracket C_1 \rrbracket \eta \mathbf{w}) = \max(\eta \mathbf{w}') + h_1$ and for all $\eta' \in \mathbb{I}$ with $\eta \sqsubseteq \eta'$

$$\max(\llbracket C_1 \rrbracket \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}') + h_1 \quad (4.2)$$

Summing up:

$$\begin{aligned} \max(\llbracket C_1; C_2 \rrbracket \eta \mathbf{y}) &= \max(\llbracket C_2 \rrbracket (\llbracket C_1 \rrbracket \eta) \mathbf{y}) \\ &= \max(\llbracket C_2 \rrbracket \eta_1 \mathbf{y}) \\ &= \max(\eta_1 \mathbf{w}) + h_2 \\ &= \max(\llbracket C_1 \rrbracket \eta \mathbf{w}) + h_2 \\ &= \max(\eta \mathbf{w}') + h_1 + h_2. \end{aligned}$$

Now, for all $\eta' \in \mathbb{I}$ with $\eta \sqsubseteq \eta'$ we have that:

$$\begin{aligned} \max(\llbracket C_1; C_2 \rrbracket \eta' \mathbf{y}) &= \\ \max(\llbracket C_2 \rrbracket (\llbracket C_1 \rrbracket \eta') \mathbf{y}) &\geq \\ \max(\llbracket C_1 \rrbracket \eta' \mathbf{w}) + h_2 &\geq \quad \text{by (4.1), since } \eta_1 = \llbracket C_1 \rrbracket \eta \sqsubseteq \llbracket C_1 \rrbracket \eta' \text{ and monotonicity} \\ (\max(\eta' \mathbf{w}') + h_1) + h_2 &\quad \text{by (4.2)} \end{aligned}$$

Thus, the thesis holds with $h = h_1 + h_2$, as $|h| = |h_1 + h_2| \leq |h_1| + |h_2| \leq (C_1)^b + (C_2)^b = (C_1; C_2)^b$, as needed.

Case $(\text{fix}(C))$. Let $\eta \in \mathbb{I}$ such that $\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) \neq +\infty$. Recall that $\llbracket \text{fix}(C) \rrbracket \eta = \text{lfp}(\lambda \mu. (\llbracket C \rrbracket \mu \sqcup \eta))$. Observe that the least fixpoint of $\lambda \mu. (\llbracket C \rrbracket \mu \sqcup \eta)$ coincides with the least fixpoint of $\lambda \mu. (\llbracket C \rrbracket \mu \sqcup \mu) = \lambda \mu. \llbracket C + \text{true} \rrbracket \mu$ above η . Hence, if

$$\begin{aligned} \eta_0 &\triangleq \eta \\ \text{for all } i \in \mathbb{N} \quad \eta_{i+1} &\triangleq \llbracket C \rrbracket \eta_i \sqcup \eta_i = \llbracket C + \text{true} \rrbracket \eta_i \sqsupseteq \eta_i \end{aligned}$$

then we define an increasing chain $\{\eta_i\}_{i \in \mathbb{N}} \subseteq \mathbb{I}$ such that

$$\llbracket \text{fix}(C) \rrbracket \eta = \bigsqcup_{i \in \mathbb{N}} \eta_i.$$

Since $\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) \neq +\infty$, we have that for all $i \in \mathbb{N}$, $\max(\eta_i \mathbf{y}) \neq +\infty$. Moreover, $\bigsqcup_{i \in \mathbb{N}} \eta_i$ on \mathbf{y} is finitely reached in the chain $\{\eta_i\}_{i \in \mathbb{N}}$, i.e., there exists $m \in \mathbb{N}$ such that for all $i \geq m+1$

$$\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) = \max(\eta_i \mathbf{y}).$$

The inductive hypothesis holds for C and true , hence for $C + \text{true}$, therefore for all $\mathbf{x} \in Var$ and $j \in \{0, 1, \dots, m\}$, if $\max(\eta_{j+1} \mathbf{x}) > (C + \text{true})^b = (C)^b$ then there exist $\mathbf{z} \in Var$ and $h \in \mathbb{Z}$ such that $|h| \leq (C)^b$ and

- (a) $+\infty \neq \max(\eta_{j+1} \mathbf{x}) = \max(\eta_j \mathbf{z}) + h$,
- (b) $\forall \eta' \sqsupseteq \eta_j. \max(\llbracket C + \text{true} \rrbracket \eta' \mathbf{x}) \geq \max(\eta' \mathbf{z}) + h$.

To shortly denote that the two conditions (a) and (b) hold, we write

$$(\mathbf{z}, j) \rightarrow_h (\mathbf{x}, j+1)$$

Now, assume that for some variable $\mathbf{y} \in Var$

$$\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) = \max(\eta_{m+1} \mathbf{y}) > (\text{fix}(C))^b = (n+1)(C)^b$$

where $n = |\text{vars}(\mathbf{C})|$. We want to show that the thesis holds, i.e., that there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ with $|h| \leq (\text{fix}(\mathbf{C}))^b$ such that:

$$\max(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{y}) = \max(\eta' \mathbf{z}) + h \quad (4.3)$$

and for all $\eta' \sqsupseteq \eta$,

$$\max(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta' \mathbf{y}) \geq \max(\eta' \mathbf{z}) + h \quad (4.4)$$

Let us consider (4.3). We first observe that we can define a path

$$\sigma \triangleq (\mathbf{y}_0, 0) \rightarrow_{h_0} (\mathbf{y}_1, 1) \rightarrow_{h_1} \dots \rightarrow_{h_m} (\mathbf{y}_{m+1}, m+1) \quad (4.5)$$

such that $\mathbf{y}_{m+1} = \mathbf{y}$ and for all $j \in \{0, \dots, m+1\}$, $\mathbf{y}_j \in \text{Var}$ and $\max(\eta_j \mathbf{y}_j) > (\mathbf{C})^b$. In fact, if, by contradiction, this were not the case, there would exist an index $i \in \{0, \dots, m\}$ (as $\max(\eta_{m+1} \mathbf{y}_{m+1}) > (\mathbf{C})^b$ already holds) such that $\max(\eta_i \mathbf{y}_i) \leq (\mathbf{C})^b$, while for all $j \in \{i+1, \dots, m+1\}$, $\max(\eta_j \mathbf{y}_j) > (\mathbf{C})^b$. Thus, in such a case, we consider the nonempty path:

$$\pi \triangleq (\mathbf{y}_i, i) \rightarrow_{h_i} (\mathbf{y}_{i+1}, i+1) \rightarrow_{h_{i+1}} \dots \rightarrow_{h_m} (\mathbf{y}_{m+1}, m+1) \quad (4.6)$$

and we have that:

$$\begin{aligned} \sum_{j=i}^m h_j &= \\ \sum_{j=i}^m \max(\eta_{j+1} \mathbf{y}_{j+1}) - \max(\eta_j \mathbf{y}_j) &= \\ \max(\eta_{m+1} \mathbf{y}_{m+1}) - \max(\eta_i \mathbf{y}_i) &= \\ \max(\eta_{m+1} \mathbf{y}) - \max(\eta_i \mathbf{y}_i) &> \\ (n+1)(\mathbf{C})^b - (\mathbf{C})^b &= n(\mathbf{C})^b \end{aligned}$$

with $|h_j| \leq (\mathbf{C})^b$ for $j \in \{i, \dots, m\}$. Hence we can apply Lemma 4.4 to the projection π_p of the nodes of this path π to the variable component to deduce that π_p has a subpath which is a cycle with a strictly positive weight. More precisely, there exist $i \leq k_1 < k_2 \leq m+1$ such that $\mathbf{y}_{k_1} = \mathbf{y}_{k_2}$ and $h = \sum_{j=k_1}^{k_2-1} h_j > 0$. If we denote $\mathbf{w} = \mathbf{y}_{k_1} = \mathbf{y}_{k_2}$, then we have that

$$\begin{aligned} \max(\eta_{k_2} \mathbf{w}) &= h_{k_2-1} + \max(\eta_{k_2-1} \mathbf{w}) \\ &= h_{k_2-1} + h_{k_2-2} + \max(\eta_{k_2-2} \mathbf{w}) \\ &= \sum_{j=k_1}^{k_2-1} h_j + \max(\eta_{k_1} \mathbf{w}) \\ &= h + \max(\eta_{k_1} \mathbf{w}) \end{aligned}$$

Thus,

$$\max(\llbracket \mathbf{C} + \text{true} \rrbracket^{k_2-k_1} \eta_{k_1} \mathbf{w}) = \max(\eta_{k_1} \mathbf{w}) + h$$

Observe that for all $\eta' \sqsupseteq \eta_{k_1}$

$$\max(\llbracket \mathbf{C} + \text{true} \rrbracket^{k_2-k_1} \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}) + h \quad (4.7)$$

Let us show Property (4.7) by induction on $\ell = k_2 - k_1 \geq 1$.

Case ($\ell = 1$). Notice that by (b) used to build π in (4.6) it holds that $\forall \eta' \sqsupseteq \eta_{k_1} \sqsupseteq \eta$

$$\max(\llbracket \mathbf{C} + \text{true} \rrbracket \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}) + h$$

hence the thesis holds.

Case ($\ell \implies \ell + 1$). Recall that

$$(\llbracket \mathbf{C} + \text{true} \rrbracket)^{\ell+1} \eta' = (\llbracket \mathbf{C} + \text{true} \rrbracket) \left((\llbracket \mathbf{C} + \text{true} \rrbracket)^\ell \eta' \right)$$

and by inductive hypothesis $\max\left(\llbracket C + \text{true} \rrbracket^\ell \eta' \mathbf{w}\right) \geq \max(\eta' \mathbf{w}) + h$. Recall that for all $\eta'' \in \dot{\mathbb{I}} \llbracket C + \text{true} \rrbracket \eta'' = \eta'' \sqcup \llbracket C \rrbracket \eta''$. Hence we can notice that $\max(\llbracket C + \text{true} \rrbracket \eta'') \mathbf{x} \geq \max(\eta'') \mathbf{x}$ for all $\mathbf{x} \in \text{Var}$. Therefore

$$\max\left(\llbracket C + \text{true} \rrbracket \left(\llbracket C + \text{true} \rrbracket^\ell \eta'\right) \mathbf{w}\right) \geq \max\left(\llbracket C + \text{true} \rrbracket^\ell \eta' \mathbf{w}\right) \geq \max(\eta' \mathbf{w}) + h$$

which is our thesis for Property (4.7).

Then, an inductive argument allows us to show that for all $r \in \mathbb{N}$:

$$\max(\llbracket C + \text{true} \rrbracket^{r(k_2 - k_1)} \eta_{k_1} \mathbf{w}) \geq \max(\eta_{k_1} \mathbf{w}) + rh \quad (4.8)$$

In fact, for $r = 0$ the claim trivially holds. Assuming the validity for $r \geq 0$ then we have that

$$\begin{aligned} \max(\llbracket C + \text{true} \rrbracket^{(r+1)(k_2 - k_1)} \eta_{k_1} \mathbf{w}) &= \\ \max(\llbracket C + \text{true} \rrbracket^{k_2 - k_1} (\llbracket C + \text{true} \rrbracket^{r(k_2 - k_1)} \eta_{k_1}) \mathbf{w}) &\geq \text{by (4.7) as } \eta_{k_1} \sqsubseteq \llbracket C + \text{true} \rrbracket^{r(k_2 - k_1)} \eta_{k_1} \\ \max(\llbracket C + \text{true} \rrbracket^{r(k_2 - k_1)} \eta_{k_1} \mathbf{w}) + h &\geq \text{by inductive hypothesis} \\ \max(\eta_{k_1} \mathbf{w}) + rh + h &\geq \max(\eta_{k_1} \mathbf{w}) + (r+1)h \end{aligned}$$

However, this would contradict the hypothesis $\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) \neq \infty$. In fact the inequality (4.8) would imply

$$\begin{aligned} \llbracket \text{fix}(C) \rrbracket \eta \mathbf{w} &= \bigsqcup_{i \in \mathbb{N}} \llbracket C + \text{true} \rrbracket^i \eta \mathbf{w} = \\ &= \bigsqcup_{i \in \mathbb{N}} \llbracket C + \text{true} \rrbracket^i \eta_{k_1} \mathbf{w} \\ &= \bigsqcup_{r \in \mathbb{N}} \llbracket C + \text{true} \rrbracket^{r(k_2 - k_1)} \eta_{k_1} \mathbf{w} \\ &= +\infty \end{aligned}$$

Now, from (4.5) we deduce that for all $\eta' \sqsupseteq \eta_{k_1}$, for $j \in \{k_1, \dots, m\}$, if we let $\mu_{k_1} = \eta'$ and $\mu_{j+1} = \llbracket C + \text{true} \rrbracket \mu_j$, by the choice of the subsequence, since $k_1 \geq i$, we have that

$$\max(\mu_{j+1} \mathbf{y}_{j+1}) \geq \max(\mu_{j+1} \mathbf{y}_j) + h_j$$

and thus

$$\llbracket C + \text{true} \rrbracket^{m-k_1+1} \eta' \mathbf{y} = \mu_{m+1} \mathbf{y}_{m+1} \geq \max(\mathbf{y}_{k_1}) + \sum_{i=k_1}^m h_i = \max(\eta' \mathbf{w}) + \sum_{i=k_1}^m h_i$$

Since $\eta' = \llbracket \text{fix}(C) \rrbracket \eta \sqsupseteq \eta_{k_1}$ we conclude

$$\begin{aligned} \max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) &= \max\left(\llbracket C + \text{true} \rrbracket^{m-k_1+1} \llbracket \text{fix}(C) \rrbracket \eta \mathbf{w}\right) \\ &= \max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{w}) + \sum_{i=k_1}^m h_i \\ &\geq +\infty + \sum_{i=k_1}^m h_i = +\infty \end{aligned}$$

contradicting the assumption.

Therefore, the path σ of (4.5) must exist, and consequently

$$\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) = \max(\eta_{m+1} \mathbf{y}) = \max(\eta \mathbf{y}_0) + \sum_{i=0}^m h_i$$

and $\sum_{i=0}^m h_i \leq (\text{fix}(C))^b = (n+1)(C)^b$, otherwise we could use the same argument above for inferring the contradiction $\max(\llbracket \text{fix}(C) \rrbracket \eta \mathbf{y}) = +\infty$.

Let us now show (4.4). Given $\eta' \sqsupseteq \eta$ from (4.5) we deduce that for all $j \in \{0, \dots, m\}$, if we let $\mu_0 = \eta'$ and $\mu_{j+1} = \llbracket C + \text{true} \rrbracket \mu_j$, we have that

$$\max(\mu_{j+1} \mathbf{y}_{j+1}) \geq \max(\mu_{j+1} \mathbf{y}_j) + h_j.$$

Therefore, since $\llbracket \text{fix}(C) \rrbracket \eta' \sqsupseteq \mu_{m+1}$ (observe that the convergence of $\llbracket \text{fix}(C) \rrbracket \eta'$ could be at an index greater than $m+1$), we conclude that:

$$\max(\llbracket \text{fix}(C) \rrbracket \eta' \mathbf{y}) \geq \max(\mu_{m+1} \mathbf{y}) = \max(\mu_{m+1} \mathbf{y}_{m+1}) \geq \max(\eta' \mathbf{y}_0) + \sum_{i=0}^m h_i$$

as desired. \square

We can now notice that this proof also works for the min value of each variable's interval. I.e., the following property also holds:

Lemma 4.7. *Let $C \in \text{Imp}$.*

For all $\eta \in \dot{\mathbb{I}}$ and $\mathbf{y} \in \text{Var}$, if $\min(\llbracket C \rrbracket \eta \mathbf{y}) \neq -\infty$ and $\min(\llbracket C \rrbracket \eta \mathbf{y}) < -(C)_b$ then there exist a variable $\mathbf{z} \in \text{Var}$ and an integer $h \in \mathbb{Z}$ s.t. $|h| \leq (C)_b$ s.t. the following two properties hold:

$$(i) \min(\llbracket C \rrbracket \eta \mathbf{y}) = \min(\eta \mathbf{z}) + h;$$

$$(ii) \text{ for all } \eta' \in \dot{\mathbb{I}}, \text{ if } \eta' \sqsupseteq \eta \text{ then } \min(\llbracket C \rrbracket \eta' \mathbf{y}) \leq \min(\eta' \mathbf{z}) + h.$$

Proof. The full proof is available at Appendix A.1, as Lemma A.2. Intuitively the proof works by considering the integers \mathbb{Z} with the reverse ordering $<$ and a new bound, $(C)_b$, computed by considering the reverse ordering. \square

4.3 Computing interval semantics

Lemma 4.6 and Lemma 4.7 provide us an useful insight to compute the abstract interval semantics of Imp programs. Combined, they state that each variable in a program can increment (or decrement) by a maximal value which depends on the initial state η and the constants that appear in the program itself: the bound $(C)_b$. In this section we use this idea to compute the interval semantics in another abstract domain, which does not contain infinite ascending or descending chains, and therefore using the simple Kleene iteration to compute least fixpoints ensures termination.

To start we define bounded abstract interval domains:

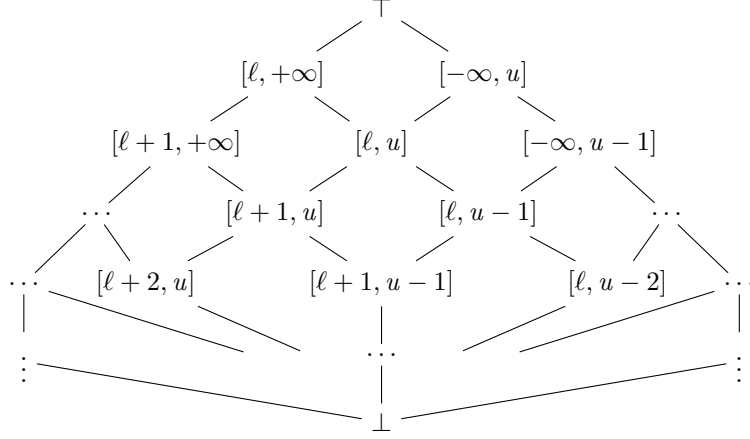
Definition 4.8 (Bounded interval). We define $\dot{\mathbb{I}}_\ell^u \triangleq (\text{Var}_C \rightarrow \mathbb{I}_\ell^u) \cup \{\perp\}$ where

$$\begin{aligned} \mathbb{I}_\ell^u &\triangleq \{[a, b] \mid a, b \in \mathbb{Z} \wedge \ell \leq a \leq b \leq u\} \\ &\cup \{[a, +\infty] \mid a \in \mathbb{Z} \wedge a \geq \ell\} \\ &\cup \{[-\infty, b] \mid b \in \mathbb{Z} \wedge b \leq u\} \end{aligned}$$

We visualize the Hasse diagram of the bounded integer domain in Figure 4.1 and notice that by definition there are no infinite ascending chains. Now we can notice that given $\ell, u \in \mathbb{Z}$ we can build a Galois Connection (Definition 1.16) between the interval domain $\dot{\mathbb{I}}$ and the bounded interval domain $\dot{\mathbb{I}}_\ell^u$ playing here the role of concrete and abstract domain respectively. To do so we first need to define a concretization and abstraction maps.

Definition 4.9. Given $\ell, u \in \mathbb{Z}$ we define a concretization map $\gamma_{\ell, u} : \mathbb{I}_\ell^u \rightarrow \dot{\mathbb{I}}$ as the function

$$\forall a \in \dot{\mathbb{I}}_\ell^u \quad \gamma_{\ell, u}(a) = a$$

Figure 4.1: \mathbb{I}_ℓ^u Hasse diagram

i.e., $\gamma_{\ell,u} = \text{id}$. While we define an abstraction map $\alpha_{\ell,u} : \mathbb{I} \rightarrow \mathbb{I}_\ell^u$ in the following way

$$\alpha_{\ell,u}(\perp) = \perp$$

$$\alpha_{\ell,u}([a, b]) = \begin{cases} [a, b] & \text{if } a \geq \ell \wedge b \leq u \\ [-\infty, b] & \text{if } a < \ell \wedge b \leq u \\ [a, +\infty] & \text{if } a \geq \ell \wedge b > u \\ [-\infty, +\infty] & \text{otherwise} \end{cases}$$

Next, we prove that given $\ell, u \in \mathbb{Z}$ we in fact have a Galois Connection:

Lemma 4.10. *Given $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$*

$$\langle \mathbb{I}, \sqsubseteq \rangle \xleftrightarrow[\alpha_{\ell,u}]{\text{id}} \langle \mathbb{I}_\ell^u, \sqsubseteq \rangle$$

i.e., $\langle \alpha_{\ell,u}, \mathbb{I}, \mathbb{I}_\ell^u, \text{id} \rangle$ is a Galois Connection.

Proof. We want to prove that id and $\alpha_{\ell,u}$ satisfy the property as in Theorem 1.17:

- (1) $\alpha_{\ell,u}, \text{id}$ are monotonic;
- (2) $\text{id} \circ \alpha_{\ell,u}$ is extensive, i.e., $\forall \iota \in \mathbb{I}$ it holds that $\iota \sqsubseteq \text{id}(\alpha_{\ell,u}(\iota))$;
- (3) $\alpha_{\ell,u} \circ \text{id}$ is reductive, i.e., $\forall \iota_b \in \mathbb{I}_\ell^u$ it holds that $\alpha_{\ell,u}(\text{id}(\iota_b)) \sqsubseteq \iota_b$.

Let us show (1). The function id is monotone since $\forall \iota, \kappa \in \mathbb{I}_\ell^u$ it holds that $\iota \sqsubseteq \kappa \implies \iota \sqsubseteq \kappa$. For $\alpha_{\ell,u}$ we have to prove that for all $\iota, \kappa \in \mathbb{I}$ it holds that $\iota \sqsubseteq \kappa \implies \alpha_{\ell,u}(\iota) \sqsubseteq \alpha_{\ell,u}(\kappa)$. Now notice that $\iota \sqsubseteq \kappa$ means that $\min(\iota) \geq \min(\kappa)$ and $\max(\iota) \leq \max(\kappa)$. Hence, by Definition 4.9 of $\alpha_{\ell,u}$ it holds that $\alpha_{\ell,u}(\iota) \sqsubseteq \alpha_{\ell,u}(\kappa)$, which is our thesis.

Let us now show (2). We have to prove that $\forall \iota \in \mathbb{I}$ it holds that $\iota \sqsubseteq \gamma(\alpha_{\ell,u}(\iota))$. By hypothesis $\gamma = \text{id}$, hence we just have to prove that $\iota \sqsubseteq \alpha_{\ell,u}(\iota)$. Based on the definition of $\alpha_{\ell,u}$ from Definition 4.9 both the following hold:

$$\begin{aligned} \min(\alpha_{\ell,u}(\iota)) &\leq \min(\iota) \\ \max(\alpha_{\ell,u}(\iota)) &\geq \max(\iota) \end{aligned}$$

Hence it holds that

$$\iota \sqsubseteq \alpha_{\ell,u}(\iota) \tag{4.9}$$

We can finally prove (3): $\alpha_{\ell,u} \circ \gamma$ is reductive, i.e., $\forall \iota_b \in \mathbb{I}_\ell^u, \alpha_{\ell,u}(\text{id}(\iota_b)) \sqsubseteq \iota_b$. Notice that $\forall \iota_b \in \mathbb{I}_\ell^u$ it holds that

$$\alpha_{\ell,u}(\iota_b) = \iota_b \quad (4.10)$$

hence it holds that $\alpha_{\ell,u}(\iota_b) \sqsubseteq \iota_b$ \square

Notice that because of Equation (4.10) holds we know that $\alpha_{\ell,u} \circ \gamma_{\ell,u} = \text{id}$, hence we not only have a Galois Connection but a Galois insertion (Definition 1.20):

$$\langle \mathbb{I}, \sqsubseteq \rangle \xleftarrow[\alpha_{\ell,u}]{\text{id}} \langle \mathbb{I}_\ell^u, \sqsubseteq \rangle \quad (4.11)$$

because of the latter observation and since $\dot{\mathbb{I}}, \dot{\mathbb{I}}_\ell^u$ are the point(variable)-wise lifting of $\mathbb{I}, \mathbb{I}_\ell^u$ respectively, by Theorem 1.22 it holds that

$$\langle \dot{\mathbb{I}}, \sqsubseteq \rangle \xleftarrow[\dot{\alpha}_{\ell,u}]{\dot{\gamma}_{\ell,u}} \langle \dot{\mathbb{I}}_\ell^u, \sqsubseteq \rangle \quad (4.12)$$

where $\dot{\alpha}_{\ell,u}(\eta) = \lambda x. \alpha_{\ell,u}(\eta x)$ and $\dot{\gamma}_{\ell,u} = \text{id}$. We can therefore define our analysis in $\dot{\mathbb{I}}_\ell^u$ by means of best correct approximations over $\dot{\mathbb{I}}_\ell^u$.

Notation 4.11. For the following definition and whenever we will need to talk about the abstract semantics over the interval domain bounded over some constants $\ell, u \in \mathbb{Z}$ we will write $\llbracket \cdot \rrbracket_\ell^u$ to refer to $\llbracket \cdot \rrbracket_{\dot{\mathbb{I}}_\ell^u}$.

Definition 4.12 (Bounded interval analysis). Let $\rho \in \dot{\mathbb{I}}_\ell^u$ for some $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$ and $C \in \text{Imp}$. We define $((e))^{\dot{\mathbb{I}}_\ell^u} \rho$ as the best correct approximation of base expressions over $\dot{\mathbb{I}}$:

$$((e))^{\dot{\mathbb{I}}_\ell^u} \triangleq \dot{\alpha}_{\ell,u} \circ ((e))^{\dot{\mathbb{I}}} \circ \dot{\gamma}_{\ell,u}$$

where $e \in \text{Exp}$.

Notice that for basic expressions we are using the best correct approximation $\dot{\alpha}_{\ell,u} \circ \llbracket e \rrbracket \circ \text{id}$, which allows us to prove the soundness of the analysis over $\dot{\mathbb{I}}_\ell^u$ w.r.t. the analysis over $\dot{\mathbb{I}}$:

Lemma 4.13. For all $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$, $\rho \in \dot{\mathbb{I}}_\ell^u$

$$\llbracket C \rrbracket \rho \sqsubseteq \llbracket C \rrbracket_\ell^u \rho$$

i.e., with $\dot{\mathbb{I}}_\ell^u$ we have an over-approximation of $\dot{\mathbb{I}}$.

Proof. The theorem follows from the fact that there is a Galois connection

$$\dot{\mathbb{I}} \xleftarrow[\dot{\alpha}_{\ell,u}]{\text{id}} \dot{\mathbb{I}}_\ell^u$$

between $\dot{\mathbb{I}}$ and $\dot{\mathbb{I}}_\ell^u$ for all $\ell, u \in \mathbb{Z} \mid \ell \leq u$ and the fact that base expressions are defined with the best correct approximation. Hence by Corollary 3.3 follows that for all $C \in \text{Imp}, \rho \in \dot{\mathbb{I}}_\ell^u$

$$(\llbracket C \rrbracket \circ \text{id}) \rho \sqsubseteq (\text{id} \circ \llbracket C \rrbracket_\ell^u) \rho$$

hence our thesis. \square

Now we define a new bound, it will be useful later in Theorem 4.17.

Definition 4.14. Let $C \in \text{Imp}$. Then $(\cdot)_+^b : \text{Imp} \rightarrow \mathbb{N}$ is the *updated bound*, recursively defined as follows:

$$\begin{aligned} (e)_+^b &\triangleq (e)^b \\ (C_1 + C_2)_+^b &\triangleq (C_1)_+^b + (C_2)_+^b \\ (C_1; C_2)_+^b &\triangleq (C_1)_+^b + (C_2)_+^b \\ (\text{fix}(C))_+^b &\triangleq (n + 2)(C)_+^b \end{aligned}$$

where $n = \text{vars}(C)$. Similarly, $(\cdot)_b^+ : \text{Imp} \rightarrow \mathbb{N}$ is the *updated lower bound* of commands, recursively defined as follows:

$$\begin{aligned} (e)_b^+ &\triangleq (e)_b \\ (C_1 + C_2)_b^+ &\triangleq (C_1)_b^+ + (C_2)_b^+ \\ (C_1; C_2)_b^+ &\triangleq (C_1)_b^+ + (C_2)_b^+ \\ (\text{fix}(C))_b^+ &\triangleq (n + 2)(C)_b^+ \end{aligned}$$

Notice that the updated bounds differ to bounds of Definition 4.3 only in the case of the $\text{fix}(C)$ command. With this consideration we can now proceed to prove that the analysis on our bounded lattice $\dot{\mathbb{I}}_{C,\rho}$ produces the same result as the analysis on $\dot{\mathbb{I}}$. We preliminarily define min and max values of an environment. Intuitively they are the min and max values that variables can assume without diverging.

Definition 4.15 (min and max). Given a command C , the corresponding finite set of variables $\text{Var}_C \triangleq \text{vars}(C)$, and an interval environment $\rho : \text{Var}_C \rightarrow \mathbb{I}$, we define both the min and the max value of an interval environment:

$$\begin{aligned} \max(\rho) &\triangleq \max \{ \max\{\max(\rho(x)) \mid x \in \text{Var}_C \wedge \max(\rho(x)) \neq +\infty\}, 0 \} \\ \min(\rho) &\triangleq \min \{ \min\{\min(\rho(x)) \mid x \in \text{Var}_C \wedge \min(\rho(x)) \neq -\infty\}, 0 \} \end{aligned}$$

Notice that it holds that $\max(\rho) \geq 0$ and $\min(\rho) \leq 0$. This will later be useful for the base cases of Lemma 4.6 and Lemma 4.18. Now, when computing $\llbracket C \rrbracket \rho$ on such ρ having a finite domain, we can restrict to an interval domain bounded by some constant $k \in \mathbb{N}$: Thanks to the latter definition, we can now also define the notion of domain bounded by initial state and program.

Definition 4.16. Let $C \in \text{Imp}$ and $\rho \in \dot{\mathbb{I}}$. Then the *bounded interval domain* $\dot{\mathbb{I}}_{C,\rho}$ is a bounded interval domain $\dot{\mathbb{I}}_\ell^u$ where

$$\begin{aligned} \ell &= \min(\rho) - (C)_b^+ \\ u &= \max(\rho) + (C)_+^b \end{aligned}$$

Theorem 4.17. Let $C \in \text{Imp}$ be a command. Then, for all finitely supported $\rho : \text{Var} \rightarrow \mathbb{I}$ and $\ell, u \in \mathbb{Z}$ s.t. $\dot{\mathbb{I}}_{C,\rho} \subseteq \dot{\mathbb{I}}_\ell^u$, i.e., $\ell \leq \min(\rho) - (C)_b^+$ and $u \geq \max(\rho) + (C)_+^b$

$$\llbracket C \rrbracket \rho = \llbracket C \rrbracket_\ell^u \rho \quad (4.13)$$

i.e., the abstract semantics $\llbracket C \rrbracket \rho$ computed in $\dot{\mathbb{I}}$ and the one computed in $\dot{\mathbb{I}}_\ell^u$ coincide.

Proof. Notice that because of Lemma 4.13 the statement $\llbracket C \rrbracket \rho \subseteq \llbracket C \rrbracket_\ell^u \rho$ already holds. Therefore what we are left to prove is that

$$\llbracket C \rrbracket \rho \supseteq \llbracket C \rrbracket_\ell^u \rho$$

The proof will proceed by induction on the command $C \in \text{Imp}$.

Case $(x \in S)$. In this case we want to prove that $\llbracket x \in S \rrbracket \rho \sqsupseteq \llbracket x \in S \rrbracket_\ell^u \rho$. Recall that we are considering $\ell \leq \min(\rho) - (x \in S)_b^+ = \min(\rho) - (x \in S)_b$ and $u \geq \max(\rho) + (x \in S)_+^b = \max(\rho) + (x \in S)_+^b$. Notice that either $\rho x \sqcap S = \perp$, which implies that $\llbracket x \in S \rrbracket \rho = \perp$, and therefore $\alpha_{\ell,u}(\llbracket x \in S \rrbracket \rho) = \alpha_{\ell,u}(\perp) = \perp$ and therefore $\perp \sqsupseteq \perp$ holds, or $\rho x \sqcap S = [a, b] \neq \perp$, but in this case $\llbracket x \in S \rrbracket \rho = \rho[x \mapsto \rho x \sqcap S]$ and we can observe that both the following hold:

$$\begin{aligned} \min(\rho) - (x \in S)_b &\leq \min(\rho) \leq \min(\rho x \sqcap S) \\ \max(\rho x \sqcap S) &\leq \max(\rho) \leq \max(\rho) + (x \in S)_+^b \end{aligned}$$

hence

$$\llbracket x \in S \rrbracket \rho = \rho[x \mapsto \rho x \sqcap S] = \dot{\alpha}_{\ell,u}(\llbracket x \in S \rrbracket \rho) = \llbracket x \in S \rrbracket_\ell^u \rho$$

which is our thesis.

Case $(x := k)$. In this case we have to prove that $\llbracket x := k \rrbracket \rho \sqsupseteq \llbracket x := k \rrbracket_\ell^u \rho$. Recall that we are considering $\ell \leq \min(\rho) - (x := k)_b^+ = \min(\rho) - (x := k)_b$ and $u \geq \max(\rho) + (x := k)_+^b$. We can notice similarly to the previous case, that because of the values of ℓ and u it holds that

$$\llbracket x := k \rrbracket \rho = \rho[x \mapsto [k, k]] = \dot{\alpha}_{\ell,u}(\llbracket x := k \rrbracket \rho) = \llbracket x := k \rrbracket_\ell^u \rho$$

hence our thesis holds.

Case $(x := y + k)$. In this case we have to prove that $\llbracket x := y + k \rrbracket \rho \sqsupseteq \llbracket x := y + k \rrbracket_\ell^u \rho$. Recall that we are considering $\ell \leq \min(\rho) - (x := y + k)_b^+ = \min(\rho) - (x := y + k)_b$ and $u \geq \max(\rho) + (x := y + k)_+^b$. Notice also that $(x := y + k)_b^+ = k = (x := y + k)_+^b$ and since $\llbracket x := y + k \rrbracket \rho = \rho[x \mapsto \rho y + k]$ we can notice that for each variable $w \in \text{Var}$ it holds that

$$\begin{aligned} \min(\rho) - k &\leq \min(\rho[x \mapsto \rho y + k]w) \\ \max(\rho) + k &\geq \max(\rho[x \mapsto \rho y + k]w) \end{aligned}$$

hence

$$\llbracket x := y + k \rrbracket_\ell^u \rho = \dot{\alpha}_{\ell,u}(\rho[x \mapsto \rho y + k]) = \rho[x \mapsto \rho y + k] = \llbracket x := y + k \rrbracket \rho$$

which is our thesis

Case $(C_1 + C_2)$. In this case we have to prove that $\llbracket C_1 + C_2 \rrbracket \rho \sqsupseteq \llbracket C_1 + C_2 \rrbracket_\ell^u \rho$. Recall that we are considering $\ell \leq \min(\rho) - (C_1 + C_2)_b^+ = \min(\rho) - (C_1 + C_2)_b$ and $u \geq \max(\rho) + (C_1 + C_2)_+^b$. By inductive hypothesis it holds that

$$\llbracket C_1 \rrbracket \rho = \llbracket C_1 \rrbracket_\ell^u \rho$$

for all $\ell \leq \min(\rho) - (C_1)_b^+$ and $u \geq \max(\rho) + (C_1)_+^b$. Again by inductive hypothesis it holds that

$$\llbracket C_2 \rrbracket \rho = \llbracket C_2 \rrbracket_\ell^u \rho$$

for all $\ell \leq \min(\rho) - (C_2)_b^+$ and $u \geq \max(\rho) + (C_2)_+^b$. In particular, both hold for

$$\begin{aligned} \ell &\leq \min(\rho) - (C_1)_b^+ - (C_2)_b^+ = \min(\rho) - (C_1 + C_2)_b^+ \\ u &\geq \max(\rho) + (C_1)_+^b + (C_2)_+^b = \max(\rho) + (C_1 + C_2)_+^b \end{aligned}$$

i.e., our initial choice of ℓ, u . We can conclude by closure over \sqcup

$$\llbracket C_1 + C_2 \rrbracket \rho = \llbracket C_1 \rrbracket \rho \sqcup \llbracket C_2 \rrbracket \rho = \llbracket C_1 \rrbracket_\ell^u \rho \sqcup \llbracket C_2 \rrbracket_\ell^u \rho = \llbracket C_1 + C_2 \rrbracket_\ell^u \rho$$

which is our thesis.

Case $(C_1; C_2)$. In this case we have to prove that $\llbracket C_1; C_2 \rrbracket \rho \sqsupseteq \llbracket C_1; C_2 \rrbracket_\ell^u \rho$ for all $\ell \leq \min(\rho) - (C_1; C_2)_b^+$ and $u \geq \max(\rho) + (C_1; C_2)_+^b$. Recall that $\llbracket C_1; C_2 \rrbracket \rho = (\llbracket C_2 \rrbracket \circ \llbracket C_1 \rrbracket) \rho$. By inductive hypothesis it holds that

$$\llbracket C_1 \rrbracket \rho = \llbracket C_1 \rrbracket_\ell^u \rho \quad \forall \ell \leq \min(\rho) - (C_1)_b^+ \wedge u \geq \max(\rho) + (C_1)_+^b \quad (4.14)$$

$$\llbracket C_2 \rrbracket \rho' = \llbracket C_2 \rrbracket_{k_3}^{k_4} \rho' \quad \forall k_3 \leq \min(\rho') - (C_2)_b^+ \wedge k_4 \geq \max(\rho') + (C_2)_+^b \quad (4.15)$$

where $\rho' = \llbracket C_1 \rrbracket \rho$. In particular notice that both (4.14) and (4.15) hold for all n, m s.t.

$$\begin{aligned} m &\leq \min(\rho) - (C_1)_b^+ - (C_2)_b^+ \\ n &\geq \max(\rho) + (C_1)_+^b + (C_2)_+^b \end{aligned}$$

Hence

$$\llbracket C_1; C_2 \rrbracket \rho = (\llbracket C_2 \rrbracket \circ \llbracket C_1 \rrbracket) \rho = (\llbracket C_2 \rrbracket_m^n \circ \llbracket C_1 \rrbracket_m^n) \rho = \llbracket C_1; C_2 \rrbracket_m^n \rho$$

which is our thesis.

Case $(\text{fix}(C))$. What we want to prove in this case is that $\llbracket \text{fix}(C) \rrbracket \rho \sqsupseteq \llbracket \text{fix}(C) \rrbracket_\ell^u \rho$ for all $\ell \leq \min(\rho) - (\text{fix}(C))_b^+$ and $u \geq \max(\rho) + (\text{fix}(C))_+^b$. Recall that by Lemma 3.4 $\llbracket \text{fix}(C) \rrbracket$ is syntactic sugar for $\llbracket (C + \text{true})^* \rrbracket$, therefore

$$\llbracket \text{fix}(C) \rrbracket \rho = \llbracket (C + \text{true})^* \rrbracket \rho = \bigsqcup_{i \in \mathbb{N}} (\llbracket C + \text{true} \rrbracket)^i \rho \quad (4.16)$$

$$\llbracket \text{fix}(C) \rrbracket_\ell^u \rho = \llbracket (C + \text{true})^* \rrbracket_\ell^u \rho = \bigsqcup_{i \in \mathbb{N}} (\llbracket C + \text{true} \rrbracket_\ell^u)^i \rho \quad (4.17)$$

By latter equation we want to prove that for every $i \in \mathbb{N}$ it holds that

$$\llbracket \text{fix}(C) \rrbracket \rho \sqsupseteq (\llbracket C + \text{true} \rrbracket_\ell^u)^i \rho \quad (4.18)$$

Case $(i = 0)$. In this case we can observe that our thesis

$$\llbracket \text{fix}(C) \rrbracket \rho \sqsupseteq (\llbracket \text{fix}(C) \rrbracket_\ell^u)^0 \rho = \text{id}(\rho) = \rho$$

holds by (4.16).

Case $(i \implies i + 1)$. In this case we can first notice that

$$\begin{aligned} \llbracket C + \text{true} \rrbracket (\llbracket \text{fix}(C) \rrbracket \rho) &= \llbracket C \rrbracket (\llbracket \text{fix}(C) \rrbracket \rho) \sqcup (\llbracket \text{fix}(C) \rrbracket \rho) && \text{by definition of } C + \text{true} \\ &= \llbracket C \rrbracket (\text{lfp}(\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu)) \sqcup (\llbracket \text{fix}(C) \rrbracket \rho) \end{aligned} \quad (4.19)$$

by definition of $\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu$ it holds that $\text{lfp}(\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu) \sqsupseteq \rho$, hence

$$\begin{aligned} \llbracket C \rrbracket (\text{lfp}(\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu)) &= \rho \sqcup \llbracket C \rrbracket (\text{lfp}(\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu)) \\ &= \text{lfp}(\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu) \\ &= \llbracket \text{fix}(C) \rrbracket \rho \end{aligned} \quad (4.20)$$

therefore in (4.19)

$$\begin{aligned} \llbracket C \rrbracket (\text{lfp}(\lambda \mu. \rho \sqcup \llbracket C \rrbracket \mu)) \sqcup (\llbracket \text{fix}(C) \rrbracket \rho) &= \llbracket \text{fix}(C) \rrbracket \rho \sqcup \llbracket \text{fix}(C) \rrbracket \rho && \text{by (4.20)} \\ &= \llbracket \text{fix}(C) \rrbracket \rho. \end{aligned}$$

We can now continue. By calling $\llbracket \text{fix}(C) \rrbracket \rho = \beta$ we have to prove that

$$\llbracket C + \text{true} \rrbracket \beta \sqsupseteq \llbracket C + \text{true} \rrbracket_\ell^u \beta. \quad (4.21)$$

for all $\ell \leq \min(\rho) - (\text{fix}(C))_b^+$ and $u \geq \max(\rho) + (\text{fix}(C))_+^b$. In other words what we want to prove is that for every $y \in \text{Var}_C$ both

$$\begin{aligned} \max(\llbracket C + \text{true} \rrbracket \beta y) &\leq u \\ \min(\llbracket C + \text{true} \rrbracket \beta y) &\geq \ell \end{aligned}$$

To start notice that $\max(\beta y) \leq \max(\rho) + (\text{fix}(\mathbf{C}))^b$ by Lemma 4.6. Hence by Definition 4.15 $\max(\beta) \leq \max(\rho) + (\text{fix}(\mathbf{C}))^b$, and by calling $n = \text{vars}(\mathbf{C})$ we can notice the following:

$$\begin{aligned}
 \max(\llbracket \mathbf{C} + \text{true} \rrbracket \beta) &\leq \max(\beta) + (\mathbf{C})^b && \text{by Lemma 4.6} \\
 &\leq \max(\rho) + (\text{fix}(\mathbf{C}))^b + (\mathbf{C})^b \\
 &= \max(\rho) + (n + 2)(\mathbf{C})^b \\
 &\leq \max(\rho) + (n + 2)(\mathbf{C})_+^b \\
 &= \max(\rho) + (\text{fix}(\mathbf{C}))_+^b = u
 \end{aligned}$$

A similar procedure can be applied on the minimum to observe that

$$\min(\llbracket \mathbf{C} + \text{true} \rrbracket \beta) \geq \min(\rho) - (\mathbf{C})_b^+ = \ell$$

Hence we can conclude by observing that

$$\begin{aligned}
 \beta &= \llbracket \mathbf{C} + \text{true} \rrbracket \beta \supseteq \llbracket \mathbf{C} + \text{true} \rrbracket_\ell^u \beta && \text{by (4.21)} \\
 &\supseteq \llbracket \mathbf{C} + \text{true} \rrbracket_\ell^u (\llbracket \mathbf{C} + \text{true} \rrbracket_\ell^u)^i \rho && \text{by induction on } i \\
 &= (\llbracket \mathbf{C} + \text{true} \rrbracket_\ell^u)^{i+1} \rho
 \end{aligned}$$

Therefore for all $i \in \mathbb{N}$ $\llbracket \text{fix}(\mathbf{C}) \rrbracket \rho \supseteq (\llbracket \mathbf{C} + \text{true} \rrbracket_\ell^u)^i \rho$. By this we can deduce that

$$\beta = \llbracket \text{fix}(\mathbf{C}) \rrbracket \rho \supseteq \bigsqcup_{i \in \mathbb{N}} (\llbracket \mathbf{C} + \text{true} \rrbracket_\ell^u)^i \rho = \llbracket \text{fix}(\mathbf{C}) \rrbracket_\ell^u \rho$$

which is our thesis. □

Our last theorem proved that by bounding the interval domain according to the constants that appear in a program and its initial state we can ensure termination of the analysis while achieving the most precise abstract invariant for the program, effectively solving Problem 4.1. The result is analogous to the findings of [Gaw+09], but we achieve it by bounding the abstract domain. Notice that we achieved such result only by looking at maximal and minimal values of the analysis w.r.t. the considered program \mathbf{C} and the initial environment $\eta \in \mathbb{I}$. By this observation one might think that the same result would be possible to achieve for the non-relational collecting semantics $\langle\langle \cdot \rangle\rangle$. While our guess is that such thing should be possible, in the next section we will provide some examples that show that by using a simple Kleene iteration technique it is not possible to infer the most precise abstract invariant in \mathbb{C} , it is only possible to infer the termination of the abstract interpreter $\langle\langle \cdot \rangle\rangle$.

4.4 Bounded non-relational collecting semantics

In this section on non-relational collecting semantics, we demonstrate that while it is not possible to compute the most precise invariant for any given program in Imp , it is feasible to compute a more abstract semantics. This abstract semantics provides valuable insights into the termination behavior of the non-relational collecting analyzer. Specifically, we show that the two abstract semantics align when the abstract analyzer for the non-relational collecting semantics terminates.

For an easier reading, we will refer to $\llbracket \cdot \rrbracket^{\mathbf{C}}$ with the same notation we used in Section 3.2.1: $\langle\langle \cdot \rangle\rangle = \llbracket \cdot \rrbracket^{\mathbf{C}}$.

Lemma 4.18. *Let $\mathbf{C} \in \text{Imp}$. For all $\eta \in \mathbb{C}$ and $y \in \text{Var}$, if $\max(\langle\langle \mathbf{C} \rangle\rangle \eta y) \neq +\infty$ and $\max(\langle\langle \mathbf{C} \rangle\rangle \eta y) > (\mathbf{C})^b$ then there exist a variable $z \in \text{Var}$ and an integer $h \in \mathbb{Z}$ such that $|h| \leq (\mathbf{C})^b$ and the following two properties hold:*

$$(i) \max(\langle\langle C \rangle\rangle \eta y) = \max(\eta z) + h;$$

$$(ii) \text{ for all } \eta' \in \mathbb{C}, \text{ if } \eta' \dot{\supseteq} \eta \text{ then } \max(\langle\langle C \rangle\rangle \eta' y) \geq \max(\eta' z) + h.$$

Proof. The proof is left in Appendix A.2, since it is analogous to the proof of Lemma 4.6. \square

Remark 4.19. The key point is that in the base case ($x \in I$), we can draw similar conclusions as we did for intervals. This is because the filtering occurs on an interval $I \in \mathbb{I}$ rather than on an arbitrary decidable set. It is important to note that if we were to allow a generic decidable set in a guard, the result would not hold. Specifically, let $S \in \wp(\mathbb{Z})$ be a decidable set and consider $y = x$; under these conditions, we observe that

$$\max(\langle\langle x \in S \rangle\rangle \eta y) = \max(\eta[x \mapsto \eta x \cap S]x) = \max(\eta x \cap S) \quad (4.22)$$

Since S is generally non-convex what happens is that from $\eta x \cap S \neq \emptyset$ and $\max(S) = +\infty$ we can only deduce

$$\max(\eta x \cap S) \leq \max(\eta x) \quad (4.23)$$

and not equality, providing a potential counterexample to the Lemma. For example consider the program ($x \in \mathbb{P}$) where \mathbb{P} is the set of even numbers and an initial environment $\eta \triangleq [x \mapsto \mathbb{D} \cup \{2\}]$, where \mathbb{D} is the set of odd numbers. Then

$$\langle\langle x \in \mathbb{P} \rangle\rangle \eta x = \{2\}$$

and $\max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) = 2$, while $(\langle\langle x \in \mathbb{P} \rangle\rangle)^b = 0$.

Hence both $\max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) \neq +\infty$ and $\max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) > (\langle\langle x \in \mathbb{P} \rangle\rangle)^b$ hold. The lemma would state that $\exists w \in \text{Var}$ and $h \in \mathbb{Z} \mid |h| \leq (x \in \mathbb{P})^b$ s.t.

$$(i) \max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) = \max(\eta w) + h$$

$$(ii) \forall \eta' \supseteq \eta \quad \max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta' y) \geq \max(\eta' w) + h$$

Here we can observe that the hypothesis would hold:

$$\max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) = 2 \neq +\infty$$

$$\max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) > (x \in \mathbb{P})^b = 0$$

By $(\cdot)^b$ definition

but (i) does not hold, in fact $\max(\eta x) = 0$ since $\eta x = \mathbb{D} \cup \{2\}$ and therefore

$$\max(\langle\langle x \in \mathbb{P} \rangle\rangle \eta x) = 2 \neq 0 = \max(\eta x) + (x \in \mathbb{P})^b$$

With intervals the theorem instead holds, the rough idea is that the intersection between an interval and a generic subset of \mathbb{Z} S preserves the maximal element of S whenever the interval is open on the right, and preserves the minimal element when it is open on the left, otherwise the preconditions for the lemma do not hold. This is not the case for generic subsets of \mathbb{Z} in the guard as we showed with this remark.

The same applies for the increment on the lower bound, in a similar fashion as for the intervals:

Lemma 4.20. *Let $C \in \text{Imp}$. For all $\eta \in \mathbb{C}$ and $y \in \text{Var}$, if $\min(\langle\langle C \rangle\rangle \eta y) \neq -\infty$ and $\min(\langle\langle C \rangle\rangle \eta y) < -(C)^b$ then there exist a variable $z \in \text{Var}$ and an integer $h \in \mathbb{Z}$ such that $|h| \leq (C)^b$ and the following two properties hold:*

$$(i) \min(\langle\langle C \rangle\rangle \eta y) = \min(\eta z) + h;$$

$$(ii) \text{ for all } \eta' \in \mathbb{C}, \text{ if } \eta' \dot{\supseteq} \eta \text{ then } \min(\langle\langle C \rangle\rangle \eta' y) \leq \min(\eta' z) + h.$$

The proof is again left in Appendix A.3 as Lemma A.4.

4.5 Computing non-relational collecting semantics

In this section, we demonstrate the ability to compute a further abstraction of non-relational collecting semantics to infer the termination of non-relational collecting analysis. We illustrate that merely bounding individual variables, as done with intervals, is insufficient. Intuitively, we can no longer deduce that when one variable diverges, it does not impact the behavior of other variables. To start, we rely on Definition 4.15 of min and max values of an abstract environment ρ to bound the non relational collecting domain \mathbb{C} in this way:

Definition 4.21 (Bounded non-relational collecting domain). We define $\mathbb{C}_\ell^u \triangleq \text{Var}_C \rightarrow \wp^*(\mathbb{Z})_\ell^u$ where

$$\wp^*(\mathbb{Z})_\ell^u = \{S \subseteq \mathbb{Z} \mid S \neq \emptyset \wedge \forall x \in S \quad \ell \leq x \leq u\} \cup \{\top\}$$

Notice that differently from what happened in the case of intervals, for the domain \mathbb{C} we have no appropriate way of representing an unbounded element: unbounded intervals could be represented with elements with the shape $[a, +\infty]$, $[-\infty, b]$ with $a, b \in \mathbb{Z}$. For arbitrary subsets of \mathbb{Z} we have instead to rely on a single \top element.

A first observation we can make is that by definition there are no infinite ascending nor descending chains, as every chain is bounded by above by some $u \in \mathbb{Z}$ and below by some $\ell \in \mathbb{Z}$. Moreover, there is a Galois connection between this abstract domain and its unbounded counterpart $\wp(\mathbb{Z})$. First let's define the concretization and abstraction maps

Definition 4.22. Let $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$. We define a concretization map $\gamma_{\ell,u}^* : \wp^*(\mathbb{Z})_\ell^u \rightarrow \wp^*(\mathbb{Z})$ as the identity function

$$\gamma_{\ell,u}^* = \text{id}$$

similarly we define an abstraction map $\alpha_{\ell,u}^* : \wp^*(\mathbb{Z}) \rightarrow \wp^*(\mathbb{Z})_\ell^u$ in the following way

$$\alpha_{\ell,u}^*(S) = \begin{cases} S & \text{if } \inf(S) \geq \ell \wedge \sup(S) \leq u \\ \top & \text{otherwise} \end{cases}$$

Lemma 4.23. Given $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$.

$$\langle \wp^*(\mathbb{Z}), \subseteq \rangle \xleftrightarrow[\alpha_{\ell,u}^*]{\gamma_{\ell,u}^*} \langle \wp^*(\mathbb{Z})_\ell^u, \subseteq \rangle$$

i.e., $\langle \alpha_{\ell,u}^*, \wp^*(\mathbb{Z}), \wp^*(\mathbb{Z})_\ell^u, \gamma_{\ell,u}^* \rangle$ is a Galois connection.

Proof. The proof consists in showing that $\gamma_{\ell,u}^*$ and $\alpha_{\ell,u}^*$ satisfy the properties of Theorem 1.18:

- (1) $\alpha_{\ell,u}^*, \text{id}$ are monotonic;
- (2) $\text{id} \circ \alpha_{\ell,u}^*$ is extensive, i.e., $\sigma \subseteq \alpha_{\ell,u}^*(\sigma)$ for all $\sigma \in \wp^*(\mathbb{Z})$;
- (3) $\alpha_{\ell,u}^* \circ \gamma^*$ is reductive, i.e., $\alpha_{\ell,u}^*(\sigma_b) \subseteq \sigma_b$ for all $\sigma_b \in \wp^*(\mathbb{Z})_\ell^u$.

To start let's prove (1). Of course id is monotone by definition. For $\alpha_{\ell,u}^*$ we have to prove that given any $\sigma, \tau \in \wp^*(\mathbb{Z})$ s.t. $\sigma \subseteq \tau$ it holds that $\alpha_{\ell,u}^*(\sigma) \subseteq \alpha_{\ell,u}^*(\tau)$. Notice that since $\sigma \subseteq \tau$ it holds that $\max(\sigma) \leq \max(\tau)$ and $\min(\sigma) \geq \min(\tau)$, which means by Definition 4.22 $\alpha_{\ell,u}^*\sigma \subseteq \alpha_{\ell,u}^*\tau$, which is our thesis.

Both (2) and (3) follow from Definition 4.22. For (2) for all $\sigma \in \wp^*(\mathbb{Z})$ either $\alpha_{\ell,u}^*(\sigma) = \sigma$ or $\alpha_{\ell,u}^*(\sigma) = \top$, hence in both cases $\sigma \subseteq \alpha_{\ell,u}^*(\sigma)$ holds. For (3), for all $\sigma_b \in \wp^*(\mathbb{Z})_\ell^u$ it holds that $\max(\sigma_b) \leq u$ and $\min(\sigma_b) \geq \ell$, hence

$$\alpha_{\ell,u}^*(\sigma_b) = \sigma_b \tag{4.24}$$

and therefore $\alpha_{\ell,u}^*(\sigma_b) \subseteq \sigma_b$ holds. Moreover, (4.24) means that for all $\sigma_b \in \wp^*(\mathbb{Z})_\ell^u$ that $\alpha_{\ell,u}^* \circ \text{id} = \text{id}$, which means by Definition 1.20 that we formed a Galois insertion:

$$\langle \wp^*(\mathbb{Z}), \subseteq \rangle \xleftrightarrow[\alpha_{\ell,u}^*]{\gamma_{\ell,u}^*} \langle \wp^*(\mathbb{Z})_\ell^u, \subseteq \rangle \quad \square$$

Notice that since \mathbb{C} and \mathbb{C}_ℓ^u are respectively the point-wise lifting of $\wp^*(\mathbb{Z})$ and $\wp^*(\mathbb{Z})_\ell^u$ there is also a Galois insertion between them:

$$\langle \mathbb{C}, \dot{\subseteq} \rangle \xleftarrow[\alpha_{\ell,u}]{\gamma_{\ell,u}} \langle \mathbb{C}_\ell^u, \dot{\subseteq} \rangle$$

Where $\alpha_{\ell,u}(\eta) = \lambda x. \alpha_{\ell,u}^*(\eta x)$ and $\gamma_{\ell,u} = \text{id}$. Since we have a Galois connection between the non relational collecting domain \mathbb{C} and its bounded version \mathbb{C}_ℓ^u we can define an abstract inductive semantics which is sound by construction:

Definition 4.24 (Bounded non-relational collecting semantics). Let $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$. We define basic expressions over the bounded non relational collecting semantics $((\cdot))^{\mathbb{C}_\ell^u} : \text{Exp} \rightarrow \mathbb{C}_\ell^u \rightarrow \mathbb{C}_\ell^u$ as

$$((e))^{\mathbb{C}_\ell^u} \triangleq \alpha_{\ell,u} \circ ((e))^{\mathbb{C}}$$

i.e. the best correct abstraction.

Lemma 4.25 (Bounded non-relational collecting is sound). Let $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$. For all $\eta^\sharp \in \mathbb{C}_\ell^u$ it holds that

$$(\langle \mathbb{C} \rangle \circ \text{id}) \eta^\sharp \dot{\subseteq} (\text{id} \circ \langle \mathbb{C} \rangle_\ell^u) \eta^\sharp$$

i.e., $\langle \cdot \rangle_\ell^u$ is sound w.r.t. $\langle \cdot \rangle$.

Proof. The proof follows from the fact that $\langle \cdot \rangle_\ell^u$ is defined as the bca on basic expressions over \mathbb{C} and there is a Galois connection

$$\langle \mathbb{C}, \dot{\subseteq} \rangle \xleftarrow[\alpha_{\ell,u}]{\gamma_{\ell,u}} \langle \mathbb{C}_\ell^u, \dot{\subseteq} \rangle$$

Hence by Lemma 3.3 our thesis

$$(\langle \mathbb{C} \rangle \circ \text{id}) \eta^\sharp \dot{\subseteq} (\text{id} \circ \langle \mathbb{C} \rangle_\ell^u) \eta^\sharp$$

holds. □

By using ℓ, u properly, we can introduce a notion of order between bounded non relational collecting domain. More in detail, given $a, b, c, d \in \mathbb{Z}$ s.t. $a \leq b$ and $c \leq d$. Then \preceq is a relation order s.t.

$$\mathbb{C}_a^b \preceq \mathbb{C}_c^d \iff a \leq c \wedge d \leq b$$

We bounded our analysis the same way we did with interval analysis in Definition 4.8. This initial solution however has a problem. Consider the following code snippet:

```

1  /* Pa */
2  x := 0
3  y := 0
4  /* Pb */
5  while (x < 1)
6    x := x + 1
7    y := y + 2
8  /* Pc */
9  if (y = 1)
10   x := 2

```

Code 4.2: Snippet where bounded analysis diverges from the unbounded counterpart

This example highlights the main problem of just considering the bounds for each variable. When a variable exceeds the program bound and reaches \top this implies a loss of information, which leads to a difference between the non-relational collecting semantics and its bounded counterpart, even when the first one stays inside the bounds.

To highlight the problem let's compute the exact semantics $\langle \mathbb{P} \rangle$, to better address the problem, consider $P = P_a; P_b; P_c$ and an initial environment $[x \mapsto \top, y \mapsto \top] = \eta$.

$\langle P_a \rangle \eta$ infers the invariant $\eta_a = [x \mapsto \{0\}, y \mapsto \{0\}]$. This is our starting point.

$\langle\langle P_b \rangle\rangle \eta_a$ infers the invariant $\eta_b = [x \mapsto \{0, 1\}, y \mapsto \mathbb{P} \cap \mathbb{N}]$. The analysis filters for all elements greater than 1 for the variable x , while variable y grows indefinitely, each time increasing by 2, which means that it can assume any positive even number, hence $\mathbb{P} \cap \mathbb{N}$.

$\langle\langle P_c \rangle\rangle \eta_b$ infers the final result

$$\eta_c = [x \mapsto \{0, 1\}, y \mapsto \mathbb{P} \cap \mathbb{N}]$$

The guard $y = 1$ in the **if** statement filters for $\mathbb{P} \cap \mathbb{N} \cap \{1\} = \emptyset$ and therefore the variable x is never assigned to 2, leaving the result unchanged.

Let's consider instead an hypothetical analysis over \mathbb{C}_ℓ^u , with ℓ, u chosen accordingly to the program.

$\langle\langle P_b \rangle\rangle_\ell^u \eta_a$ infers $\eta'_b = [x \mapsto \{0, 1\}, y \mapsto \top]$, since at some point y will exceed the bound, hence the iteration will widen to \top .

$\langle\langle P_c \rangle\rangle_\ell^u \eta'_b$ now has to filter again for the guard $y = 1$, but in this case $\top \cap \{1\} = \{1\}$, hence the final invariant becomes

$$\eta'_c = [x \mapsto \{0, 1, 2\}, y \mapsto \top]$$

hence $\eta'_c \neq \eta_c$, which means that $\langle\langle \cdot \rangle\rangle_\ell^u$ and $\langle\langle \cdot \rangle\rangle$ diverge even when some variable in $\langle\langle \cdot \rangle\rangle$ stays inside the program bounds.

We hypothesize that it is feasible to deduce the exact infinite invariant, given that all the necessary information for its generation is syntactically accessible. Prior research on this topic, such as the work of [Lef+24], primarily addresses Presburger arithmetic, which is beyond the scope of this thesis.

For this reason our approach consists in smashing the \top element of our analysis. Remember that the original problem we want to solve (roughly) is the non-termination of the analysis,

Definition 4.26 (Smashed \top non relational collecting). Let

$$\wp(\mathbb{Z})_\ell^u \triangleq \{S \subseteq \mathbb{Z} \mid S \neq \emptyset \wedge \forall x \in S \quad \ell \leq x \leq u\}.$$

We define $\overline{\mathbb{C}}_\ell^u$ as

$$\overline{\mathbb{C}}_\ell^u \triangleq (Var \rightarrow \wp(\mathbb{Z})_\ell^u) \cup \{\perp, \top\}$$

we can build a Galois connection with \mathbb{C}_ℓ^u for some fixed $\ell, u \in \mathbb{Z}$:

Definition 4.27. Let $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$, $\eta \in \mathbb{C}_\ell^u$ and $\bar{\eta} \in \overline{\mathbb{C}}_\ell^u$. Then the abstraction map $\bar{\alpha}_{\ell, u} : \mathbb{C}_\ell^u \rightarrow \overline{\mathbb{C}}_\ell^u$ is defined as

$$\bar{\alpha}_{\ell, u}(\eta) = \begin{cases} \top & \text{if } \exists x \in Var \text{ s.t. } \eta x = \top \\ \eta & \text{otherwise} \end{cases}$$

while concretization map $\bar{\gamma}_{\ell, u} : \overline{\mathbb{C}}_\ell^u \rightarrow \mathbb{C}_\ell^u$ is defined as

$$\begin{aligned} \bar{\gamma}_{\ell, u}(\top) &= \lambda x \in Var. \top \\ \bar{\gamma}_{\ell, u}(\bar{\eta}) &= \bar{\eta} \end{aligned}$$

We can now define base expressions based on the bca with the bounded non relational collecting semantics:

Definition 4.28. Let $e \in \text{Exp}$. The semantics of base expressions over $\overline{\mathbb{C}}_\ell^u$ is defined as

$$\begin{aligned} ((e))_{\ell}^{\overline{\mathbb{C}}_\ell^u}(\top) &\triangleq \top \\ ((e))_{\ell}^{\overline{\mathbb{C}}_\ell^u}(\eta) &\triangleq (\bar{\alpha}_{\ell, u} \circ ((e))^{\mathbb{C}_\ell^u}) \eta \end{aligned}$$

Notice that this is *not* the b.c.a. on the basic expressions: the \top element is preserved. This is proved in Lemma 4.29.

Once again, $\llbracket \cdot \rrbracket_{\ell}^{\overline{\mathbb{C}}_{\ell}^u}$ is defined accordingly to the abstract inductive semantics of Definition 3.1. Notice that contrary to latter definition of \mathbb{C}_{ℓ}^u in this case we have a smashed top element. The idea is that whenever a variable diverges we infer that the whole precise analysis diverges, in order to solve Problem 4.1 and decide analysis termination. For simplicity, from now on we will refer to $\llbracket \cdot \rrbracket_{\ell}^{\overline{\mathbb{C}}_{\ell}^u}$ as $\overline{\llbracket \cdot \rrbracket}_{\ell}^u$.

We preliminarily prove a simple but useful property of $\overline{\llbracket \cdot \rrbracket}_{\ell}^u$: it preserves the \top element.

Lemma 4.29 ($\overline{\llbracket \cdot \rrbracket}_{\ell}^u$ preserves \top). *Let $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$ and $C \in \text{Imp}$*

$$\overline{\llbracket C \rrbracket}_{\ell}^u \top = \top$$

Proof. We proceed by induction on the program C to prove that

$$\overline{\llbracket C \rrbracket}_{\ell}^u \top = \top$$

Case (e). In the case of base expressions $e \in \text{Exp}$ the Lemma follows from the definition of base expressions over the top element.

$$\begin{aligned} \overline{\llbracket e \rrbracket}_{\ell}^u \top &= (\llbracket e \rrbracket_{\ell}^{\overline{\mathbb{C}}_{\ell}^u}(\top)) \\ &= \top \end{aligned} \quad \text{by Definition 4.28}$$

Case ($C_1 + C_2$). In this case

$$\begin{aligned} \overline{\llbracket C_1 + C_2 \rrbracket}_{\ell}^u \top &= \overline{\llbracket C_1 \rrbracket}_{\ell}^u \top \dot{\cup} \overline{\llbracket C_2 \rrbracket}_{\ell}^u \top \\ &= \top \dot{\cup} \top \\ &= \top \end{aligned} \quad \text{By inductive hypothesis}$$

Case ($C_1; C_2$). In this case

$$\begin{aligned} \overline{\llbracket C_1; C_2 \rrbracket}_{\ell}^u \top &= \overline{\llbracket C_2 \rrbracket}_{\ell}^u (\overline{\llbracket C_1 \rrbracket}_{\ell}^u \top) \\ &= \overline{\llbracket C_2 \rrbracket}_{\ell}^u \top \\ &= \top \end{aligned} \quad \begin{array}{l} \text{By induction on } C_1 \\ \text{By induction on } C_2 \end{array}$$

Case ($\text{fix}(C)$). In this case

$$\begin{aligned} \overline{\llbracket \text{fix}(C) \rrbracket}_{\ell}^u \top &= \text{lfp} \left(\lambda \mu. \top \dot{\cup} \overline{\llbracket C \rrbracket}_{\ell}^u \mu \right) \dot{\supseteq} \top \\ &= \top \end{aligned} \quad \begin{array}{l} \text{By definition of lfp} \\ \text{By definition of } \top \end{array}$$

□

Theorem 4.30. *Let $C \in \text{Imp}$ be a program. Then, for all finitely supported $\eta \in \overline{\mathbb{C}}_{\ell}^u$ and $\ell, u \in \mathbb{Z}$ s.t. $\mathbb{C}_{C,\eta} \preceq \mathbb{C}_{\ell}^u$, i.e., $\ell \leq \min(\eta) - (C)_b^+$ and $u \geq \max(\eta) + (C)_+^b$ then*

$$\overline{\llbracket C \rrbracket}_{\ell}^u \eta \neq \top \implies \llbracket C \rrbracket \eta = \overline{\llbracket C \rrbracket}_{\ell}^u \eta$$

i.e., if the analysis over $\overline{\mathbb{C}}_{\ell}^u$ does not diverge, then the analysis over \mathbb{C} converges to the same result.

Proof. The proof will proceed by induction on the program C , covering first the base cases of Exp expressions and then the inductive cases of Imp . Notice that because of Lemma 4.25

$$\llbracket C \rrbracket \overline{\gamma}_{\ell,u}(\overline{\eta}) \dot{\subseteq} \llbracket C \rrbracket_{\ell}^{\overline{\mathbb{C}}_{\ell}^u} \overline{\gamma}_{\ell,u}(\overline{\eta}) \dot{\subseteq} \overline{\gamma}_{\ell,u}(\overline{\llbracket C \rrbracket}_{\ell}^u \overline{\eta})$$

already holds for every $\ell, u \in \mathbb{Z}$ s.t. $\ell \leq u$ and $\bar{\eta} \in \overline{\mathbb{C}}_\ell^u$, hence what we have to prove for every case is that

$$\overline{\langle \mathbb{C} \rangle}_\ell^u \eta \neq \top \implies \langle \mathbb{C} \rangle \eta \supseteq \overline{\langle \mathbb{C} \rangle}_\ell^u \eta$$

First notice that it cannot be $\bar{\eta} = \top$, otherwise by Lemma 4.29 $\overline{\langle \mathbb{C} \rangle}_\ell^u \top = \top$ and therefore the hypothesis $\overline{\langle \mathbb{C} \rangle}_\ell^u \top \neq \top$ is not respected. Furthermore, notice that $\overline{\langle \mathbb{C} \rangle}_\ell^u \bar{\eta} \neq \top$ implies that $\bar{\gamma}_{\ell,u}(\overline{\langle \mathbb{C} \rangle}_\ell^u \bar{\eta}) = \overline{\langle \mathbb{C} \rangle}_\ell^u \bar{\eta}$, due to the definition of $\bar{\gamma}_{\ell,u}$.

Case (e). In this case we have to prove that

$$\overline{\langle \mathbf{e} \rangle}_\ell^u \bar{\eta} \neq \top \implies \langle \mathbf{e} \rangle \bar{\gamma}_{\ell,u}(\bar{\eta}) = \bar{\gamma}_{\ell,u}(\overline{\langle \mathbf{e} \rangle}_\ell^u \bar{\eta})$$

First, if $\bar{\eta} = \perp$ we can notice that

$$(\bar{\alpha}_{\ell,u} \circ \alpha_{\ell,u} \circ \langle \mathbf{e} \rangle) \perp = \perp = \langle \mathbf{e} \rangle \bar{\gamma}_{\ell,u}(\perp)$$

which is our thesis.

The last case is when $\top \neq \bar{\eta} \neq \perp$ and therefore for all $\mathbf{x} \in \text{Var}$ $\bar{\eta} \mathbf{x} \in \wp(\mathbb{Z})_\ell^u$. In this case by Lemma 4.18 and Lemma 4.20 for all $\mathbf{y} \in \text{Var}$ both

$$\begin{aligned} \max(\langle \mathbf{e} \rangle \bar{\gamma}_{\ell,u}(\bar{\eta}) \mathbf{y}) &\leq \max(\bar{\gamma}_{\ell,u}(\bar{\eta})) + (\mathbf{e})^b \leq \max(\bar{\gamma}_{\ell,u}(\bar{\eta})) + (\mathbf{e})_+^b = u \\ \min(\langle \mathbf{e} \rangle \bar{\gamma}_{\ell,u}(\bar{\eta}) \mathbf{y}) &\geq \min(\bar{\gamma}_{\ell,u}(\bar{\eta})) - (\mathbf{e})_b \geq \min(\bar{\gamma}_{\ell,u}(\bar{\eta})) - (\mathbf{e})_b^+ = \ell \end{aligned}$$

hence by definition of $\alpha_{\ell,u}$ and $\bar{\alpha}_{\ell,u}$

$$(\bar{\alpha}_{\ell,u} \circ \alpha_{\ell,u}) (\langle \mathbf{e} \rangle \bar{\gamma}_{\ell,u}(\bar{\eta})) = \langle \mathbf{e} \rangle \bar{\gamma}_{\ell,u}(\bar{\eta})$$

which is our thesis.

Case $(\mathbb{C}_1 + \mathbb{C}_2)$. In this case we have to prove that

$$\overline{\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle}_\ell^u \eta \neq \top \implies \langle \mathbb{C}_1 + \mathbb{C}_2 \rangle \eta = \overline{\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle}_\ell^u \eta$$

with $\ell \leq \min(\eta) - (\mathbb{C}_1 + \mathbb{C}_2)_b^+$ and $u \geq \max(\eta) + (\mathbb{C}_1 + \mathbb{C}_2)_+^b$. First we can notice that since $\overline{\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x} = \overline{\langle \mathbb{C}_1 \rangle}_\ell^u \eta \mathbf{x} \cup \overline{\langle \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x}$ our hypothesis $\overline{\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x} \neq \top$ implies both $\overline{\langle \mathbb{C}_1 \rangle}_\ell^u \eta \mathbf{x} \neq \top$ and $\overline{\langle \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x} \neq \top$. Hence by choice of ℓ and u we can use the inductive hypothesis and state that

$$\begin{aligned} \langle \mathbb{C}_1 \rangle \eta \mathbf{x} &= \overline{\langle \mathbb{C}_1 \rangle}_\ell^u \eta \mathbf{x} \\ \langle \mathbb{C}_2 \rangle \eta \mathbf{x} &= \overline{\langle \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x} \end{aligned}$$

and by closure over \cup

$$\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle \eta \mathbf{x} = \langle \mathbb{C}_1 \rangle \eta \mathbf{x} \cup \langle \mathbb{C}_2 \rangle \eta \mathbf{x} = \overline{\langle \mathbb{C}_1 \rangle}_\ell^u \eta \mathbf{x} \cup \overline{\langle \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x} = \overline{\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle}_\ell^u \eta \mathbf{x}$$

which is our thesis.

Case $(\mathbb{C}_1; \mathbb{C}_2)$. In this case we have to prove that

$$\overline{\langle \mathbb{C}_1; \mathbb{C}_2 \rangle}_\ell^u \eta \neq \top \implies \langle \mathbb{C}_1; \mathbb{C}_2 \rangle \eta = \overline{\langle \mathbb{C}_1; \mathbb{C}_2 \rangle}_\ell^u \eta$$

for all $\ell \leq \min(\eta) - (\mathbb{C}_1; \mathbb{C}_2)_b^+$ and $u \geq \max(\eta) + (\mathbb{C}_1; \mathbb{C}_2)_+^b$. First let's recall that

$$\overline{\langle \mathbb{C}_1; \mathbb{C}_2 \rangle}_\ell^u \bar{\eta} = \overline{\langle \mathbb{C}_2 \rangle}_\ell^u \left(\overline{\langle \mathbb{C}_1 \rangle}_\ell^u \bar{\eta} \right)$$

and we are under the hypothesis $\overline{\langle \mathbb{C}_1; \mathbb{C}_2 \rangle}_\ell^u \bar{\eta} \neq \top$, which means that $\overline{\langle \mathbb{C}_2 \rangle}_\ell^u \bar{\eta} \neq \top \neq \overline{\langle \mathbb{C}_2 \rangle}_\ell^u \bar{\eta}'$ where $\bar{\eta}' = \overline{\langle \mathbb{C}_1 \rangle}_\ell^u \bar{\eta}$, otherwise by Lemma 4.29 we would contradict the hypothesis $\overline{\langle \mathbb{C}_1 + \mathbb{C}_2 \rangle}_\ell^u \bar{\eta} \neq \top$.

\top . Then, by inductive hypothesis $\langle\langle C_1 \rangle\rangle \bar{\eta} = \overline{\langle\langle C_1 \rangle\rangle_\ell}^u \bar{\eta}$ for all $\ell \leq \min(\eta) - (C_1)_b^+$ and $u \geq \max(\eta) + (C_2)_+^b$. We can now call $\bar{\eta}' = \overline{\langle\langle C_1 \rangle\rangle_\ell}^u \bar{\eta}$ and by inductive hypothesis again:

$$\overline{\langle\langle C_2 \rangle\rangle_\ell}^u \bar{\eta}' \neq \top \implies \langle\langle C_2 \rangle\rangle \bar{\eta}' = \overline{\langle\langle C_2 \rangle\rangle_\ell}^u \bar{\eta}'$$

for all $\ell \leq \min(\eta') - (C_2)_b^+$ and $u \geq \max(\eta') + (C)_+^b$. Notice that $\min(\eta') \geq \min(\eta) - (C_1)_b^+$ and $\max(\eta') \leq \max(\eta) + (C_1)_+^b$ and therefore we can chose $\ell \leq \min(\eta) - (C_1)_b^+ - (C_2)_b^+$ and $u \geq \max(\eta) + (C_1)_+^b + (C_2)_+^b$. Since both inductive hypothesis hold, then following holds

$$\overline{\langle\langle C_1; C_2 \rangle\rangle_\ell}^u \bar{\eta} \neq \top \implies \langle\langle C_1; C_2 \rangle\rangle \bar{\eta} = \overline{\langle\langle C_1; C_2 \rangle\rangle_\ell}^u \bar{\eta}$$

which is our thesis.

Case $(\text{fix}(C))$. In this case we want to prove that

$$\overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta} \neq \top \implies \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} = \overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta}$$

for all $\ell \geq \min(\bar{\eta}) - (\text{fix}(C))_b^+$ and $u \leq \max(\bar{\eta}) + (\text{fix}(C))_+^b$. Recall that by Lemma 4.25 it always holds that

$$\langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \dot{\subseteq} \overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta}$$

We have therefore to prove that

$$\overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta} \neq \top \implies \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \dot{\supseteq} \overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta} \quad (4.25)$$

for all $\ell \leq \min(\bar{\eta}) - (\text{fix}(C))_b^+$ and $u \geq \max(\bar{\eta}) + (\text{fix}(C))_+^b$. To start notice that according to Lemma 3.4 $\langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} = \langle\langle (C + \text{true})^* \rangle\rangle \bar{\eta}$, hence we can alternatively prove that

$$\overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta} \neq \top \implies \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \dot{\supseteq} \bigcup_{i \in \mathbb{N}} \left(\overline{\langle\langle C + \text{true} \rangle\rangle_\ell}^u \right)^i \bar{\eta}$$

which implies Equation 4.25. To start we will initially prove that for every $i \in \mathbb{N}$ it holds that

$$\langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \neq \top \implies \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \dot{\supseteq} \left(\overline{\langle\langle C + \text{true} \rangle\rangle_\ell}^u \right)^i \bar{\eta}$$

to then prove the first one by closure over \cup . We will prove it by induction on i :

Case $(i = 0)$. In this case we have to prove that

$$\overline{\langle\langle \text{fix}(C) \rangle\rangle_\ell}^u \bar{\eta} \neq \top \implies \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \dot{\supseteq} \bar{\eta}$$

We can notice that by definition of $\langle\langle \text{fix}(C) \rangle\rangle$ the thesis holds.

Case $(i \implies i + 1)$. In this case we have to prove that

$$\langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \neq \top \implies \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \dot{\supseteq} \left(\overline{\langle\langle C \rangle\rangle_\ell}^u + \text{true} \right)^{i+1} \bar{\eta}$$

First we can notice that

$$\begin{aligned} \langle\langle C + \text{true} \rangle\rangle (\langle\langle \text{fix}(C) \rangle\rangle \bar{\eta}) &= \langle\langle C \rangle\rangle (\langle\langle \text{fix}(C) \rangle\rangle \bar{\eta}) \cup \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \\ &= \langle\langle C \rangle\rangle (\text{lfp}(\lambda \mu. \bar{\eta} \cup \langle\langle C \rangle\rangle \mu)) \cup \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \\ &= \bar{\eta} \cup \langle\langle C \rangle\rangle (\text{lfp}(\lambda \mu. \bar{\eta} \cup \langle\langle C \rangle\rangle \mu)) \cup \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} && \text{since } \bar{\eta} \subseteq \text{lfp}(\lambda \mu. \bar{\eta} \cup \langle\langle C \rangle\rangle \mu) \\ &= (\lambda \mu. \bar{\eta} \cup \langle\langle C \rangle\rangle \mu) (\text{lfp}(\lambda \mu. \bar{\eta} \cup \langle\langle C \rangle\rangle \mu)) \cup \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \\ &= (\text{lfp}(\lambda \mu. \bar{\eta} \cup \langle\langle C \rangle\rangle \mu)) \cup \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} && \text{by def. of lfp} \\ &= \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \cup \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \\ &= \langle\langle \text{fix}(C) \rangle\rangle \bar{\eta} \end{aligned}$$

Now we can preliminarily observe that by calling $\beta = \langle\langle \text{fix}(\mathbb{C}) \rangle\rangle \bar{\eta}$

$$\langle\langle \mathbb{C} + \text{true} \rangle\rangle \beta \supseteq \overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u \beta} \quad (4.26)$$

In fact, for all $\mathbf{x} \in \text{Var}$

$$\begin{aligned} \max(\langle\langle \mathbb{C} + \text{true} \rangle\rangle \beta \mathbf{x}) &\leq \max(\beta) + (\mathbb{C} + \text{true})^b = \max(\beta) + (\mathbb{C})^b \\ &\leq \max(\bar{\eta}) + (\text{fix}(\mathbb{C}))^b + (\mathbb{C})^b && \text{by Lemma 4.18} \\ &\leq \max(\bar{\eta}) + (n+2)(\mathbb{C})^b \\ &\leq \max(\bar{\eta}) + (n+2)(\mathbb{C})_+^b \\ &\leq \max(\bar{\eta}) + (\text{fix}(\mathbb{C}))_+^b = u \end{aligned}$$

similarly for the min value

$$\min(\langle\langle \mathbb{C} + \text{true} \rangle\rangle \beta \mathbf{x}) \geq \min(\bar{\eta}) - (\text{fix}(\mathbb{C}))_b^+ = \ell$$

hence

$$\begin{aligned} \beta &= \langle\langle \mathbb{C} + \text{true} \rangle\rangle \beta \supseteq \overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u \beta} && \text{by (4.26)} \\ &\supseteq \overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u \left(\overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u}^i \bar{\eta} \right)} && \text{by induction on } i \\ &= \left(\overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u} \right)^{i+1} \bar{\eta} \end{aligned}$$

Hence our thesis, for all $i \in \mathbb{N}$

$$\overline{\langle\langle \text{fix}(\mathbb{C}) \rangle\rangle_\ell^u \bar{\eta}} \top \implies \langle\langle \text{fix}(\mathbb{C}) \rangle\rangle \bar{\eta} \supseteq \left(\overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u} \right)^i \bar{\eta}$$

holds. We can now conclude by noticing that our original thesis

$$\overline{\langle\langle \text{fix}(\mathbb{C}) \rangle\rangle_\ell^u \bar{\eta}} \neq \top \implies \langle\langle \text{fix}(\mathbb{C}) \rangle\rangle \bar{\eta} \supseteq \bigcup_{i \in \mathbb{N}} \left(\overline{\langle\langle \mathbb{C} + \text{true} \rangle\rangle_\ell^u} \right)^i \bar{\eta} = \overline{\langle\langle \text{fix}(\mathbb{C}) \rangle\rangle_\ell^u \bar{\eta}}$$

also holds.

□

The latter theorem is a result similar to the result for the interval domain with Theorem 4.17. In its essence it states that when doing static analysis with abstract interpretation using the non relational collecting domain \mathbb{C} for some program $\mathbb{C} \in \text{Imp}$ and an initial environment $\eta \in \mathbb{C}$ we can consider a bounded version of the domain $\bar{\mathbb{C}}_\ell^u$ with $\ell = \min(\eta) - (\mathbb{C})_b^+$ and $u = \max(\eta) + (\mathbb{C})_+^b$ (hence computed accordingly to \mathbb{C} and η). Such domain has no infinite ascending or descending chains, hence the Kleene iteration method is guaranteed to halt in finite time. The result however is not as precise as the bounded version of the interval analysis: while for intervals we could compute the exact invariant, for the non-relational collecting domain this is the case only when the bounded semantics is not \top . Intuitively this happens when all invariants inferred in the computation are finite, i.e., if the analysis over \mathbb{C} halts in finite time. To put it differently, we can decide whether the analysis over \mathbb{C} halts in finite time or not.

CHAPTER 5

Conclusions

In this study, we investigated some decidability questions in the setting of abstract interpretation, the most prevalent framework for developing static analysis tools. The general question concerned the possibility of effectively computing the abstract semantics. The question is non trivial when the abstract domain can have infinite ascending chains and thus the fixpoint computations can diverge. We focused on a simple imperative language **Imp**, offering a set of constructs sufficient to make it Turing complete. We defined its concrete denotational semantics in its concrete domain \mathcal{C} and small step semantics. We first showed that, as expected, all non-trivial properties of the collecting semantics are undecidable due to Rice's Theorem. We then built the abstraction chain in Figure 5.1. First we abstracted program execution using the non-relational collecting semantics \mathbb{C} , the finest non-relational domain. Subsequently, we defined and established the fundamental properties of the intervals domain \mathbb{I} , a well-known domain in abstract interpretation literature. We proved that in the aforementioned abstract domains, variables are constrained to increase up to a limit denoted as $(C)^b$, which is dependent on the program C . We therefore introduced the bounded interval domain \mathbb{I}_ℓ^u : such domain does not contain infinite ascending and descending chains, and therefore fixpoints are computable in finite time. We then proved that given a program and an initial environment $C \in \text{Imp}, \eta \in \mathbb{I}$ and by crafting appropriate bounds for \mathbb{I}_ℓ^u , namely $\ell = \min(\eta) - (C)_b^+$ and $u = \max(\eta) + (C)_+^b$, we not only could compute $\llbracket C \rrbracket_{\mathbb{I}_\ell^u}^u \eta$ but also show that

$$\llbracket C \rrbracket_{\mathbb{I}_\ell^u}^u \eta = \llbracket C \rrbracket_{\mathbb{I}} \eta$$

This means that we were able to compute exact interval analysis in finite time. Then we tried to adopt the same technique to the non-relational collecting semantics. Indeed one can introduce a bounded version \mathbb{C}_ℓ^u of \mathbb{C} where a \top element is used for variables that exceed the bounds. However in this way we loose too much information to infer equality with the non-relational collecting semantics. Therefore we introduced $\overline{\mathbb{C}}_\ell^u$, bounded non-relational collecting with a smashed top element. With this analysis we could infer equality up to the top element, meaning that if the bounded analysis with smashed top resulted in something different than \top then its result was the same as the non-relational collecting, i.e.,

$$\overline{\llbracket C \rrbracket}_\ell^u \eta \neq \top \implies \llbracket C \rrbracket \eta = \overline{\llbracket C \rrbracket}_\ell^u \eta$$

Intuitively, this implies that we can decide the termination of non-relational collecting analysis. The bounded version with the smashed \top element, only infers this \top element if, during the analysis, a variable exceeds the predefined bounds and follows an infinite sequence. Consequently, the original non-relational collecting analysis does not terminate under these conditions using the Kleene iteration. While if something other than \top is inferred, it means that it is the most precise invariant.

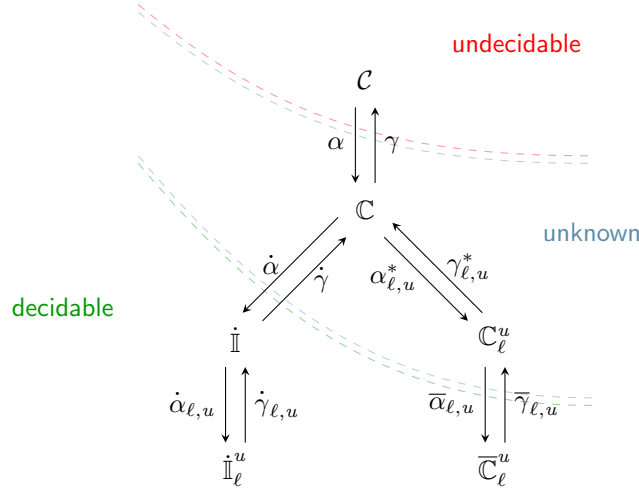


Figure 5.1: Abstractions chain we build throughout Chapter 3

Future work. Throughout the development of this thesis, we encountered various obstacles, which we present here as potential extensions and directions for future research. Addressing all of these issues was beyond the scope of this thesis due to time constraints.

The first and most prominent question is whether it is possible to precisely compute the non-relational collecting semantics. Our hypothesis is that it is indeed feasible, as suggested by the work of [Lef+24]. Their research builds on Presburger arithmetic from [Pre29], which, although beyond the scope of this thesis, presents interesting results, particularly its decidability.

A second follow up question might involve the extension of the language to support non-linear expressions. Our hypothesis in this case is that invariants might become undecidable, as we are able to write diophantine equations. Take for example the following code snippet

```

1   $\vec{x} := \text{null\_vec\_size}(n);$  /* where n is the number of
2                               variables in the diofrantine
3                               equation */
4  while (true) do
5       $(x_i++) + (x_i--)$ 
6       $z = P(\vec{x})$  /* Where  $P(\vec{x})$  is a diofantine equation */

```

Code 5.1: Program with diofantine equations

If the non-relational collecting semantics were decidable in a language that permitted non-linear equations, we could determine the solution of any Diophantine equation by deciding whether $0 \in \llbracket P_d \rrbracket$. However, this is known to be an undecidable problem, as demonstrated in [Mat70].

Finally, future work can be focused on the application to real-world scenarios. Aforementioned tools such as Astreé [Cou+05] and MOPSA [MOM23] use the traditional widening technique, which, while sound, does not infer the most precise invariant in the domain of intervals. In contrast, we have demonstrated that it is possible to achieve more precise invariants while ensuring the termination of the analyzer using the domain of intervals.

APPENDIX A

Additional proofs

A.1 Lemma 4.7 proof

We preliminarily observe that we can also prove a dual property of the Lemma 4.4:

Lemma A.1 (Negative cycles in weighted directed graphs). *Let p be a finite path*

$$p = x_0 \rightarrow_{h_0} x_1 \rightarrow_{h_1} x_2 \rightarrow_{h_2} \cdots \rightarrow_{h_{\ell-1}} x_\ell$$

with $m \triangleq \max\{|h_j| \mid j \in \{0, \dots, \ell-1\}\} \in \mathbb{N}$ and $w(p) < -(|X| - 1)m$. Then, p has a subpath which is a cycle having a strictly negative weight.

Proof. First note that $w(p) = \sum_{k=0}^{\ell-1} h_k < -m(|X| - 1)$ implies that $|p| = \ell \geq |X|$. Then, we show our claim by induction on $|p| = \ell \geq |X|$.

Case ($|p| = |X|$). Since the path p includes exactly $|X| + 1 = \ell + 1$ nodes, there exist indices $0 \leq i < j \leq \ell$ such that $x_i = x_j$, i.e., $p_{i,j}$ is a subpath of p which is a cycle. Moreover, since this cycle $p_{i,j}$ includes at least one edge, we have that

$$\begin{aligned} w(p_{i,j}) &= w(p) - (\sum_{k=0}^{i-1} h_k + \sum_{k=j}^{\ell-1} h_k) < && \text{as } w(p) < -m(|X| - 1) \\ &-m(|X| - 1) - (\sum_{k=0}^{i-1} h_k + \sum_{k=j}^{\ell-1} h_k) \leq && \text{as } \sum_{k=0}^{i-1} h_k + \sum_{k=j}^{\ell-1} h_k \geq -m(\ell - 1) \\ &-m(|X| - 1) - (-m(\ell - 1)) = && \text{as } \ell = |X| \\ &-m(|X| - 1) + m(|X| - 1) = 0 \end{aligned}$$

so that $w(p_{i,j}) < 0$ holds.

($|p| > |X|$): Since the path p includes at least $|X| + 2$ nodes, as in the base case, we have that p has a subpath which is a cycle. Then, we consider a cycle $p_{i,j}$ in p , for some indices $0 \leq i < j \leq \ell$, which is maximal, i.e., such that if $p_{i',j'}$ is a cycle in p , for some $0 \leq i' < j' \leq \ell$, then $p_{i,j}$ is not a proper subpath of $p_{i',j'}$.

If $w(p_{i,j}) < 0$ then we are done. Otherwise we have that $w(p_{i,j}) \geq 0$ and we consider the path p' obtained from p by stripping off the cycle $p_{i,j}$, i.e.,

$$p' \equiv \overbrace{x_0 \rightarrow_{h_0} x_1 \rightarrow_{h_1} \cdots \rightarrow_{h_{i-1}} x_i}^{p'_{0,i}} = \overbrace{x_j \rightarrow_{h_{j+1}} \cdots \rightarrow_{h_{\ell-1}} x_\ell}^{p'_{j+1,\ell}}$$

Since $|p'| < |p|$ and $w(p') = w(p) - w(p_{i,j}) \leq w(p) < -m(|X| - 1)$, we can apply the inductive hypothesis on p' . We therefore derive that p' has a subpath q which is a cycle having strictly positive weight. This cycle q is either entirely in $p'_{0,i}$ or in $p'_{j+1,\ell}$, otherwise q would include the cycle $p_{i,j}$ thus contradicting the maximality of $p_{i,j}$. Hence, q is a cycle in the original path p having a strictly negative weight. \square

For the following proof consider the $\min : \mathbb{I} \rightarrow \mathbb{Z}$ function, inductively defined as follows:

$$\begin{aligned}\min(\perp) &= +\infty \\ \min([a, b]) &= a\end{aligned}$$

and recall the Lemma 4.7 statement:

Lemma A.2. *Let $C \in \text{Imp}$.*

For all $\eta \in \mathbb{I}$ and $y \in \text{Var}$, if $\min(\llbracket C \rrbracket \eta y) \neq -\infty$ and $\min(\llbracket C \rrbracket \eta y) < -(C)_b$ then there exist a variable $z \in \text{Var}$ and an integer $h \in \mathbb{Z}$ s.t. $|h| \leq (C)_b$ s.t. the following two properties hold:

- (i) $\min(\llbracket C \rrbracket \eta y) = \min(\eta z) + h;$
- (ii) *for all $\eta' \in \mathbb{I}$, if $\eta' \supseteq \eta$ then $\min(\llbracket C \rrbracket \eta' y) \leq \min(\eta' z) + h.$*

Proof. The proof is by structural induction on the command $C \in \text{Imp}$. We preliminarily observe that we can safely assume $\eta \neq \perp$. In fact, if $\eta = \perp$ then $\llbracket C \rrbracket \perp = \perp$ and thus $\min(\llbracket C \rrbracket \eta y) = +\infty \geq (C)_b$, against the hypothesis $\min(\llbracket C \rrbracket \eta y) < -(C)_b$. Moreover, when quantifying over η' such that $\eta' \supseteq \eta$ in (i), if $\min(\llbracket C \rrbracket \eta' y) = -\infty$ holds, then $\min(\llbracket C \rrbracket \eta' y) \leq \min(\eta' z) + h$ trivially holds, hence we will sometimes silently omit to consider this case.

Case $(x \in I)$

Take $\eta \in \mathbb{I}$ and assume $-\infty \neq \min(\llbracket x \in I \rrbracket \eta y) < -(x \in I)_b$. Clearly $\llbracket x \in I \rrbracket \eta \neq \perp$, otherwise we would get the contradiction $\min(\llbracket x \in I \rrbracket \eta y) = +\infty \geq (x \in I)_b$. We distinguish two cases:

- If $y \neq x$, then for all $\eta' \in \mathbb{I}$ such that $\eta \subseteq \eta'$ it holds

$$\perp \neq \llbracket x \in I \rrbracket \eta' = \eta' [x \mapsto \eta x \sqcap I]$$

and thus

$$\min(\llbracket x \in I \rrbracket \eta' y) = \min(\eta' y) = \min(\eta' y) + 0$$

hence the thesis follows with $z = y$ and $h = 0$.

- If $y = x$ then $\eta(x) \in \mathbb{I}$ and

$$\min(\llbracket x \in I \rrbracket \eta y) = \min(\eta x \sqcap I)$$

Note that it cannot be $\min(I) \in \mathbb{Z}$. Otherwise, by Definition 4.3, $\min(\eta x \sqcap I) \geq \min(I) = (x \in I)_b$, violating the assumption $\min(\llbracket x \in I \rrbracket \eta y) < -(x \in I)_b$. Hence, $\min(S) = -\infty$ must hold and therefore $\min(\eta x \sqcap I) = \min(\eta(x)) = \min(\eta(x)) + 0$. It is immediate to check that the same holds for all $\eta' \supseteq \eta$, i.e.,

$$\min(\eta' x \sqcap I) = \min(\eta' x) + 0$$

and thus the thesis follows with $z = y = x$ and $h = 0$.

Case $(x := k)$ Take $\eta \in \mathbb{I}$ and assume $\min(\llbracket x := k \rrbracket \eta y) < -(x := k)_b = |k|$.

Observe that it cannot be $x = y$. In fact, since $\llbracket x := k \rrbracket \eta = \eta [x \mapsto \alpha_{\mathbb{I}}(\{k\})]$, we would have $\llbracket x := k \rrbracket \eta y = [k, k]$ and thus

$$\min(\llbracket x := k \rrbracket \eta y) = k \geq |k| = (x := k)_b$$

violating the assumption. Therefore, it must be $y \neq x$. Now, for all $\eta' \supseteq \eta$, we have $\llbracket x := k \rrbracket \eta' y = \eta' y$ and thus

$$\min(\llbracket x := k \rrbracket \eta' y) = \min(\eta' y) = \min(\eta' y) + 0,$$

hence the thesis holds with $h = 0 \leq |k| = (x := k)_b$ and $z = y$.

Case $(x := w + k)$ Take $\eta \in \mathbb{I}$ and assume $\min(\llbracket x := w + k \rrbracket \eta y) < -(x := w + k)_b = -|k|$. Recall that $\llbracket x := w + k \rrbracket \eta = \eta [x \mapsto \eta w + k]$.

We distinguish two cases:

- If $y \neq x$, then for all $\eta' \supseteq \eta$, we have $\llbracket x := w + k \rrbracket \eta' y = \eta' y$ and thus

$$\min(\llbracket x := w + k \rrbracket \eta' y) = \min(\eta' y) + 0$$

hence the thesis follows with $h = 0 \leq (x := w + k)_b$ and $z = y$.

- If $x = y$ then for all $\eta' \supseteq \eta$, we have $\llbracket x := w + k \rrbracket \eta' y = \eta' w + k$ and thus

$$\min(\llbracket x := w + k \rrbracket \eta' y) = \min(\eta' w) + k$$

hence, the thesis follows with $h = k$ (recall that $k \leq |k| = (x := w + k)_b$) and $z = w$.

Case $(C_1 + C_2)$ Take $\eta \in \dot{\mathbb{I}}$ and assume $\min(\llbracket C_1 + C_2 \rrbracket \eta) < -(C_1 + C_2)_b = -(C_1)_b - (C_2)_b$. Recall that $\llbracket C_1 + C_2 \rrbracket \eta = \llbracket C_1 \rrbracket \eta \sqcup \llbracket C_2 \rrbracket \eta$. Hence, since $\min(\llbracket C_1 + C_2 \rrbracket \eta y) \neq -\infty$, we have that $\min(\llbracket C_1 \rrbracket \eta y) \neq -\infty \neq \min(\llbracket C_2 \rrbracket \eta y)$. Moreover

$$\begin{aligned} \min(\llbracket C_1 + C_2 \rrbracket \eta y) &= \min(\llbracket C_1 \rrbracket \eta y \sqcup \llbracket C_2 \rrbracket \eta y) \\ &= \min\{\min(\llbracket C_1 \rrbracket \eta y), \min(\llbracket C_2 \rrbracket \eta y)\} \end{aligned}$$

Thus $\min(\llbracket C_1 + C_2 \rrbracket \eta y) = \min(\llbracket C_i \rrbracket \eta y)$ for some $i \in \{1, 2\}$. We can assume, without loss of generality, that the maximum is realized by the first component, i.e., $\min(\llbracket C_1 + C_2 \rrbracket \eta y) = \min(\llbracket C_1 \rrbracket \eta y) < -(C_1 + C_2)_b$. Hence we can use the inductive hypothesis on C_1 and state that there exists $h \in \mathbb{Z}$ with $|h| \leq (C_1)_b$ and $z \in \text{Var}$ such that $\min(\llbracket C_1 \rrbracket \eta y) = \min(\eta z) + h$ and for all $\eta' \in \dot{\mathbb{I}}$, $\eta \sqsubseteq \eta'$,

$$\min(\llbracket C_1 \rrbracket \eta' y) \leq \min(\eta' z) + h$$

Therefore

$$\min(\llbracket C_1 + C_2 \rrbracket \eta y) = \min(\llbracket C_1 \rrbracket \eta y) = \min(\eta z) + h$$

and for all $\eta' \in \dot{\mathbb{I}}$, $\eta \sqsubseteq \eta'$,

$$\begin{aligned} \min(\llbracket C_1 + C_2 \rrbracket \eta' y) &= \min\{\min(\llbracket C_1 \rrbracket \eta' y), \min(\llbracket C_2 \rrbracket \eta' y)\} \\ &\leq \min(\llbracket C_1 \rrbracket \eta' y) \\ &\leq \min(\eta' z) + h \end{aligned}$$

with $|h| \leq (C_1)_b \leq (C_1 + C_2)_b$, as desired.

Case $(C_1; C_2)$ Take $\eta \in \dot{\mathbb{I}}$ and assume $\min(\llbracket C_1; C_2 \rrbracket \eta y) < -(C_1; C_2)_b = -(C_1)_b - (C_2)_b$.

Recall that $\llbracket C_1; C_2 \rrbracket \eta = \llbracket C_2 \rrbracket (\llbracket C_1 \rrbracket \eta)$. If we define

$$\llbracket C_1 \rrbracket \eta = \eta_1$$

since $\min(\llbracket C_2 \rrbracket \eta_1 y) \neq -\infty$ and $\min(\llbracket C_2 \rrbracket \eta_1 y) < -(C_1; C_2)_b \leq (C_2)_b$, by inductive hypothesis on C_2 , there are $|h_2| \leq (C_2)_b$ and $w \in \text{Var}$ such that $\min(\llbracket C_2 \rrbracket \eta_1 y) = \min(\eta_1 w) + h_2$ and for all $\eta'_1 \in \dot{\mathbb{I}}$ with $\eta_1 \sqsubseteq \eta'_1$

$$\min(\llbracket C_2 \rrbracket \eta'_1 y) \leq \min(\eta'_1 w) + h_2 \tag{A.1}$$

Now observe that $\min(\llbracket C_1 \rrbracket \eta w) = \min(\eta_1 w) < -(C_1)_b$. Otherwise, if it were $\min(\eta_1 w) \geq -(C_1)_b$ we would have

$$\min(\llbracket C_2 \rrbracket \eta_1 y) = \min(\eta_1 w) + h_2 \geq -(C_1)_b - (C_2)_b = -(C_1; C_2)_b,$$

violating the hypotheses. Moreover, $\min(\llbracket C_1 \rrbracket \eta w) \neq -\infty$, otherwise we would have $\min(\llbracket C_2 \rrbracket \eta_1 y) = \min(\eta_1 w) + h_2 = -\infty$, contradicting the hypotheses. Therefore we can apply the inductive hypothesis also to C_1 and deduce that there are $|h_1| \leq (C_1)_b$ and $w' \in \text{Var}$ such that $\min(\llbracket C_1 \rrbracket \eta w) = \min(\eta w') + h_1$ and for all $\eta' \in \dot{\mathbb{I}}$ with $\eta \sqsubseteq \eta'$

$$\min(\llbracket C_1 \rrbracket \eta' w) \leq \min(\eta' w') + h_1 \tag{A.2}$$

Now, for all $\eta' \in \mathbb{I}$ with $\eta \sqsubseteq \eta'$ we have that:

$$\begin{aligned} \min(\llbracket C_1; C_2 \rrbracket \eta y) &= \min(\llbracket C_2 \rrbracket (\llbracket C_1 \rrbracket \eta) y) \\ &= \min(\llbracket C_2 \rrbracket \eta_1 y) \\ &= \min(\eta_1 \mathbf{w}) + h_2 \\ &= \min(\llbracket C_1 \rrbracket \eta \mathbf{w}) + h_2 \\ &= \min(\eta \mathbf{w}') + h_1 + h_2 \end{aligned}$$

and

$$\begin{aligned} \min(\llbracket C_1; C_2 \rrbracket \eta' y) &= \\ \min(\llbracket C_2 \rrbracket (\llbracket C_1 \rrbracket \eta') y) &\leq \\ \min(\llbracket C_1 \rrbracket \eta' \mathbf{w}) + h_2 &\leq \quad \text{by (A.1), since } \eta_1 = \llbracket C_1 \rrbracket \eta \sqsubseteq \llbracket C_1 \rrbracket \eta' \\ (\min(\eta' \mathbf{w}') + h_1) + h_2 &\quad \text{by (A.2)} \end{aligned}$$

Thus, the thesis holds with $h = h_1 + h_2$, as $|h| = |h_1 + h_2| \leq |h_1| + |h_2| \leq (C_1)_b + (C_2)_b = (C_1; C_2)_b$, as needed.

Case (fix(C)) Let $\eta \in \mathbb{I}$ such that $\min(\llbracket \text{fix}(C) \rrbracket \eta y) \neq -\infty$. Recall that $\llbracket \text{fix}(C) \rrbracket \eta = \text{lfp } \lambda \mu. (\llbracket C \rrbracket \mu \sqcup \eta)$. Observe that the least fixpoint of $\lambda \mu. (\llbracket C \rrbracket \mu \sqcup \eta)$ coincides with the least fixpoint of $\lambda \mu. (\llbracket C \rrbracket \mu \sqcup \mu) = \lambda \mu. \llbracket C + \text{true} \rrbracket \mu$ above η . Hence, if

- $\eta_0 \triangleq \eta$,
- for all $i \in \mathbb{N}$, $\eta_{i+1} \triangleq \llbracket C \rrbracket \eta_i \sqcup \eta_i = \llbracket C + \text{true} \rrbracket \eta_i \sqsupseteq \eta_i$,

then we define an increasing chain $\{\eta_i\}_{i \in \mathbb{N}} \subseteq \mathbb{I}$ such that

$$\llbracket \text{fix}(C) \rrbracket \eta = \bigsqcup_{i \in \mathbb{N}} \eta_i.$$

Since $\min(\llbracket \text{fix}(C) \rrbracket \eta y) \neq -\infty$, we have that for all $i \in \mathbb{N}$, $\min(\eta_i y) \neq -\infty$. Moreover, $\bigsqcup_{i \in \mathbb{N}} \eta_i$ on y is finitely reached in the chain $\{\eta_i\}_{i \in \mathbb{N}}$, i.e., there exists $m \in \mathbb{N}$ such that for all $i \geq m + 1$

$$\llbracket \text{fix}(C) \rrbracket \eta y = \eta_i y.$$

The inductive hypothesis holds for C and true , hence for $C + \text{true}$, therefore for all $\mathbf{x} \in \text{Var}$ and $j \in \{0, 1, \dots, m\}$, if $\min(\eta_{j+1} \mathbf{x}) < -(C + \text{true})_b = -(C)_b$ then there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ such that $|h| \leq (C)_b$ and

- (a) $-\infty \neq \min(\eta_{j+1} \mathbf{x}) = \min(\eta_j \mathbf{z}) + h$,
- (b) $\forall \eta' \sqsupseteq \eta_j. \min(\llbracket C + \text{true} \rrbracket \eta' \mathbf{x}) \leq \min(\eta' \mathbf{z}) + h$.

To shortly denote that the two conditions (a) and (b) hold, we write

$$(\mathbf{z}, j) \rightarrow_h (\mathbf{x}, j + 1)$$

Now, assume that for some variable $y \in \text{Var}$

$$\min(\llbracket \text{fix}(C) \rrbracket \eta y) = \min(\eta_{m+1} y) < -(\text{fix}(C))_b = -(n + 1)(C)_b$$

where $n = |\text{vars}(C)|$. We want to show that the thesis holds, i.e., that there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ with $|h| \leq (\text{fix}(C))_b$ such that:

$$\min(\llbracket \text{fix}(C) \rrbracket \eta y) = \min(\eta \mathbf{z}) + h \tag{A.3}$$

and for all $\eta' \sqsupseteq \eta$,

$$\min(\llbracket \text{fix}(C) \rrbracket \eta' y) \leq \min(\eta' \mathbf{z}) + h \tag{A.4}$$

Let us consider (i). We first observe that we can define a path

$$\sigma \triangleq (y_0, 0) \rightarrow_{h_0} (y_1, 1) \rightarrow_{h_1} \dots \rightarrow_{h_m} (y_{m+1}, m+1) \quad (\text{A.5})$$

such that $y_{m+1} = y$ and for all $j \in \{0, \dots, m+1\}$, $y_j \in \text{Var}$ and $\min(\eta_j y_j) < -(C)_b$. In fact, if, by contradiction, this is not the case, there would exist an index $i \in \{0, \dots, m\}$ (as

$$\min(\eta_{m+1} y_{m+1}) < -(C)_b$$

already holds) such that $\min(\eta_i y_i) \geq -(C)_b$, while for all $j \in \{i+1, \dots, m+1\}$, $\min(\eta_j y_j) < -(C)_b$. Thus, in such a case, we consider the nonempty path:

$$\pi \triangleq (y_i, i) \rightarrow_{h_i} (y_{i+1}, i+1) \rightarrow_{h_{i+1}} \dots \rightarrow_{h_m} (y_{m+1}, m+1)$$

and we have that:

$$\begin{aligned} \sum_{j=i}^m h_j &= \\ \sum_{j=i}^m \min(\eta_{j+1} y_{j+1}) - \min(\eta_j y_j) &= \\ \min(\eta_{m+1} y_{m+1}) - \min(\eta_i y_i) &= \\ \min(\eta_{m+1} y) - \min(\eta_i y_i) &< \\ -(n+1)(C)_b + (C)_b &= -n(C)_b \end{aligned}$$

with $|h_j| \leq (C)_b$ for $j \in \{i, \dots, m\}$. Hence we can apply Lemma A.1 to the projection π_p of the nodes of this path π to the variable component to deduce that π_p has a subpath which is a cycle with a strictly negative weight. More precisely, there exist $i \leq k_1 < k_2 \leq m+1$ such that $y_{k_1} = y_{k_2}$ and $h = \sum_{j=k_1}^{k_2-1} h_j < 0$. If we denote $w = y_{k_1} = y_{k_2}$, then we have that

$$\begin{aligned} \min(\eta_{k_2} w) &= h_{k_2-1} + \min(\eta_{k_2-1} w) \\ &= h_{k_2-1} + h_{k_2-2} + \min(\eta_{k_2-2} w) \\ &= \sum_{j=k_1}^{k_2-1} h_j + \min(\eta_{k_1} w) \\ &= \min(\eta_{k_1} w) + h \end{aligned} \quad \text{recall } h = \sum_{j=k_1}^{k_2-1} h_j < 0$$

Thus,

$$\min(\llbracket C + \text{true} \rrbracket^{k_2-k_1} \eta_{k_1} w) = \min(\eta_{k_1} w) + h$$

Observe that for all $\eta' \sqsupseteq \eta_{k_1}$

$$\min(\llbracket C + \text{true} \rrbracket^{k_2-k_1} \eta' w) \leq \min(\eta' w) + h \quad (\text{A.6})$$

This property (A.6) can be shown by induction on $k_2 - k_1 \geq 1$.

Then, an inductive argument allows us to show that for all $r \in \mathbb{N}$:

$$\min(\llbracket C + \text{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1} w) \leq \min(\eta_{k_1} w) + rh \quad (\text{A.7})$$

In fact, for $r = 0$ the claim trivially holds. Assuming the validity for $r \geq 0$ then we have that

$$\begin{aligned} \min(\llbracket C + \text{true} \rrbracket^{(r+1)(k_2-k_1)} \eta_{k_1} w) &= \\ \min(\llbracket C + \text{true} \rrbracket^{k_2-k_1} (\llbracket C + \text{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1} w)) &\leq \quad \text{by (A.6) as } \eta_{k_1} \sqsubseteq \llbracket C + \text{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1} \\ \min(\llbracket C + \text{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1} w) + h &\leq \quad \text{by inductive hypothesis} \\ \min(\eta_{k_1} w) + rh + h &\leq \min(\eta_{k_1} w) + (r+1)h \end{aligned}$$

However, This would contradict the hypothesis $\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{y} \neq -\infty$. In fact the inequality (A.7) would imply

$$\begin{aligned} \min(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{w}) &= \min \left(\bigsqcup_{i \in \mathbb{N}} \llbracket \mathbf{C} + \text{true} \rrbracket^i \eta \mathbf{w} \right) \\ &= \min \left(\bigsqcup_{i \in \mathbb{N}} \llbracket \mathbf{C} + \text{true} \rrbracket^i \eta_{k_1} \mathbf{w} \right) \\ &= \min \left(\bigsqcup_{r \in \mathbb{N}} \llbracket \mathbf{C} + \text{true} \rrbracket^{r(k_2 - k_1)} \eta_{k_1} \mathbf{w} \right) \\ &= -\infty \end{aligned}$$

Now, from (A.5) we deduce that for all $\eta' \sqsupseteq \eta_{k_1}$, for $j \in \{k_1, \dots, m\}$, if we let $\mu_{k_1} = \eta'$ and $\mu_{j+1} = \llbracket \mathbf{C} + \text{true} \rrbracket \mu_j$, we have that $\min(\mu_{j+1} \mathbf{y}_{j+1}) \leq \min(\mu_{j+1} \mathbf{y}_j) + h_j$ and thus

$$\llbracket \mathbf{C} + \text{true} \rrbracket^{m-k_1+1} \eta' \mathbf{y} = \mu_{m+1} \mathbf{y}_{m+1} \leq \min(\mathbf{y}_{k_1}) + \sum_{i=k_1}^m h_i = \min(\eta' \mathbf{w}) + \sum_{i=k_1}^m h_i$$

Since $\eta' = \llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \sqsupseteq \eta_{k_1}$ we conclude

$$\begin{aligned} \min(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{y}) &= \min \left(\llbracket \mathbf{C} + \text{true} \rrbracket^{m-k_1+1} \llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{y} \right) \\ &\leq -\infty + \sum_{i=k_1}^m h_i = -\infty \end{aligned}$$

contradicting the assumption. Therefore, the path σ of (A.5) must exist, and consequently

$$\min(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{y}) = \min(\eta_{m+1} \mathbf{y}) = \min(\eta \mathbf{y}_0) + \sum_{i=0}^m h_i$$

and $|\sum_{i=0}^m h_i| \leq (\text{fix}(\mathbf{C}))_{\mathbf{b}} = (n+1)(\mathbf{C})_{\mathbf{b}}$, otherwise we could use the same argument above for inferring the contradiction $\min(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta \mathbf{y}) = -\infty$.

Let us now show (ii). Given $\eta' \sqsupseteq \eta$ from (A.5) we deduce that for all $j \in \{0, \dots, m\}$, if we let $\mu_0 = \eta'$ and $\mu_{j+1} = \llbracket \mathbf{C} + \text{true} \rrbracket \mu_j$, we have that

$$\min(\mu_{j+1} \mathbf{y}_{j+1}) \leq \min(\mu_{j+1} \mathbf{y}_j) + h_j.$$

Therefore, since $\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta' \sqsupseteq \mu_{m+1}$ (observe that the convergence of $\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta'$ could be at an index greater than $m+1$), we conclude that:

$$\min(\llbracket \text{fix}(\mathbf{C}) \rrbracket \eta' \mathbf{y}) \leq \min(\mu_{m+1} \mathbf{y}) = \min(\mu_{m+1} \mathbf{y}_{m+1}) \leq \min(\eta' \mathbf{y}_0) + \sum_{i=0}^m h_i$$

as desired. \square

A.2 Lemma 4.18 proof

Lemma A.3. *Let $\mathbf{C} \in \text{Imp}$. For all $\eta \in \mathbb{C}$ and $\mathbf{y} \in \text{Var}$, if $\max(\langle \mathbf{C} \rangle \eta \mathbf{y}) \neq +\infty$ and $\max(\langle \mathbf{C} \rangle \eta \mathbf{y}) > (\mathbf{C})^{\mathbf{b}}$ then there exist a variable $\mathbf{z} \in \text{Var}$ and an integer $h \in \mathbb{Z}$ such that $|h| \leq (\mathbf{C})^{\mathbf{b}}$ and the following two properties hold:*

$$(i) \max(\langle \mathbf{C} \rangle \eta \mathbf{y}) = \max(\eta \mathbf{z}) + h;$$

$$(ii) \text{ for all } \eta' \in \mathbb{C}, \text{ if } \eta' \dot{\sqsupseteq} \eta \text{ then } \max(\langle \mathbf{C} \rangle \eta' \mathbf{y}) \geq \max(\eta' \mathbf{z}) + h.$$

Proof. The proof is by structural induction on the command $\mathbf{C} \in \text{Imp}$. We preliminarily observe that we can safely assume $\eta \neq \perp$. In fact, if $\eta = \perp$ then $\langle \mathbf{C} \rangle \perp = \perp$ and thus $\max(\langle \mathbf{C} \rangle \eta \mathbf{y}) = -\infty \leq (\mathbf{C})^{\mathbf{b}}$, against the hypothesis $\max(\langle \mathbf{C} \rangle \eta \mathbf{y}) > (\mathbf{C})^{\mathbf{b}}$. Moreover, when quantifying over η' such that $\eta' \dot{\sqsupseteq} \eta$ in (ii), if $\max(\langle \mathbf{C} \rangle \eta' \mathbf{y}) = +\infty$ holds, then $\max(\langle \mathbf{C} \rangle \eta' \mathbf{y}) \geq \max(\eta' \mathbf{z}) + h$ trivially holds, hence we will sometimes silently omit to consider this case.

Case $(x \in I)$. Take $\eta \in \mathbb{C}$ and assume $+\infty \neq \max(\langle\langle x \in I \rangle\rangle \eta y) > (x \in I)^b$. Recall that $I \in \mathbb{I}$. Clearly $\langle\langle x \in I \rangle\rangle \eta \neq \perp$, otherwise we would get the contradiction $\max(\langle\langle x \in I \rangle\rangle \eta y) = -\infty \leq (x \in I)^b$. We distinguish two cases:

- If $y \neq x$, then for all $\eta' \in \mathbb{C}$ such that $\eta \dot{\subseteq} \eta'$ it holds

$$\perp \neq \langle\langle x \in I \rangle\rangle \eta' = \eta'[x \mapsto \eta'x \dot{\cap} \gamma_{\mathbb{I}}(I)]$$

and thus

$$\max(\langle\langle x \in I \rangle\rangle \eta' y) = \max(\eta' y) = \max(\eta' y) + 0$$

hence the thesis follows with $z = y$ and $h = 0$.

- If $y = x$ then

$$\max(\langle\langle x \in I \rangle\rangle \eta y) = \max(\eta x \dot{\cap} \gamma_{\mathbb{I}}(I))$$

Note that it cannot be $\max(I) \in \mathbb{Z}$. Otherwise, by Definition 4.3, $\max(\eta x \dot{\cap} \gamma_{\mathbb{I}}(I)) \leq \max(I) = (x \in I)^b$, violating the assumption $\max(\langle\langle x \in I \rangle\rangle \eta y) > (x \in I)^b$. Hence, $\max(I) = +\infty$ must hold and therefore $\max(\eta x \dot{\cap} \gamma_{\mathbb{I}}(I)) = \max(\eta(x)) = \max(\eta(x)) + 0$. It is immediate to check that the same holds for all $\eta' \dot{\supseteq} \eta$, i.e.,

$$\max(\eta' x \dot{\cap} \gamma_{\mathbb{I}}(I)) = \max(\eta' x) + 0$$

and thus the thesis follows with $z = y = x$ and $h = 0$.

Case $(x := k)$. Take $\eta \in \mathbb{C}$ and assume $\max(\langle\langle x := k \rangle\rangle \eta y) > (x := k)^b = |k|$.

Observe that it cannot be $x = y$. In fact, since $\langle\langle x := k \rangle\rangle \eta = \eta[x \mapsto \{k\}]$, we would have $\langle\langle x := k \rangle\rangle \eta y = \{k\}$ and thus

$$\max(\langle\langle x := k \rangle\rangle \eta y) = k \leq (x := k)^b$$

violating the assumption. Therefore, it must be $y \neq x$. Now, for all $\eta' \dot{\supseteq} \eta$, we have $\langle\langle x := k \rangle\rangle \eta' y = \eta' y$ and thus

$$\max(\langle\langle x := k \rangle\rangle \eta' y) = \max(\eta' y) = \max(\eta' y) + 0,$$

hence the thesis holds with $h = 0 \leq (x := k)^b$ and $z = y$.

Case $(x := w + k)$. Take $\eta \in \mathbb{C}$ and assume $\max(\langle\langle x := w + k \rangle\rangle \eta y) > (x := w + k)^b = |k|$. Recall that $\langle\langle x := w + k \rangle\rangle \eta = \eta[x \mapsto \eta w + k]$.

We distinguish two cases:

- If $y \neq x$, then for all $\eta' \dot{\supseteq} \eta$, we have $\langle\langle x := w + k \rangle\rangle \eta' y = \eta' y$ and thus

$$\max(\langle\langle x := w + k \rangle\rangle \eta' y) = \max(\eta' y)$$

hence the thesis follows with $h = 0 \leq (x := w + k)^b$ and $z = y$.

- If $x = y$ then for all $\eta' \dot{\supseteq} \eta$, we have $\langle\langle x := w + k \rangle\rangle \eta' y = \eta' w + k$ and thus

$$\max(\langle\langle x := w + k \rangle\rangle \eta' y) = \max(\eta' w) + k$$

hence, the thesis follows with $h = k$ (recall that $k \leq |k| = (x := w + k)^b$) and $z = w$.

Case $(C_1 + C_2)$. Take $\eta \in \mathbb{C}$ and assume $\max(\langle\langle C_1 + C_2 \rangle\rangle \eta) > (C_1 + C_2)^b = (C_1)^b + (C_2)^b$. Recall that $\langle\langle C_1 + C_2 \rangle\rangle \eta = \langle\langle C_1 \rangle\rangle \eta \dot{\cup} \langle\langle C_2 \rangle\rangle \eta$. Hence, since $\max(\langle\langle C_1 + C_2 \rangle\rangle \eta y) \neq +\infty$, we have that $\max(\langle\langle C_1 \rangle\rangle \eta y) \neq +\infty \neq \max(\langle\langle C_2 \rangle\rangle \eta y)$. Moreover

$$\begin{aligned} \max(\langle\langle C_1 + C_2 \rangle\rangle \eta y) &= \max(\langle\langle C_1 \rangle\rangle \eta y \dot{\cup} \langle\langle C_2 \rangle\rangle \eta y) \\ &= \max\{\max(\langle\langle C_1 \rangle\rangle \eta y), \max(\langle\langle C_2 \rangle\rangle \eta y)\} \end{aligned}$$

Thus $\max(\langle\langle C_1 + C_2 \rangle\rangle \eta y) = \max(\langle\langle C_i \rangle\rangle \eta y)$ for some $i \in \{1, 2\}$. We can assume, without loss of generality, that the maximum is realized by the first component, i.e., $\max(\langle\langle C_1 + C_2 \rangle\rangle \eta y) = \max(\langle\langle C_1 \rangle\rangle \eta y) > (C_1 + C_2)^b$. Hence we can use the inductive hypothesis on C_1 and state that there exists $h \in \mathbb{Z}$ with $|h| \leq (C_1)^b$ and $z \in \text{Var}$ such that $\max(\langle\langle C_1 \rangle\rangle \eta y) = \max(\eta z) + h$ and for all $\eta' \in \mathbb{C}$, $\eta \dot{\subseteq} \eta'$,

$$\max(\langle\langle C_1 \rangle\rangle \eta' y) \geq \max(\eta' z) + h$$

Therefore

$$\max(\langle\langle C_1 + C_2 \rangle\rangle \eta y) = \max(\langle\langle C_1 \rangle\rangle \eta y) = \max(\eta z) + h$$

and for all $\eta' \in \mathbb{C}$, $\eta \dot{\subseteq} \eta'$,

$$\begin{aligned} \max(\langle\langle C_1 + C_2 \rangle\rangle \eta' y) &= \max\{\max(\langle\langle C_1 \rangle\rangle \eta' y), \max(\langle\langle C_2 \rangle\rangle \eta' y)\} \\ &\geq \max(\langle\langle C_1 \rangle\rangle \eta' y) \\ &\geq \max(\eta' z) + h \end{aligned}$$

with $|h| \leq (C_1)^b \leq (C_1 + C_2)^b$, as desired.

Case $(C_1; C_2)$. Take $\eta \in \mathbb{C}$ and assume $\max(\langle\langle C_1; C_2 \rangle\rangle \eta y) > (C_1; C_2)^b = (C_1)^b + (C_2)^b$. Recall that $\langle\langle C_1; C_2 \rangle\rangle \eta = \langle\langle C_2 \rangle\rangle (\langle\langle C_1 \rangle\rangle \eta)$. If we define

$$\langle\langle C_1 \rangle\rangle \eta = \eta_1$$

since $\max(\langle\langle C_2 \rangle\rangle \eta_1 y) \neq +\infty$ and $\max(\langle\langle C_2 \rangle\rangle \eta_1 y) > (C_1; C_2)^b \geq (C_2)^b$, by inductive hypothesis on C_2 , there are $|h_2| \leq (C_2)^b$ and $w \in \text{Var}$ such that $\max(\langle\langle C_2 \rangle\rangle \eta_1 y) = \max(\eta_1 w) + h_2$ and for all $\eta'_1 \in \mathbb{C}$ with $\eta_1 \dot{\subseteq} \eta'_1$

$$\max(\langle\langle C_2 \rangle\rangle \eta'_1 y) \geq \max(\eta'_1 w) + h_2 \tag{A.8}$$

Now observe that $\max(\langle\langle C_1 \rangle\rangle \eta w) = \max(\eta_1 w) > (C_1)^b$. Otherwise, if it were $\max(\eta_1 w) \leq (C_1)^b$ we would have

$$\max(\langle\langle C_2 \rangle\rangle \eta_1 y) = \max(\eta_1 w) + h_2 \leq (C_1)^b + (C_2)^b = (C_1; C_2)^b,$$

violating the hypotheses. Moreover, $\max(\langle\langle C_1 \rangle\rangle \eta w) \neq +\infty$, otherwise we would have $\max(\langle\langle C_2 \rangle\rangle \eta_1 y) = \max(\eta_1 w) + h_2 = +\infty$, contradicting the hypotheses. Therefore we can apply the inductive hypothesis also to C_1 and deduce that there are $|h_1| \leq (C_1)^b$ and $w' \in \text{Var}$ such that $\max(\langle\langle C_1 \rangle\rangle \eta w) = \max(\eta w') + h_1$ and for all $\eta' \in \mathbb{C}$ with $\eta \dot{\subseteq} \eta'$

$$\max(\langle\langle C_1 \rangle\rangle \eta' w) \geq \max(\eta' w') + h_1 \tag{A.9}$$

Summing up:

$$\begin{aligned} \max(\langle\langle C_1; C_2 \rangle\rangle \eta y) &= \max(\langle\langle C_2 \rangle\rangle (\langle\langle C_1 \rangle\rangle \eta) y) \\ &= \max(\langle\langle C_2 \rangle\rangle \eta_1 y) \\ &= \max(\eta_1 w) + h_2 \\ &= \max(\langle\langle C_1 \rangle\rangle \eta w) + h_2 \\ &= \max(\eta w') + h_1 + h_2. \end{aligned}$$

Now, for all $\eta' \in \mathbb{C}$ with $\eta \dot{\subseteq} \eta'$ we have that:

$$\begin{aligned} \max(\langle\langle C_1; C_2 \rangle\rangle \eta' y) &= \\ \max(\langle\langle C_2 \rangle\rangle (\langle\langle C_1 \rangle\rangle \eta') y) &\geq \\ \max(\langle\langle C_1 \rangle\rangle \eta' w) + h_2 &\geq \quad \text{by (A.8), since } \eta_1 = \langle\langle C_1 \rangle\rangle \eta \dot{\subseteq} \langle\langle C_1 \rangle\rangle \eta' \text{ and monotonicity} \\ (\max(\eta' w') + h_1) + h_2 &\quad \text{by (A.9)} \end{aligned}$$

Thus, the thesis holds with $h = h_1 + h_2$, as $|h| = |h_1 + h_2| \leq |h_1| + |h_2| \leq (C_1)^b + (C_2)^b = (C_1; C_2)^b$, as needed.

Case ($\text{fix}(\mathbf{C})$). Let $\eta \in \mathbb{C}$ such that $\max(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta \mathbf{y}) \neq +\infty$. Recall that $\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta = \text{lfp} \left(\lambda \mu. (\langle\langle \mathbf{C} \rangle\rangle \mu \dot{\cup} \eta) \right)$.

Observe that the least fixpoint of $\lambda \mu. (\langle\langle \mathbf{C} \rangle\rangle \mu \dot{\cup} \eta)$ coincides with the least fixpoint of $\lambda \mu. (\langle\langle \mathbf{C} \rangle\rangle \mu \dot{\cup} \mu) = \lambda \mu. \langle\langle \mathbf{C} + \text{true} \rangle\rangle \mu$ above η . Hence, if

$$\begin{aligned} \eta_0 &\triangleq \eta \\ \text{for all } i \in \mathbb{N} \quad \eta_{i+1} &\triangleq \langle\langle \mathbf{C} \rangle\rangle \eta_i \dot{\cup} \eta_i = \langle\langle \mathbf{C} + \text{true} \rangle\rangle \eta_i \dot{\supseteq} \eta_i \end{aligned}$$

then we define an increasing chain $\{\eta_i\}_{i \in \mathbb{N}} \subseteq \mathbb{C}$ such that

$$\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta = \dot{\bigcup}_{i \in \mathbb{N}} \eta_i.$$

Since $\max(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta \mathbf{y}) \neq +\infty$, we have that for all $i \in \mathbb{N}$, $\max(\eta_i \mathbf{y}) \neq +\infty$. Moreover, $\dot{\bigcup}_{i \in \mathbb{N}} \eta_i$ on \mathbf{y} is finitely reached in the chain $\{\eta_i\}_{i \in \mathbb{N}}$, i.e., there exists $m \in \mathbb{N}$ such that for all $i \geq m+1$

$$\max(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta \mathbf{y}) = \max(\eta_i \mathbf{y}).$$

The inductive hypothesis holds for \mathbf{C} and true , hence for $\mathbf{C} + \text{true}$, therefore for all $\mathbf{x} \in \text{Var}$ and $j \in \{0, 1, \dots, m\}$, if $\max(\eta_{j+1} \mathbf{x}) > (\mathbf{C} + \text{true})^b = (\mathbf{C})^b$ then there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ such that $|h| \leq (\mathbf{C})^b$ and

- (a) $+\infty \neq \max(\eta_{j+1} \mathbf{x}) = \max(\eta_j \mathbf{z}) + h$,
- (b) $\forall \eta' \dot{\supseteq} \eta_j. \max(\langle\langle \mathbf{C} + \text{true} \rangle\rangle \eta' \mathbf{x}) \geq \max(\eta' \mathbf{z}) + h$.

To shortly denote that the two conditions (a) and (b) hold, we write

$$(\mathbf{z}, j) \rightarrow_h (\mathbf{x}, j+1)$$

Now, assume that for some variable $\mathbf{y} \in \text{Var}$

$$\max(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta \mathbf{y}) = \max(\eta_{m+1} \mathbf{y}) > (\text{fix}(\mathbf{C}))^b = (n+1)(\mathbf{C})^b$$

where $n = |\text{vars}(\mathbf{C})|$. We want to show that the thesis holds, i.e., that there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ with $|h| \leq (\text{fix}(\mathbf{C}))^b$ such that:

$$\max(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta \mathbf{y}) = \max(\eta \mathbf{z}) + h \tag{A.10}$$

and for all $\eta' \dot{\supseteq} \eta$,

$$\max(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta' \mathbf{y}) \geq \max(\eta' \mathbf{z}) + h \tag{A.11}$$

Let us consider (A.10). We first observe that we can define a path

$$\sigma \triangleq (\mathbf{y}_0, 0) \rightarrow_{h_0} (\mathbf{y}_1, 1) \rightarrow_{h_1} \dots \rightarrow_{h_m} (\mathbf{y}_{m+1}, m+1) \tag{A.12}$$

such that $\mathbf{y}_{m+1} = \mathbf{y}$ and for all $j \in \{0, \dots, m+1\}$, $\mathbf{y}_j \in \text{Var}$ and $\max(\eta_j \mathbf{y}_j) > (\mathbf{C})^b$. In fact, if, by contradiction, this were not the case, there would exist an index $i \in \{0, \dots, m\}$ (as $\max(\eta_{m+1} \mathbf{y}_{m+1}) > (\mathbf{C})^b$ already holds) such that $\max(\eta_i \mathbf{y}_i) \leq (\mathbf{C})^b$, while for all $j \in \{i+1, \dots, m+1\}$, $\max(\eta_j \mathbf{y}_j) > (\mathbf{C})^b$. Thus, in such a case, we consider the nonempty path:

$$\pi \triangleq (\mathbf{y}_i, i) \rightarrow_{h_i} (\mathbf{y}_{i+1}, i+1) \rightarrow_{h_{i+1}} \dots \rightarrow_{h_m} (\mathbf{y}_{m+1}, m+1) \tag{A.13}$$

and we have that:

$$\begin{aligned} \Sigma_{j=i}^m h_j &= \\ \Sigma_{j=i}^m \max(\eta_{j+1} \mathbf{y}_{j+1}) - \max(\eta_j \mathbf{y}_j) &= \\ \max(\eta_{m+1} \mathbf{y}_{m+1}) - \max(\eta_i \mathbf{y}_i) &= \\ \max(\eta_{m+1} \mathbf{y}) - \max(\eta_i \mathbf{y}_i) &> \\ (n+1)(\mathbf{C})^b - (\mathbf{C})^b &= n(\mathbf{C})^b \end{aligned}$$

with $|h_j| \leq (C)^b$ for $j \in \{i, \dots, m\}$. Hence we can apply Lemma 4.4 to the projection π_p of the nodes of this path π to the variable component to deduce that π_p has a subpath which is a cycle with a strictly positive weight. More precisely, there exist $i \leq k_1 < k_2 \leq m+1$ such that $y_{k_1} = y_{k_2}$ and $h = \sum_{j=k_1}^{k_2-1} h_j > 0$. If we denote $\mathbf{w} = y_{k_1} = y_{k_2}$, then we have that

$$\begin{aligned} \max(\eta_{k_2} \mathbf{w}) &= h_{k_2-1} + \max(\eta_{k_2-1} \mathbf{w}) \\ &= h_{k_2-1} + h_{k_2-2} + \max(\eta_{k_2-2} \mathbf{w}) \\ &= \sum_{j=k_1}^{k_2-1} h_j + \max(\eta_{k_1} \mathbf{w}) \\ &= h + \max(\eta_{k_1} \mathbf{w}) \end{aligned}$$

Thus,

$$\max(\langle\langle C + \text{true} \rangle\rangle^{k_2-k_1} \eta_{k_1} \mathbf{w}) = \max(\eta_{k_1} \mathbf{w}) + h$$

Observe that for all $\eta' \dot{\supseteq} \eta_{k_1}$

$$\max(\langle\langle C + \text{true} \rangle\rangle^{k_2-k_1} \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}) + h \quad (\text{A.14})$$

Let us show that Equation (A.14) holds. We do so by induction on $\ell = k_2 - k_1 \geq 1$.

Case ($\ell = 1$). Notice that by (b) used to build π in (A.13) it holds that $\forall \eta' \dot{\supseteq} \eta_{k_1} \dot{\supseteq} \eta$

$$\max(\langle\langle C + \text{true} \rangle\rangle \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}) + h$$

hence the thesis holds.

Case ($\ell \Rightarrow \ell + 1$). Recall that

$$(\langle\langle C + \text{true} \rangle\rangle)^{\ell+1} \eta' = (\langle\langle C + \text{true} \rangle\rangle) \left((\langle\langle C + \text{true} \rangle\rangle)^\ell \eta' \right)$$

and by inductive hypothesis $\max(\langle\langle C + \text{true} \rangle\rangle^\ell \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}) + h$. Recall that for all $\eta'' \in C$ we know that $\langle\langle C + \text{true} \rangle\rangle \eta'' = \eta'' \dot{\cup} \langle\langle C \rangle\rangle \eta''$. Hence we can notice that $\max(\langle\langle C + \text{true} \rangle\rangle \eta'' \mathbf{x}) \geq \max(\eta'' \mathbf{x})$ for all $\mathbf{x} \in \text{Var}$. Therefore

$$\max(\langle\langle C + \text{true} \rangle\rangle (\langle\langle C + \text{true} \rangle\rangle^\ell \eta' \mathbf{w})) \geq \max(\langle\langle C + \text{true} \rangle\rangle^\ell \eta' \mathbf{w}) \geq \max(\eta' \mathbf{w}) + h$$

which is our thesis for Property (A.14).

Then, an inductive argument allows us to show that for all $r \in \mathbb{N}$:

$$\max(\langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w}) \geq \max(\eta_{k_1} \mathbf{w}) + rh \quad (\text{A.15})$$

In fact, for $r = 0$ the claim trivially holds. Assuming the validity for $r \geq 0$ then we have that

$$\begin{aligned} \max(\langle\langle C + \text{true} \rangle\rangle^{(r+1)(k_2-k_1)} \eta_{k_1} \mathbf{w}) &= \\ \max(\langle\langle C + \text{true} \rangle\rangle^{k_2-k_1} (\langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w})) &\geq \quad \text{by (A.14) as } \eta_{k_1} \dot{\subseteq} \langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \\ \max(\langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w}) + h &\geq \quad \text{by inductive hypothesis} \\ \max(\eta_{k_1} \mathbf{w}) + rh + h &\geq \max(\eta_{k_1} \mathbf{w}) + (r+1)h \end{aligned}$$

However, this would contradict the hypothesis $\langle\langle \text{fix}(C) \rangle\rangle \eta y \neq \infty$. In fact the Inequality (A.15) would imply

$$\begin{aligned} \langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{w} &= \bigcup_{i \in \mathbb{N}} \langle\langle C + \text{true} \rangle\rangle^i \eta \mathbf{w} = \\ &= \bigcup_{i \in \mathbb{N}} \langle\langle C + \text{true} \rangle\rangle^i \eta_{k_1} \mathbf{w} \\ &= \bigcup_{r \in \mathbb{N}} \langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w} \\ &= +\infty \end{aligned}$$

Now, from (A.12) we deduce that for all $\eta' \dot{\supseteq} \eta_{k_1}$, for $j \in \{k_1, \dots, m\}$, if we let $\mu_{k_1} = \eta'$ and $\mu_{j+1} = \langle\langle C + \text{true} \rangle\rangle \mu_j$, by the choice of the subsequence, since $k_1 \geq i$, we have that

$$\max(\mu_{j+1} \mathbf{y}_{j+1}) \geq \max(\mu_{j+1} \mathbf{y}_j) + h_j$$

and thus

$$\langle\langle C + \text{true} \rangle\rangle^{m-k_1+1} \eta' \mathbf{y} = \mu_{m+1} \mathbf{y}_{m+1} \geq \max(\mathbf{y}_{k_1}) + \sum_{i=k_1}^m h_i = \max(\eta' \mathbf{w}) + \sum_{i=k_1}^m h_i$$

Since $\eta' = \langle\langle \text{fix}(C) \rangle\rangle \eta \dot{\supseteq} \eta_{k_1}$ we conclude

$$\begin{aligned} \max(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) &= \max\left(\langle\langle C + \text{true} \rangle\rangle^{m-k_1+1} \langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{w}\right) \\ &= \max(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{w}) + \sum_{i=k_1}^m h_i \\ &\geq +\infty + \sum_{i=k_1}^m h_i = +\infty \end{aligned}$$

contradicting the assumption.

Therefore, the path σ of (A.12) must exist, and consequently

$$\max(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) = \max(\eta_{m+1} \mathbf{y}) = \max(\eta \mathbf{y}_0) + \sum_{i=0}^m h_i$$

and $\sum_{i=0}^m h_i \leq (\text{fix}(C))^b = (n+1)(C)^b$, otherwise we could use the same argument above for inferring the contradiction $\max(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) = +\infty$.

Let us now show (A.11). Given $\eta' \dot{\supseteq} \eta$ from (A.12) we deduce that for all $j \in \{0, \dots, m\}$, if we let $\mu_0 = \eta'$ and $\mu_{j+1} = \langle\langle C + \text{true} \rangle\rangle \mu_j$, we have that

$$\max(\mu_{j+1} \mathbf{y}_{j+1}) \geq \max(\mu_{j+1} \mathbf{y}_j) + h_j.$$

Therefore, since $\langle\langle \text{fix}(C) \rangle\rangle \eta' \dot{\supseteq} \mu_{m+1}$ (observe that the convergence of $\langle\langle \text{fix}(C) \rangle\rangle \eta'$ could be at an index greater than $m+1$), we conclude that:

$$\max(\langle\langle \text{fix}(C) \rangle\rangle \eta' \mathbf{y}) \geq \max(\mu_{m+1} \mathbf{y}) = \max(\mu_{m+1} \mathbf{y}_{m+1}) \geq \max(\eta' \mathbf{y}_0) + \sum_{i=0}^m h_i$$

as desired. □

A.3 Lemma 4.20 proof

Lemma A.4. *Let $C \in \text{Imp}$.*

For all $\eta \in \mathbb{C}$ and $\mathbf{y} \in \text{Var}$, if $\min(\langle\langle C \rangle\rangle \eta \mathbf{y}) \neq -\infty$ and $\min(\langle\langle C \rangle\rangle \eta \mathbf{y}) < -(C)_b$ then there exist a variable $\mathbf{z} \in \text{Var}$ and an integer $h \in \mathbb{Z}$ s.t. $|h| \leq (C)_b$ s.t. the following two properties hold:

(i) $\min(\langle\langle C \rangle\rangle \eta \mathbf{y}) = \min(\eta \mathbf{z}) + h;$

(ii) *for all $\eta' \in \mathbb{C}$, if $\eta' \dot{\supseteq} \eta$ then $\min(\langle\langle C \rangle\rangle \eta' \mathbf{y}) \leq \min(\eta' \mathbf{z}) + h$.*

Proof. The proof is by structural induction on the command $C \in \text{Imp}$. We preliminarily observe that we can safely assume $\eta \neq \perp$. In fact, if $\eta = \perp$ then $\langle\langle C \rangle\rangle \perp = \perp$ and thus $\min(\langle\langle C \rangle\rangle \eta \mathbf{y}) = +\infty \geq (C)_b$, against the hypothesis $\min(\langle\langle C \rangle\rangle \eta \mathbf{y}) < -(C)_b$. Moreover, when quantifying over η' such that $\eta' \dot{\supseteq} \eta$ in (i), if $\min(\langle\langle C \rangle\rangle \eta' \mathbf{y}) = -\infty$ holds, then $\min(\langle\langle C \rangle\rangle \eta' \mathbf{y}) \leq \min(\eta' \mathbf{z}) + h$ trivially holds, hence we will sometimes silently omit to consider this case.

Case ($\mathbf{x} \in I$)

Take $\eta \in \mathbb{C}$ and assume $-\infty \neq \min(\langle\langle \mathbf{x} \in I \rangle\rangle \eta \mathbf{y}) < -(\mathbf{x} \in I)_b$. Clearly $\langle\langle \mathbf{x} \in I \rangle\rangle \eta \neq \perp$, otherwise we would get the contradiction $\min(\langle\langle \mathbf{x} \in I \rangle\rangle \eta \mathbf{y}) = +\infty \geq (\mathbf{x} \in I)_b$. We distinguish two cases:

- If $y \neq x$, then for all $\eta' \in \mathbb{C}$ such that $\eta \dot{\subseteq} \eta'$ it holds

$$\perp \neq \langle\langle x \in I \rangle\rangle \eta' = \eta' [x \mapsto \eta x \dot{\cap} \gamma_{\mathbb{I}}(I)]$$

and thus

$$\min(\langle\langle x \in I \rangle\rangle \eta' y) = \min(\eta' y) = \min(\eta' y) + 0$$

hence the thesis follows with $z = y$ and $h = 0$.

- If $y = x$ then $\eta(x) \in \mathbb{I}$ and

$$\min(\langle\langle x \in I \rangle\rangle \eta y) = \min(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta x) \cap \gamma_{\mathbb{I}}(S)))$$

Note that it cannot be $\min(S) \in \mathbb{Z}$. Otherwise, by Definition 4.3, $\min(\eta x \dot{\cap} \gamma_{\mathbb{I}}(I)) \geq \min(I) = (x \in I)_{\mathbf{b}}$, violating the assumption $\min(\langle\langle x \in I \rangle\rangle \eta y) < -(x \in I)_{\mathbf{b}}$. Hence, $\min(S) = -\infty$ must hold and therefore $\min(\eta x \dot{\cap} \gamma_{\mathbb{I}}(I)) = \min(\eta(x)) = \min(\eta(x)) + 0$. It is immediate to check that the same holds for all $\eta' \dot{\supseteq} \eta$, i.e.,

$$\min(\eta' x \dot{\cap} \gamma_{\mathbb{I}}(I)) = \min(\eta' x) + 0$$

and thus the thesis follows with $z = y = x$ and $h = 0$.

Case $(x := k)$ Take $\eta \in \mathbb{C}$ and assume $\min(\langle\langle x := k \rangle\rangle \eta y) < -(x := k)_{\mathbf{b}} = |k|$.

Observe that it cannot be $x = y$. In fact, since $\langle\langle x := k \rangle\rangle \eta = \eta[x \mapsto \{k\}]$

$$\min(\langle\langle x := k \rangle\rangle \eta y) = k \geq |k| = (x := k)_{\mathbf{b}}$$

violating the assumption. Therefore, it must be $y \neq x$. Now, for all $\eta' \dot{\supseteq} \eta$, we have $\langle\langle x := k \rangle\rangle \eta' y = \eta' y$ and thus

$$\min(\langle\langle x := k \rangle\rangle \eta' y) = \min(\eta' y) = \min(\eta' y) + 0,$$

hence the thesis holds with $h = 0 \leq |k| = (x := k)_{\mathbf{b}}$ and $z = y$.

Case $(x := w + k)$ Take $\eta \in \mathbb{C}$ and assume $\min(\langle\langle x := w + k \rangle\rangle \eta y) < -(x := w + k)_{\mathbf{b}} = -|k|$. Recall that $\langle\langle x := w + k \rangle\rangle \eta = \eta[x \mapsto \eta w + k]$.

We distinguish two cases:

- If $y \neq x$, then for all $\eta' \dot{\supseteq} \eta$, we have $\langle\langle x := w + k \rangle\rangle \eta' y = \eta' y$ and thus

$$\min(\langle\langle x := w + k \rangle\rangle \eta' y) = \min(\eta' y) + 0$$

hence the thesis follows with $h = 0 \leq (x := w + k)_{\mathbf{b}}$ and $z = y$.

- If $x = y$ then for all $\eta' \dot{\supseteq} \eta$, we have $\langle\langle x := w + k \rangle\rangle \eta' y = \eta' w + k$ and thus

$$\min(\langle\langle x := w + k \rangle\rangle \eta' y) = \min(\eta' w) + k$$

hence, the thesis follows with $h = k$ (recall that $k \leq |k| = (x := w + k)_{\mathbf{b}}$) and $z = w$.

Case $(C_1 + C_2)$ Take $\eta \in \mathbb{C}$ and assume $\min(\langle\langle C_1 + C_2 \rangle\rangle \eta) < -(C_1 + C_2)_{\mathbf{b}} = -(C_1)_{\mathbf{b}} - (C_2)_{\mathbf{b}}$. Recall that $\langle\langle C_1 + C_2 \rangle\rangle \eta = \langle\langle C_1 \rangle\rangle \eta \dot{\cup} \langle\langle C_2 \rangle\rangle \eta$. Hence, since $\min(\langle\langle C_1 + C_2 \rangle\rangle \eta y) \neq -\infty$, we have that $\min(\langle\langle C_1 \rangle\rangle \eta y) \neq -\infty \neq \min(\langle\langle C_2 \rangle\rangle \eta y)$. Moreover

$$\begin{aligned} \min(\langle\langle C_1 + C_2 \rangle\rangle \eta y) &= \min(\langle\langle C_1 \rangle\rangle \eta y \dot{\cup} \langle\langle C_2 \rangle\rangle \eta y) \\ &= \min\{\min(\langle\langle C_1 \rangle\rangle \eta y), \min(\langle\langle C_2 \rangle\rangle \eta y)\} \end{aligned}$$

Thus $\min(\langle\langle C_1 + C_2 \rangle\rangle \eta y) = \min(\langle\langle C_i \rangle\rangle \eta y)$ for some $i \in \{1, 2\}$. We can assume, without loss of generality, that the maximum is realized by the first component, i.e., $\min(\langle\langle C_1 + C_2 \rangle\rangle \eta y) = \min(\langle\langle C_1 \rangle\rangle \eta y) < -(C_1 + C_2)_{\mathbf{b}}$. Hence we can use the inductive hypothesis on C_1 and state that

there exists $h \in \mathbb{Z}$ with $|h| \leq (C_1)_b$ and $z \in Var$ such that $\min(\langle\langle C_1 \rangle\rangle \eta y) = \min(\eta z) + h$ and for all $\eta' \in \mathbb{C}$, $\eta \dot{\subseteq} \eta'$,

$$\min(\langle\langle C_1 \rangle\rangle \eta' y) \leq \min(\eta' z) + h$$

Therefore

$$\min(\langle\langle C_1 + C_2 \rangle\rangle \eta y) = \min(\langle\langle C_1 \rangle\rangle \eta y) = \min(\eta z) + h$$

and for all $\eta' \in \mathbb{C}$, $\eta \dot{\subseteq} \eta'$,

$$\begin{aligned} \min(\langle\langle C_1 + C_2 \rangle\rangle \eta' y) &= \min\{\min(\langle\langle C_1 \rangle\rangle \eta' y), \min(\langle\langle C_2 \rangle\rangle \eta' y)\} \\ &\leq \min(\langle\langle C_1 \rangle\rangle \eta' y) \\ &\leq \min(\eta' z) + h \end{aligned}$$

with $|h| \leq (C_1)_b \leq (C_1 + C_2)_b$, as desired.

Case $(C_1; C_2)$ Take $\eta \in \mathbb{C}$ and assume $\min(\langle\langle C_1; C_2 \rangle\rangle \eta y) < -(C_1; C_2)_b = -(C_1)_b - (C_2)_b$.

Recall that $\langle\langle C_1; C_2 \rangle\rangle \eta = \langle\langle C_2 \rangle\rangle (\langle\langle C_1 \rangle\rangle \eta)$. If we define

$$\langle\langle C_1 \rangle\rangle \eta = \eta_1$$

since $\min(\langle\langle C_2 \rangle\rangle \eta_1 y) \neq -\infty$ and $\min(\langle\langle C_2 \rangle\rangle \eta_1 y) < -(C_1; C_2)_b \leq (C_2)_b$, by inductive hypothesis on C_2 , there are $|h_2| \leq (C_2)_b$ and $w \in Var$ such that $\min(\langle\langle C_2 \rangle\rangle \eta_1 y) = \min(\eta_1 w) + h_2$ and for all $\eta'_1 \in \mathbb{C}$ with $\eta_1 \dot{\subseteq} \eta'_1$

$$\min(\langle\langle C_2 \rangle\rangle \eta'_1 y) \leq \min(\eta'_1 w) + h_2 \quad (\text{A.16})$$

Now observe that $\min(\langle\langle C_1 \rangle\rangle \eta w) = \min(\eta_1 w) < -(C_1)_b$. Otherwise, if it were $\min(\eta_1 w) \geq -(C_1)_b$ we would have

$$\min(\langle\langle C_2 \rangle\rangle \eta_1 y) = \min(\eta_1 w) + h_2 \geq -(C_1)_b - (C_2)_b = -(C_1; C_2)_b,$$

violating the hypotheses. Moreover, $\min(\langle\langle C_1 \rangle\rangle \eta w) \neq -\infty$, otherwise we would have $\min(\langle\langle C_2 \rangle\rangle \eta_1 y) = \min(\eta_1 w) + h_2 = -\infty$, contradicting the hypotheses. Therefore we can apply the inductive hypothesis also to C_1 and deduce that there are $|h_1| \leq (C_1)_b$ and $w' \in Var$ such that $\min(\langle\langle C_1 \rangle\rangle \eta w) = \min(\eta w') + h_1$ and for all $\eta' \in \mathbb{C}$ with $\eta \dot{\subseteq} \eta'$

$$\min(\langle\langle C_1 \rangle\rangle \eta' w) \leq \min(\eta' w') + h_1 \quad (\text{A.17})$$

Now, for all $\eta' \in \mathbb{C}$ with $\eta \dot{\subseteq} \eta'$ we have that:

$$\begin{aligned} \min(\langle\langle C_1; C_2 \rangle\rangle \eta y) &= \min(\langle\langle C_2 \rangle\rangle (\langle\langle C_1 \rangle\rangle \eta) y) \\ &= \min(\langle\langle C_2 \rangle\rangle \eta_1 y) \\ &= \min(\eta_1 w) + h_2 \\ &= \min(\langle\langle C_1 \rangle\rangle \eta w) + h_2 \\ &= \min(\eta w') + h_1 + h_2 \end{aligned}$$

and

$$\begin{aligned} \min(\langle\langle C_1; C_2 \rangle\rangle \eta' y) &= \\ \min(\langle\langle C_2 \rangle\rangle (\langle\langle C_1 \rangle\rangle \eta') y) &\leq \\ \min(\langle\langle C_1 \rangle\rangle \eta' w) + h_2 &\leq \quad \text{by (A.16), since } \eta_1 = \langle\langle C_1 \rangle\rangle \eta \dot{\subseteq} \langle\langle C_1 \rangle\rangle \eta' \\ (\min(\eta' w') + h_1) + h_2 &\quad \text{by (A.17)} \end{aligned}$$

Thus, the thesis holds with $h = h_1 + h_2$, as $|h| = |h_1 + h_2| \leq |h_1| + |h_2| \leq (C_1)_b + (C_2)_b = (C_1; C_2)_b$, as needed.

Case $(\text{fix}(C))$ Let $\eta \in \mathbb{C}$ such that $\min(\langle\langle \text{fix}(C) \rangle\rangle \eta y) \neq -\infty$. Recall that $\langle\langle \text{fix}(C) \rangle\rangle \eta = \text{lfp} \lambda \mu. (\langle\langle C \rangle\rangle \mu \dot{\cup} \eta)$. Observe that the least fixpoint of $\lambda \mu. (\langle\langle C \rangle\rangle \mu \dot{\cup} \eta)$ coincides with the least fixpoint of $\lambda \mu. (\langle\langle C \rangle\rangle \mu \dot{\cup} \mu) = \lambda \mu. \langle\langle C + \text{true} \rangle\rangle \mu$ above η . Hence, if

- $\eta_0 \triangleq \eta$,
- for all $i \in \mathbb{N}$, $\eta_{i+1} \triangleq \langle\langle C \rangle\rangle \eta_i \dot{\cup} \eta_i = \langle\langle C + \text{true} \rangle\rangle \eta_i \dot{\supseteq} \eta_i$,

then we define an increasing chain $\{\eta_i\}_{i \in \mathbb{N}} \subseteq \mathbb{C}$ such that

$$\langle\langle \text{fix}(C) \rangle\rangle \eta = \bigsqcup_{i \in \mathbb{N}} \eta_i.$$

Since $\min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) \neq -\infty$, we have that for all $i \in \mathbb{N}$, $\min(\eta_i \mathbf{y}) \neq -\infty$. Moreover, $\bigsqcup_{i \in \mathbb{N}} \eta_i$ on \mathbf{y} is finitely reached in the chain $\{\eta_i\}_{i \in \mathbb{N}}$, i.e., there exists $m \in \mathbb{N}$ such that for all $i \geq m+1$

$$\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y} = \eta_i \mathbf{y}.$$

The inductive hypothesis holds for C and true , hence for $C + \text{true}$, therefore for all $\mathbf{x} \in \text{Var}$ and $j \in \{0, 1, \dots, m\}$, if $\min(\eta_{j+1} \mathbf{x}) < -(C + \text{true})_{\mathbf{b}} = -(C)_{\mathbf{b}}$ then there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ such that $|h| \leq (C)_{\mathbf{b}}$ and

- (a) $-\infty \neq \min(\eta_{j+1} \mathbf{x}) = \min(\eta_j \mathbf{z}) + h$,
- (b) $\forall \eta' \dot{\supseteq} \eta_j. \min(\langle\langle C + \text{true} \rangle\rangle \eta' \mathbf{x}) \leq \min(\eta' \mathbf{z}) + h$.

To shortly denote that the two conditions (a) and (b) hold, we write

$$(\mathbf{z}, j) \rightarrow_h (\mathbf{x}, j+1)$$

Now, assume that for some variable $\mathbf{y} \in \text{Var}$

$$\min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) = \min(\eta_{m+1} \mathbf{y}) < -(\text{fix}(C))_{\mathbf{b}} = -(n+1)(C)_{\mathbf{b}}$$

where $n = |\text{vars}(C)|$. We want to show that the thesis holds, i.e., that there exist $\mathbf{z} \in \text{Var}$ and $h \in \mathbb{Z}$ with $|h| \leq (\text{fix}(C))_{\mathbf{b}}$ such that:

$$\min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) = \min(\eta \mathbf{z}) + h \tag{A.18}$$

and for all $\eta' \dot{\supseteq} \eta$,

$$\min(\langle\langle \text{fix}(C) \rangle\rangle \eta' \mathbf{y}) \leq \min(\eta' \mathbf{z}) + h \tag{A.19}$$

Let us consider (i). We first observe that we can define a path

$$\sigma \triangleq (\mathbf{y}_0, 0) \rightarrow_{h_0} (\mathbf{y}_1, 1) \rightarrow_{h_1} \dots \rightarrow_{h_m} (\mathbf{y}_{m+1}, m+1) \tag{A.20}$$

such that $\mathbf{y}_{m+1} = \mathbf{y}$ and for all $j \in \{0, \dots, m+1\}$, $\mathbf{y}_j \in \text{Var}$ and $\min(\eta_j \mathbf{y}_j) < -(C)_{\mathbf{b}}$. In fact, if, by contradiction, this is not the case, there would exist an index $i \in \{0, \dots, m\}$ (as

$$\min(\eta_{m+1} \mathbf{y}_{m+1}) < -(C)_{\mathbf{b}}$$

already holds) such that $\min(\eta_i \mathbf{y}_i) \geq -(C)_{\mathbf{b}}$, while for all $j \in \{i+1, \dots, m+1\}$, $\min(\eta_j \mathbf{y}_j) < -(C)_{\mathbf{b}}$. Thus, in such a case, we consider the nonempty path:

$$\pi \triangleq (\mathbf{y}_i, i) \rightarrow_{h_i} (\mathbf{y}_{i+1}, i+1) \rightarrow_{h_{i+1}} \dots \rightarrow_{h_m} (\mathbf{y}_{m+1}, m+1)$$

and we have that:

$$\begin{aligned} \sum_{j=i}^m h_j &= \\ \sum_{j=i}^m \min(\eta_{j+1} \mathbf{y}_{j+1}) - \min(\eta_j \mathbf{y}_j) &= \\ \min(\eta_{m+1} \mathbf{y}_{m+1}) - \min(\eta_i \mathbf{y}_i) &= \\ \min(\eta_{m+1} \mathbf{y}) - \min(\eta_i \mathbf{y}_i) &< \\ -(n+1)(C)_{\mathbf{b}} + (C)_{\mathbf{b}} &= -n(C)_{\mathbf{b}} \end{aligned}$$

with $|h_j| \leq (C)_b$ for $j \in \{i, \dots, m\}$. Hence we can apply Lemma A.1 to the projection π_p of the nodes of this path π to the variable component to deduce that π_p has a subpath which is a cycle with a strictly negative weight. More precisely, there exist $i \leq k_1 < k_2 \leq m+1$ such that $y_{k_1} = y_{k_2}$ and $h = \sum_{j=k_1}^{k_2-1} h_j < 0$. If we denote $\mathbf{w} = y_{k_1} = y_{k_2}$, then we have that

$$\begin{aligned} \min(\eta_{k_2} \mathbf{w}) &= h_{k_2-1} + \min(\eta_{k_2-1} \mathbf{w}) \\ &= h_{k_2-1} + h_{k_2-2} + \min(\eta_{k_2-2} \mathbf{w}) \\ &= \sum_{j=k_1}^{k_2-1} h_j + \min(\eta_{k_1} \mathbf{w}) \\ &= \min(\eta_{k_1} \mathbf{w}) + h \end{aligned} \quad \text{recall } h = \sum_{j=k_1}^{k_2-1} h_j < 0$$

Thus,

$$\min(\langle\langle C + \text{true} \rangle\rangle^{k_2-k_1} \eta_{k_1} \mathbf{w}) = \min(\eta_{k_1} \mathbf{w}) + h$$

Observe that for all $\eta' \dot{\supseteq} \eta_{k_1}$

$$\min(\langle\langle C + \text{true} \rangle\rangle^{k_2-k_1} \eta' \mathbf{w}) \leq \min(\eta' \mathbf{w}) + h \quad (\text{A.21})$$

This property (A.21) can be shown by induction on $k_2 - k_1 \geq 1$.

Then, an inductive argument allows us to show that for all $r \in \mathbb{N}$:

$$\min(\langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w}) \leq \min(\eta_{k_1} \mathbf{w}) + rh \quad (\text{A.22})$$

In fact, for $r = 0$ the claim trivially holds. Assuming the validity for $r \geq 0$ then we have that

$$\begin{aligned} \min(\langle\langle C + \text{true} \rangle\rangle^{(r+1)(k_2-k_1)} \eta_{k_1} \mathbf{w}) &= \\ \min(\langle\langle C + \text{true} \rangle\rangle^{k_2-k_1} (\langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w})) &\leq \quad \text{by (A.21) as } \eta_{k_1} \dot{\subseteq} \langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \\ \min(\langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w}) + h &\leq \quad \text{by inductive hypothesis} \\ \min(\eta_{k_1} \mathbf{w}) + rh + h &\leq \min(\eta_{k_1} \mathbf{w}) + (r+1)h \end{aligned}$$

However, This would contradict the hypothesis $\min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) \neq -\infty$. In fact the inequality (A.22) would imply

$$\begin{aligned} \min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{w}) &= \min \left(\bigsqcup_{i \in \mathbb{N}} \langle\langle C + \text{true} \rangle\rangle^i \eta \mathbf{w} \right) \\ &= \min \left(\bigsqcup_{i \in \mathbb{N}} \langle\langle C + \text{true} \rangle\rangle^i \eta_{k_1} \mathbf{w} \right) \\ &= \min \left(\bigsqcup_{r \in \mathbb{N}} \langle\langle C + \text{true} \rangle\rangle^{r(k_2-k_1)} \eta_{k_1} \mathbf{w} \right) \\ &= -\infty \end{aligned}$$

Now, from (A.20) we deduce that for all $\eta' \dot{\supseteq} \eta_{k_1}$, for $j \in \{k_1, \dots, m\}$, if we let $\mu_{k_1} = \eta'$ and $\mu_{j+1} = \langle\langle C + \text{true} \rangle\rangle \mu_j$, we have that $\min(\mu_{j+1} y_{j+1}) \leq \min(\mu_j y_j) + h_j$ and thus

$$\langle\langle C + \text{true} \rangle\rangle^{m-k_1+1} \eta' \mathbf{y} = \mu_{m+1} y_{m+1} \leq \min(y_{k_1}) + \sum_{i=k_1}^m h_i = \min(\eta' \mathbf{w}) + \sum_{i=k_1}^m h_i$$

Since $\eta' = \langle\langle \text{fix}(C) \rangle\rangle \eta \dot{\supseteq} \eta_{k_1}$ we conclude

$$\begin{aligned} \min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) &= \min \left(\langle\langle C + \text{true} \rangle\rangle^{m-k_1+1} \langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y} \right) \\ &\leq -\infty + \sum_{i=k_1}^m h_i = -\infty \end{aligned}$$

contradicting the assumption. Therefore, the path σ of (A.20) must exist, and consequently

$$\min(\langle\langle \text{fix}(C) \rangle\rangle \eta \mathbf{y}) = \min(\eta_{m+1} \mathbf{y}) = \min(\eta y_0) + \sum_{i=0}^m h_i$$

and $|\sum_{i=0}^m h_i| \leq (\text{fix}(\mathbf{C}))_{\mathbf{b}} = (n+1)(\mathbf{C})_{\mathbf{b}}$, otherwise we could use the same argument above for inferring the contradiction $\min(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta \mathbf{y}) = -\infty$.

Let us now show (ii). Given $\eta' \dot{\supseteq} \eta$ from (A.20) we deduce that for all $j \in \{0, \dots, m\}$, if we let $\mu_0 = \eta'$ and $\mu_{j+1} = \langle\langle \mathbf{C} + \text{true} \rangle\rangle \mu_j$, we have that

$$\min(\mu_{j+1} \mathbf{y}_{j+1}) \leq \min(\mu_{j+1} \mathbf{y}_j) + h_j.$$

Therefore, since $\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta' \dot{\supseteq} \mu_{m+1}$ (observe that the convergence of $\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta'$ could be at an index greater than $m+1$), we conclude that:

$$\min(\langle\langle \text{fix}(\mathbf{C}) \rangle\rangle \eta' \mathbf{y}) \leq \min(\mu_{m+1} \mathbf{y}) = \min(\mu_{m+1} \mathbf{y}_{m+1}) \leq \min(\eta' \mathbf{y}_0) + \sum_{i=0}^m h_i$$

as desired. □

Bibliography

- [CC77] Patrick Cousot and Radhia Cousot. “Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints”. In: *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*. POPL ’77. Los Angeles, California: Association for Computing Machinery, 1977, pp. 238–252. ISBN: 9781450373500. DOI: [10.1145/512950.512973](https://doi.org/10.1145/512950.512973). URL: <https://doi.org/10.1145/512950.512973> (cit. on pp. iii, 2, 7, 8, 11).
- [CC79] Patrick Cousot and Radhia Cousot. “Systematic Design of Program Analysis Frameworks”. In: *Proceedings of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*. POPL ’79. San Antonio, Texas: Association for Computing Machinery, 1979, pp. 269–282. ISBN: 9781450373579. DOI: [10.1145/567752.567778](https://doi.org/10.1145/567752.567778). URL: <https://doi.org/10.1145/567752.567778> (cit. on pp. 2, 7).
- [CC92] Patrick Cousot and Radhia Cousot. “Comparing the Galois connection and widening/narrowing approaches to abstract interpretation”. In: *Programming Language Implementation and Logic Programming*. Ed. by Maurice Bruynooghe and Martin Wirsing. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 269–295. ISBN: 978-3-540-47297-1 (cit. on p. 3).
- [Cou+05] Patrick Cousot et al. “The ASTREÉ Analyzer”. In: *Programming Languages and Systems*. Ed. by Mooly Sagiv. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 21–30. ISBN: 978-3-540-31987-0 (cit. on pp. 2, 68).
- [Cut80] Nigel Cutland. *Computability: An introduction to recursive function theory*. Cambridge university press, 1980 (cit. on pp. 5, 22, 23).
- [Dis+19] Dino Distefano et al. “Scaling static analyses at Facebook”. In: *Commun. ACM* 62.8 (July 2019), pp. 62–70. ISSN: 0001-0782. DOI: [10.1145/3338112](https://doi.org/10.1145/3338112). URL: <https://doi.org/10.1145/3338112> (cit. on p. 1).
- [Dow97] Mark Dowson. “The Ariane 5 software failure”. In: *SIGSOFT Softw. Eng. Notes* 22.2 (Mar. 1997), p. 84. ISSN: 0163-5948. DOI: [10.1145/251880.251992](https://doi.org/10.1145/251880.251992). URL: <https://doi.org/10.1145/251880.251992> (cit. on p. 1).
- [Eis+89] T. Eisenberg et al. “The Cornell commission: on Morris and the worm”. In: *Commun. ACM* 32.6 (June 1989), pp. 706–709. ISSN: 0001-0782. DOI: [10.1145/63526.63530](https://doi.org/10.1145/63526.63530). URL: <https://doi.org/10.1145/63526.63530> (cit. on p. 1).
- [Gaw+09] Thomas Gawlitza et al. *Polynomial Precise Interval Analysis Revisited*. Ed. by Susanne Albers, Helmut Alt, and Stefan Näher. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 422–437. ISBN: 978-3-642-03456-5. DOI: [10.1007/978-3-642-03456-5_28](https://doi.org/10.1007/978-3-642-03456-5_28). URL: https://doi.org/10.1007/978-3-642-03456-5_28 (cit. on pp. 4, 57).
- [GR22] Roberto Giacobazzi and Francesco Ranzato. “History of Abstract Interpretation”. In: *IEEE Annals of the History of Computing* 44.2 (2022), pp. 33–43. DOI: [10.1109/MAHC.2021.3133136](https://doi.org/10.1109/MAHC.2021.3133136) (cit. on p. 7).

- [HM08] Tony Hoare and Jay Misra. “Verified Software: Theories, Tools, Experiments Vision of a Grand Challenge Project”. In: *Verified Software: Theories, Tools, Experiments: First IFIP TC 2/WG 2.3 Conference, VSTTE 2005, Zurich, Switzerland, October 10-13, 2005, Revised Selected Papers and Discussions*. Ed. by Bertrand Meyer and Jim Woodcock. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–18. ISBN: 978-3-540-69149-5. DOI: [10.1007/978-3-540-69149-5_1](https://doi.org/10.1007/978-3-540-69149-5_1). URL: https://doi.org/10.1007/978-3-540-69149-5_1 (cit. on p. 1).
- [JOW06] C. Jones, P. O’Hearn, and J. Woodcock. “Verified software: a grand challenge”. In: *Computer* 39.4 (2006), pp. 93–95. DOI: [10.1109/MC.2006.145](https://doi.org/10.1109/MC.2006.145) (cit. on p. 1).
- [Koc+19] Paul Kocher et al. “Spectre Attacks: Exploiting Speculative Execution”. In: *40th IEEE Symposium on Security and Privacy*. 2019 (cit. on p. 1).
- [Kön26] Dénes König. “Sur les correspondances multivoques des ensembles”. In: *Fundamenta Mathematicae* 8 (1926), pp. 114–134. DOI: [10.4064/fm-8-1-114-134](https://doi.org/10.4064/fm-8-1-114-134) (cit. on p. 27).
- [Koz97] Dexter Kozen. “Kleene Algebra with Tests”. In: *ACM Trans. Program. Lang. Syst.* 19.3 (May 1997), pp. 427–443. ISSN: 0164-0925. DOI: [10.1145/256167.256195](https://doi.org/10.1145/256167.256195). URL: <https://doi.org/10.1145/256167.256195> (cit. on p. 13).
- [Lac+98] Ph. Lacan et al. “ARIANE 5 - The Software Reliability Verification Process”. In: *DASIA 98 - Data Systems in Aerospace*. Ed. by B. Kaldeich-Schürmann. Vol. 422. ESA Special Publication. July 1998, p. 201 (cit. on p. 1).
- [Le 97] G. Le Lann. “An analysis of the Ariane 5 flight 501 failure-a system engineering perspective”. In: *Proceedings International Conference and Workshop on Engineering of Computer-Based Systems*. 1997, pp. 339–346. DOI: [10.1109/ECBS.1997.581900](https://doi.org/10.1109/ECBS.1997.581900) (cit. on p. 1).
- [Lef+24] Engel Lefauchaux et al. “Porous invariants for linear systems”. In: *Formal Methods in System Design* (Feb. 2024). ISSN: 1572-8102. DOI: [10.1007/s10703-024-00444-3](https://doi.org/10.1007/s10703-024-00444-3). URL: <https://doi.org/10.1007/s10703-024-00444-3> (cit. on pp. 61, 68).
- [Lip+18] Moritz Lipp et al. “Meltdown: Reading Kernel Memory from User Space”. In: *27th USENIX Security Symposium (USENIX Security 18)*. 2018 (cit. on p. 1).
- [Mat70] Yu. V. Matiyasevich. “Enumerable sets are diophantine”. English. In: *Sov. Math., Dokl.* 11 (1970), pp. 354–358. ISSN: 0197-6788 (cit. on p. 68).
- [Min18] Antonie Miné. *Static Inference of Numeric Invariants by Abstract Interpretation*. Université Pierre et Marie Curie, Paris, France, 2018 (cit. on pp. 6, 7).
- [MOM23] Raphaël Monat, Abdelraouf Ouadjaout, and Antoine Miné. “Mopsa-C: Modular Domains and Relational Abstract Interpretation for C Programs (Competition Contribution)”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by Sriram Sankaranarayanan and Natasha Sharygina. Cham: Springer Nature Switzerland, 2023, pp. 565–570. ISBN: 978-3-031-30820-8 (cit. on pp. 2, 68).
- [OHe19] Peter W. O’Hearn. “Incorrectness logic”. In: *Proc. ACM Program. Lang.* 4.POPL (Dec. 2019). DOI: [10.1145/3371078](https://doi.org/10.1145/3371078). URL: <https://doi.org/10.1145/3371078> (cit. on p. 1).
- [Orm03] H. Orman. “The Morris worm: a fifteen-year perspective”. In: *IEEE Security & Privacy* 1.5 (2003), pp. 35–43. DOI: [10.1109/MSECP.2003.1236233](https://doi.org/10.1109/MSECP.2003.1236233) (cit. on p. 1).
- [Pre29] M. Presburger. “Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchen die Addition als einzige Operation hervortritt”. In: *Comptes-Rendus du ler Congres des Mathématiciens des Pays Slavs* (1929). URL: <https://cir.nii.ac.jp/crid/1571698599431503232> (cit. on p. 68).
- [Ric53] Henry Gordon Rice. “Classes of recursively enumerable sets and their decision problems”. In: *Transactions of the American Mathematical society* 74.2 (1953), pp. 358–366 (cit. on pp. 1, 25).

- [See89] Donn Seeley. “A Tour of the Worm”. In: *Proceedings of 1989 Winter USENIX Conference, Usenix Association, San Diego, CA, February*. 1989 (cit. on p. 1).
- [Spa89] Eugene H. Spafford. “The internet worm program: an analysis”. In: *SIGCOMM Comput. Commun. Rev.* 19.1 (Jan. 1989), pp. 17–57. ISSN: 0146-4833. DOI: [10.1145/66093.66095](https://doi.org/10.1145/66093.66095). URL: <https://doi.org/10.1145/66093.66095> (cit. on p. 1).
- [SW05] Zhendong Su and David Wagner. “A class of polynomially solvable range constraints for interval analysis without widenings”. In: *Theoretical Computer Science* 345.1 (2005). Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2004), pp. 122–138. ISSN: 0304-3975. DOI: <https://doi.org/10.1016/j.tcs.2005.07.035>. URL: <https://www.sciencedirect.com/science/article/pii/S0304397505003889> (cit. on p. 4).
- [Tar55] Alfred Tarski. “A lattice-theoretical fixpoint theorem and its applications.” In: (1955) (cit. on p. 7).
- [Tur21] Alan Mathison Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem (1936)”. In: *Ideas That Created the Future: Classic Papers of Computer Science*. The MIT Press, Feb. 2021. ISBN: 9780262363174. DOI: [10.7551/mitpress/12274.003.0008](https://doi.org/10.7551/mitpress/12274.003.0008). eprint: https://direct.mit.edu/book/chapter-pdf/2248314/9780262363174/_c000500.pdf. URL: <https://doi.org/10.7551/mitpress/12274.003.0008> (cit. on p. 1).
- [Woo06] J. Woodcock. “First Steps in the Verified Software Grand Challenge”. In: *Computer* 39.10 (2006), pp. 57–64. DOI: [10.1109/MC.2006.340](https://doi.org/10.1109/MC.2006.340) (cit. on p. 1).

Acknowledgments

To my family.