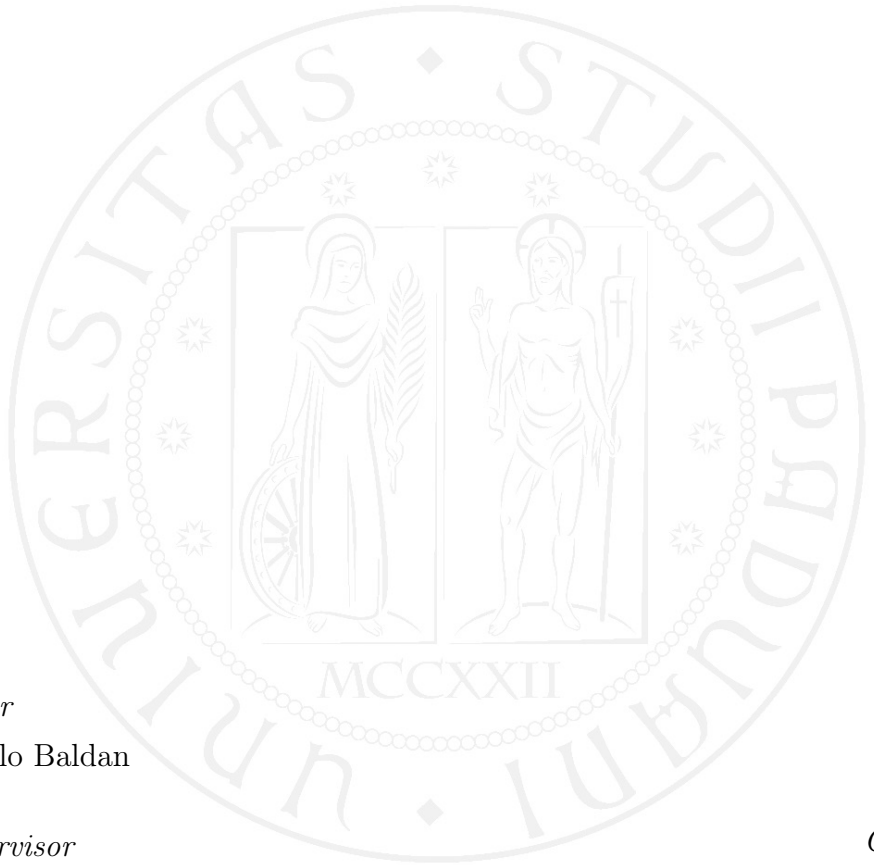
University of Padova

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 trough 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 ensuring 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 thesis focuses on a special abstract domain, called the *intervals* domain, where each variable of program is assigned to an interval over the integer numbers. The thesis argues that in such a context widening and narrowing operators can be replaced by another method, that relies on deciding program divergence by looking at the behavior of variables in the context of the program.

Acknowledgments

To my family.

Contents

Chapter 1

Background

The following chapter aims to provide context, notation and the important external references for the work that will follow on. It is structured as follows:

- In Section 1.1 we make an introduction on the topic, exploring its importance and significance;
- in Section 1.2 we will introduce some notation and the important aspect of recursion theory needed to understand the following chapters;
- finally in Section 1.3 we explore order theory and set the notation that we will use in the rest of the thesis to talk about this topic.

1.1 Introduction and related work

Abstract interpretation has faced a difficult challenge since the beginning: the termination of the analyzer. When viewed from another perspective, every interpreter can be considered the most precise possible static analyzer. However, the critical issue is its non-termination. Cousot addresses this problem from the outset in [CC77] by introducing two operators that have since become standard: Widening and narrowing. The former is used to infer properties related to the termination of a loop, albeit at the cost of analysis precision (while still maintaining soundness). However, is it possible to forgo the use of widening operators and still achieve a sound analysis with the same precision as the one utilizing these operators? [Gaw+09] is the first to introduce and demonstrate that this is indeed feasible, presenting an algorithm for calculating fixed-point equation systems using operations and the abstract domain of intervals in polynomial time. It relies on a generalization of the Bellman-Ford algorithm from [Bel58] to find a least solution for system of equations with addition and least upper bound. The method is then extended until the authors build a cubic time algorithm for the class of interval equations (equations with variables in the interval domain).

1.2 Recursion theory

This first section aims to provide background and terminology for the parts in recursion theory that will follow. More in detail, we'll take some notation from [Cut80] and introduce some new 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

$$X \to Y$$

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. Hence

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

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 1.1 (partial and total recursive functions)**.** By $\mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$ we denote the set of partial recursive functions on natural numbers, while by $\mathbb{N} \overset{r}{\to} \mathbb{N}$ we denote the set of partial 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 1.3 (Recursively enumerable and recursive sets)**.** A set $A \subseteq \mathbb{N}^k$ is *recursively enumerable* (r.e. or semi-decidable) if $A = dom(f)$ for some $f \in \mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$.

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

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

Lemma 1.1 (Computable function over a recursive set)**.** *Given $f : A \overset{r}{\to} B$, let the domain A to be recursive. B is at least r. e.*

Proof. $f : A \overset{r}{\to} B$ total recursive function over a recursive set A. We can write the function

$$\mathcal{X}_B = \lambda x.sg(\mu z.|f(z) - x|)$$

which is computable as it is composition of computable functions. In other terms, this function

$$\mathcal{X}_B(x) = \begin{cases} 1 & x \in B \\ \uparrow & \text{otherwise} \end{cases}$$

is the semi-decision function for B □

Observation 1.1. In general, B is not recursive, as it would means that both $A \leqslant_m B$ and $B \leqslant_m A$, which is not always the case, but it is r.e., as we could always write the inverse function as in lemma 1.1 and derive a semi-decision function for the image of the function.

1.3 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 1.4 (Partiall 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. $x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y$ (antisymmetry)
3. $x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z$ (transitivity)

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

Definition 1.5 (Least upper bound)**.** Let $\langle X, \sqsubseteq \rangle$ be a POSet and let $Z \subseteq X$. We say that $\overline{z} \in Z$ is an *upper bound* of Z if $\forall z \in Z \; z \sqsubseteq \overline{z}$. It is the *least upper bound* of Z (denoted as $\sqcup Z$) if $\forall z'$ upper bounds of Z, $\overline{z} \sqsubseteq z'$.

Definition 1.6 (Greatest lower bound)**.** Let $\langle X, \sqsubseteq \rangle$ be a POSet and let $Z \subseteq X$. We say that $\overline{z} \in Z$ is a *lower bound* of Z if $\forall z \in Z \; \overline{z} \sqsubseteq z$. It is the *greatest lower bound* of Z (denoted as $\sqcap Z$) if $\forall z'$ upper bounds of Z, $z' \sqsubseteq \overline{z}$.

Usually then we are talking about least and greatest lower bound the host set is often implicit, and we therefore simply write $\cup Z$ and $\cap Z$. 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 1.7 (Complete lattice)**.** A POSet $\langle X, \sqsubseteq \rangle$ is called a *complete lattice* if

$$\forall Y \subseteq X \quad \exists \cup Y \wedge \exists \cap Y$$

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 1.8 (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 1.9. $\langle D, \sqsubseteq \rangle$ is a chain complete partially ordered set (ccpo) if 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 1.10 (Monotone functions)**.** Let $\langle D, \sqsubseteq \rangle$ and $\langle D', \sqsubseteq' \rangle$ be complete lattices. The total function $f : D \to D'$ is *monotone* if

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

Monotonicity however does not preserve upper bounds, just orders. In particular if we take a chain $Y \subseteq D$ of some ccpo $\langle D, \subseteq \rangle$ and some monotone function $f : D \to D$, 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 1.11 (Continuous functions)**.** Let $\langle X, \sqsubseteq \rangle$ and $\langle X', \sqsubseteq' \rangle$ be ccpos. The total function $f : D \to D'$ 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 1.1 (Kleene fixed-point)**.** *Let $f : D \to D$ be a continuous function over a chain complete partial order $\langle D, \sqsubseteq \rangle$ with the lest element $\bot$. Then*

$$lfp(f) = \sqcup\{f^n(\bot) \mid n \in \mathbb{N}\}$$

where

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

is the least fix point of f.

1.4 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] and 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 aspect 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].

1.4.1 General concepts

Abstract interpretation heavily relies on order theory, which we introduced in Section 1.3, 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 $\langle C, \leqslant \rangle$ (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

Definition 1.12 (Concretization)**.** A concretization function $\gamma : \langle A, \sqsubseteq \rangle \to \langle C, \leqslant \rangle$ is a *monotonic* function, i.e.

$$\forall a, a' \in A \quad a \sqsubseteq a' \Rightarrow \gamma(a) \leqslant \gamma(a')$$

Trough concretization we have a first notion of *soundness*

Definition 1.13 (Soundness)**.** $a \in A$ is a *sound* abstraction of $c \in C$ iff $c \leqslant \gamma(a)$.

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* $\alpha : \langle C, \leqslant \rangle \to \langle A, \sqsubseteq \rangle$, such that $\langle \alpha, C, A, \gamma \rangle$ forms a Galois connection:

Definition 1.14 (Galois connection)**.** Given two partially ordered sets $\langle C, \leqslant \rangle, \langle A, \sqsubseteq \rangle$, the tuple $\langle \alpha, C, A, \gamma \rangle$ is a Galois connection if

- A, C are complete lattices;

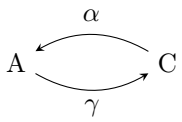


Figure 1.1: Galois connection between an abstract domain A and a concrete domain C

- $\alpha : \langle C, \leqslant \rangle \to \langle A, \sqsubseteq \rangle$ and $\gamma : \langle A, \sqsubseteq \rangle \to \langle C, \leqslant \rangle$ are monotonic;
- for all $a \in A, c \in C$,

$$c \leqslant \gamma(a) \iff \alpha(c) \sqsubseteq a. \tag{1.1}$$

We denote $\langle \alpha, C, A, \gamma \rangle$ as $\langle C, \leqslant \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq \rangle$.

Usually though we use an alternative characterization of Galois Connections, which is easier to prove:

Theorem 1.2. *$\langle C, \leqslant \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq \rangle$ is a Galois connection iff the function pair $\langle \alpha, \gamma \rangle$ satisfies all the following properties:*

(1) α, γ are monotonic;

(2) $\forall c \in C \quad c \leqslant \gamma(\alpha(c))$ 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 1.2 hold.

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

$$c \leqslant \gamma(\alpha(c)) \tag{1.2}$$

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

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

$$\alpha(\gamma(a)) \sqsubseteq a \tag{1.3}$$

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

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

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

(⇒) First assume that $c \leqslant \gamma(a)$. Then $\alpha(c) \sqsubseteq \alpha(\gamma(a))$ by monotonicity of α and $\alpha(\gamma(a)) \sqsubseteq a$ by reductivity, hence $\alpha(c) \sqsubseteq a$.

(⇐) Likewise, assume that $\alpha(c) \sqsubseteq a$. Then $\gamma(\alpha(c)) \leqslant \gamma(a)$ by monotonicity of γ and $c \leqslant \gamma(\alpha(c))$ by extensivity, hence $c \leqslant \gamma(a)$.

□

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 1.3 (Galois connection properties)**.** *Given a Galois connection* $\langle C, \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq\rangle$ *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. $\forall c \in C\ \alpha(c) = \sqcap\{a \mid c \leqslant \gamma(a)\}$*;*
4. $\forall a \in A\ \gamma(a) = \vee\{c \mid \alpha(c) \sqsubseteq a\}$*;*
5. α *maps concrete lubs to abstract lubs:*

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

6. γ *maps abstract glbs to concrete glbs:*

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

Theorem 1.3 states two important properties of Galois connections: *soundness* and *optimality*. Recall that $c \leqslant \gamma(a)$ means by Definition 1.13 that a is a *sound* approximation of c. Then, given $c \in C$ recall that $c \leqslant \gamma(\alpha(c))$, 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 1.1 (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 1.15 (Galois Insertion)**.** A Galois connection $\langle C, \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq\rangle$ 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: $\forall a, a' \in A\ \gamma(a) = \gamma(a') \Rightarrow a = a'$;
3. $\alpha \circ \gamma$ is the identity function.

We denote Galois insertions as $\langle C, \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows\!\!\!\!\rightarrow}} \langle A, \sqsubseteq\rangle$.

Nexr, 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-wilse lifting.

Theorem 1.4. *Given a Galois connection* $\langle C, \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq\rangle$ *we can derive a new Galois connection with the* point wise lifting*, i.e.* $\langle S \to C, \dot{\leqslant}\rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle S \to A, \dot{\sqsubseteq}\rangle$ *where*

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

and S is an arbitrary set.

Proof.

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

□

from this, we can derive the following

Theorem 1.5. *if* $\langle C, \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq\rangle$ *is a Galois insertion, then its point-wise lifting is again a Galois insertion:*

$$\langle S \to C, \dot{\leqslant}\rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle S \to A, \dot{\sqsubseteq}\rangle$$

Proof. By Th. 1.4 it holds that $\langle S \to C, \dot{\leqslant}\rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle S \to A, \dot{\sqsubseteq}\rangle$. What we have to prove is that $\dot{\alpha} \circ \dot{\gamma} = \mathsf{id}$ kowing that $\alpha \circ \gamma = \mathsf{id}$.

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

hence $\dot{\alpha} \circ \dot{\gamma} = \mathsf{id}$, which means $\langle S \to C, \dot{\leqslant}\rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle S \to A, \dot{\sqsubseteq}\rangle$. □

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

Definition 1.16 (Sound and correct operator abstraction)**.** Let $\gamma : \langle A \sqsubseteq\rangle \to \langle C, \leqslant\rangle$ be a concretization map from an abstract domain $\langle A, \sqsubseteq\rangle$ to a concrete domain $\langle C, \leqslant\rangle$, $f : C \to C$ be a concrete operator and $g : A \to A$ an abstract operator.

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

Notice that a correct 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 1.17 (Best correct approximation (bca))**.** Given a Galois connection $\langle \alpha, C, A, \gamma\rangle$ and a concrete operator $f : C \to C$, 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 1.6 (Operator composition)**.** *Let* $f, f' : C \to C$ *be concrete operators and* $g, g' A \to A$ *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 correct abstractions of f and f' respectively then $g \circ g'$ is a correct abstraction of $f \circ f'$.

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

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

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

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

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

□

1.4.2 Fixpoint approximations

Critical parts of program semantics are defined trough the idea of least fixpoints $\mathrm{lfp} f$ of some monotonic function or continuous operator $f : C \to C$ over the concrete domain $\langle C, \leqslant \rangle$. In order to abstract the computation of $\mathrm{lfp} f$ in the abstract domain $\langle A, \sqsubseteq \rangle$ the natural idea is to start with a sound abstraction $g : A \to A$ of f. Then there are many ways of approximate the fixpoint computation trough 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 $\mathrm{lfp} f$ as the limit of an iteration sequence from Kleene's Theorem 1.1:

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

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

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

i.e., the least fixpoint of a monotonic function f over a complete lattice f is the greatest lower bound of the postfixpoints. The observation is that any abstract postfixpoint of a sound abstract rapresents, trough γ, a concrete fixpoint (by Theorem 1.3). The theorem however does not state how to compute a postfixpoint 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-existstent at all.

As mentioned before, in order to solve the convergence problem in abstract domains,Patrick and Radhia Cousot in [CC77] prposed to use Tarski's characterization to provide post fixpoint of infinite ascending chains, introducing a binary operator: the *widening* operator $\triangledown$.

Definition 1.18 (Widening operator)**.** A binary operator $\triangledown : A \times X \to A$ is a *widening operator* in an abstract domain $\langle A, \sqsubseteq \rangle$ if

1. it computes upper bounds:

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

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: $\exists k \geqslant 0 \mid x^{k+1} = x^k$.

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

Theorem 1.7. *Let f be a monotonic operator in a complete concrete lattice and g a sonud abstraction of f. Then the following iteration*

$$x^0 \triangleq \bot$$
$$x^{i+1} \triangleq x^i \triangledown g(x^i)$$

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

Proof. Convergence is ensured by Property 2 of Definition 1.18, while the soundness is because of Equation (1.4) given that, when encountering a stable value $x^{i+1} = x^i$, then $f(x^i) \sqsubseteq x^i \triangledown f(x^i) = x^{i+1}$, i.e., x^i is an abstract postfixpoint. □

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

2.1 The Imp language

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]. We denote by $\mathbb{Z}$ the set of naturals with the usual order, extended with the least and greatest elements $-\infty$ and $+\infty$, s.t. $-\infty \leqslant z \leqslant +\infty$ 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.

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

where $\mathtt{x}, \mathtt{y} \in \mathit{Var}$ a finite set of variables of interest, i.e., the variables appearing in the considered program, $S \in \mathbb{I}$ an *interval* (as defined in Definition 3.2), $a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}, a \leqslant b$ 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: $\mathsf{Env} \triangleq \{\rho \mid \rho : \mathit{Var} \to \mathbb{Z}\}$. For basic expressions $e \in \mathsf{Exp}$ the *concrete semantics* $(\!|\cdot|\!) : \mathsf{Exp} \to \mathsf{Env} \to \mathsf{Env}_\bot$ is inductively defined by:

$$\begin{aligned} (\!|\mathtt{x} \in S|\!)\rho &\triangleq \begin{cases} \rho & \rho(\mathtt{x}) \in S \\ \bot & \text{otherwise} \end{cases} \\ (\!|\mathtt{x} := k|\!)\rho &\triangleq \rho[\mathtt{x} \mapsto k] \\ (\!|\mathtt{x} := \mathtt{y} + k|\!)\rho &\triangleq \begin{cases} \rho[\mathtt{x} \mapsto \rho(\mathtt{y}) + k] & \rho \neq \bot \\ \bot & \text{otherwise} \end{cases} \\ (\!|\mathtt{x} := \mathtt{y} - k|\!)\rho &\triangleq \begin{cases} \rho[\mathtt{x} \mapsto \rho(\mathtt{y}) - k] & \rho \neq \bot \\ \bot & \text{otherwise} \end{cases} \end{aligned}$$

where with $\mathsf{Env}_\bot$ we mean $\mathsf{Env} \cup \{\bot\}$.

The next building block is the concrete collecting semantics for the language, it associates each program in 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 $\mathbb{C} \triangleq \langle 2^{\mathsf{Env}}, \subseteq \rangle$ be a complete lattice called *concrete collecting domain*. The *concrete collecting semantics* for Imp is given by the total function $\langle\cdot\rangle : \text{Imp} \to \mathbb{C} \to \mathbb{C}$ which maps each program $\mathsf{C} \in \text{Imp}$ to a total function over the complete lattice $\mathbb{C}$, inductively defined as follows: given $X \in \mathbb{C}$

$$\begin{aligned}
\langle \mathsf{e} \rangle X &\triangleq \{ (\!|\mathsf{e}|\!)\rho \mid \rho \in X, (\!|\mathsf{e}|\!)\rho \neq \bot \} \\
\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X &\triangleq \langle \mathsf{C}_1 \rangle X \cup \langle \mathsf{C}_2 \rangle X \\
\langle \mathsf{C}_1 ; \mathsf{C}_2 \rangle X &\triangleq \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X) \\
\langle \mathsf{C}^* \rangle X &\triangleq \bigcup_{i \in \mathbb{N}} \langle \mathsf{C} \rangle^i X \\
\langle \mathsf{fix}(\mathsf{C}) \rangle X &\triangleq \text{lfp}(\lambda Y \in 2^{\mathsf{Env}}.(X \cup \langle \mathsf{C} \rangle Y))
\end{aligned}$$

We observe that the semantics we described is additive:

Observation 2.1 (Additivity)**.** Given $\mathsf{C} \in \text{Imp}$, $X, Y \in \mathbb{C}$,

$$\langle \mathsf{C} \rangle (X \cup Y) = \langle \mathsf{C} \rangle X \cup \langle \mathsf{C} \rangle Y$$

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

Case (e). Therefore

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

Next we can explore the inductive cases.

Case ($\mathsf{C}_1 + \mathsf{C}_2$). Therefore

$$\begin{aligned}
\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle (X \cup Y) &= \langle \mathsf{C}_1 \rangle (X \cup Y) \cup \langle \mathsf{C}_2 \rangle (X \cup Y) && \text{by definition} \\
&= \langle \mathsf{C}_1 \rangle X \cup \langle \mathsf{C}_1 \rangle Y \cup \langle \mathsf{C}_2 \rangle X \cup \langle \mathsf{C}_2 \rangle Y && \text{by inductive hypothesis} \\
&= \langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X \cup \langle \mathsf{C}_1 + \mathsf{C}_2 \rangle Y
\end{aligned}$$

Case ($\mathsf{C}_1 ; \mathsf{C}_2$). Therefore

$$\begin{aligned}
\langle \mathsf{C}_1 ; \mathsf{C}_2 \rangle (X \cup Y) &= \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle (X \cup Y)) && \text{by definition} \\
&= \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X \cup \langle \mathsf{C}_1 \rangle Y) && \text{by inductive hypothesis} \\
&= \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X) \cup \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle Y) && \text{by inductive hypothesis}
\end{aligned}$$

Case (C^*). Therefore

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

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

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

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 $i \Rightarrow i + 1$:

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

Therefore we can use the inductive hypothesis internally and say

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

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

Lemma 2.1. *Given a program $\mathsf{C} \in$ Imp, the semantic function $\langle \mathsf{C} \rangle : \mathbb{C} \to \mathbb{C}$ is monotone.*

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

Case (e). In this case

$$\begin{aligned}
\langle \mathsf{e} \rangle X &= \{ (\!| \mathsf{e} |\!) \rho \mid \rho \in X, (\!| \mathsf{e} |\!) \rho \neq \bot \} \\
\langle \mathsf{e} \rangle Y &= \{ (\!| \mathsf{e} |\!) \rho \mid \rho \in Y, (\!| \mathsf{e} |\!) \rho \neq \bot \}
\end{aligned}$$

$X \subseteq Y$ therefore $\rho \in X \Rightarrow \rho \in Y$ which also means that $\rho' \in \langle \mathsf{e} \rangle X \Rightarrow \rho' \in \langle \mathsf{e} \rangle Y$, therefore $\langle \mathsf{e} \rangle X \subseteq \langle \mathsf{e} \rangle Y$

Case ($\mathsf{C}_1 + \mathsf{C}_2$). In this case we need to show that $\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X \subseteq \langle \mathsf{C}_1 + \mathsf{C}_2 \rangle Y$

$$\begin{aligned}
\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X &= \langle \mathsf{C}_1 \rangle X \cup \langle \mathsf{C}_2 \rangle X \\
\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle Y &= \langle \mathsf{C}_1 \rangle Y \cup \langle \mathsf{C}_2 \rangle Y
\end{aligned}$$

by inductive hypothesis both $\langle \mathsf{C}_1 \rangle X \subseteq \langle \mathsf{C}_1 \rangle Y$ and $\langle \mathsf{C}_2 \rangle X \subseteq \langle \mathsf{C}_2 \rangle Y$ and therefore $\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X \subseteq \langle \mathsf{C}_1 + \mathsf{C}_2 \rangle Y$.

Case ($\mathsf{C}_1; \mathsf{C}_2$). Therefore we need to show that $\langle \mathsf{C}_1; \mathsf{C}_2 \rangle X \subseteq \langle \mathsf{C}_1; \mathsf{C}_2 \rangle Y$

$$\begin{aligned}
\langle \mathsf{C}_1; \mathsf{C}_2 \rangle X &= \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X) \\
\langle \mathsf{C}_1; \mathsf{C}_2 \rangle Y &= \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle Y)
\end{aligned}$$

By induction hypothesis $\langle \mathsf{C}_1 \rangle X \subseteq \langle \mathsf{C}_1 \rangle Y$, and by induction hypothesis again $\langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X) \subseteq \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle Y)$ which means $\langle \mathsf{C}_1; \mathsf{C}_2 \rangle X \subseteq \langle \mathsf{C}_1; \mathsf{C}_2 \rangle Y$.

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

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

what we need to prove is that

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

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 $j \Rightarrow j + 1$. Notice that it holds that

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

  and also for Y

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

  Also notice that

  (i) $X \subseteq Y$ by hypothesis;

  (ii) $\bigcup_{i=0}^{k} \langle \mathsf{C} \rangle^i X \subseteq \bigcup_{i=0}^{k} \langle \mathsf{C} \rangle^i Y$ by inductive hypothesis;

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

  Therefore

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

□

Since concrete semantics is additive, the Kleene star (C^*) and the fixpoint ($\mathsf{fix}(\mathsf{C})$) have the same concrete semantics $\langle \mathsf{C}^* \rangle = \langle \mathsf{fix}(\mathsf{C}) \rangle$. In order to notice this, let $X \in \mathbb{C}, f = \lambda Y \in \mathbb{C}.\, (X \cup \langle \mathsf{C} \rangle Y)$ and recall that $f^0 X = X$ and $f^{n+1} X = X \cup \langle \mathsf{C} \rangle (f^n X)$.

$$\begin{aligned} \langle \mathsf{fix}(\mathsf{C}) \rangle X = \mathrm{lfp}(f) &= \bigcup \{ f^n \perp \mid n \in \mathbb{N} \} && \text{by fixpoint theorem (1.1)} \\ &= \bigcup_{i \in \mathbb{N}} \left(X \cup \langle \mathsf{C} \rangle^i X \right) && \text{by definition} \\ &= X \cup (X \cup \langle \mathsf{C} \rangle X) \cup \left(X \cup \langle \mathsf{C} \rangle X \cup \langle \mathsf{C} \rangle^2 X \right) \cup \dots \\ &= \bigcup_{i \in \mathbb{N}} \langle \mathsf{C} \rangle^i X \\ &= \langle \mathsf{C}^* \rangle X \end{aligned}$$

This will not be the case for the abstract semantics (cf. example 3.1), 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 $\mathsf{fix}(\mathsf{C})$. Since for a given program C and a set of initial states $X \in \mathbb{C}$ the collecting semantics $\langle \mathsf{C} \rangle X$ expresses properties that hold at the end of the execution of C we will in the following chapters usually refer to $\langle \mathsf{C} \rangle X$ as program *invariant*.

Notation 2.1 (Singleton shorthand)**.** Sometimes we need to consider the semantics over the singleton set $\{\rho\}$. In these cases we will write $\langle \mathsf{C} \rangle \rho$ instead of $\langle \mathsf{C} \rangle \{\rho\}$.

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.

$$\begin{aligned}
\mathtt{x} \in [a,b] &= \mathtt{x} \in S && \text{with } S = [a,b] \\
\mathtt{x} \leqslant k &= \mathtt{x} \in (-\infty, k] \\
\mathtt{x} > k &= \mathtt{x} \in [k+1, +\infty) \\
\mathsf{true} &= \mathtt{x} \in \mathbb{Z} \\
\mathsf{false} &= \mathtt{x} \in \varnothing \\
\mathtt{x} \in S_1 \vee \mathtt{x} \in S_2 &= (\mathtt{x} \in S_1) + (\mathtt{x} \in S_2) \\
\mathtt{x} \in S_1 \wedge \mathtt{x} \in S_2 &= (\mathtt{x} \in S_1); (\mathtt{x} \in S_2) \\
\mathsf{if}\ b\ \mathsf{then}\ C_1\ \mathsf{else}\ C_2 &= (\mathsf{e}; \mathsf{C}_1) + (\neg \mathsf{e}; \mathsf{C}_2) \\
\mathsf{while}\ b\ \mathsf{do}\ C &= \mathsf{fix}(b; \mathsf{C}); \neg b \\
\mathtt{x}\mathtt{+}\mathtt{+} &= \mathtt{x} := \mathtt{x} + 1
\end{aligned}$$

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*: $\mathsf{State} \triangleq \mathrm{Imp} \times \mathsf{Env}$ tuples of programs and program environments. With states we can define our small step semantics:

Definition 2.3 (Small step semantics)**.** The small step transition relation for the language Imp $\rightarrow$: $\mathsf{State} \times (\mathsf{State} \cup \mathsf{Env})$ is defined by the following rules:

$$\frac{(\!|e|\!)\rho \neq \bot}{\langle \mathsf{e}, \rho \rangle \rightarrow (\!|e|\!)\rho}\ \text{expr}$$

$$\frac{}{\langle \mathsf{C}_1 + \mathsf{C}_2, \rho \rangle \rightarrow \langle \mathsf{C}_1, \rho \rangle}\ \text{sum}_1 \qquad \frac{}{\langle \mathsf{C}_1 + \mathsf{C}_2, \rho \rangle \rightarrow \langle \mathsf{C}_2, \rho \rangle}\ \text{sum}_2$$

$$\frac{\langle \mathsf{C}_1, \rho \rangle \rightarrow \langle \mathsf{C}_1', \rho' \rangle}{\langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \rightarrow \langle \mathsf{C}_1'; \mathsf{C}_2, \rho' \rangle}\ \text{comp}_1 \qquad \frac{\langle \mathsf{C}_1, \rho \rangle \rightarrow \rho'}{\langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \rightarrow \langle \mathsf{C}_2, \rho' \rangle}\ \text{comp}_2$$

$$\frac{}{\langle \mathsf{C}^*, \rho \rangle \rightarrow \langle \mathsf{C}; \mathsf{C}^*, \rho \rangle}\ \text{star} \qquad \frac{}{\langle \mathsf{C}^*, \rho \rangle \rightarrow \rho}\ \text{star}_{\text{fix}}$$

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 2.2. *For any* $\mathsf{C} \in \mathit{Imp}, X \in 2^{\mathsf{Env}}$

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

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 $\langle \mathsf{e} \rangle X = \{(\!|e|\!)\rho \mid \rho \in X \wedge (\!|e|\!)\rho \neq \bot\}$, $\forall \rho \in X. \langle \mathsf{e}, \rho \rangle \to (\!|e|\!)\rho$ if $(\!|e|\!)\rho \neq \bot$, and because of the expr rule

$$\langle \mathsf{e} \rangle X = \{(\!|e|\!)\rho \mid \rho \in X \wedge (\!|e|\!)\rho \neq \bot\} = \{\rho' \in \mathsf{Env} \mid \rho \in X \langle \mathsf{e}, \rho \rangle \to \rho'\}$$

Case ($\mathsf{C}_1 + \mathsf{C}_2$). In this case $\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X = \langle \mathsf{C}_1 \rangle X \cup \langle \mathsf{C}_2 \rangle X$, $\forall \rho \in X. \langle \mathsf{C}_1 + \mathsf{C}_2, \rho \rangle \to \langle \mathsf{C}_1, \rho \rangle \vee \langle \mathsf{C}_1 + \mathsf{C}_2, \rho \rangle \to \langle \mathsf{C}_2, \rho \rangle$ respectively according to rules sum_1 and sum_2. By inductive hypothesis

$$\langle \mathsf{C}_1 \rangle X = \{\rho' \in \mathsf{Env} \mid \rho \in X, \langle \mathsf{C}_1, \rho \rangle \to^* \rho'\} \quad \langle \mathsf{C}_2 \rangle X = \{\rho' \in \mathsf{Env} \mid \rho \in X, \langle \mathsf{C}_2, \rho \rangle \to^* \rho'\}$$

Therefore

$$\begin{aligned}
\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle X &= \langle \mathsf{C}_1 \rangle X \cup \langle \mathsf{C}_2 \rangle X && \text{(by definition)} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}_1, \rho \rangle \to^* \rho'\} \cup \{\rho' \in \mathsf{Env} \mid \rho \in X, \langle \mathsf{C}_2, \rho \rangle \to^* \rho'\} && \text{(by ind. hp)} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}_1, \rho \rangle \to^* \rho' \vee \langle \mathsf{C}_2, \rho \rangle \to^* \rho'\} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}_1 + \mathsf{C}_2, \rho \rangle \to^* \rho'\}
\end{aligned}$$

Case ($\mathsf{C}_1; \mathsf{C}_2$). In this case $\langle \mathsf{C}_1; \mathsf{C}_2 \rangle X = \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X)$. By inductive hp $\langle \mathsf{C}_1 \rangle X = \{\rho' \in \mathsf{Env} \mid \rho \in X, \langle \mathsf{C}_1, \rho \rangle \to^* \rho'\} = Y$, by inductive hp again $\langle \mathsf{C}_2 \rangle Y = \{\rho' \in \mathsf{Env} \mid \rho \in Y, \langle \mathsf{C}_2, \rho \rangle \to^* \rho'\}$. Therefore

$$\begin{aligned}
\langle \mathsf{C}_1; \mathsf{C}_2 \rangle X &= \langle \mathsf{C}_2 \rangle (\langle \mathsf{C}_1 \rangle X) && \text{(by definition)} \\
&= \{\rho' \in \mathsf{Env} \mid \rho'' \in \{\rho''' \mid \rho \in X, \langle \mathsf{C}_1, \rho \rangle \to^* \rho'''\}, \langle \mathsf{C}_2, \rho'' \rangle \to^* \rho'\} && \text{(by ind. hp)} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}_1, \rho \rangle \to^* \rho'' \wedge \langle \mathsf{C}_2, \rho'' \rangle \to^* \rho'\} && \text{(by composition lemma)} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \to^* \rho'\}
\end{aligned}$$

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

$$\begin{aligned}
\langle \mathsf{C}^* \rangle X &= X \cup \langle \mathsf{C} \rangle X \cup \langle \mathsf{C} \rangle^2 X \cup \ldots && \text{(by definition)} \\
&= X \cup \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}, \rho \rangle \to^* \rho'\} \cup \ldots && \text{(by ind. hp)} \\
&= \cup_{i \in \mathbb{N}} \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}^i, \rho \rangle \to^* \rho'\} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \vee_{i \in \mathbb{N}} \langle \mathsf{C}^i, \rho \rangle \to^* \rho'\} \\
&= \{\rho' \in \mathsf{Env} \mid \rho \in X. \langle \mathsf{C}^*, \rho \rangle \to^* \rho'\} && \square
\end{aligned}$$

Note that $\langle \mathsf{C} \rangle X = \varnothing \iff \nexists \rho' \in \mathsf{Env}, \rho \in X \mid \langle \mathsf{C}, \rho \rangle \to^* \rho'$, in other words the collecting semantics of some program C starting from some states $X \in \mathbb{C}$ 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 operational semantics $\to$ 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 2.4 (Transition system)**.** The transition system for the language Imp is

$$\mathsf{TS} \triangleq \langle \mathsf{State} \cup \mathsf{Env}, \mathsf{Env}, \to \rangle$$

where

- $\mathsf{State} \cup \mathsf{Env}$ is the set of configurations in the system;
- Env is the set of terminal states;
- $\rightarrow$ is the small step semantics defined in Definition 2.3, 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 2.5 (Partial termination)**.** Let $\mathsf{C} \in \text{Imp}, \rho \in \mathsf{Env}$. We say C *partially halts* on ρ when there is at least one derivation sequence of finite length in the transition system starting with $\langle \mathsf{C}, \rho \rangle$ and ending in some state ρ':

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

Dually

$$\langle \mathsf{C}, \rho \rangle \Uparrow \iff \neg \langle \mathsf{C}, \rho \rangle \downarrow$$

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

Definition 2.6 (Universal termination)**.** Let $\mathsf{C} \in \text{Imp}, \rho \in \mathsf{Env}$. We say$\mathsf{C}$ *partially loops* on ρ when there is at least one derivation sequence of infinite length in the transition system tarting from $\langle \mathsf{C}, \rho \rangle$:

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

Dually

$$\langle \mathsf{C}, \rho \rangle \Downarrow \iff \neg \langle \mathsf{C}, \rho \rangle \uparrow$$

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

Example 2.3 shows a program that partially halts, while Example 2.2 shows a program that always loops. Notice that the absence of infinite derivation sequences implies that $\mathsf{TS}(\langle \mathsf{C}, \rho \rangle)$ is finite. Example 2.3 shows a program that partially loops, while example 2.1 shows a program that universally halts.

Example 2.1. Consider the program

$$\mathtt{x} := 0;$$

it universally halts, since $\forall \rho \in \mathsf{Env}, \rho \neq \bot$

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

according to the expr rule in definition 2.3. Therefore $\langle (\mathtt{x} := 0), \rho \rangle \Downarrow \forall \rho \in \mathsf{Env} \setminus \{\bot\}$.

Example 2.2. Consider the program P

$$(\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; \mathtt{x} < 0$$

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

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

$$\cdots \rightarrow^* \langle (\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; x < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + k] \rangle \rightarrow^* \dots$$

is always present.

$$\langle (\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; \mathtt{x} < 0, \rho\rangle \longrightarrow \langle \mathtt{x} < 0, \rho\rangle \not\to$$
$$\downarrow$$
$$\langle \mathtt{x} \geqslant; 0\mathtt{x{+}{+}}; (\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; \mathtt{x} < 0, \rho\rangle$$
$$\downarrow^*$$
$$\langle (\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + 1]\rangle \longrightarrow \langle \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + 1]\rangle \not\to$$
$$\downarrow^*$$
$$\langle (\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + k]\rangle \longrightarrow \langle \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + k]\rangle \not\to$$

Figure 2.1: Transition system of $(\mathtt{x} \geqslant 0; \mathtt{x{+}{+}})^*; \mathtt{x} < 0$

Example 2.3. Consider the program

$$(\mathtt{x{+}{+}})^*$$

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

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

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

$$\langle (\mathtt{x{+}{+}})^*, \rho[\mathtt{x} \mapsto 0]\rangle \to^* \langle (\mathtt{x{+}{+}})^*, \rho[\mathtt{x} \mapsto 1]\rangle \to^* \langle (\mathtt{x{+}{+}})^*, \rho[\mathtt{x} \mapsto 2]\rangle \to^* \ldots$$

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

Lemma 2.3 (Decomposition lemma)**.** *If $\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle \to^k \rho''$, then there exists a state ρ' and a natural number k_1, k_2 s.t. $\langle \mathsf{C}_1, \rho\rangle \to^{k_1} \rho'$ and $\langle \mathsf{C}_2, \rho'\rangle \to^{k_2} \rho''$, 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

$$\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle \to^0 \rho''$$

holds vacuously since $\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle$ and ρ'' are different.

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

$$\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle \to^{k+1} \rho''$$

can be written as

$$\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle \to \gamma \to^k \rho''$$

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

Case [comp_1]. Then $\gamma = \langle \mathsf{C}_1'; \mathsf{C}_2, \rho'\rangle$ and $\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle \to \langle \mathsf{C}_1'; \mathsf{C}_2, \rho'\rangle$ because $\langle \mathsf{C}_1, \rho\rangle \to \langle \mathsf{C}_1', \rho'\rangle$. Therefore we have

$$\langle \mathsf{C}_1'; \mathsf{C}_2, \rho'\rangle \to^k \rho''$$

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

$$\langle \mathsf{C}_1'; \rho'\rangle \to^{k_1} \rho''' \quad \wedge \quad \langle \mathsf{C}_2, \rho'''\rangle \to^{k_2} \rho''$$

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

$$\langle \mathsf{C}_1, \rho\rangle \to^{k_1+1} \rho'''$$

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

$$\langle \mathsf{C}_1; \mathsf{C}_2, \rho\rangle \to^{k+1} \rho''$$

which is our thesis.

Case [comp_2]. In this case $\gamma = \langle \mathsf{C}_2, \rho' \rangle$ because $\langle \mathsf{C}_1, \rho \rangle \to \rho'$ and it holds that

$$\langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \to \langle \mathsf{C}_2, \rho' \rangle \to^k \rho''$$

Hence our thesis follows by using the inductive hypothesis on $\langle \mathsf{C}_2, \rho' \rangle$ and by choosing $k_1 = 1, k_2 = k$. □

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

Corollary 2.1. *If* $\langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \to^* \rho''$ *then* $\exists \rho'$ *s.t.* $\langle \mathsf{C}_1, \rho \rangle \to^* \rho'$ *and* $\langle \mathsf{C}_2, \rho' \rangle \to^* \rho''$.

The second lemma states a similar but inverted property:

Lemma 2.4 (Composition lemma)**.** *If* $\langle \mathsf{C}_1, \rho \rangle \to^k \rho'$ *then* $\langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \to^k \langle \mathsf{C}_2, \rho' \rangle$

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 $\langle \mathsf{C}_1, \rho \rangle$ and ρ' are different.

Case ($k \Rightarrow k + 1$). In this case .... □

Corollary 2.2. *If* $\langle \mathsf{C}_1, \rho \rangle \to^* \rho'$ *then* $\langle \mathsf{C}_1; \mathsf{C}_2, \rho \rangle \to^* \langle \mathsf{C}_2, \rho' \rangle$.

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

Definition 2.7 (Reducts)**.** Let Imp^* denotes the set whose elements are statements in Imp. The reduction function $\mathsf{red} : \text{Imp} \to \text{Imp}^*$ is recursively defined by the following clauses:

$$\begin{aligned} \mathsf{red}(\mathsf{e}) &\triangleq \{\mathsf{e}\} \\ \mathsf{red}(\mathsf{C}_1 + \mathsf{C}_2) &\triangleq \{\mathsf{C}_1 + \mathsf{C}_2\} \cup \mathsf{red}(\mathsf{C}_1) \cup \mathsf{red}(\mathsf{C}_2) \\ \mathsf{red}(\mathsf{C}_1; \mathsf{C}_2) &\triangleq (\mathsf{red}(\mathsf{C}_1); \mathsf{C}_2) \cup \mathsf{red}(\mathsf{C}_2) \\ \mathsf{red}(\mathsf{C}^*) &\triangleq \{\mathsf{C}^*\} \cup (\mathsf{red}(\mathsf{C}); \mathsf{C}^*) \end{aligned}$$

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

$$\begin{aligned} \varnothing; \mathsf{C} &\triangleq \varnothing \\ \{\mathsf{C}_1, \ldots, \mathsf{C}_k\}; \mathsf{C} &\triangleq \{\mathsf{C}_1; \mathsf{C}, \ldots, \mathsf{C}_k; \mathsf{C}\} \end{aligned}$$

Notice that the set of reduction of any finite program $\mathsf{C} \in \text{Imp}$ 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 2.8 (Partially recursive functions)**.** The class $\mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$ of *partially recursive functions* is the least class of functions on the natural numbers which contains

(a) the zero function:

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

(b) the successor function

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

(c) the projection function

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

and is closed under

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

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

(2) primitive recursion: given $f : \mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$ and $g : \mathbb{N}^{k+2} \overset{r}{\hookrightarrow} \mathbb{N}$ we define $h : \mathbb{N}^{k+1} \overset{r}{\hookrightarrow} \mathbb{N}$ 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 $f : \mathbb{N}^{k+1} \overset{r}{\hookrightarrow} \mathbb{N}$, $h : \mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$ defined trough *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 $(a_1, \ldots, a_k)$ as input for an Imp program. We do this by special input states and variables: we can consider initial states $\rho = [\mathtt{x}_1 \mapsto a_1, \ldots, \mathtt{x}_k \mapsto a_k]$ where each special variable $\mathtt{x}_k$ 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 2.2 (Program output)**.** Let $\mathsf{Env} \ni \rho = [\mathtt{x}_1 \mapsto a_1, \ldots, \mathtt{x}_n \mapsto a_n]$. We say

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

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

Definition 2.9 (Imp computability)**.** Let $f : \mathbb{N}^k \to \mathbb{N}$ be a function. We say that f is Imp computable if

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

with $\rho = [\mathtt{x}_1 \mapsto a_1, \ldots, \mathtt{x}_k \mapsto a_k]$.

We argue that the class of function computed by Imp is the same as the set of partially recursive functions $\mathbb{N} \overset{r}{\hookrightarrow} \mathbb{N}$ (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 2.10 (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 2.8.

Lemma 2.5 (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:

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

  is Imp-computable:

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

- The successor function

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

  is Imp-computable:

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

- The projection function

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

  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 minimalization.

Lemma 2.6. *let $f : \mathbb{N}^k \to \mathbb{N}$, $g_1, \dots, g_k : \mathbb{N}^n \to \mathbb{N}$ and consider the composition*

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

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. □

Lemma 2.7. *Given $f : \mathbb{N}^k \to \mathbb{N}$ and $g : \mathbb{N}^{k+2} \to \mathbb{N}$ Imp computable, we argue that $h : \mathbb{N}^{k+1} \to \mathbb{N}$*

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

defined trough primitive recursion is Imp-computable.

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

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

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

Lemma 2.8. *Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be a Imp-computable function. Then the function $h : \mathbb{N}^k \to \mathbb{N}$ defined trough unbounded minimalization*

$$
h(\vec{x}) = \mu y.f(\vec{x}, y) = \begin{cases} \textit{least } z \textit{ s.t.} & \begin{cases} f(\vec{x}, z) = 0 \\ f(\vec{x}, z) \downarrow \quad f(\vec{x}, z') \neq 0 \quad \forall z < z' \end{cases} \\ \uparrow & \textit{otherwise} \end{cases} \tag{2.1}
$$

is Imp-computable.

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

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

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

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. □

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

$$
f \in \mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N} \Rightarrow \exists \mathsf{C} \in \text{Imp} \mid \langle \mathsf{C}, \rho \rangle \downarrow\downarrow b \iff f(a_1, \ldots, a_k) \downarrow b
$$

with $\rho = [\mathtt{x}_1 \mapsto a_1, \ldots, \mathtt{x}_k \mapsto a_k]$.

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

Definition 2.11 (Saturated set)**.** $A \subseteq \mathbb{N}$ is *saturated* (or *extensional*) is for all $x, y \in \mathbb{N}$

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

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

Meaning that deciding weather 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 2.3. *$\langle \mathsf{C}, \rho \rangle \uparrow\uparrow$ (i.e., $\langle \mathsf{C} \rangle \rho = \varnothing$) is undecidable.*

Proof. The set of functions $f \in \mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$ 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]. □

Corollary 2.4. *$\langle \mathsf{C}, \rho \rangle \downarrow\downarrow$ is undecidable.*

Proof. The set of functions $f \in \mathbb{N}^k \overset{r}{\hookrightarrow} \mathbb{N}$ 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]. □

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 $\langle \mathsf{C} \rangle X$ is finite for some program $\mathsf{C} \in \text{Imp}$ and some initial states $X \in \mathbb{C}$, we could decide whether $\langle \mathsf{C}, \rho \rangle \downdownarrows$ 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 2.9. *If $\mathsf{D} \in Imp_s$, and $X \in 2^{env}$ is finite, then*

(i). $\langle \mathsf{D} \rangle X$ *is finite;*

(ii). $\forall \rho \in X \ \langle \mathsf{D}, \rho \rangle \downdownarrows$

(iii). $|\mathsf{TS}(\langle \mathsf{D}, \rho \rangle)| < \infty$ *for all* $\rho \in X$.

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

Proof. By induction on the program D:

Base case:
$\mathsf{D} \equiv e$, therefore

(i). $\langle \mathsf{e} \rangle X = \{ (\!| e |\!) \rho \mid \rho \in X, (\!| e |\!) \rho \neq \bot \}$, which is finite, since X is finite;

(ii). by expr rule $\forall \rho \in X$ either $\langle e, \rho \rangle \to (\!| e |\!) \rho$ or $\langle e, \rho \rangle \not\to$. In both cases there are no infinite derivation sequences, and therefore $\mathsf{TS}(\langle \mathsf{e}, \rho \rangle) \downdownarrows$;

(iii). Notice that $\forall \rho \in X$ either by the expr rule $\langle \mathsf{e}, \rho \rangle \to (\!| \mathsf{e} |\!) \rho$ or $\langle \mathsf{e}, \rho \rangle \not\to$ therefore

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

Inductive cases:

1. $\mathsf{D} \equiv \mathsf{D}_1 + \mathsf{D}_2$, therefore

   (i). $\langle \mathsf{D}_1 + \mathsf{D}_2 \rangle X = \langle \mathsf{D}_1 \rangle X \cup \langle \mathsf{D}_2 \rangle X$. By inductive hypothesis, both $\langle \mathsf{D}_1 \rangle X, \langle \mathsf{D}_2 \rangle X$ are finite, as they are sub expressions of D. Since the union of finite sets is finite, $\langle \mathsf{D}_1 + \mathsf{D}_2 \rangle X$ is finite;

   (ii). by inductive hypothesis again $\forall \rho \in X \ \langle \mathsf{D}_1, \rho \rangle \downdownarrows$ and $\langle \mathsf{D}_2, \rho \rangle \downdownarrows$. By sum_1 rule $\langle \mathsf{D}_1 + \mathsf{D}_2, \rho \rangle \to \langle \mathsf{D}_1, \rho \rangle$ and by sum_2 $\langle \mathsf{D}_1 + \mathsf{D}_2, \rho \rangle \to \langle \mathsf{D}_2, \rho \rangle$. Therefore $\langle \mathsf{D}_1 + \mathsf{D}_2 \rangle \rho \downdownarrows$.

   (iii). For the latter argument, since both $\langle \mathsf{D}_1 \rangle \rho$ and $\langle \mathsf{D}_2 \rangle \rho$ are finite and composed of finite derivation sequences $|\mathsf{TS}(\langle \mathsf{D}_1 + \mathsf{D}_2, \rho \rangle)| < \infty$.

2. $\mathsf{D} \equiv \mathsf{D}_1; \mathsf{D}_2$, therefore

   (i). $\langle \mathsf{D}_1; \mathsf{D}_2 \rangle X = \langle \mathsf{D}_2 \rangle (\langle \mathsf{D}_1 \rangle X)$. By inductive hypothesis $\langle \mathsf{D}_1 \rangle X = Y$. By inductive hypothesis again $\langle \mathsf{D}_2 \rangle Y$ is finite;

   (ii). by inductive hypothesis both $\forall \rho \in X \ \langle \mathsf{D}_1, \rho \rangle \downdownarrows$ and $\forall \rho' \in Y \ \langle \mathsf{D}_2, \rho' \rangle \downdownarrows$, therefore by composition lemma $\langle \mathsf{D}_1; \mathsf{D}_2, \rho \rangle \downdownarrows$

   (iii). by inductive hypothesis both $|\mathsf{TS}(\langle \mathsf{D}_1, \rho \rangle)| < \infty$ and $|\mathsf{TS}(\langle \mathsf{D}_2, \rho' \rangle)| < \infty \ \forall \rho \in X, \rho' \in \langle \mathsf{D}_1 \rangle X$. For all derivation sequences starting from $\langle \mathsf{D}_1, \rho \rangle$ where

   $$\langle \mathsf{D}_1, \rho \rangle \to^* \rho'$$

   with $\rho' \in \langle \mathsf{D}_1 \rangle X$ we can apply the composition lemma and state that

   $$\langle \mathsf{D}_1; \mathsf{D}_2, \rho \rangle \to^* \langle \mathsf{D}_2, \rho' \rangle \quad \forall \rho \in X$$

   from there we can notice that since $|\langle \mathsf{D}_2, \rho' \rangle| < \infty$ then $|\langle \mathsf{D}_1; \mathsf{D}_2, \rho' \rangle| < \infty$

□

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

Lemma 2.10. *Let* $\mathsf{D} \in Imp_s$ *and* $\rho \in \mathsf{Env}$*. If*

$$\langle \mathsf{D} \rangle^{k+1} \rho \subseteq \bigcup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho \quad \textit{for some } k \in \mathbb{N} \tag{2.2}$$

then

$$\forall j \in \mathbb{N} \quad \langle \mathsf{D} \rangle^{k+1+j} \rho \subseteq \bigcup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho \tag{2.3}$$

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

Proof. We can show (2.3) by induction on j:

- if $j = 0$ then we want to show that $\langle \mathsf{D} \rangle^{k+1} \rho \subseteq \cup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho$, which is true by hypothesis (2.2);
- In the inductive case we have to show that if the statement holds for j, it also holds for $j+1$. We know that

$$\begin{aligned} \bigcup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho &= \bigcup_{i=0}^{k+1} \langle \mathsf{D} \rangle^{i} \rho && \text{since by (2.2) } \langle \mathsf{D} \rangle^{k+1} \rho \subseteq \cup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho \\ &= \rho \cup \bigcup_{i=1}^{k+1} \langle \mathsf{D} \rangle^{i} \rho \\ &= \rho \cup \langle \mathsf{D} \rangle \left(\bigcup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho \right) && \text{by additivity of } \langle \mathsf{D} \rangle \end{aligned}$$

  By inductive hypothesis

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

  so, by monotonicity of $\langle \mathsf{D} \rangle$

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

  and therefore

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

□

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

Lemma 2.11 (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 2.10 and Lemma 2.11 we can prove the following.

Lemma 2.12. *Given* $\mathsf{D} \in Imp_s$*, and* $\rho \in \mathsf{Env}$*, the predicate "*$\langle \mathsf{D}^* \rangle \rho$ *is finite" is undecidable.*

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

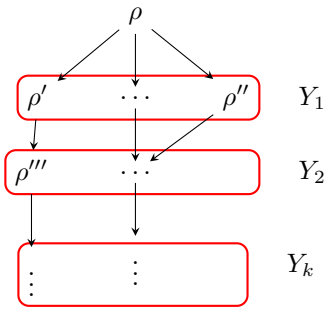


Figure 2.2: Example of $\to^D$ relations between elements of Env.

- Suppose that $\langle \mathsf{D}^* \rangle \rho$ is infinite, then we can observe that because Lemma 2.10

$$\forall k \in \mathbb{N} \quad \langle \mathsf{D} \rangle^{k+1} \rho \not\subseteq \bigcup_{i=0}^{k} \langle \mathsf{D} \rangle^{i} \rho \tag{2.4}$$

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

$$Y_0 = \{\rho\}$$
$$Y_{k+1} = \left(\langle \mathsf{D} \rangle^{k+1} \rho \right) \setminus \left(\bigcup_{i=0}^{k} \langle \mathsf{D} \rangle \rho \right)$$

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 $\to^D$ relations and visualizes the levels Y_k. We can therefore make the following observations:

(i) The tree is rooted in $\rho \in Y_0$. In fact $\forall \rho' \in Y_1\ \rho \to^D \rho'$ by definition and $\forall \rho''' \in Y_{k+1} \exists \rho'' \in Y_k \mid \rho'' \to^D \rho'''$;

(ii) since $\forall k \in \mathbb{N}\ \langle \mathsf{D} \rangle^{k+1} \rho \not\subseteq \cup_{i=0}^{k} \langle \mathsf{D} \rangle^i \rho$ by (2.4), each level Y_k is non empty. Each level is also finite because of Lemma 2.1.(i). Therefore there is an infinite quantity of levels, where each node has a finite quantity of children.

what is left to do is show that there is a derivation sequence from $\langle \mathsf{D}^*, \rho \rangle$ of infinite length.

We can do this by using König's Lemma 2.11 and deduce that there exists an infinite derivation sequence from ρ of $\to^D$ relations

$$\rho \to^D \rho' \to^D \rho'' \to^D \dots$$

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 $\rho', \rho'' \in \mathsf{Env}$ s.t. $\rho' \to^D \rho''$ since $\langle \mathsf{D}, \rho' \rangle \to^* \rho''$ we can apply Corollary 2.2 of Lemma 2.4 and derive that $\langle \mathsf{D}; \mathsf{D}^*, \rho' \rangle \to^* \langle \mathsf{D}^*, \rho'' \rangle$ and because of the star rule $\langle \mathsf{D}^*, \rho' \rangle \to \langle \mathsf{D}; \mathsf{D}^*, \rho' \rangle$. We can therefore say that

$$\langle \mathsf{D}^*, \rho' \rangle \to^* \langle \mathsf{D}^*, \rho'' \rangle$$

Therefore, there exists an infinite derivation sequence

$$\langle \mathsf{D}^*, \rho \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle \to^* \langle \mathsf{D}^*, \rho'' \rangle \to^* \dots$$

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

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

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

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

  $$\langle \mathsf{D}^*, \rho \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle \to^+ \langle \mathsf{D}^*, \rho' \rangle \to^* \rho''$$

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

  $$\langle \mathsf{D}^*, \rho \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle \to^+ \langle \mathsf{D}^*, \rho' \rangle \to^+ \langle \mathsf{D}^*, \rho' \rangle \to^+ \dots$$

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

  ($\Leftarrow$) In this case we want to prove that if $\langle \mathsf{D}^* \rangle \rho$ is finite and there are no cycles in any derivation sequence starting from $\langle \mathsf{D}, \rho \rangle$ then $\langle \mathsf{D}, \rho \rangle \downarrow\downarrow$. We work again by contradiction. Suppose that we have an infinite derivation sequence starting from $\langle \mathsf{D}^*, \rho \rangle$. 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

  $$\langle \mathsf{D}^*, \rho \rangle \to^* \langle \mathsf{D}^*, \rho_1 \rangle \to^* \langle \mathsf{D}^*, \rho_2 \rangle \to^* \langle \mathsf{D}^*, \rho_3 \rangle \to^* \dots$$

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

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

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

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

    $$\langle \mathsf{D}^*, \rho' \rangle \to \rho'$$

    and therefore in a finite number of reached states;
  - or we apply the star rule

    $$\langle \mathsf{D}^*, \rho' \rangle \to \langle \mathsf{D}; \mathsf{D}^*, \rho' \rangle$$

    by lemma 2.9 we know that $\langle \mathsf{D}, \rho' \rangle \downarrow\downarrow$ and $|\mathsf{TS}(\langle \mathsf{D}, \rho' \rangle)| < \infty$, therefore there is a finite number of environments ρ'' s.t. $\langle \mathsf{D}, \rho' \rangle \to^* \rho''$. For each one of them we can use the composition lemma and observe that

    $$\langle \mathsf{D}; \mathsf{D}^*, \rho' \rangle \to^* \langle \mathsf{D}^*, \rho'' \rangle$$

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

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

□

Chapter 3

Abstract domains

In the following chapters we present two domains that we will discuss: 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. This chapter's sections are organized as follows:

Expand

- Section 3.2 will talk about the interval domain, with its characterization in §3.2.1 and the domain properties in Section 3.2.3.
- Section 3.3 will talk about the non-relational collecting abstraction, first by introducing the definition of the non-relational collecting domain in Section 3.3.1 and finally by showing some properties of the abstraction in Section 3.3.3

3.1 Abstract semantics

In order to talk about analysis over some abstract domain $\mathbb{A}$, we preliminarly introduce the abstract semantic.

Definition 3.1. Given an abstract domain $\mathbb{A}$, with an abstraction map $\alpha : 2^{\mathbb{Z}} \to \mathbb{A}$ and a concretization map $\gamma : \mathbb{A} \to 2^{\mathbb{Z}}$ the *analysis semantics* over $\mathbb{A}$ is defined as the strict (i.e., preserving $\bot$) extension of the following function $[\![\cdot]\!]^{\mathbb{A}} : \mathrm{Imp} \to \mathbb{A} \to \mathbb{A}$. For all $\eta \in \mathbb{A}$

$$
\begin{aligned}
[\![\mathtt{x} \in S]\!]^{\mathbb{A}}\eta &\triangleq \begin{cases} \eta[\mathtt{x} \mapsto \alpha(\gamma(\eta(\mathtt{x})) \cap \gamma_{\mathbb{I}}(S))] & \text{if } \gamma(\eta(\mathtt{x})) \cap \gamma_{\mathbb{I}}(S) \neq \varnothing \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathtt{x} := k]\!]^{\mathbb{A}}\eta &\triangleq \eta[\mathtt{x} \mapsto \alpha(\{k\})] \\
[\![\mathtt{x} := \mathtt{y} + k]\!]^{\mathbb{A}}\eta &\triangleq \eta[\mathtt{x} \mapsto \eta(\mathtt{y}) + k] \\
[\![\mathtt{x} := \mathtt{y} - k]\!]^{\mathbb{A}}\eta &\triangleq \eta[\mathtt{x} \mapsto \eta(\mathtt{y}) - k] \\
[\![\mathsf{C}_1 + \mathsf{C}_2]\!]^{\mathbb{A}}\eta &\triangleq [\![\mathsf{C}_1]\!]^{\mathbb{A}}\eta \sqcup [\![\mathsf{C}_2]\!]^{\mathbb{A}}\eta \\
[\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\mathbb{A}}\eta &\triangleq [\![\mathsf{C}_2]\!]^{\mathbb{A}}([\![\mathsf{C}_1]\!]^{\mathbb{A}}\eta) \\
[\![\mathsf{C}^*]\!]^{\mathbb{A}}\eta &\triangleq \bigsqcup_{i \in \mathbb{N}} \left([\![\mathsf{C}]\!]^{\mathbb{A}}\right)^i(\eta) \\
[\![\mathsf{fix}(\mathsf{C})]\!]^{\mathbb{A}}\eta &\triangleq \mathrm{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\mathbb{A}}\mu))
\end{aligned}
$$

where $\gamma_{\mathbb{I}}$ is the interval concretization map as defined in Definition 3.3 and $S \in \mathbb{I}$, defined in Definition 3.2.

3.2 Interval domain

Interval analysis are among the most well known abstract interpretation standard abstract domains. 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 to prove the fact that interval analysis is decidable without a widening operator, i.e., infinite ascending chains can be decided.

3.2.1 Definition

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

Definition 3.2 (Integer intervals)**.** We call

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

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 $\mathbb{C}$ we need to define concretization and abstraction maps for the intervals domain

Definition 3.3. We define the *concretization map* $\gamma : \mathbb{I} \to 2^{\mathbb{Z}}$ as

$$\begin{aligned} \gamma([a,b]) &\triangleq \{x \in \mathbb{Z} \mid a \leqslant x \leqslant b\} \\ \gamma(\bot) &\triangleq \varnothing \end{aligned}$$

And the *abstraction map* $\alpha : 2^{\mathbb{Z}} \to \mathbb{I}$ as

$$\begin{aligned} \alpha(\varnothing) &\triangleq \bot \\ \alpha(X) &\triangleq \begin{cases} \bot & \text{if } X = \varnothing \\ [\min(X), \max(X)] & \text{otherwise} \end{cases} \end{aligned}$$

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 3.4 ($\mathbb{I}$ partial order)**.** Let $I, J \in \mathbb{I}$. Then

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

We also define least upper bound and greatest lower bound on the $\mathbb{I}$ domain. Let $[a,b], [c,d] \in \mathbb{I}$

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

Observe that because of $\sqsubseteq$ and $\sqcup$ definitions $\langle \mathbb{I}, \sqsubseteq \rangle$ is a complete lattice. The next building block is the definition of some more operations on intervals, namely the addition and subtraction of an integer constant:

Definition 3.5 (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 $\pm\infty + c = \pm\infty - c = \pm\infty$).

3.2.2 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 3.6 (Abstract integer domain)**.** Let $\mathbb{I}_* \triangleq \mathbb{I} \setminus \{\bot\}$. The abstract domain $\dot{\mathbb{I}}$ for program analysis is the variable-wise lifting of $\mathbb{I}$:

$$\dot{\mathbb{I}} \triangleq (\mathit{Var} \to \mathbb{I}_*) \cup \{\bot\}$$

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 3.7. We define the *concretization map* of abstract environments $\eta \in \dot{\mathbb{I}}$, i.e., $\gamma : \dot{\mathbb{I}} \to 2^{\mathsf{Env}}$ as follows

$$\begin{aligned}\dot{\gamma}(\bot) &\triangleq \varnothing \\ \dot{\gamma}(\eta) &\triangleq \{\rho \in \mathsf{Env} \mid \forall \mathtt{x} \in \mathit{Var} \quad \rho(\mathtt{x}) \in \gamma(\eta\mathtt{x})\}\end{aligned}$$

and the *abstraction map* of sets of concrete environments $X \in 2^{\mathsf{Env}}$, i.e., $\alpha : 2^{\mathsf{Env}} \to \dot{\mathbb{I}}$ as

$$\begin{aligned}\dot{\alpha}(\varnothing) &\triangleq \bot \\ \dot{\alpha}(X) &\triangleq \lambda \mathtt{x} \,.\, \alpha(\{\rho(\mathtt{x}) \mid \rho \in X\})\end{aligned}$$

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

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

Notice that because of the definition of the concretization map (Definition 3.7)

$$\eta \sqsubseteq \vartheta \iff \forall \mathtt{x} \in \mathit{Var} \quad \eta(\mathtt{x}) \sqsubseteq \vartheta(\mathtt{x})$$

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

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

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 $\eta \sqcap \vartheta$.

3.2.3 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 3.1. 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    if x = 2
3      then x := x+6;
4    endif;
5    x := x-3;
6    if x <= 0
7      then x:=0;
8    endif;
9  done;
```

Code 3.1: fix(C) and C^* difference

starting with the finite interval $[3,4]$ we get the following loop invariants:

$$\text{Kleene: } \sqcup\{[3,4],[0,1],[0,0],[0,0],\ldots\} = [0,4]$$
$$\text{Fix: } \sqcup\{\bot,[3,4],[0,4],[0,5],[0,5],\ldots\} = [0,5]$$

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.

Lemma 3.1 ($\mathsf{fix}(\mathsf{C})$ **is syntactic sugar**). *For all* η, $[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\eta = [\![(\mathsf{true}+\mathsf{C})^*]\!]^{\dot{\mathbb{I}}}\eta$.

Proof. Let us first show by induction that

$$\forall i \geqslant 0.\ (\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot = (\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i}\eta \qquad (\sharp)$$

$i=0$: $(\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^1\bot = \eta \sqcup \bot \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\bot = \eta = (\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^0\eta$.
$i+1$:

$$\begin{aligned}
(\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\eta &= \\
(\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})((\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i}\eta) &= \\
((\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i}\eta) \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}((\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i}\eta) &= && \text{By induction} \\
(\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}((\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot) &= && \text{As } \eta \sqsubseteq (\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot \\
\eta \sqcup (\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}((\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot) &= \\
(\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})((\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot) &= \\
(\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+2}\bot &
\end{aligned}$$

Let us also show that:

$$\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) = \mathrm{lfp}\lambda\mu.(\eta \sqcup \mu \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) \qquad (\diamond)$$

Observe that $\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) = \eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}(\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu))$, so that we have that:

$$\eta \sqcup \mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}(\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu)) \sqsubseteq \mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu)$$

As a consequence, $\mathrm{lfp}\lambda\mu.(\eta \sqcup \mu \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) \sqsubseteq \mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu)$ holds. The reverse inequality follows because, for all μ, $\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu \sqsubseteq \eta \sqcup \mu \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu$.

Then, we have that:

$$\begin{aligned}
[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\eta &= \\
\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) &= && \text{By } (\diamond) \\
\mathrm{lfp}\lambda\mu.(\eta \sqcup \mu \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\mu) &= && \text{By Knaster-Tarski Theorem} \\
\bigsqcup_{i\in\mathbb{N}} (\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i}\bot &= \\
\bot \sqcup \bigsqcup_{i\in\mathbb{N}} (\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i+1}\bot &= && \text{By (4.3)} \\
\bigsqcup_{i\in\mathbb{N}} (\mathsf{true} \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}})^{i}\eta &= \\
[\![(\mathsf{true}+\mathsf{C})^*]\!]^{\dot{\mathbb{I}}}\eta. &
\end{aligned}$$

□

Remark 3.1. Let us remark that in case we were interested in studying termination of the abstract interpreter, we could assume that the input of a program will always be a finite interval in such a way that non termination can be identified with the impossibility of converging to a finite interval for some variable. In fact, starting from an environment η which maps each variable to a finite interval, $[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\eta$ might be infinite on some variable when C includes a either Kleene or fix iteration which does not converge in finitely many steps.

3.3 Non relational collecting

intro da fare

3.3.1 Definition

We first define *non-relational collecting* analysis the the Imp language in a standard way, taking again the best correct approximation (bca) for the basic expressions in Exp. Unlike the Intervals domain, where we needed to define the set of intervals, the non-relational collecting analysis will rely on $2^{\mathbb{Z}}$ for the abstract values of each variable in the variable-wise lifting. Notice that since we are already in $2^{\mathbb{Z}}$ we do not need to abstract and concretize the values of our lattice. As we already observed, $\langle 2^{\mathbb{Z}}, \subseteq \rangle$ is a complete lattice, where the notions of $\cap$ and $\cup$ are well-known. The next building block is the definition of some more operations on intervals, namely the addition and subtraction of an integer constant:

Definition 3.8 (Set addition and subtraction)**.** For a nonempty set $S \in 2^{\mathbb{Z}}$ and $c \in \mathbb{N}$ define $S \pm c \triangleq \{x \pm c \mid x \in S\}$ (recall that $\pm\infty + c = \pm\infty - c = \pm\infty$).

3.3.2 Variable-wise lifting

We can therefore proceed to introduce the variable-wise lifting of the $2^{\mathbb{Z}}$ domain, building the abstract domain $\mathbb{C}^{\mathsf{c}}$:

Definition 3.9 (Abstract Non relational collecting domain)**.** Let $2^{\mathbb{Z}}_{*} = 2^{\mathbb{Z}} \setminus \{\varnothing\}$. The abstract domain $\mathbb{C}^{\mathsf{c}}$ for program analysis is the variable-wise lifting of $2^{\mathbb{Z}}$:

$$\mathbb{C}^{\mathsf{c}} \triangleq (\mathit{Var} \to 2^{\mathbb{Z}}_{*}) \cup \{\bot\}$$

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

Definition 3.10. We define the *concretization map* of abstract environments $\eta \in \mathbb{C}^{\mathsf{c}}$, i.e., $\gamma : \mathbb{C}^{\mathsf{c}} \to 2^{\mathsf{Env}}$ as follows

$$\begin{aligned} \gamma(\bot) &\triangleq \varnothing \\ \gamma(\eta) &\triangleq \{\rho \in \mathsf{Env} \mid \forall \mathtt{x} \in \mathit{Var} \quad \rho(\mathtt{x}) \in \eta\mathtt{x}\} \end{aligned}$$

and the *abstraction map* of sets of concrete environments $X \in 2^{\mathsf{Env}}$, i.e., $\alpha : 2^{\mathsf{Env}} \to \mathbb{C}^{\mathsf{c}}$ as

$$\begin{aligned} \alpha(\varnothing) &\triangleq \bot \\ \alpha(X) &\triangleq \lambda \mathtt{x} \,.\, \{\rho(\mathtt{x}) \mid \rho \in X\} \end{aligned}$$

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

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

Notice that because of the definition of the concretization map (Definition 3.10)

$$\eta \sqsubseteq \vartheta \iff \forall \mathtt{x} \in \mathit{Var}\ \eta(\mathtt{x}) \sqsubseteq \vartheta(\mathtt{x})$$

Again, we can define least upper bounds and greatest lower bounds by lifting the $\sqcup$ and $\sqcap$ operations. Let again $\eta, \vartheta \in \mathbb{C}^{\mathsf{c}}$, then

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

Again we can notice that $\langle \mathbb{C}^{\mathsf{c}}, \sqsubseteq \rangle$ is a complete lattice, as for every two elements $\eta, \vartheta \in \mathbb{C}^{\mathsf{c}}$ there exists both $\eta \sqcup \vartheta$ and $\eta \sqcap \vartheta$.

3.3.3 Properties

We can notice that the semantics we just defined has some properties similar to the interval semantics but also the concrete collecting semantics from Definition 2.2. The main property is additivity, which we lost with the interval semantics and got back with the non-relational collecting.

Let's denote as $\langle\!\langle \cdot \rangle\!\rangle$ the abstract semantics over $\mathbb{C}^{\mathsf{c}}$, i.e., $\langle\!\langle \cdot \rangle\!\rangle = [\![\cdot]\!]^{\mathbb{C}^{\mathsf{c}}}$.

Lemma 3.2 (Additivity)**.** *Let $\eta, \vartheta \in \mathbb{C}^{\mathsf{c}}$, $\mathsf{C} \in$ Imp then*

$$\langle\!\langle \mathsf{C} \rangle\!\rangle (\eta \sqcup \vartheta) = (\langle\!\langle \mathsf{C} \rangle\!\rangle \eta) \sqcup (\langle\!\langle \mathsf{C} \rangle\!\rangle \vartheta)$$

Proof. We can work by induction on C:

Case ($\mathtt{x} \in S$). Then

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

Case ($\mathtt{x} := k$). Then

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

Case ($\mathtt{x} := \mathtt{y} + k$). Then

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

Case ($\mathsf{C} \equiv \mathsf{C}_1 + \mathsf{C}_2$). Then

$$\begin{aligned}
\langle\!\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle\!\rangle (\eta \sqcup \sigma) &= \langle\!\langle \mathsf{C}_1 \rangle\!\rangle (\eta \sqcup \sigma) \sqcup \langle\!\langle \mathsf{C}_2 \rangle\!\rangle (\eta \sqcup \sigma) && \text{by definition} \\
&= \langle\!\langle \mathsf{C}_1 \rangle\!\rangle \eta \sqcup \langle\!\langle \cup_1 \rangle\!\rangle \vartheta \sqcup \langle\!\langle \mathsf{C}_2 \rangle\!\rangle \eta \sqcup \langle\!\langle \cup_2 \rangle\!\rangle \vartheta && \text{by inductive hypothesis} \\
&= \langle\!\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle\!\rangle \eta \sqcup \langle\!\langle \mathsf{C}_1 + \mathsf{C}_2 \rangle\!\rangle \vartheta
\end{aligned}$$

Case ($\mathsf{C} \equiv \mathsf{C}_1; \mathsf{C}_2$). Then

$$\begin{aligned}
\langle\!\langle \mathsf{C}_1; \mathsf{C}_2 \rangle\!\rangle (\eta \sqcup \sigma) &= \langle\!\langle \mathsf{C}_2 \rangle\!\rangle (\langle\!\langle \mathsf{C}_1 \rangle\!\rangle (\eta \sqcup \vartheta)) \\
&= \langle\!\langle \mathsf{C}_2 \rangle\!\rangle (\langle\!\langle \mathsf{C}_1 \rangle\!\rangle \eta \sqcup \langle\!\langle \cup_1 \rangle\!\rangle \vartheta) && \text{by inductive hypothesis} \\
&= \langle\!\langle \mathsf{C}_2 \rangle\!\rangle (\langle\!\langle \mathsf{C}_1 \rangle\!\rangle \eta) \sqcup \langle\!\langle \mathsf{C}_2 \rangle\!\rangle (\langle\!\langle \mathsf{C}_1 \rangle\!\rangle \vartheta) && \text{by inductive hypothesis}
\end{aligned}$$

Case ($\mathsf{C} \equiv \mathsf{C}^*$). Then

$$\langle\!\langle \mathsf{C}^* \rangle\!\rangle (\eta \sqcup \vartheta) = \bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^i (\eta \sqcup \vartheta)$$

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

- $i = 0$. Then
$$\langle\!\langle \mathsf{C} \rangle\!\rangle^0 (\eta \sqcup \vartheta) = \eta \sqcup \vartheta = \langle\!\langle \mathsf{C} \rangle\!\rangle^0 \eta \sqcup \langle\!\langle \mathsf{C} \rangle\!\rangle^0 \vartheta$$
and the statement holds.

- $i \Rightarrow i+1$. Notice that

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

Therefore

$$\begin{aligned}
\langle\!\langle \mathsf{C}^* \rangle\!\rangle(\eta \sqcup \vartheta) &= \bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}(\eta \sqcup \vartheta) \\
&= \bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}(\eta \sqcup \vartheta) \\
&= \bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}\eta \sqcup \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}\vartheta \\
&= \left(\bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}\eta \right) \sqcup \left(\bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}\vartheta \right) \\
&= \langle\!\langle \mathsf{C}^* \rangle\!\rangle\eta \sqcup \langle\!\langle \mathsf{C}^* \rangle\!\rangle\vartheta
\end{aligned}$$

□

Again, because of additivity we can notice that the analysis of the fixpoint and the Kleene star is the same. Let $f = \lambda\mu.(\eta \sqcup \langle\!\langle \mathsf{C} \rangle\!\rangle\mu)$

$$\begin{aligned}
\langle\!\langle \mathsf{fix}(\mathsf{C}) \rangle\!\rangle\eta &= \mathrm{lfp}(f) \\
&= \bigsqcup_{i \in \mathbb{N}} \{f^n \bot \mid n \in \mathbb{N}\} && \text{by fixpoint theorem 1.1} \\
&= \eta \sqcup (\eta \sqcup \langle\!\langle \mathsf{C} \rangle\!\rangle\eta) \sqcup \ldots && \text{by definition} \\
&= \bigsqcup_{i \in \mathbb{N}} \langle\!\langle \mathsf{C} \rangle\!\rangle^{i}\eta \\
&= \langle\!\langle \mathsf{C}^* \rangle\!\rangle\eta
\end{aligned}$$

therefore we will omit one of the two cases based on preference and ease of reading, but in reality they are the same

Chapter 4

Program bounds and analysis termination

In this chapter we argue that for the language Imp the abstract semantics is computable in finite time without widening for abstract domains with some properties. Observe that the exact computation 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 intervals abstract domain, 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 $\mathsf{fix}(\mathsf{C})$ 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. First let's formalize the problem we want to solve.

Problem 4.1 (Analysis termination)**.** Given a program $\mathsf{C} \in$ Imp and an abstract domain $\mathbb{A}$ with $\eta \in \mathbb{A}$, decide

$$[\![\mathsf{C}]\!]\eta =^{?} \top$$

To do so we present a novel technique, based on the idea of *bounds*. 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 within its bound, the behavior of the program with respect to that variable becomes stable.

4.1 Program bounds

Definition 4.1 (**Program bound**). The *upper bound* associated with a command $\mathsf{C} \in \text{Imp}$ is an integer number, denoted $(\mathsf{C})^{\mathsf{b}} \in \mathbb{N}$, defined inductively as follows:

$$
\begin{aligned}
(\mathtt{x} \in S)^{\mathsf{b}} &\triangleq \begin{cases} |\min(S)| & \text{if } \max(S) = +\infty \\ |\max(S)| & \text{if } \max(S) \in \mathbb{Z} \end{cases} \\
(\mathtt{x} := k)^{\mathsf{b}} &\triangleq |k| \\
(\mathtt{x} := \mathtt{y} + k)^{\mathsf{b}} &\triangleq |k| \\
(\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}} &\triangleq (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} \\
(\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}} &\triangleq (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} \\
(\mathsf{fix}(\mathsf{C}))^{\mathsf{b}} &\triangleq (|\mathit{vars}(\mathsf{C})| + 1)(\mathsf{C})^{\mathsf{b}}
\end{aligned}
$$

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

$$
\begin{aligned}
(\mathtt{x} \in S)_{\mathsf{b}} &\triangleq \begin{cases} |\max(S)| & \text{if } \min(S) = -\infty \\ |\min(S)| & \text{if } \min(S) \in \mathbb{Z} \end{cases} \\
(\mathtt{x} := k)_{\mathsf{b}} &\triangleq |k| \\
(\mathtt{x} := \mathtt{y} + k)_{\mathsf{b}} &\triangleq |k| \\
(\mathsf{C}_1 + \mathsf{C}_2)_{\mathsf{b}} &\triangleq (\mathsf{C}_1)_{\mathsf{b}} + (\mathsf{C}_2)_{\mathsf{b}} \\
(\mathsf{C}_1; \mathsf{C}_2)_{\mathsf{b}} &\triangleq (\mathsf{C}_1)_{\mathsf{b}} + (\mathsf{C}_2)_{\mathsf{b}} \\
(\mathsf{fix}(\mathsf{C}))_{\mathsf{b}} &\triangleq (|\mathit{vars}(\mathsf{C})| + 1)(\mathsf{C})_{\mathsf{b}}
\end{aligned}
$$

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

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

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 $\langle X, \to \rangle$ where $\to \subseteq X \times \mathbb{Z} \times X$ and $x \to_h x'$ denotes that $(x, h, x') \in \to$. Consider a finite path in $\langle X, \to \rangle$

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

where:

(i). $\ell \geqslant 1$

(ii). the carrier size of p is $s(p) \triangleq |\{x_0, ..., x_\ell\}|$

(iii). the weight of p is $w(p) \triangleq \Sigma_{k=0}^{\ell-1} h_k$

(iv). the length of p is $|p| \triangleq \ell$

(v). given indices $0 \leqslant i < j \leqslant \ell$, $p_{i,j}$ denotes the subpath of p given by $x_i \to_{h_i} x_{i+1} \to_{i+1} \cdots \to_{h_{j-1}} x_j$ whose length is $j - i$; $p_{i,j}$ is a cycle if $x_i = x_j$.

Lemma 4.1 (**Positive cycles in weighted directed graphs**). *Let p be a finite path*

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

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

Proof. First note that $w(p) = \Sigma_{k=0}^{\ell-1} h_k > (|X| - 1)m$ implies that $|p| = \ell \geqslant |X|$. Then, we show our claim by induction on $|p| = \ell \geqslant |X|$.
($|p| = |X|$): Since the path p includes exactly $|X|+1 = \ell+1$ nodes, there exist indices $0 \leqslant i < j \leqslant \ell$ 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

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

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 $0 \leqslant i < j \leqslant \ell$, which is maximal, i.e., such that if $p_{i',j'}$ is a cycle in p, for some $0 \leqslant i' < j' \leqslant \ell$, 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 $w(p_{i,j}) \leqslant 0$ and we consider the path p' obtained from p by stripping off the cycle $p_{i,j}$, i.e.,

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

Since $|p'| < |p|$ and $w(p') = w(p) - w(p_{i,j}) \geqslant w(p) > (|X| - 1)m$, 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. □

We also preliminarily need to define the max function for intervals. More in detail $\max : \mathbb{I} \to \mathbb{Z}$ is defined as follows

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

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

Lemma 4.2. *Let $\mathsf{C} \in$ Imp. For all $\eta \in \dot{\mathbb{I}}$ and $\mathtt{y} \in$ Var, if $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) \neq +\infty$ and $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) > (\mathsf{C})^{\mathsf{b}}$ then there exist a variable $\mathtt{z} \in$ Var and an integer $h \in \mathbb{Z}$ such that $|h| \leqslant (\mathsf{C})^{\mathsf{b}}$ and the following two properties hold:*

(i) $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) = \max(\eta\mathtt{z}) + h$;

(ii) for all $\eta' \in \dot{\mathbb{I}}$, if $\eta' \sqsupseteq \eta$ then $\max([\![\mathsf{C}]\!]\eta'\mathtt{y}) \geqslant \max(\eta'\mathtt{z}) + h$.

Proof. Preliminarily let's state that we will refer to $[\![\cdot]\!]^{\dot{\mathbb{I}}}$ as $[\![\cdot]\!]$. This is done in order to have a less crowded notation. The proof is by structural induction on the command $\mathsf{C} \in$ Imp. We preliminarily observe that we can safely assume $\eta \neq \bot$. In fact, if $\eta = \bot$ then $[\![\mathsf{C}]\!]\bot = \bot$ and thus $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) = -\infty \leqslant (\mathsf{C})^{\mathsf{b}}$, against the hypothesis $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) > (\mathsf{C})^{\mathsf{b}}$. Moreover, when quantifying over η' such that $\eta' \sqsupseteq \eta$ in (ii), if $\max([\![\mathsf{C}]\!]\eta'\mathtt{y}) = +\infty$ holds, then $\max([\![\mathsf{C}]\!]\eta'\mathtt{y}) \geqslant \max(\eta'\mathtt{z}) + h$ trivially holds, hence we will sometimes silently omit to consider this case.

Case $(\mathtt{x} \in S)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $+\infty \neq \max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) > (\mathtt{x} \in S)^{\mathsf{b}}$. Clearly $[\![\mathtt{x} \in S]\!]\eta \neq \bot$, otherwise we would get the contradiction $\max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) = -\infty \leqslant (\mathtt{x} \in S)^{\mathsf{b}}$.

We distinguish two cases:

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

$$\bot \neq [\![\mathtt{x} \in S]\!]\eta' = \eta'[\mathtt{x} \mapsto \alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta'(\mathtt{x})) \cap \gamma_{\mathbb{I}}(S))]$$

  and thus

$$\max([\![\mathtt{x} \in S]\!]\eta'\mathtt{y}) = \max(\eta'\mathtt{y}) = \max(\eta'\mathtt{y}) + 0$$

  hence the thesis follows with $\mathtt{z} = \mathtt{y}$ and $h = 0$.

- If $\mathtt{y} = \mathtt{x}$ then

$$\max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) = \max(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta\mathtt{x}) \cap \gamma_{\mathbb{I}}(S)))$$

  Note that it cannot be $\max(S) \in \mathbb{Z}$. Otherwise, by Definition **??**, $\max(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta\mathtt{x}) \cap \gamma_{\mathbb{I}}(S))) \leqslant \max(S) = (\mathtt{x} \in S)^{\mathsf{b}}$, violating the assumption $\max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) > (\mathtt{x} \in S)^{\mathsf{b}}$. Hence, $\max(S) = +\infty$ must hold and therefore $\max(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta\mathtt{x}) \cap \gamma_{\mathbb{I}}(S))) = \max(\eta(\mathtt{x})) = \max(\eta(\mathtt{x}))+ 0$. It is immediate to check that the same holds for all $\eta' \sqsupseteq \eta$, i.e.,

$$\max(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta'\mathtt{x}) \cap \gamma_{\mathbb{I}}(S))) = \max(\eta'\mathtt{x}) + 0$$

  and thus the thesis follows with $\mathtt{z} = \mathtt{y} = \mathtt{x}$ and $h = 0$.

Case $(\mathtt{x} := k)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max([\![\mathtt{x} := k]\!]\eta\mathtt{y}) > (\mathtt{x} := k)^{\mathsf{b}} = |k|$.

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

$$\max([\![\mathtt{x} := k]\!]\eta\mathtt{y}) = k \leqslant (\mathtt{x} := k)^{\mathsf{b}}$$

violating the assumption. Therefore, it must be $\mathtt{y} \neq \mathtt{x}$. Now, for all $\eta' \sqsupseteq \eta$, we have $[\![\mathtt{x} := k]\!]\eta'\mathtt{y} = \eta'\mathtt{y}$ and thus

$$\max([\![\mathtt{x} := k]\!]\eta'\mathtt{y}) = \max(\eta'\mathtt{y}) = \max(\eta'\mathtt{y}) + 0,$$

hence the thesis holds with $h = 0 \leqslant (\mathtt{x} := k)^{\mathsf{b}}$ and $\mathtt{z} = \mathtt{y}$.

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

We distinguish two cases:

- If $\mathtt{y} \neq \mathtt{x}$, then for all $\eta' \sqsupseteq \eta$, we have $[\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y} = \eta'\mathtt{y}$ and thus

$$\max([\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y}) = \max(\eta'\mathtt{y})$$

  hence the thesis follows with $h = 0 \leqslant (\mathtt{x} := \mathtt{w} + k)^{\mathsf{b}}$ and $\mathtt{z} = \mathtt{y}$.

- If $\mathtt{x} = \mathtt{y}$ then for all $\eta' \sqsupseteq \eta$, we have $[\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y} = \eta'\mathtt{w} + k$ and thus

$$\max([\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y}) = \max(\eta'\mathtt{w}) + k$$

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

Case $(\mathsf{C}_1 + \mathsf{C}_2)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta) > (\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}} = (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}}$. Recall that $[\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta = [\![\mathsf{C}_1]\!]\eta \sqcup [\![\mathsf{C}_2]\!]\eta$. Hence, since $\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) \neq +\infty$, we have that $\max([\![\mathsf{C}_1]\!]\eta\mathtt{y}) \neq \infty \neq \max([\![\mathsf{C}_2]\!]\eta\mathtt{y})$. Moreover

$$\begin{aligned}\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) &= \max([\![\mathsf{C}_1]\!]\eta\mathtt{y} \sqcup [\![\mathsf{C}_2]\!]\eta\mathtt{y}) \\ &= \max\{\max([\![\mathsf{C}_1]\!]\eta\mathtt{y}), \max([\![\mathsf{C}_2]\!]\eta\mathtt{y})\}\end{aligned}$$

Thus $\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \max([\![\mathsf{C}_i]\!]\eta\mathtt{y})$ for some $i \in \{1, 2\}$. We can assume, without loss of generality, that the maximum is realized by the first component, i.e., $\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \max([\![\mathsf{C}_1]\!]\eta\mathtt{y}) > (\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}}$. Hence we can use the inductive hypothesis on C_1 and state that there exists $h \in \mathbb{Z}$ with $|h| \leqslant (\mathsf{C}_1)^{\mathsf{b}}$ and $\mathtt{z} \in \mathit{Var}$ such that $\max([\![\mathsf{C}_1]\!]\eta\mathtt{y}) = \max(\eta\mathtt{z}) + h$ and for all $\eta' \in \dot{\mathbb{I}}$, $\eta \sqsubseteq \eta'$,

$$\max([\![\mathsf{C}_1]\!]\eta'\mathtt{y}) \geqslant \max(\eta'\mathtt{z}) + h$$

Therefore

$$\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \max([\![\mathsf{C}_1]\!]\eta\mathtt{y}) = \max(\eta\mathtt{z}) + h$$

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

$$\begin{aligned}\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta'\mathtt{y}) &= \max\{\max([\![\mathsf{C}_1]\!]\eta'\mathtt{y}), \max([\![\mathsf{C}_2]\!]\eta'\mathtt{y})\}\\ &\geqslant \max([\![\mathsf{C}_1]\!]\eta'\mathtt{y})\\ &\geqslant \max(\eta'\mathtt{z}) + h\end{aligned}$$

with $|h| \leqslant (\mathsf{C}_1)^{\mathsf{b}} \leqslant (\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}}$, as desired.

Case $(\mathsf{C}_1; \mathsf{C}_2)$. Take $\eta \in \dot{\mathbb{I}}$ and assume $\max([\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta\mathtt{y}) > (\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}} = (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}}$. Recall that $[\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta = [\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta)$. If we define

$$[\![\mathsf{C}_1]\!]\eta = \eta_1$$

since $\max([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) \neq \infty$ and $\max([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) > (\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}} \geqslant (\mathsf{C}_2)^{\mathsf{b}}$, by inductive hypothesis on C_2, there are $|h_2| \leqslant (\mathsf{C}_2)^{\mathsf{b}}$ and $\mathtt{w} \in \mathit{Var}$ such that $\max([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) = \max(\eta_1\mathtt{w}) + h_2$ and for all $\eta_1' \in \dot{\mathbb{I}}$ with $\eta_1 \sqsubseteq \eta_1'$

$$\max([\![\mathsf{C}_2]\!]\eta_1'\mathtt{y}) \geqslant \max(\eta_1'\mathtt{w}) + h_2 \qquad (\dagger)$$

Now observe that $\max([\![\mathsf{C}_1]\!]\eta\mathtt{w}) = \max(\eta_1\mathtt{w}) > (\mathsf{C}_1)^{\mathsf{b}}$. Otherwise, if it were $\max(\eta_1\mathtt{w}) \leqslant (\mathsf{C}_1)^{\mathsf{b}}$ we would have

$$\max([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) = \max(\eta_1\mathtt{w}) + h_2 \leqslant (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} = (\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}},$$

violating the hypotheses. Moreover, $\max([\![\mathsf{C}_1]\!]\eta\mathtt{w}) \neq +\infty$, otherwise we would have $\max([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) = \max(\eta_1\mathtt{w}) + h_2 = +\infty$, contradicting the hypotheses. Therefore we can apply the inductive hypothesis also to C_1 and deduce that there are $|h_1| \leqslant (\mathsf{C}_1)^{\mathsf{b}}$ and $\mathtt{w}' \in \mathit{Var}$ such that $\max([\![\mathsf{C}_1]\!]\eta\mathtt{w}) = \max(\eta\mathtt{w}') + h_1$ and for all $\eta' \in \dot{\mathbb{I}}$ with $\eta \sqsubseteq \eta'$

$$\max([\![\mathsf{C}_1]\!]\eta'\mathtt{w}) \geqslant \max(\eta'\mathtt{w}') + h_1 \qquad (\ddagger)$$

Summing up:

$$\begin{aligned}\max([\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta\mathtt{y}) &= \max([\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta)\mathtt{y})\\ &= \max([\![\mathsf{C}_2]\!]\eta_1\mathtt{y})\\ &= \max(\eta_1\mathtt{w}) + h_2\\ &= \max([\![\mathsf{C}_1]\!]\eta\mathtt{w}) + h_2\\ &= \max(\eta\mathtt{w}') + h_1 + h_2.\end{aligned}$$

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

$$\begin{aligned}&\max([\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta'\mathtt{y}) =\\ &\max([\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta')\mathtt{y}) \geqslant\\ &\max([\![\mathsf{C}_1]\!]\eta'\mathtt{w}) + h_2 \geqslant && \text{by } (\dagger), \text{ since } \eta_1 = [\![\mathsf{C}_1]\!]\eta \sqsubseteq [\![\mathsf{C}_1]\!]\eta' \text{ and monotonicity}\\ &(\max(\eta'\mathtt{w}') + h_1) + h_2 && \text{by } (\ddagger)\end{aligned}$$

Thus, the thesis holds with $h = h_1 + h_2$, as $|h| = |h_1 + h_2| \leqslant |h_1| + |h_2| \leqslant (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} = (\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}}$, as needed.

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

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

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

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

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

$$\llbracket \mathsf{fix}(\mathsf{C}) \rrbracket \eta \mathtt{y} = \eta_i \mathtt{y}.$$

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

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

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

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

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

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

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

$$\max(\llbracket \mathsf{fix}(\mathsf{C}) \rrbracket \eta \mathtt{y}) = \max(\eta \mathtt{z}) + h \tag{i}$$

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

$$\max(\llbracket \mathsf{fix}(\mathsf{C}) \rrbracket \eta' \mathtt{y}) \geqslant \max(\eta' \mathtt{z}) + h \tag{ii}$$

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

$$\sigma \triangleq (\mathtt{y}_0, 0) \rightarrow_{h_0} (\mathtt{y}_1, 1) \rightarrow_{h_1} \ldots \rightarrow_{h_m} (\mathtt{y}_{m+1}, m+1) \tag{4.1}$$

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

$$\pi \triangleq (\mathtt{y}_i, i) \rightarrow_{h_i} (\mathtt{y}_{i+1}, i+1) \rightarrow_{h_{i+1}} \ldots \rightarrow_{h_m} (\mathtt{y}_{m+1}, m+1) \tag{4.2}$$

and we have that:

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

with $|h_j| \leqslant (\mathsf{C})^{\mathsf{b}}$ for $j \in \{i, \ldots, m\}$. Hence we can apply Lemma 4.1 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 positive weight. More precisely, there exist $i \leqslant k_1 < k_2 \leqslant m+1$ such that $\mathtt{y}_{k_1} = \mathtt{y}_{k_2}$ and $h = \Sigma_{j=k_1}^{k_2-1} h_j > 0$. If we denote $\mathtt{w} = \mathtt{y}_{k_1} = \mathtt{y}_{k_2}$, then we have that

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

Thus,

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

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

$$\max\left(\llbracket \mathsf{C} + \mathsf{true} \rrbracket^{k_2-k_1} \eta'\mathtt{w}\right) \geqslant \max(\eta'\mathtt{w}) + h \tag{4.3}$$

Let us show Property (4.3) by induction on $\ell = k_2 - k_1 \geqslant 1$.

Case ($\ell = 1$). Notice that by (b) used to build π in (4.2) it holds that $\forall \eta' \sqsupseteq \eta_{k_1} \sqsupseteq \eta$

$$\max\left(\llbracket \mathsf{C} + \mathsf{true} \rrbracket \eta'\mathtt{w}\right) \geqslant \max(\eta'\mathtt{w}) + h$$

hence the thesis holds.

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

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

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

$$\max\left(\llbracket \mathsf{C} + \mathsf{true} \rrbracket \left(\left(\llbracket \mathsf{C} + \mathsf{true} \rrbracket\right)^{\ell} \eta'\right)\mathtt{w}\right) \geqslant \max\left(\left(\llbracket \mathsf{C} + \mathsf{true} \rrbracket\right)^{\ell} \eta'\mathtt{w}\right) \geqslant \max(\eta'\mathtt{w}) + h$$

which is our thesis for Property (4.3).

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

$$\max(\llbracket \mathsf{C} + \mathsf{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1}\mathtt{w}) \geqslant \max(\eta_{k_1}\mathtt{w}) + rh \tag{4.4}$$

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

$$\begin{aligned}&\max(\llbracket \mathsf{C} + \mathsf{true} \rrbracket^{(r+1)(k_2-k_1)} \eta_{k_1}\mathtt{w}) = \\ &\max(\llbracket \mathsf{C} + \mathsf{true} \rrbracket^{k_2-k_1} (\llbracket \mathsf{C} + \mathsf{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1})\mathtt{w}) \geqslant \quad \text{by (4.3) as } \eta_{k_1} \sqsubseteq \llbracket \mathsf{C} + \mathsf{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1} \\ &\max(\llbracket \mathsf{C} + \mathsf{true} \rrbracket^{r(k_2-k_1)} \eta_{k_1}\mathtt{w}) + h \geqslant \quad \text{by inductive hypothesis} \\ &\max(\eta_{k_1}\mathtt{w}) + rh + h \geqslant \max(\eta_{k_1}\mathtt{w}) + (r+1)h\end{aligned}$$

However, this would contradict the hypothesis $\llbracket \mathsf{fix}(\mathsf{C}) \rrbracket \eta\mathtt{y} \neq \infty$. In fact the inequality (4.4) would imply

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

Now, from (4.1) we deduce that for all $\eta' \sqsupseteq \eta_{k_1}$, for $j \in \{k_1, \ldots, m\}$, if we let $\mu_{k_1} = \eta'$ and $\mu_{j+1} = [\![\mathsf{C} + \mathsf{true}]\!]\mu_j$, by the choice of the subsequence, since $k_1 \geqslant i$, we have that

$$\max(\mu_{j+1}\mathtt{y}_{j+1}) \geqslant \max(\mu_{j+1}\mathtt{y}_j) + h_j$$

and thus

$$[\![\mathsf{C} + \mathsf{true}]\!]^{m-k_1+1}\eta'\mathtt{y} = \mu_{m+1}\mathtt{y}_{m+1} \geqslant \max(\mathtt{y}_{k_1}) + \Sigma_{i=k_1}^{m} h_i = \max(\eta'\mathtt{w}) + \Sigma_{i=k_1}^{m} h_i$$

Since $\eta' = [\![\mathsf{fix}(\mathsf{C})]\!]\eta \sqsupseteq \eta_{k_1}$ we conclude

$$\begin{aligned}\max\left([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}\right) &= \max\left([\![\mathsf{C} + \mathsf{true}]\!]^{m-k_1+1}[\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{w}\right)\\ &= \max\left([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{w}\right) + \Sigma_{i=k_1}{}^{m} h_i\\ &\geqslant +\infty + \Sigma_{i=k_1}^{m} h_i = +\infty\end{aligned}$$

contradicting the assumption.

Therefore, the path σ of (4.1) must exist, and consequently

$$\max([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) = \max(\eta_{m+1}\mathtt{y}) = \max(\eta\mathtt{y}_0) + \Sigma_{i=0}^{m} h_i$$

and $\Sigma_{i=0}^{m} h_i \leqslant (\mathsf{fix}(\mathsf{C}))^{\mathsf{b}} = (n+1)(\mathsf{C})^{\mathsf{b}}$, otherwise we could use the same argument above for inferring the contradiction $\max([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) = +\infty$.

Let us now show (ii). Given $\eta' \sqsupseteq \eta$ from (4.1) we deduce that for all $j \in \{0, \ldots, m\}$, if we let $\mu_0 = \eta'$ and $\mu_{j+1} = [\![\mathsf{C} + \mathsf{true}]\!]\mu_j$, we have that

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

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

$$\max([\![\mathsf{fix}(\mathsf{C})]\!]\eta'\mathtt{y}) \geqslant \max(\mu_{m+1}\mathtt{y}) = \max(\mu_{m+1}\mathtt{y}_{m+1}) \geqslant \max(\eta'\mathtt{y}_0) + \Sigma_{i=0}^{m} h_i$$

as desired.

□

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 4.3. *Let* $\mathsf{C} \in$ *Imp.*
For all $\eta \in \dot{\mathbb{I}}$ *and* $\mathtt{y} \in$ *Var, if* $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) \neq -\infty$ *and* $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) < -(\mathsf{C})_{\mathsf{b}}$ *then there exist a variable* $\mathtt{z} \in$ *Var and an integer* $h \in \mathbb{Z}$ *s.t.* $|h| \leqslant (\mathsf{C})_{\mathsf{b}}$ *s.t. the following two properties hold:*

(i) $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) = \min(\eta\mathtt{z}) + h$*;*

(ii) for all $\eta' \in \dot{\mathbb{I}}$*, if* $\eta' \sqsupseteq \eta$ *then* $\min([\![\mathsf{C}]\!]\eta'\mathtt{y}) \leqslant \min(\eta'\mathtt{z}) + h$*.*

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, $(\mathsf{C})_{\mathsf{b}}$, computed by considering the reverse ordering. □

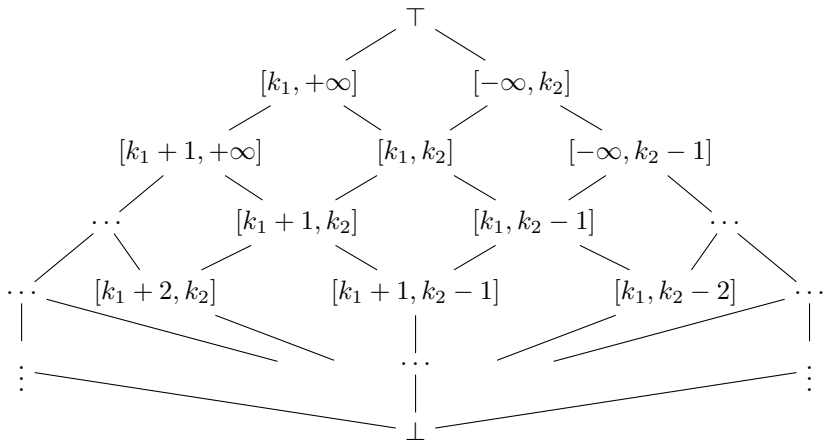


Figure 4.1: $\mathbb{I}_{k_1}^{k_2}$ Hasse diagram

4.3 Computing interval semantics

Lemma 4.2 provides an effective algorithm for computing the abstract semantics of commands. This means that we can apply Lemma 4.2 on the intervals domain $\dot{\mathbb{I}}$. First, given a command C, the corresponding finite set of variables $Var_{\mathsf{C}} \triangleq vars(\mathsf{C})$, and an interval environment $\rho : Var_{\mathsf{C}} \to \mathbb{I}$, we define both the min and the max value of an interval environment:

$$\max(\rho) \triangleq \max\{\max(\rho(\mathtt{x})) \mid \mathtt{x} \in Var_{\mathsf{C}}\}$$
$$\min(\rho) \triangleq \max\{\min(\rho(\mathtt{x})) \mid \mathtt{x} \in Var_{\mathsf{C}}\}$$

Then, when computing $[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\rho$ on such ρ having a finite domain, we can restrict to an interval domain bounded by some constant $k \in \mathbb{N}$:

Definition 4.2 (Bounded interval)**.** We define $\dot{\mathbb{I}}_{k_1}^{k_2} \triangleq (Var_{\mathsf{C}} \to \mathbb{I}_{k_1}^{k_2}) \cup \{\top, \bot\}$ where

$$\begin{aligned}\mathbb{I}_{k_1}^{k_2} \triangleq\ & \{[a,b] \mid a,b \in \mathbb{Z}\ \wedge\ k_1 \leqslant a \leqslant b \leqslant k_2\} \\ & \cup \{[a,+\infty] \mid a \in \mathbb{Z}\ \wedge\ a \geqslant k_1\} \\ & \cup \{[-\infty,b] \mid b \in \mathbb{Z}\ \wedge\ b \leqslant k_2\}\end{aligned}$$

We can visualize the Hasse diagram of the bounded integer domain in Figure 4.1 and notice that there are no infinite ascending chains by definition. Now we can notice that given $k_1, k_2 \in \mathbb{Z}$ we can build a Galois Connection (Definition 1.14) between the interval domain $\dot{\mathbb{I}}$ (the *concrete* domain) and the bounded interval domain $\dot{\mathbb{I}}_{k_1}^{k_2}$ (the *abstract* domain). To do so we first need to define a concretization and abstraction maps.

Definition 4.3. Given $k_1, k_2 \in \mathbb{Z}$ we define a concretization map $\gamma_{k_1,k_2} : \mathbb{I}_{k_1}^{k_2} \to \mathbb{I}$ as the identity function

$$\gamma_{k_1,k_2} = \mathsf{id}$$

while we define an abstraction map $\alpha_{k_1,k_2} : \mathbb{I} \to \mathbb{I}_{k_1}^{k_2}$ in the following way

$$\alpha_{k_1,k_2}(\bot) = \bot$$
$$\alpha_{k_1,k_2}([a,b]) = \begin{cases}[a,b] & \text{if } a \geqslant k_1 \wedge b \leqslant k_2 \\ [-\infty,b] & \text{if } a < k_1 \wedge b \leqslant k_2 \\ [a,+\infty] & \text{if } a \geqslant k_1 \wedge b > k_2 \\ [-\infty,+\infty] & \text{otherwise}\end{cases}$$

Next, we prove that given $k_1, k_2 \in \mathbb{Z}$ we in fact have a Galois Connection:

Lemma 4.4. *Given $k_1, k_2 \in \mathbb{Z}$ s.t. $k_1 \leqslant k_2$*

$$\langle \mathbb{I}, \sqsubseteq \rangle \underset{\alpha_{k_1,k_2}}{\overset{\mathsf{id}}{\leftrightarrows}} \langle \mathbb{I}_{k_1}^{k_2}, \sqsubseteq \rangle$$

i.e., $\langle \alpha_{k_1,k_2}, \mathbb{I}, \mathbb{I}_{k_1}^{k_2}, \mathsf{id} \rangle$ is a Galois Connection.

Proof. We want to prove that id and α_{k_1,k_2} satisfy the property of Theorem 1.2:

(1) $\alpha_{k_1,k_2}, \mathsf{id}$ are monotonic;

(2) $\mathsf{id} \circ \alpha_{k_1,k_2}$ is extensive, i.e., $\forall \iota \in \mathbb{I}$ it holds that $\iota \sqsubseteq \mathsf{id}(\alpha_{k_1,k_2}(\iota))$;

(3) $\alpha_{k_1,k_2} \circ \mathsf{id}$ is reductive, i.e., $\forall \iota_b \in \mathbb{I}_{k_1}^{k_2}$ it holds that $\alpha_{k_1,k_2}(\mathsf{id}(\iota_b)) \sqsubseteq \iota_b$.

Let us show (1). id is monotone since $\forall \iota, \kappa \in \mathbb{I}_{k_1}^{k_2}$ it holds that $\iota \sqsubseteq \kappa \Rightarrow \iota \sqsubseteq \kappa$. For α_{k_1,k_2} we have to prove that for all $\iota, \kappa \in \mathbb{I}$ it holds that $\iota \sqsubseteq \kappa \Rightarrow \alpha_{k_1,k_2}(\iota) \sqsubseteq \alpha_{k_1,k_2}(\kappa)$. Now notice that $\iota \sqsubseteq \kappa$ means that $\min(\iota) \geqslant \min(\kappa)$ and $\max(\iota) \leqslant \max(\kappa)$. Hence, by Definition 4.3 of α_{k_1,k_2} it holds that $\alpha_{k_1,k_2}(\iota) \sqsubseteq \alpha_{k_1,k_2}(\kappa)$, which is our thesis.

Let us now show (2). We have to prove that $\forall \iota \in \mathbb{I}$ it holds that $\iota \sqsubseteq \gamma(\alpha_{k_1,k_2}(\iota))$. By hypothesis $\gamma = \mathsf{id}$, hence we just have to prove that $\iota \sqsubseteq \alpha_{k_1,k_2}(\iota)$. Based on the definition of α_{k_1,k_2} from Definition 4.3 both the following hold:

$$\min(\alpha_{k_1,k_2}(\iota)) \leqslant \min(\iota)$$
$$\max(\alpha_{k_1,k_2}(\iota)) \geqslant \max(\iota)$$

Hence it holds that

$$\iota \sqsubseteq \alpha_{k_1,k_2}(\iota) \tag{4.5}$$

We can finally prove (3): $\alpha_{k_1,k_2} \circ \gamma$ is reductive, i.e., $\forall \iota_b \in \mathbb{I}_{k_1}^{k_2}, \alpha_{k_1,k_2}(\mathsf{id}(\iota_b)) \sqsubseteq \iota_b$. Notice that $\forall \iota_b \in \mathbb{I}_{k_1}^{k_2}$ it holds that

$$\alpha_{k_1,k_2}(\iota_b) = \iota_b \tag{4.6}$$

hence it holds that $\alpha_{k_1,k_2}(\iota_b) \sqsubseteq \iota_b$ □

Notice that because of Equation (4.6) holds we know that $\alpha_{k_1,k_2} \circ \mathsf{id} = \mathsf{id}$, hence we not only have a Galois Connection but a Galois Injection (Definition 1.15):

$$\langle \mathbb{I}, \sqsubseteq \rangle \underset{\alpha_{k_1,k_2}}{\overset{\mathsf{id}}{\leftrightarrows\!\!\!\!\!\!\rightarrow}} \langle \mathbb{I}_{k_1}^{k_2}, \sqsubseteq \rangle \tag{4.7}$$

because of the latter observation and since $\dot{\mathbb{I}}, \dot{\mathbb{I}}_{k_1}^{k_2}$ are the point(variable)-wise lifting of $\mathbb{I}, \mathbb{I}_{k_1}^{k_2}$ respectively, by Theorem 1.5 it holds that

$$\langle \dot{\mathbb{I}}, \sqsubseteq \rangle \underset{\dot{\alpha}_{k_1,k_2}}{\overset{\mathsf{id}}{\leftrightarrows\!\!\!\!\!\!\rightarrow}} \langle \dot{\mathbb{I}}_{k_1}^{k_2}, \sqsubseteq \rangle \tag{4.8}$$

where $\dot{\alpha}_{k_1,k_2}(\eta) = \lambda \mathtt{x}.\alpha_{k_1,k_2}(\eta \mathtt{x})$. We can therefore define our analysis in $\dot{\mathbb{I}}_{k_1}^{k_2}$ by means of best correct approximations over $\dot{\mathbb{I}}_{k_1}^{k_2}$:

Definition 4.4 (Bounded interval analysis)**.** Let $\rho \in \dot{\mathbb{I}}_{k_1}^{k_2}$ for some $k_1, k_2 \in \mathbb{Z}$ s.t. $k_1 \leqslant k_2$ and $\mathsf{C} \in \mathrm{Imp}$. We define $[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho$ as follows

$$
\begin{aligned}
[\![\mathsf{e}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho &\triangleq \dot{\alpha}_{k_1,k_2}\left([\![\mathsf{e}]\!]^{\dot{\mathbb{I}}} \rho\right) \\
[\![\mathsf{C}_1 + \mathsf{C}_2]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho &\triangleq [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho \sqcup [\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho \\
[\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho &\triangleq \left([\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \circ [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\right) \rho \\
[\![\mathsf{C}^*]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho &\triangleq \bigsqcup_{i \in \mathbb{N}} \left([\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\right)^i \rho \\
[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho &\triangleq \mathrm{lfp}\left(\lambda \mu. \rho \sqcup [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \mu\right)
\end{aligned}
$$

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

Notice that for basic expressions we're using the best correct approximation $\dot{\alpha}_{k_1,k_2} \circ [\![\mathsf{e}]\!]^{\dot{\mathbb{I}}} \circ \mathsf{id}$, which allows us to prove the soundness of the analysis over $\dot{\mathbb{I}}_{k_1}^{k_2}$ w.r.t. the analysis over $\dot{\mathbb{I}}$:

Lemma 4.5. *for all $k_1, k_2 \in \mathbb{Z}$ s.t. $k_1 \leqslant k_2$, $\rho \in \dot{\mathbb{I}}_{k_1}^{k_2}$*

$$[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}} \rho \sqsubseteq [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}} \rho$$

i.e., with $\dot{\mathbb{I}}_{k_1}^{k_2}$ we have an over-approximation of $\dot{\mathbb{I}}$.

Proof. The theorem follows from the fact that for basic expressions we're using best correct approximations (from Definition 4.4), the Galois connection ensures that composition preserves soundness (by Theorem 1.6) and that least upper bounds are sound (by Property 5 of Theorem 1.3). □

Now we define a new bound, it will be useful for the latter Theorem 4.1.

Definition 4.5. Let $\mathsf{C} \in \mathrm{Imp}$. Then $(\cdot)_+^{\mathsf{b}} : \mathrm{Imp} \to \mathbb{N}$ is the *updated bound*, recursively defined as follows:

$$
\begin{aligned}
(\mathsf{e})_+^{\mathsf{b}} &\triangleq (\mathsf{e})^{\mathsf{b}} \\
(\mathsf{C}_1 + \mathsf{C}_2)_+^{\mathsf{b}} &\triangleq (\mathsf{C}_1)_+^{\mathsf{b}} + (\mathsf{C}_2)_+^{\mathsf{b}} \\
(\mathsf{C}_1; \mathsf{C}_2)_+^{\mathsf{b}} &\triangleq (\mathsf{C}_1)_+^{\mathsf{b}} + (\mathsf{C}_2)_+^{\mathsf{b}} \\
(\mathsf{fix}(\mathsf{C}))_+^{\mathsf{b}} &\triangleq (n+2)(\mathsf{C})_+^{\mathsf{b}}
\end{aligned}
$$

where $n = \mathit{vars}(\mathsf{C})$. Similarly, $(\cdot)_{\mathsf{b}}^+ : \mathrm{Imp} \to \mathbb{N}$ is the *updated lower bound* of commands, recursively defined as follows:

$$
\begin{aligned}
(\mathsf{e})_{\mathsf{b}}^+ &\triangleq (\mathsf{e})_{\mathsf{b}} \\
(\mathsf{C}_1 + \mathsf{C}_2)_{\mathsf{b}}^+ &\triangleq (\mathsf{C}_1)_{\mathsf{b}}^+ + (\mathsf{C}_2)_{\mathsf{b}}^+ \\
(\mathsf{C}_1; \mathsf{C}_2)_{\mathsf{b}}^+ &\triangleq (\mathsf{C}_1)_{\mathsf{b}}^+ + (\mathsf{C}_2)_{\mathsf{b}}^+ \\
(\mathsf{fix}(\mathsf{C}))_{\mathsf{b}}^+ &\triangleq (n+2)(\mathsf{C})_{\mathsf{b}}^+
\end{aligned}
$$

Notice that the updated bounds differ to bounds of Definition 4.1 only in the case of the $\mathsf{fix}(\mathsf{C})$ command. Thanks to the latter definition, we can now also define the notion of domain bounded by initial state and program.

Definition 4.6. Let $\mathsf{C} \in \text{Imp}$ and $\rho \in \dot{\mathbb{I}}$. Then the *bounded interval domain* $\dot{\mathbb{I}}_{\mathsf{C},\rho}$ is a bounded interval $\dot{\mathbb{I}}_{k_1}^{k_2}$ where

$$k_1 = \min(\rho) - (\mathsf{C})_{\mathsf{b}}^{+}$$
$$k_2 = \max(\rho) + (\mathsf{C})_{+}^{\mathsf{b}}$$

With this consideration we can now proceed to prove that the analysis on our bounded lattice $\dot{\mathbb{I}}_{\mathsf{C},\rho}$ produces the same result as the analysis on $\dot{\mathbb{I}}$.

Theorem 4.1. *Let $\mathsf{C} \in$ Imp be a command. Then, for all finitely supported $\rho : Var \to \mathbb{I}$ and $k_1, k_2 \in \mathbb{Z}$ s.t. $\dot{\mathbb{I}}_{\mathsf{C},\rho} \sqsubseteq \dot{\mathbb{I}}_{k_1}^{k_2}$, i.e., $k_1 \leqslant \min(\rho) - (\mathsf{C})_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{C})^{\mathsf{b}}$*

$$[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho \tag{4.9}$$

i.e., the abstract semantics $[\![\mathsf{C}]\!]\rho$ computed in $\dot{\mathbb{I}}$ and the one computed in $\dot{\mathbb{I}}_{k_1}^{k_2}$ coincide.

Proof. Notice that because of Lemma 4.5 the statement $[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\rho \sqsubseteq [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$ already holds. Therefore what we're left to prove is that

$$[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$$

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

Case $(\mathtt{x} \in S)$. In this case we want to prove that $[\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$. Recall that we are considering $k_1 \leqslant \min(\rho) - (\mathtt{x} \in S)_{\mathsf{b}} = \min(\rho) - (\mathtt{x} \in S)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathtt{x} \in S)^{\mathsf{b}} = \max(\rho) + (\mathtt{x} \in S)^{\mathsf{b}}$. Notice that either $\rho\mathtt{x} \sqcap S = \bot$, which implies that $[\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}}\rho = \bot$, and therefore $\alpha_{k_1,k_2}([\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}})\rho = \alpha_{k_1,k_2}(\bot) = \bot$ and therefore $\bot \sqsupseteq \bot$ holds, or $\rho\mathtt{x} \sqcap S = [a, b] \neq \bot$, but in this case $[\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}}\rho = \rho[\mathtt{x} \mapsto \rho\mathtt{x} \sqcap S]$ and we can observe that both the following hold:

$$\min(\rho) - (\mathtt{x} \in S)_{\mathsf{b}} \leqslant \min(\rho) \leqslant \min(\rho\mathtt{x} \sqcap S)$$
$$\max(\rho\mathtt{x} \sqcap S) \leqslant \max(\rho) \leqslant \max(\rho) + (\mathtt{x} \in S)^{\mathsf{b}}$$

hence

$$[\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}}\rho = \rho[\mathtt{x} \mapsto \rho\mathtt{x} \sqcap S] = \dot{\alpha}_{k_1,k_2}\left([\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}}\rho\right) = [\![\mathtt{x} \in S]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$$

which is our thesis.

Case $(\mathtt{x} := k)$. In this case we have to prove that $[\![\mathtt{x} := k]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathtt{x} := k]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$. Recall that we are considering $k_1 \leqslant \min(\rho) - (\mathtt{x} := k)_{\mathsf{b}} = \min(\rho) - (\mathtt{x} := k)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathtt{x} := k)^{\mathsf{b}}$. We can notice similarly to the previous case, that because of the values of k_1 and k_2 it holds that

$$[\![\mathtt{x} := k]\!]^{\dot{\mathbb{I}}}\rho = \rho[\mathtt{x} \mapsto [k, k]] = \dot{\alpha}_{k_1,k_2}\left([\![\mathtt{x} := k]\!]^{\dot{\mathbb{I}}}\rho\right) = [\![\mathtt{x} := k]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$$

hence our thesis holds.

Case $(\mathtt{x} := \mathtt{y} + k)$. In this case we have to prove that $[\![\mathtt{x} := \mathtt{y} + k]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathtt{x} := \mathtt{y} + k]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho$. Recall that we are considering $k_1 \leqslant \min(\rho) - (\mathtt{x} := \mathtt{y} + k)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathtt{x} := \mathtt{y} + k)^{\mathsf{b}}$. Notice also that $(\mathtt{x} := \mathtt{y} + k)_{\mathsf{b}} = k = (\mathtt{x} := \mathtt{y} + k)^{\mathsf{b}}$ and since $[\![\mathtt{x} := \mathtt{y} + k]\!]^{\dot{\mathbb{I}}}\rho = \rho[\mathtt{x} \mapsto \rho\mathtt{y} + k]$ we can notice that for each variable $\mathtt{w} \in Var$ it holds that

$$\min(\rho) - k \leqslant (\rho[\mathtt{x} \mapsto \rho\mathtt{y} + k])(\mathtt{w}) \leqslant \max(\rho) + k$$

hence

$$[\![\mathtt{x} := \mathtt{y} + k]\!]^{\dot{\mathbb{I}}_{k_1}^{k_2}}\rho = \dot{\alpha}_{k_1,k_2}(\rho[\mathtt{x} \mapsto \rho\mathtt{y} + k]) = \rho[\mathtt{x} \mapsto \rho\mathtt{y} + k] = [\![\mathtt{x} := \mathtt{y} + k]\!]^{\dot{\mathbb{I}}}\rho$$

which is our thesis

Case $(\mathsf{C}_1 + \mathsf{C}_2)$. In this case we have to prove that $[\![\mathsf{C}_1 + \mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathsf{C}_1 + \mathsf{C}_2]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$. Recall that we are considering $k_1 \leqslant \min(\rho) - (\mathsf{C}_1 + \mathsf{C}_2)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}}$. By inductive hypothesis it holds that

$$[\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$$

for all $k_1 \leqslant \min(\rho) - (\mathsf{C}_1)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{C}_1)^{\mathsf{b}}$. Again by inductive hypothesis it holds that

$$[\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$$

for all $k_1 \leqslant \min(\rho) - (\mathsf{C}_2)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{C}_2)^{\mathsf{b}}$. In particular, both hold for $k_1 \leqslant \min(\rho) - (\mathsf{C}_1)_{\mathsf{b}} - (\mathsf{C}_2)_{\mathsf{b}} = \min(\rho) - (\mathsf{C}_1 + \mathsf{C}_2)_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} = \max(\rho) + (\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}}$, i.e., our initial choice of k_1, k_2. We can conclude by closure over $\sqcup$

$$[\![\mathsf{C}_1 + \mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}}\rho \sqcup [\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho \sqcup [\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho = [\![\mathsf{C}_1 + \mathsf{C}_2]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$$

which is our thesis.

Case $(\mathsf{C}_1; \mathsf{C}_2)$. In this case we have to prove that $[\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$ for all $k_1 \leqslant \min(\rho) - (\mathsf{C}_1; \mathsf{C}_2)_{\mathsf{b}}$ and $k_2 \geqslant \max(\mathsf{C}) + (\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}}$. Recall that $[\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho = \left([\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}} \circ [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}}\right)\rho$. By inductive hypothesis it holds that

$$[\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho \qquad \forall k_1 \leqslant \min(\rho) - (\mathsf{C}_1)_{\mathsf{b}} \ \wedge \ k_2 \geqslant \max(\rho) + (\mathsf{C}_1)^{\mathsf{b}} \qquad (\dagger)$$

$$[\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho' = [\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}^{k_4}_{k_3}}\rho' \qquad \forall k_3 \leqslant \min(\rho') - (\mathsf{C}_2)_{\mathsf{b}} \ \wedge \ k_4 \geqslant \max(\rho') + (\mathsf{C}_2)^{\mathsf{b}} \qquad (\ddagger)$$

where $\rho' = [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}}\rho$. In particular notice that both $(\dagger)$ and $(\ddagger)$ hold for all n, m s.t.

$$m \leqslant \min(\rho) - (\mathsf{C}_1)_{\mathsf{b}} - (\mathsf{C}_2)_{\mathsf{b}}$$
$$n \geqslant \max(\rho) + (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}}$$

Hence

$$[\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\dot{\mathbb{I}}}\rho = \left([\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}} \circ [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}}\right)\rho = \left([\![\mathsf{C}_2]\!]^{\dot{\mathbb{I}}^n_m} \circ [\![\mathsf{C}_1]\!]^{\dot{\mathbb{I}}^n_m}\right)\rho = [\![\mathsf{C}_1; \mathsf{C}_2]\!]^{\dot{\mathbb{I}}^n_m}\rho$$

which is our thesis.

Case $(\mathsf{fix}(\mathsf{C}))$. What we want to prove in this case is that $[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho \sqsupseteq [\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$ for all $k_1 \leqslant \min(\rho) - (\mathsf{fix}(\mathsf{C}))_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{fix}(\mathsf{C}))^{\mathsf{b}}$. Recall that by Lemma 3.1 $[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}$ is syntactic sugar for $[\![(\mathsf{C} + \mathsf{true})^*]\!]^{\dot{\mathbb{I}}}$, therefore

$$[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho = [\![(\mathsf{C} + \mathsf{true})^*]\!]^{\dot{\mathbb{I}}}\rho = \bigsqcup_{i \in \mathbb{N}} \left([\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\right)^i \rho \qquad (4.10)$$

and we can build the chain of iterands ρ_i inductively in the following way

$$\rho_0 \triangleq \rho$$
$$\rho_{i+1} \triangleq [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\rho_i = \left([\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\right)^{i+1} \rho$$

based on the value of $\max([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y})$ we want to prove that both $\max([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y}) \geqslant \max([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho)$ and $\min([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y}) \leqslant \min([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho)$. For every variable $\mathtt{y} \in \mathit{Var}_{\mathsf{C}}$ we have 2 cases:

(i) If $\max([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y}) = +\infty$ (or $\min([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y}) = -\infty$) then we can recall Lemma 4.5 and notice that it holds that $[\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho \sqsubseteq [\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$, hence

$$+\infty = \max([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y}) \leqslant \max([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathtt{y}) = +\infty$$
$$-\infty = \min([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}}\rho\mathtt{y}) \geqslant \min([\![\mathsf{fix}(\mathsf{C})]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathtt{y}) = -\infty$$

but this means that $+\infty = \max([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) = \max\left([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathsf{y}\right)$ and $-\infty = \min([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) = \min\left([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathsf{y}\right)$ hold and the following inequality also hold:

$$+\infty = \max([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) \geqslant \max([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathsf{y}) = +\infty$$
$$-\infty = \min([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) \leqslant \min([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathsf{y}) = -\infty$$

(ii) Otherwise we have to prove that $\max([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) \geqslant \max([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho)$ and $\min([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) \leqslant \min([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho\mathsf{y})$ when respectively $\max([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) \neq +\infty$ and $\min([\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho\mathsf{y}) \neq -\infty$. In this case we also build the chain of iterands over the bounded analysis

$$\rho'_0 \triangleq \rho$$
$$\rho'_{i+1} \triangleq [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho'_i$$

What we want to prove, is that $\rho_{i+1} = \rho'_{i+1}$ holds for every $i \in \mathbb{N}$, this way the two chains hold the same result. To do so we work by induction on i:

Case $(i = 0)$. In this case, by induction on $\mathsf{C} + \mathsf{true}$ we know that

$$[\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\rho = [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$$

for all $k_1 \leqslant \min(\rho) - (\mathsf{C} + \mathsf{true})_{\mathsf{b}} = \min(\rho) - (\mathsf{C})_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (\mathsf{C} + \mathsf{true})^{\mathsf{b}} = \max(\rho) + (\mathsf{C})^{\mathsf{b}}$. Hence by our choice of k_1, k_2 it holds.

Case $(i \Rightarrow i + 1)$. In this case we have to prove

$$\rho_{i+1} = [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\rho_i \quad = \quad [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho'_i = \rho'_{i+1}$$

To start, by induction on i we can say that $\rho_i = \rho'_i = \rho''$. By induction on $\mathsf{C} + \mathsf{true}$ we can say that

$$[\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\rho'' = [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho'' \tag{4.11}$$

for all $k_1 \leqslant \min(\rho'') - (\mathsf{C})_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho'') + (\mathsf{C})^{\mathsf{b}}$. Now we can observe that because of (4.10) and Lemma 4.2 it holds that

$$\min(\rho'') \geqslant \min(\rho) - (\mathsf{fix(C)})_{\mathsf{b}} = \min(\rho) - (n+1)(\mathsf{C})_{\mathsf{b}}$$
$$\max(\rho'') \leqslant \max(\rho) + (\mathsf{fix(C)})^{\mathsf{b}} = \max(\rho) + (n+1)(\mathsf{C})^{\mathsf{b}}$$

hence by choosing

$$k_1 \leqslant \min(\rho) - (\mathsf{fix(C)})_{\mathsf{b}} - (\mathsf{C})_{\mathsf{b}} = \min(\rho) - (n+2)(\mathsf{C})_{\mathsf{b}}$$
$$k_2 \geqslant \max(\rho) + (\mathsf{fix(C)})^{\mathsf{b}} + (\mathsf{C})^{\mathsf{b}} = \max(\rho) + (n+2)(\mathsf{C})^{\mathsf{b}}$$

the inductive hypothesis holds and we can conclude that

$$\rho_{i+1} = [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\rho_i = [\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho'_i = \rho'_{i+1} \tag{4.12}$$

Hence the following

$$[\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}}\rho = \bigsqcup_{i\in\mathbb{N}}\left([\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}}\right)^i\rho = \bigsqcup_{i\in\mathbb{N}}\left([\![\mathsf{C} + \mathsf{true}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\right)^i\rho = [\![\mathsf{fix(C)}]\!]^{\dot{\mathbb{I}}^{k_2}_{k_1}}\rho$$

holds for every $k_1 \leqslant \min(\rho) - (n+2)(\mathsf{C})_{\mathsf{b}}$ and $k_2 \geqslant \max(\rho) + (n+2)(\mathsf{C})^{\mathsf{b}}$, which is our thesis.

□

4.4 Computing non-relational collecting semantics

todo

Appendix A

Additional proofs

A.1 Lemma 4.3 proof

We preliminarly observe that we can also prove a dual property of the Lemma 4.1:

Lemma A.1 (**Negative cycles in weighted directed graphs**). *Let p be a finite path*

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

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

Proof. First note that $w(p) = \Sigma_{k=0}^{\ell-1} h_k < -m(|X|-1)$ implies that $|p| = \ell \geqslant |X|$. Then, we show our claim by induction on $|p| = \ell \geqslant |X|$.
Case ($|p| = |X|$). Since the path p includes exactly $|X|+1 = \ell+1$ nodes, there exist indices $0 \leqslant i < j \leqslant \ell$ 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

$$\begin{aligned}
w(p_{i,j}) = w(p) - (\Sigma_{k=0}^{i-1} h_k + \Sigma_{k=j}^{\ell-1} h_k) < & \qquad \text{as } w(p) < -m(|X|-1) \\
-m\,(|X|-1) - (\Sigma_{k=0}^{i-1} h_k + \Sigma_{k=j}^{\ell-1} h_k) \leqslant & \qquad \text{as } \Sigma_{k=0}^{i-1} h_k + \Sigma_{k=j}^{\ell-1} h_k \geqslant -m(\ell-1) \\
-m(|X|-1) - (-m(\ell-1)) = & \qquad \text{as } \ell = |X| \\
-m(|X|-1) + m(|X|-1) = 0 &
\end{aligned}$$

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 $0 \leqslant i < j \leqslant \ell$, which is maximal, i.e., such that if $p_{i',j'}$ is a cycle in p, for some $0 \leqslant i' < j' \leqslant \ell$, 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 $w(p_{i,j}) \geqslant 0$ and we consider the path p' obtained from p by stripping off the cycle $p_{i,j}$, i.e.,

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

Since $|p'| < |p|$ and $w(p') = w(p) - w(p_{i,j}) \leqslant w(p) < -m(|X|-1)$, 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. □

For the following proof consider the $\min : \mathbb{I} \to \mathbb{Z}$ function, inductively defined as follows:

$$\min(\bot) = +\infty$$
$$\min([a, b]) = a$$

and recall the Lemma 4.3 statement:

Lemma A.2. *Let* $\mathsf{C} \in$ *Imp.*
For all $\eta \in \dot{\mathbb{I}}$ *and* $\mathtt{y} \in$ *Var, if* $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) \neq -\infty$ *and* $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) < -(\mathsf{C})_{\mathsf{b}}$ *then there exist a variable* $\mathtt{z} \in$ *Var and an integer* $h \in \mathbb{Z}$ *s.t.* $|h| \leqslant (\mathsf{C})_{\mathsf{b}}$ *s.t. the following two properties hold:*

(i) $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) = \min(\eta\mathtt{z}) + h$*;*

(ii) for all $\eta' \in \dot{\mathbb{I}}$*, if* $\eta' \sqsupseteq \eta$ *then* $\min([\![\mathsf{C}]\!]\eta'\mathtt{y}) \leqslant \min(\eta'\mathtt{z}) + h$*.*

Proof. The proof is by structural induction on the command $\mathsf{C} \in$ Imp. We preliminarly observe that we can safely assume $\eta \neq \bot$. In fact, if $\eta = \bot$ then $[\![\mathsf{C}]\!]\bot = \bot$ and thus $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) = +\infty \geqslant (\mathsf{C})^{\mathsf{b}}$, against the hypothesis $\min([\![\mathsf{C}]\!]\eta\mathtt{y}) < -(\mathsf{C})_{\mathsf{b}}$. Moreover, when quantifying over η' such that $\eta' \sqsupseteq \eta$ in (i), if $\min([\![\mathsf{C}]\!]\eta'\mathtt{y}) = -\infty$ holds, then $\min([\![\mathsf{C}]\!]\eta'\mathtt{y}) \leqslant \min(\eta'\mathtt{z}) + h$ trivially holds, hence we will sometimes silently omit to consider this case.

Case $(\mathtt{x} \in S)$
Take $\eta \in \dot{\mathbb{I}}$ and assume $-\infty \neq \min([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) < -(\mathtt{x} \in S)_{\mathsf{b}}$. Clearly $[\![\mathtt{x} \in S]\!]\eta \neq \bot$, otherwise we would get the contradiction $\min([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) = +\infty \geqslant (\mathtt{x} \in S)_{\mathsf{b}}$. We distinguish two cases:

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

  $$\bot \neq [\![\mathtt{x} \in S]\!]\eta' = \eta'[\mathtt{x} \mapsto \alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta(\mathtt{x})) \cap \gamma_{\mathbb{I}}(S))]$$

  and thus

  $$\min([\![\mathtt{x} \in S]\!]\eta'\mathtt{y}) = \min(\eta'\mathtt{y}) = \min(\eta'\mathtt{y}) + 0$$

  hence the thesis follows with $\mathtt{z} = \mathtt{y}$ and $h = 0$.

- If $\mathtt{y} = \mathtt{x}$ then $\eta(\mathtt{x}) \in \mathbb{I}$ and

  $$\min([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) = \min(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta\mathtt{x}) \cap \gamma_{\mathbb{I}}(S)))$$

  Note that it cannot be $\min(S) \in \mathbb{Z}$. Otherwise, by Definition **??**, $\min(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta\mathtt{x}) \cap \gamma_{\mathbb{I}}(S))) \geqslant \min(S) = (\mathtt{x} \in S)_{\mathsf{b}}$, violating the assumption $\min([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) < -(\mathtt{x} \in S)_{\mathsf{b}}$. Hence, $\min(S) = -\infty$ must hold and therefore $\min(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta\mathtt{x}) \cap \gamma_{\mathbb{I}}(S))) = \min(\eta(\mathtt{x})) = \min(\eta(\mathtt{x})) + 0$. It is immediate to check that the same holds for all $\eta' \sqsupseteq \eta$, i.e.,

  $$\min(\alpha_{\mathbb{I}}(\gamma_{\mathbb{I}}(\eta'\mathtt{x}) \cap \gamma_{\mathbb{I}}(S))) = \min(\eta'\mathtt{x}) + 0$$

  and thus the thesis follows with $\mathtt{z} = \mathtt{y} = \mathtt{x}$ and $h = 0$.

Case $(\mathtt{x} := k)$ Take $\eta \in \dot{\mathbb{I}}$ and assume $\min([\![\mathtt{x} := k]\!]\eta\mathtt{y}) < -(\mathtt{x} := k)_{\mathsf{b}} = |k|$.

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

$$\min([\![\mathtt{x} := k]\!]\eta\mathtt{y}) = k \geqslant |k| = (\mathtt{x} := k)_{\mathsf{b}}$$

violating the assumption. Therefore, it must be $\mathtt{y} \neq \mathtt{x}$. Now, for all $\eta' \sqsupseteq \eta$, we have $[\![\mathtt{x} := k]\!]\eta'\mathtt{y} = \eta'\mathtt{y}$ and thus

$$\min([\![\mathtt{x} := k]\!]\eta'\mathtt{y}) = \min(\eta'\mathtt{y}) = \min(\eta'\mathtt{y}) + 0,$$

hence the thesis holds with $h = 0 \leqslant |k| = (\mathtt{x} := k)_\mathsf{b}$ and $\mathtt{z} = \mathtt{y}$.

Case $(\mathtt{x} := \mathtt{w} + k)$ Take $\eta \in \dot{\mathbb{I}}$ and assume $\min([\![\mathtt{x} := \mathtt{w} + k]\!]\eta\mathtt{y}) < -(\mathtt{x} := \mathtt{w} + k)_\mathsf{b} = -|k|$. Recall that $[\![\mathtt{x} := \mathtt{w} + k]\!]\eta = \eta[\mathtt{x} \mapsto \eta\mathtt{w} + k]$.

We distinguish two cases:

- If $\mathtt{y} \neq \mathtt{x}$, then for all $\eta' \sqsupseteq \eta$, we have $[\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y} = \eta'\mathtt{y}$ and thus

  $$\min([\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y}) = \min(\eta'\mathtt{y}) + 0$$

  hence the thesis follows with $h = 0 \leqslant (\mathtt{x} := \mathtt{w} + k)_\mathsf{b}$ and $\mathtt{z} = \mathtt{y}$.

- If $\mathtt{x} = \mathtt{y}$ then for all $\eta' \sqsupseteq \eta$, we have $[\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y} = \eta'\mathtt{w} + k$ and thus

  $$\min([\![\mathtt{x} := \mathtt{w} + k]\!]\eta'\mathtt{y}) = \min(\eta'\mathtt{w}) + k$$

  hence, the thesis follows with $h = k$ (recall that $k \leqslant |k| = (\mathtt{x} := \mathtt{w} + k)_\mathsf{b}$) and $\mathtt{z} = \mathtt{w}$.

Case $(\mathsf{C}_1 + \mathsf{C}_2)$ Take $\eta \in \dot{\mathbb{I}}$ and assume $\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta) < -(\mathsf{C}_1 + \mathsf{C}_2)_\mathsf{b} = -(\mathsf{C}_1)_\mathsf{b} - (\mathsf{C}_2)_\mathsf{b}$. Recall that $[\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta = [\![\mathsf{C}_1]\!]\eta \sqcup [\![\mathsf{C}_2]\!]\eta$. Hence, since $\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) \neq -\infty$, we have that $\min([\![\mathsf{C}_1]\!]\eta\mathtt{y}) \neq -\infty \neq \min([\![\mathsf{C}_2]\!]\eta\mathtt{y})$. Moreover

$$\begin{aligned}\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) &= \min([\![\mathsf{C}_1]\!]\eta\mathtt{y} \sqcup [\![\mathsf{C}_2]\!]\eta\mathtt{y}) \\ &= \min\{\min([\![\mathsf{C}_1]\!]\eta\mathtt{y}), \min([\![\mathsf{C}_2]\!]\eta\mathtt{y})\}\end{aligned}$$

Thus $\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \min([\![\mathsf{C}_i]\!]\eta\mathtt{y})$ for some $i \in \{1, 2\}$. We can assume, without loss of generality, that the maximum is realized by the first component, i.e., $\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \min([\![\mathsf{C}_1]\!]\eta\mathtt{y}) < -(\mathsf{C}_1 + \mathsf{C}_2)_\mathsf{b}$. Hence we can use the inductive hypothesis on C_1 and state that there exists $h \in \mathbb{Z}$ with $|h| \leqslant (\mathsf{C}_1)_\mathsf{b}$ and $\mathtt{z} \in \mathit{Var}$ such that $\min([\![\mathsf{C}_1]\!]\eta\mathtt{y}) = \min(\eta\mathtt{z}) + h$ and for all $\eta' \in \dot{\mathbb{I}}$, $\eta \sqsubseteq \eta'$,

$$\min([\![\mathsf{C}_1]\!]\eta'\mathtt{y}) \leqslant \min(\eta'\mathtt{z}) + h$$

Therefore

$$\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \min([\![\mathsf{C}_1]\!]\eta\mathtt{y}) = \min(\eta\mathtt{z}) + h$$

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

$$\begin{aligned}\min([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta'\mathtt{y}) &= \min\{\min([\![\mathsf{C}_1]\!]\eta'\mathtt{y}), \min([\![\mathsf{C}_2]\!]\eta'\mathtt{y})\} \\ &\leqslant \min([\![\mathsf{C}_1]\!]\eta'\mathtt{y}) \\ &\leqslant \min(\eta'\mathtt{z}) + h\end{aligned}$$

with $|h| \leqslant (\mathsf{C}_1)_\mathsf{b} \leqslant (\mathsf{C}_1 + \mathsf{C}_2)_\mathsf{b}$, as desired.

Case $(\mathsf{C}_1; \mathsf{C}_2)$ Take $\eta \in \dot{\mathbb{I}}$ and assume $\min([\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta\mathtt{y}) < -(\mathsf{C}_1; \mathsf{C}_2)_\mathsf{b} = -(\mathsf{C}_1)_\mathsf{b} - (\mathsf{C}_2)_\mathsf{b}$.

Recall that $[\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta = [\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta)$. If we define

$$[\![\mathsf{C}_1]\!]\eta = \eta_1$$

since $\min([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) \neq -\infty$ and $\min([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) < -(\mathsf{C}_1; \mathsf{C}_2)_\mathsf{b} \leqslant (\mathsf{C}_2)_\mathsf{b}$, by inductive hypothesis on C_2, there are $|h_2| \leqslant (\mathsf{C}_2)_\mathsf{b}$ and $\mathtt{w} \in \mathit{Var}$ such that $\min([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) = \min(\eta_1\mathtt{w}) + h_2$ and for all $\eta_1' \in \dot{\mathbb{I}}$ with $\eta_1 \sqsubseteq \eta_1'$

$$\min([\![\mathsf{C}_2]\!]\eta_1'\mathtt{y}) \leqslant \min(\eta_1'\mathtt{w}) + h_2 \qquad (\dagger)$$

Now observe that $\min([\![\mathsf{C}_1]\!]\eta\mathtt{w}) = \min(\eta_1\mathtt{w}) < -(\mathsf{C}_1)_\mathsf{b}$. Otherwise, if it were $\min(\eta_1\mathtt{w}) \geqslant -(\mathsf{C}_1)_\mathsf{b}$ we would have

$$\min([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) = \min(\eta_1\mathtt{w}) + h_2 \geqslant -(\mathsf{C}_1)_\mathsf{b} - (\mathsf{C}_2)_\mathsf{b} = -(\mathsf{C}_1; \mathsf{C}_2)_\mathsf{b},$$

violating the hypotheses. Moreover, $\min([\![\mathsf{C}_1]\!]\eta\mathtt{w}) \neq -\infty$, otherwise we would have $\min([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) = \min(\eta_1\mathtt{w}) + h_2 = -\infty$, contradicting the hypotheses. Therefore we can apply the inductive hypothesis also to C_1 and deduce that there are $|h_1| \leqslant (\mathsf{C}_1)_\mathsf{b}$ and $\mathtt{w}' \in \mathit{Var}$ such that $\min([\![\mathsf{C}_1]\!]\eta\mathtt{w}) = \min(\eta\mathtt{w}') + h_1$ and for all $\eta' \in \dot{\mathbb{I}}$ with $\eta \sqsubseteq \eta'$

$$\min([\![\mathsf{C}_1]\!]\eta'\mathtt{w}) \leqslant \min(\eta'\mathtt{w}') + h_1 \tag{‡}$$

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

$$\begin{aligned}\min([\![\mathsf{C}_1;\mathsf{C}_2]\!]\eta\mathtt{y}) &= \min([\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta)\mathtt{y}) \\ &= \min([\![\mathsf{C}_2]\!]\eta_1\mathtt{y}) \\ &= \min(\eta_1\mathtt{w}) + h_2 \\ &= \min([\![\mathsf{C}_1]\!]\eta\mathtt{w}) + h_2 \\ &= \min(\eta\mathtt{w}') + h_1 + h_2\end{aligned}$$

and

$$\begin{aligned}\min([\![\mathsf{C}_1;\mathsf{C}_2]\!]\eta'\mathtt{y}) &= \\ \min([\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta')\mathtt{y}) &\leqslant \\ \min([\![\mathsf{C}_1]\!]\eta'\mathtt{w}) + h_2 &\leqslant && \text{by (†), since } \eta_1 = [\![\mathsf{C}_1]\!]\eta \sqsubseteq [\![\mathsf{C}_1]\!]\eta' \\ (\min(\eta'\mathtt{w}') + h_1) + h_2 & && \text{by (‡)}\end{aligned}$$

Thus, the thesis holds with $h = h_1 + h_2$, as $|h| = |h_1 + h_2| \leqslant |h_1| + |h_2| \leqslant (\mathsf{C}_1)_\mathsf{b} + (\mathsf{C}_2)_\mathsf{b} = (\mathsf{C}_1;\mathsf{C}_2)_\mathsf{b}$, as needed.

Case ($\mathsf{fix}(\mathsf{C})$) Let $\eta \in \dot{\mathbb{I}}$ such that $\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) \neq -\infty$. Recall that $[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathrm{lfp}\lambda\mu.([\![\mathsf{C}]\!]\mu \sqcup \eta)$. Observe that the least fixpoint of $\lambda\mu.([\![\mathsf{C}]\!]\mu \sqcup \eta)$ coincides with the least fixpoint of $\lambda\mu.([\![\mathsf{C}]\!]\mu \sqcup \mu) = \lambda\mu.[\![\mathsf{C} + \mathsf{true}]\!]\mu$ above η. Hence, if

- $\eta_0 \triangleq \eta$,
- for all $i \in \mathbb{N}$, $\eta_{i+1} \triangleq [\![\mathsf{C}]\!]\eta_i \sqcup \eta_i = [\![\mathsf{C} + \mathsf{true}]\!]\eta_i \sqsupseteq \eta_i$,

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

$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \bigsqcup_{i\in\mathbb{N}} \eta_i.$$

Since $\min([\![\mathsf{fix}(\mathsf{C})]\!])\eta\mathtt{y} \neq -\infty$, we have that for all $i \in \mathbb{N}$, $\min(\eta_i\mathtt{y}) \neq -\infty$. Moreover, $\bigsqcup_{i\in\mathbb{N}} \eta_i$ on $\mathtt{y}$ is finitely reached in the chain $\{\eta_i\}_{i\in\mathbb{N}}$, i.e., there exists $m \in \mathbb{N}$ such that for all $i \geqslant m + 1$

$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y} = \eta_i\mathtt{y}.$$

The inductive hypothesis holds for C and true, hence for $\mathsf{C} + \mathsf{true}$, therefore for all $\mathtt{x} \in \mathit{Var}$ and $j \in \{0, 1, \ldots, m\}$, if $\min(\eta_{j+1}\mathtt{x}) < -(\mathsf{C} + \mathsf{true})_\mathsf{b} = -(\mathsf{C})_\mathsf{b}$ then there exist $\mathtt{z} \in \mathit{Var}$ and $h \in \mathbb{Z}$ such that $|h| \leqslant (\mathsf{C})_\mathsf{b}$ and

(a) $-\infty \neq \min(\eta_{j+1}\mathtt{x}) = \min(\eta_j\mathtt{z}) + h$,

(b) $\forall\eta' \sqsupseteq \eta_j.\, \min([\![\mathsf{C} + \mathsf{true}]\!]\eta'\mathtt{x}) \leqslant \min(\eta'\mathtt{z}) + h$.

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

$$(\mathtt{z}, j) \to_h (\mathtt{x}, j + 1)$$

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

$$\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) = \min(\eta_{m+1}\mathtt{y}) < -(\mathsf{fix}(\mathsf{C}))_\mathsf{b} = -(n + 1)(\mathsf{C})_\mathsf{b}$$

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

$$\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) = \min(\eta\mathtt{z}) + h \tag{i}$$

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

$$\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta'\mathtt{y}) \leqslant \min(\eta'\mathtt{z}) + h \tag{ii}$$

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

$$\sigma \triangleq (\mathtt{y}_0, 0) \rightarrow_{h_0} (\mathtt{y}_1, 1) \rightarrow_{h_1} \ldots \rightarrow_{h_m} (\mathtt{y}_{m+1}, m+1) \tag{A.1}$$

such that $\mathtt{y}_{m+1} = \mathtt{y}$ and for all $j \in \{0, \ldots, m+1\}$, $\mathtt{y}_j \in Var$ and $\min(\eta_j\mathtt{y}_j) < -(\mathsf{C})_{\mathsf{b}}$. In fact, if, by contradiction, this is not the case, there would exist an index $i \in \{0, \ldots, m\}$ (as

$$\min(\eta_{m+1}\mathtt{y}_{m+1}) < -(\mathsf{C})_{\mathsf{b}}$$

already holds) such that $\min(\eta_i\mathtt{y}_i) \geqslant -(\mathsf{C})_{\mathsf{b}}$, while for all $j \in \{i+1, \ldots, m+1\}$, $\min(\eta_j\mathtt{y}_j) < -(\mathsf{C})_{\mathsf{b}}$. Thus, in such a case, we consider the nonempty path:

$$\pi \triangleq (\mathtt{y}_i, i) \rightarrow_{h_i} (\mathtt{y}_{i+1}, i+1) \rightarrow_{h_{i+1}} \ldots \rightarrow_{h_m} (\mathtt{y}_{m+1}, m+1)$$

and we have that:

$$\begin{array}{l}
\Sigma_{j=i}^{m} h_j = \\
\Sigma_{j=i}^{m} \min(\eta_{j+1}\mathtt{y}_{j+1}) - \min(\eta_j\mathtt{y}_j) = \\
\min(\eta_{m+1}\mathtt{y}_{m+1}) - \min(\eta_i\mathtt{y}_i) = \\
\min(\eta_{m+1}\mathtt{y}) - \min(\eta_i\mathtt{y}_i) < \\
-(n+1)(\mathsf{C})_{\mathsf{b}} + (\mathsf{C})_{\mathsf{b}} = -n(\mathsf{C})_{\mathsf{b}}
\end{array}$$

with $|h_j| \leqslant (\mathsf{C})_{\mathsf{b}}$ for $j \in \{i, \ldots, m\}$. 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 $i \leqslant k_1 < k_2 \leqslant m+1$ such that $\mathtt{y}_{k_1} = \mathtt{y}_{k_2}$ and $h = \Sigma_{j=k_1}^{k_2-1} h_j < 0$. If we denote $\mathtt{w} = \mathtt{y}_{k_1} = \mathtt{y}_{k_2}$, then we have that

$$\begin{array}{lll}
\min(\eta_{k_2}\mathtt{w}) = h_{k_2-1} + \min(\eta_{k_2-1}\mathtt{w}) & & \\
= h_{k_2-1} + h_{k_2-2} + \min(\eta_{k_2-2}\mathtt{w}) & & \\
= \Sigma_{j=k_1}^{k_2-1} h_j + \min(\eta_{k_1}\mathtt{w}) & & \\
= \min(\eta_{k_1}\mathtt{w}) + h & & \text{recall } h = \Sigma_{j=k_1}^{k_2-1} h_j < 0
\end{array}$$

Thus,

$$\min([\![\mathsf{C} + \mathsf{true}]\!]^{k_2-k_1}\eta_{k_1}\mathtt{w}) = \min(\eta_{k_1}\mathtt{w}) + h$$

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

$$\min([\![\mathsf{C} + \mathsf{true}]\!]^{k_2-k_1}\eta'\mathtt{w}) \leqslant \min(\eta'\mathtt{w}) + h \tag{A.2}$$

This property (A.2) can be shown by induction on $k_2 - k_1 \geqslant 1$.
Then, an inductive argument allows us to show that for all $r \in \mathbb{N}$:

$$\min([\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)}\eta_{k_1}\mathtt{w}) \leqslant \min(\eta_{k_1}\mathtt{w}) + rh \tag{A.3}$$

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

$$\begin{array}{lll}
\min([\![\mathsf{C} + \mathsf{true}]\!]^{(r+1)(k_2-k_1)}\eta_{k_1}\mathtt{w}) = & & \\
\min([\![\mathsf{C} + \mathsf{true}]\!]^{k_2-k_1}([\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)}\eta_{k_1})\mathtt{w}) \leqslant & \text{by (A.2) as } \eta_{k_1} \sqsubseteq [\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)}\eta_{k_1} & \\
\min([\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)}\eta_{k_1}\mathtt{w}) + h \leqslant & & \text{by inductive hypothesis} \\
\min(\eta_{k_1}\mathtt{w}) + rh + h \leqslant \min(\eta_{k_1}\mathtt{w}) + (r+1)h & &
\end{array}$$

However, This would contradict the hypothesis $[\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y} \neq -\infty$. In fact the inequality (A.3) would imply

$$\begin{aligned}\min\left([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{w}\right) &= \min\left(\bigsqcup_{i\in\mathbb{N}} [\![\mathsf{C}+\mathsf{true}]\!]^i\eta\mathtt{w}\right)\\ &= \min\left(\bigsqcup_{i\in\mathbb{N}} [\![\mathsf{C}+\mathsf{true}]\!]^i\eta_{k_1}\mathtt{w}\right)\\ &= \min\left(\bigsqcup_{r\in\mathbb{N}} [\![\mathsf{C}+\mathsf{true}]\!]^{r(k_2-k_1)}\eta_{k_1}\mathtt{w}\right)\\ &= -\infty\end{aligned}$$

Now, from (A.1) we deduce that for all $\eta' \sqsupseteq \eta_{k_1}$, for $j \in \{k_1,\ldots,m\}$, if we let $\mu_{k_1} = \eta'$ and $\mu_{j+1} = [\![\mathsf{C}+\mathsf{true}]\!]\mu_j$, we have that $\min(\mu_{j+1}\mathtt{y}_{j+1}) \leqslant \min(\mu_{j+1}\mathtt{y}_j) + h_j$ and thus

$$[\![\mathsf{C}+\mathsf{true}]\!]^{m-k_1+1}\eta'\mathtt{y} = \mu_{m+1}\mathtt{y}_{m+1} \leqslant \min(\mathtt{y}_{k_1}) + \Sigma_{i=k_1}^m h_i = \min(\eta'\mathtt{w}) + \Sigma_{i=k_1}^m h_i$$

Since $\eta' = [\![\mathsf{fix}(\mathsf{C})]\!]\eta \sqsupseteq \eta_{k_1}$ we conclude

$$\begin{aligned}\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) &= \min\left([\![\mathsf{C}+\mathsf{true}]\!]^{m-k_1+1}[\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}\right)\\ &\leqslant -\infty + \Sigma_{i=k_1}^m h_i = -\infty\end{aligned}$$

contradicting the assumption. Therefore, the path σ of (A.1) must exist, and consequently

$$\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) = \min(\eta_{m+1}\mathtt{y}) = \min(\eta\mathtt{y}_0) + \Sigma_{i=0}^m h_i$$

and $|\Sigma_{i=0}^m h_i| \leqslant (\mathsf{fix}(\mathsf{C}))_{\mathsf{b}} = (n+1)(\mathsf{C})_{\mathsf{b}}$, otherwise we could use the same argument above for inferring the contradiction $\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y}) = -\infty$.

Let us now show (ii). Given $\eta' \sqsupseteq \eta$ from (A.1) we deduce that for all $j \in \{0,\ldots,m\}$, if we let $\mu_0 = \eta'$ and $\mu_{j+1} = [\![\mathsf{C}+\mathsf{true}]\!]\mu_j$, we have that

$$\min(\mu_{j+1}\mathtt{y}_{j+1}) \leqslant \min(\mu_{j+1}\mathtt{y}_j) + h_j.$$

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

$$\min([\![\mathsf{fix}(\mathsf{C})]\!]\eta'\mathtt{y}) \leqslant \min(\mu_{m+1}\mathtt{y}) = \min(\mu_{m+1}\mathtt{y}_{m+1}) \leqslant \min(\eta'\mathtt{y}_0) + \Sigma_{i=0}^m h_i$$

as desired. □

Bibliography

[Bel58] Richard Bellman. "On a routing problem". In: *Quarterly of Applied Mathematics* 16 (1958), pp. 87–90. DOI: 10.1090/qam/102435 (cit. on p. 1).

[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. URL: https://doi.org/10.1145/512950.512973 (cit. on pp. i, 1, 4, 8).

[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. URL: https://doi.org/10.1145/567752.567778 (cit. on p. 4).

[Cut80] Nigel Cutland. *Computability: An introduction to recursive function theory*. Cambridge university press, 1980 (cit. on pp. 1, 18, 19).

[Gaw+09] Thomas Gawlitza et al. "Polynomial Precise Interval Analysis Revisited". In: *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*. 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. URL: https://doi.org/10.1007/978-3-642-03456-5_28 (cit. on p. 1).

[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 (cit. on p. 4).

[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 (cit. on p. 23).

[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. URL: https://doi.org/10.1145/256167.256195 (cit. on p. 10).

[Min18] Antonie Miné. *Static Inference of Numeric Invariants by Abstract Interpretation*. Université Pierre et Marie Curie, Paris, France, 2018 (cit. on pp. 2, 4).

[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 p. 21).

[Tar55] Alfred Tarski. "A lattice-theoretical fixpoint theorem and its applications." In: (1955) (cit. on p. 3).