University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master degree in Computer Science

Titolo della tesi

Supervisor
Prof. Prof 1

Co. Supervisor
Prof. Prof 2

Candidate
Luca Zaninotto

Academic Year 2023-2024

Abstract

Abstract

Acknowledgments

?

Contents

Chapter 1

Framework

1.1 The Imp language

We denote by $\mathbb{Z}$ the set of integers with the usual order, extended with bottom and top elements $-\infty$ and $+\infty$, s.t. $-\infty \leqslant z \leqslant +\infty \quad \forall z \in \mathbb{Z}$. We also extend addition and subtraction by letting, for $z \in \mathbb{Z} \quad +\infty + z = +\infty - z = +\infty$ and $-\infty + z = -\infty - z = -\infty$. We focus on the following non-deterministic language.

$$
\begin{aligned}
\mathsf{Exp} \ni \mathsf{e} &::= \mathtt{x} \in S \mid \mathtt{x} \in [a, b] \mid \mathtt{x} \leqslant k \mid \mathtt{x} > k \mid \mathsf{true} \mid \mathsf{false} \mid \\
&\quad\ \mathtt{x} := k \mid \mathtt{x} := \mathtt{y} + k \mid \mathtt{x} := \mathtt{y} - k \\
\mathrm{Imp}_{\neq\star} \ni \mathsf{D} &::= \mathsf{e} \mid \mathsf{D} + \mathsf{D} \mid \mathsf{D}; \mathsf{D} \\
\mathrm{Imp} \ni \mathsf{C} &::= \mathsf{D} \mid \mathsf{C}^* \mid \mathsf{fix}(\mathsf{C})
\end{aligned}
$$

where $\mathtt{x}, \mathtt{y} \in \ Var$ a finite set of variables of interest, i.e., the variables appearing in the considered program, $S \subseteq \mathbb{Z}$ is (possibly empty) subset of numbers, $a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}, a \leqslant b, k \in \mathbb{Z}$ is any finite integer constant.

1.1.1 Syntactic sugar

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

$$
\begin{aligned}
\mathtt{x} \in S \vee \mathtt{x} \in S &= (\mathtt{x} \in S) + (\mathtt{x} \in S) \\
\mathtt{x} \in S \wedge \mathtt{x} \in S &= (\mathtt{x} \in S); (\mathtt{x} \in S) \\
\mathtt{x} \notin S &= \mathtt{x} \in \neg S \\
\mathsf{if}\ b\ \mathsf{then}\ C_1\ \mathsf{else}\ C_2 &= (\mathsf{e}; \mathsf{C}_1) + (\neg \mathsf{e}; \mathsf{C}_2) \\
\mathsf{while}\ b\ \mathsf{do}\ C &= \mathsf{fix}(\mathsf{e}; \mathsf{C}); \neg \mathsf{e} \\
\mathtt{x++} &= \mathtt{x} := \mathtt{x} + 1
\end{aligned}
$$

1.2 Semantics

The first building block is that of environments, mapst from the set of variables to their value.

Definition 1.1 (Environments)**.** Environments are (total) maps from variables to (numerical) values

$$\mathsf{Env} \triangleq \{\rho \mid \rho : \ Var \to \mathbb{Z}\}$$

Definition 1.2 (Semantics of Basic Expressions)**.** For basic expressions $e \in \mathsf{Exp}$ the concrete semantics $(\!|\cdot|\!) : \mathsf{Exp} \to \mathsf{Env} \to \mathsf{Env} \cup \{\bot\}$ is inductively defined as follows:

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

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 terminal states produced by executing the program from X.

Definition 1.3 (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 mapping

$$\langle \cdot \rangle : \text{Imp} \to \mathbb{C} \to \mathbb{C}$$

which maps each program $\mathsf{C} \in \text{Imp}$ to a total mapping over the complete lattice $\mathbb{C}$,

$$\langle C \rangle : \mathbb{C} \to \mathbb{C}$$

inductively defined as follows: given $X \in 2^{\mathsf{Env}}$

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

This concrete semantics is additive, meaning that the Kleene star (C^*) and the fixpoint ($\mathsf{fix}(C)$) have the same concrete semantics $\langle \mathsf{C}^* \rangle = \langle \mathsf{fix}(\mathsf{C}) \rangle$. This will not be the case for the abstract semantics (cf. example 2.1), 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 latter proofs we'll only explore the case C^* since it captures also $\mathsf{fix}(C)$.

Along with the collecting semantics we also define a one step transition relation.

Definition 1.4 (Program State)**.** Program states are tuples of programs and program environments:

$$\mathsf{State} \triangleq \mathrm{Imp} \times \mathsf{Env}$$

Definition 1.5 (Small step semantics)**.** The small step transition relation $\rightarrow : \mathsf{State} \times (\mathsf{State} \cup \mathsf{Env})$ is a small step semantics for the Imp language. It is defined based on the following rules

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

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

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

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

Notation 1.1. We write $\langle \mathsf{C}, \rho\rangle \rightarrow^* s$ where $s \in \mathsf{State} \cup \mathsf{Env}$ meaning

$$\exists k \in \mathbb{N} \mid \underbrace{\langle \mathsf{C}, \rho\rangle \rightarrow \langle \mathsf{C}', \rho'\rangle \rightarrow \cdots \rightarrow s}_{k \text{ transitions}}$$

Lemma 1.1 (Collecting and small step link)**.** *For any $C \in Imp, X \in 2^{\mathsf{Env}}$*

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

Therefore $\langle \mathsf{C}\rangle X = \varnothing \iff \nexists \rho' \in \mathsf{Env}, \rho \in X$ s.t. $\langle \mathsf{C}, \rho\rangle \rightarrow^* \rho'$.

Proof. by induction on C:

Base case:
$C \equiv e$: $\langle \mathsf{e}\rangle X = \{(\!|e|\!)\rho \mid \rho \in X \wedge (\!|e|\!)\rho \neq \bot\}$, $\forall \rho \in X. \langle \mathsf{e}, \rho\rangle \rightarrow (\!|e|\!)\rho$ if $(\!|e|\!)\rho \neq \bot$ because of the expr rule

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

Inductive cases:

1. $C \equiv C_1 + C_2$: $\langle \mathsf{C}_1 + \mathsf{C}_2\rangle X = \langle \mathsf{C}_1\rangle X \cup \langle \mathsf{C}_2\rangle X$, $\forall \rho \in X. \langle \mathsf{C}_1 + \mathsf{C}_2, \rho\rangle \rightarrow \langle \mathsf{C}_1, \rho\rangle \vee \langle \mathsf{C}_1 + \mathsf{C}_2, \rho\rangle \rightarrow \langle \mathsf{C}_2, \rho\rangle$ respectively according to rules sum_1 and sum_2. By inductive hypothesis

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

   Therefore

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

2. $C \equiv C_1; C_2$: $\langle \mathsf{C}_1; \mathsf{C}_2\rangle X = \langle \mathsf{C}_2\rangle(\langle \mathsf{C}_1\rangle X)$. By inductive hp $\langle \mathsf{C}_1\rangle X = \{\rho' \in \mathsf{Env} \mid \rho \in X, \langle \mathsf{C}_1, \rho\rangle \rightarrow^* \rho'\} = Y$, by inductive hp again $\langle \mathsf{C}_2\rangle Y = \{\rho' \in \mathsf{Env} \mid \rho \in Y, \langle \mathsf{C}_2, \rho\rangle \rightarrow^* \rho'\}$. Therefore

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

3. $C \equiv C^*$: $\langle \mathsf{C}^* \rangle X = \cup_{i \in \mathbb{N}} \langle \mathsf{C} \rangle^i X$

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

□

We can notice that $\langle \mathsf{C} \rangle X = \varnothing \iff \nexists \rho' \in \mathsf{Env}, \rho \in X \mid \langle \mathsf{C}, \rho \rangle \to^* \rho'$.

1.3 Transition system

With the set of states State, the set of environments Env and the small operational semantics $\to$ we define a transition system:

Definition 1.6 (Transition system)**.**

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

is a transition system for the language Imp, where

- $\mathsf{State} \cup \mathsf{Env}$ is the set of configurations in the system;
- Env is the set of terminal states;
- $\to$ is the small step semantics defined in definition 1.5, which describes the transition relations in the system.

Definition 1.7 (Paths)**.** Let $(\mathsf{State} \cup \mathsf{Env})^\infty \triangleq (\mathsf{State} \cup \mathsf{Env})^+ \cup (\mathsf{State} \cup \mathsf{Env})^\omega$ be the set of all infinitary sequences of states and environments (both finite and infinite). Then the set of *paths* in the transition system is

$$\mathsf{Path}^\infty \triangleq \{\tau \in (\mathsf{State} \cup \mathsf{Env})^\infty \mid \forall i \in [1, |\tau|).\tau_i \to \tau_{i+1}\}$$

Definition 1.8. Given $\mathsf{C} \in \text{Imp}, \rho \in \mathsf{Env}$ the *paths* in the transition system starting from $\langle \mathsf{C}, \rho \rangle$ are

$$\mathsf{C}_\rho \triangleq \{\tau \in \mathsf{Path}^\infty \mid \tau_0 = \langle \mathsf{C}, \rho \rangle\}$$

Definition 1.9 (Reductions)**.** Let Imp^* denotes the set whose elements are finite, eventually empty, ordered sequences of statements in Imp where the empty sequence is denoted by $[\,]$. Moreover let $\circ : \text{Imp}^* \times \text{Imp}^* \to \text{Imp}^*$ be the list concatenation. The reduction function $\mathsf{red} : \text{Imp} \to \text{Imp}^*$ is recursively defined by the following clauses:

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

Where the operator $\star : \text{Imp}^* \times \text{Imp} \to \text{Imp}^*$ is defined by

$$\begin{aligned}
[\,] \star \mathsf{C} &\triangleq [\,] \\
[\ \mathsf{C}_1, \dots, \mathsf{C}_\mathsf{k}\ ] \star \mathsf{C} &\triangleq [\ \mathsf{C}_1; \mathsf{C}, \dots, \mathsf{C}_\mathsf{k}; \mathsf{C}\ ]
\end{aligned}$$

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

1.4 Functions in Imp

Since we usually deal with a finite number of free variables in our programs, we can without loss of generality refer to (input) variables as $\mathtt{x}_n$ with $n \in \mathbb{N}$. Therefore the collections of states $X \in 2^{\mathsf{Env}}$ will look like

$$[\mathtt{x}_1 \mapsto v_1, \mathtt{x}_2 \mapsto v_2, \ldots, \mathtt{x}_n \mapsto v_n, \mathtt{y} \mapsto v_y, \mathtt{z} \mapsto v_z, \ldots].$$

Observe that since we're interested in finite programs, it makes sense to consider only finite collections of free variables.

Notation 1.2 (Partial termination)**.** Let $\rho = [\mathtt{x}_1 \mapsto a_1, \mathtt{x}_2 \mapsto a_2, \ldots, \mathtt{x}_k \mapsto a_k]$. We say that a program C *partially halts* on some state ρ when there's at least one path of finite length in the transition system C_ρ ending up in some state ρ':

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

Dually

$$\mathsf{C}_\rho \upuparrows \iff \neg\, \mathsf{C}_\rho \downarrow$$

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

Example 1.3 shows a program that partially halts, while example 1.2 shows a program that always loops.

Notation 1.3 (Universal termination)**.** Let $\rho = [\mathtt{x}_1 \mapsto a_1, \mathtt{x}_2 \mapsto a_2, \ldots, \mathtt{x}_k \mapsto a_k]$. We say that a program C *partially loops* on some state ρ when there's at least one path of infinite length in the transition system C_ρ:

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

Dually

$$\mathsf{C}_\rho \downdownarrows \iff \neg\, \mathsf{C}_\rho \uparrow$$

a program *universally halts* iff there's no infinite path in the transition systems.

Notice that the absence of infinite paths implies that C_ρ is finite. Example 1.3 shows a program that partially loops, while example 1.1 shows a program that universally halts.

Example 1.1. Consider the program

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

always halts, since $\forall \rho \in \mathsf{Env}, \rho \neq \bot$ builds the transition system

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

according to the expr rule in definition 1.5. Therefore $(\mathtt{x} := 0)_\rho \downdownarrows \forall \rho \in \mathsf{Env} \setminus \{\bot\}$.

Example 1.2. Consider the program P

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

The program never haltsn $\forall \rho \in \mathsf{Env}$ s.t. $\rho(\mathtt{x}) \geq 0$. In fact in these cases it builds the transition system in figure 1.1, where the infinite path

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

$$\cdots \to^* \langle (\mathtt{x} \geq 0; \mathtt{x{+}{+}})^*; x < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + k] \rangle \to^* \ldots$$

is always present.

$$\begin{tikzcd}
\langle (\mathtt{x} \geq 0; \mathtt{x++})^*; \mathtt{x} < 0, \rho\rangle \arrow[r] \arrow[d] & \langle \mathtt{x} < 0, \rho\rangle \not\to \\
\langle \mathtt{x} \geq; 0\mathtt{x++}; (\mathtt{x} \geq 0; \mathtt{x++})^*; \mathtt{x} < 0, \rho\rangle \arrow[d, "*"] & \\
\langle (\mathtt{x} \geq 0; \mathtt{x++})^*; \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + 1]\rangle \arrow[r] \arrow[d, "*"] & \langle \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + 1]\rangle \not\to \\
\langle (\mathtt{x} \geq 0; \mathtt{x++})^*; \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + k]\rangle \arrow[r] & \langle \mathtt{x} < 0, \rho[\mathtt{x} \mapsto \rho(\mathtt{x}) + k]\rangle \not\to
\end{tikzcd}$$

Figure 1.1: Transition system of $(\mathtt{x} \geq 0; \mathtt{x++})^*; \mathtt{x} < 0$

Example 1.3. Consider the program

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

it partially halts $((\mathtt{x++})^*{}_\rho \downarrow)$, 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 $((\mathtt{x++})^*{}_\rho \uparrow)$. In fact we can build the infinite path

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

Notation 1.4 (Program output)**.** Let $\mathsf{Env} \ni \rho = [\mathtt{x}_1 \mapsto a_1, \dots, \mathtt{x}_n \mapsto a_n]$. We say

$$\mathsf{C}_\rho \downdownarrows b \iff \forall \rho' \mid \langle \mathsf{C}, \rho\rangle \to^* \rho' \quad \rho'(\mathtt{y}) = b$$
$$\mathsf{C}_\rho \downarrow b \iff \exists \rho' \mid \langle \mathsf{C}, \rho\rangle \to^* \rho' \quad \rho'(\mathtt{y}) = b$$

Definition 1.10 (Imp computability)**.** let $f : \mathbb{N}^k \to \mathbb{N}$ be a function. f is Imp computable if

$$\exists \mathsf{C} \in \text{Imp} \mid \forall (a_1, \dots, a_k) \in \mathbb{N}^k \wedge b \in \mathbb{N}$$
$$\mathsf{C}_\rho \downdownarrows b \iff (a_1, \dots, a_k) \in \mathit{dom}(f) \wedge f(a_1, \dots, a_k) = b$$

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

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 it contains the zero, successor and projection functions and it is closed under composition, primitive recursion and unbounded minimalization.

Lemma 1.2 (Imp functions richness)**.** *The class of Imp-computable function is rich.*

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

- The zero function:

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

  is Imp-computable:

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

- The successor function

  $$s : \mathbb{N} \to \mathbb{N}$$
  $$x_1 \mapsto x_1 + 1$$

  is Imp-computable:

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

- The projection function

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

  is Imp-computable:

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

We'll then prove that it is closed under composition, primitive recursion and unbounded minimalization.

Lemma 1.3. *let $f : \mathbb{N}^k \to \mathbb{N}$, $g_1, \ldots, g_k : \mathbb{N}^n \to \mathbb{N}$ and consider the composition*

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

h is Imp-computable.

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

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

Which is exactly h. Therefore Imp is closed under generalised composition. □

Lemma 1.4. *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 $H(\vec{x}, x_{k+1})$:

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

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

Lemma 1.5. *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*

$$h(\vec{x}) = \mu y. f(\vec{x}, y) = \begin{cases} \textit{least } z \textit{ s.t.} & \begin{cases} f(\vec{x}, z) = 0 \\ f(\vec{x}, z) \downarrow \quad f(\vec{x}, z') \neq 0 \quad \forall z < z' \end{cases} \\ \uparrow & \textit{otherwise} \end{cases} \tag{1.1}$$

is Imp-computable.

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

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

with $\rho = [\mathtt{x}_1 \mapsto a_1, \dots, \mathtt{x}_k \mapsto a_k]$. From this we get a couple of facts that derive from well known computability results:

- $\langle \mathsf{C} \rangle X = \varnothing$ (i.e., $\mathsf{C}_\rho \upuparrows$) is undecidable. 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]);
- dually, $\langle \mathsf{C} \rangle X \neq \varnothing$ (i.e., $\mathsf{C}_\rho \downarrow$) is undecidable since is the negation of the latter statement;

1.5 Deciding invariant finiteness

Lemma 1.6. *If $\mathsf{D} \in Imp_{\neq\star}$, and $X \in 2^{env}$ is finite, then*

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

(ii). $\forall \rho \in X \; \mathsf{D}_\rho \downdownarrows$

(iii). $|\mathsf{D}_\rho| < \infty$ *for all* $\rho \in X$.

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 paths, and therefore $\mathsf{e}_\rho \downdownarrows$;

(iii). Notice that $\mathsf{e}_\rho = \{ \tau \in \mathsf{Path}^\infty \mid \tau_0 = \langle \mathsf{e}, \rho \rangle \}$ for all $\rho \in X$, therefore $|\mathsf{e}_\rho| = |X| < \infty$ because of (i).

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're 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 $\forall \rho \in X \; \mathsf{D}_{1\rho} \downdownarrows$ and $\mathsf{D}_{2\rho} \downdownarrows$. Notice that $\mathsf{D}_1 + \mathsf{D}_{2\rho} = \{ \mathsf{D}_1 + \mathsf{D}_2 \circ \tau \mid \tau \in \mathsf{D}_{1\rho} \} \cup \{ \mathsf{D}_1 + \mathsf{D}_2 \circ \tau \mid \tau \in \mathsf{D}_{2\rho} \}$ where $\circ : \text{Imp} \times \mathsf{Path}^\infty \to \mathsf{Path}^\infty$ is the appending operator. Since we're just appending $\langle \mathsf{D}_1 + \mathsf{D}_2, \rho \rangle$ in each path of the transition systems of D_1 and D_2, by inductive hypothesis $\mathsf{D}_1 + \mathsf{D}_{2\rho} \downdownarrows$ for all $\rho \in X$.

(iii). For the latter argument, since both ${\mathsf{D}_1}_\rho$ and ${\mathsf{D}_2}_\rho$ are finite and composed of finite paths $\left|(\mathsf{D}_1 + \mathsf{D}_2)_\rho\right| < \infty$.

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 $\forall \rho \in X$ ${\mathsf{D}_1}_\rho \Downarrow$ and again $\forall \rho \in Y$ ${\mathsf{D}_2}_\rho \Downarrow$;

(iii).

□

Lemma 1.7. *Given* $\mathsf{D} \in \mathit{Imp}_{\neq\star}$*, and* $\{\rho\} = X \in 2^{env}$*, the predicate "*$\langle \mathsf{D}^* \rangle X$ *is finite" is undecidable.*

Proof. Suppose we can decide $\langle \mathsf{D}^* \rangle X$ is finite. We show that we in that case we would be also able to decide wether ${\mathsf{D}^*}_\rho \Downarrow$ for some $\rho \in X$, which is undecidable.

- In case $\langle \mathsf{D}^* \rangle X$ is infinite, then it must be that $\forall k \in \mathbb{N}$ $\langle \mathsf{D} \rangle^{k+1} X \nsubseteq \cup_{i=0}^{k} \langle \mathsf{D} \rangle^i X$, otherwise we would reach a fixpoint and $\langle \mathsf{D}^* \rangle X$ would be finite. Since each application of D must create an unempty set of new environments we can build the inductive sequence of sets of environments

$$Y_0 = X$$
$$Y_{k+1} = (\langle \mathsf{D} \rangle Y_k) \setminus Y_k$$

  where $\forall \rho' \in Y_{k+1} \exists \rho \in Y_k \mid \rho' \in \langle \mathsf{D} \rangle \{\rho\}$ by definition. By lemma 1.1 $\rho' \in \{\rho'' \mid \langle \mathsf{D}, \rho \rangle \to^* \rho''\}$. This means that there must be at least one $\rho_1 \in X$ that produces an infinite path

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

  which produces new environments at each application of D: $\rho_1, \rho_2, \dots \mid \forall i, j \in \mathbb{N}$ $\rho_i \neq \rho_j$ and therefore ${\mathsf{D}^*}_{\rho_1} \uparrow$ which means that ${\mathsf{D}^*}_{\rho_1} \Downarrow$ is false.

- In case $\langle \mathsf{D}^* \rangle X$ is finite, we can notice that for all the $\rho \in X$, the states in D_ρ are

$$\mathsf{red}(\mathsf{D}^*) \times Y$$

  with $Y \subseteq \{\rho' \mid \langle \mathsf{D}, \rho \rangle \to^* \langle \mathsf{D}', \rho' \rangle \vee \langle \mathsf{D}, \rho \rangle \to^* \rho'\}$ with $\mathsf{D}' \in \mathsf{red}(\mathsf{D})$. Notice that the reductions of some commanad C are always finite, while Y is also finite: $\mathsf{D} \in \mathrm{Imp}_{\neq\star}$, therefore by lemma 1.6 $\forall \rho \in \mathsf{Env}$ $\mathsf{D}_\rho \Downarrow$ and $|\mathsf{D}_\rho| < \infty$. Since D produces a finite amount of finite paths, there is a finite amount of states, and therefore total termination reduces to the presence of some cycle. In fact if

  - $\exists \rho \in X \mid \langle \mathsf{D}^*, \rho \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle$ for some $\rho' \in \langle \mathsf{D}^* \rangle X$. In this case the semantics would still be finite, but there would be an infinite path

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

    that would imply that $\mathsf{D}_\rho \uparrow$, and therefore $\mathsf{D}_\rho \Downarrow$ would be false.

  - otherwise $\nexists \rho \in X \mid \langle \mathsf{D}^*, \rho \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle \to^* \langle \mathsf{D}^*, \rho' \rangle$ for some $\rho' \in \langle \mathsf{D}^* \rangle X$, but since there is a finite amount of states each path in D_ρ is finite, and therefore $\mathsf{D}_\rho \Downarrow$.

□

Chapter 2

Intervals

2.1 Interval Analysis

We define *interval analysis* of the above language Imp in a standard way, taking the best correct approximations (bca) for the basic expressions in Exp.

Definition 2.1 (Integer intervals)**.** We call

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

set of integer intervals.

Definition 2.2 (Concretization map)**.** We define the *concretization map* $\gamma : Int \to 2^{\mathbb{Z}}$ as

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

Observation 2.1. $\langle Int, \sqsubseteq \rangle$ is a complete lattice where for all $I, J \in Int$, $I \sqsubseteq J$ iff $\gamma(I) \subseteq \gamma(J)$.

Definition 2.3 (Abstract integer domain)**.** Let $Int_* \triangleq Int \setminus \{\bot^\sharp\}$. The abstract domain $\mathbb{A}$ for program analysis is the variable-wise lifting of Int:

$$\mathbb{A} \triangleq (Var \to Int_*) \cup \{\bot^\sharp\}$$

where the intervals for a given variable are always nonempty, while $\bot$ represents the empty set of environments. Thus, the corresponding concretization is defined as follows:

Definition 2.4 (Interval concretization)**.** We define the *concretization map* for the abstract domain $\mathbb{A}$ $\gamma_{Int} : \mathbb{A} \to 2^{\mathsf{Env}}$ as

$$\gamma_{Int}(\bot) \triangleq \varnothing$$
$$\forall \eta \neq \bot \quad \gamma_{Int}(\eta) \triangleq \{\rho \in \mathsf{Env} \mid \forall x \in Var\ \rho(x) \in \gamma(\eta(x))\}$$

Observation 2.2. If we consider the ordering $\sqsubseteq$ on $\mathbb{A}$ s.t.

$$\forall \eta, \vartheta \in \mathbb{A} \quad \eta \sqsubseteq \vartheta \iff \gamma_{Int}(\eta) \subseteq \gamma_{Int}(\vartheta)$$

then $\langle \mathbb{A}, \sqsubseteq \rangle$ is a complete lattice.

Definition 2.5 (Interval abstraction)**.** We define the *abstraction map* of some numerical set $X \subseteq \mathbb{Z}$ into the abstract domain $\mathbb{A}$: $\alpha_{Int} : 2^{\mathbb{Z}} \to \mathbb{A}$ as

$$\alpha_{Int}(X) \triangleq \begin{cases} \bot^\sharp & \text{if } X = \varnothing \\ [\min X, \max X] & \text{otherwise} \end{cases}$$

Observe that since we have both a concretization map γ_{Int} and an abstraction map α_{Int} we've built the Galois Connection

$$\langle \gamma_{Int}, \mathbb{C}, \mathbb{A}, \alpha_{Int} \rangle$$

between the concrete domain $\mathbb{C}$ and the abstract domain $\mathbb{A}$, resulting

Definition 2.6 (Abstract operations)**.** We define sound abstract operations in the $\mathbb{A}$ domain:

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

And sound abstract aritmetical operations:

$$\begin{aligned}
-^\sharp [a,b] &\triangleq [-b,-a] \\
[a,b] +^\sharp [c,d] &\triangleq [a+c, b+d] \\
[a,b] -^\sharp [c,d] &\triangleq [a-c, b-d] \\
[a,b] \times^\sharp [c,d] &\triangleq [\min(ac,ad,bc,bd), \max(ac,ad,bc,bd)]
\end{aligned}$$

Definition 2.7 (Interval sharpening)**.** For a nonempty interval $[a,b] \in \mathit{Int}$ and $c \in \mathbb{Z}$, we define two operations raising $\uparrow$ the lower bound to c and lowering $\downarrow$ the upper bound to c, respectively:

$$\begin{aligned}
[a,b] \uparrow c &\triangleq \begin{cases} [\max\{a,c\}, b] & \text{if } c \leqslant b \\ \bot & \text{if } c > b \end{cases} \\
[a,b] \downarrow c &\triangleq \begin{cases} [a, \min\{b,c\}] & \text{if } c \geq a \\ \bot & \text{if } c < a \end{cases}
\end{aligned}$$

Observe that $\max([a,b] \downarrow c) \leqslant c$ always holds. □

Definition 2.8 (Interval addition and subtraction)**.** For a nonempty interval $[a,b] \in \mathit{Int}$ and $c \in \mathbb{Z}$ define $[a,b] \pm c \triangleq [a \pm c, b \pm c]$ (recall that $\pm\infty + c = \pm\infty - c = \pm\infty$). □

Observe that for every interval $[a,b] \in \mathit{Int}$ and $c \in \mathbb{Z}$

$$\max([a,b] \uparrow c) \leqslant b \qquad \text{and} \qquad \max([a,b] \downarrow c) \leqslant c$$

that trivially holds by defining $\max(\bot) \triangleq 0$ (i.e., 0 is the maximum of an empty interval).

The *interval semantics* of Imp is defined as the strict (i.e., preserving $\bot$) extension of the following function $[\![\cdot]\!] : \exp \cup \mathrm{Imp} \to \mathbb{A} \to \mathbb{A}$. For all $\eta : \mathit{Var} \to \mathit{Int}_*$,

$$
\begin{aligned}
[\![\mathtt{x} \in S]\!]\eta &\triangleq \begin{cases} \eta[\mathtt{x} \mapsto \eta(\mathtt{x}) \sqcap \alpha_{Int}(S)] & \text{if } \eta(\mathtt{x}) \sqcap \alpha_{Int}(S) \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathtt{x} \in [a,b]]\!]\eta &\triangleq \begin{cases} \eta[\mathtt{x} \mapsto \eta(\mathtt{x}) \sqcap [a,b]] & \text{if } \eta(\mathtt{x}) \sqcap [a,b] \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathtt{x} \leqslant k]\!]\eta &\triangleq \begin{cases} \eta[\mathtt{x} \mapsto \eta(\mathtt{x}) \downarrow k] & \text{if } \eta(\mathtt{x}) \downarrow k \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathtt{x} > k]\!]\eta &\triangleq \begin{cases} \eta[\mathtt{x} \mapsto \eta(\mathtt{x}) \uparrow k+1] & \text{if } \eta(\mathtt{x}) \downarrow k \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathsf{true}]\!]\eta &\triangleq \eta \\
[\![\mathsf{false}]\!]\eta &\triangleq \bot \\
[\![\mathtt{x} := k]\!]\eta &\triangleq \eta[\mathtt{x} \mapsto [k,k]] \\
[\![\mathtt{x} := \mathtt{y} + k]\!]\eta &\triangleq \eta[\mathtt{x} \mapsto \eta(\mathtt{y}) + k] \\
[\![\mathtt{x} := \mathtt{x} - k]\!]\eta &\triangleq \eta[\mathtt{x} \mapsto \eta(\mathtt{y}) - k] \\
[\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta &\triangleq [\![\mathsf{C}_1]\!]\eta \sqcup [\![\mathsf{C}_2]\!]\eta \\
[\![\mathsf{C}_1 ; \mathsf{C}_2]\!]\eta &\triangleq [\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta) \\
[\![\mathsf{C}^*]\!]\eta &\triangleq \textstyle\bigsqcup_{i\in\mathbb{N}} [\![\mathsf{C}]\!]^i(\eta) \\
[\![\mathsf{fix}(\mathsf{C})]\!]\eta &\triangleq \mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu)
\end{aligned}
$$

The semantics is well-defined, because of the following lemma:

Lemma 2.1. *for all* $\mathsf{C} \in \mathit{Imp}$,

$$[\![\mathsf{C}]\!] : \mathbb{A} \to \mathbb{A}$$

is monotone.

Proof. What we have to proof is that given $\eta, \vartheta \in \mathbb{A}$, with $\eta \sqsubseteq \vartheta$ then $\forall C \in \text{Imp}\ [\![C]\!]\eta \sqsubseteq [\![C]\!]\vartheta$. We'll work by induction on the grammar of C:
Base cases:
We avoid cases where $\eta = \bot$ and $[\![C]\!]\eta = \bot$ as $\forall \vartheta \in \mathbb{A}\ \bot \sqsubseteq \vartheta$ and it becomes trivially true.

- $\mathsf{C} \equiv \mathtt{x} \in S$. Then

$$
\begin{aligned}
[\![\mathtt{x} \in S]\!]\eta &= \eta[x \mapsto \eta(x) \sqcap \mathit{Int}(S)] \\
[\![\mathtt{x} \in S]\!]\vartheta &= \vartheta[x \mapsto \vartheta(x) \sqcap \mathit{Int}(S)]
\end{aligned}
$$

  Since $\eta(x) \sqcap \mathit{Int}(S) \neq \bot$ and $\eta \sqsubseteq \vartheta$, then $\vartheta(x) \sqcap \mathit{Int}(S) \neq \bot$. We can see that

$$
\begin{aligned}
\eta \sqsubseteq \vartheta &\iff \gamma(\eta) \subseteq \gamma(\vartheta) \\
&\iff \{x \in \mathbb{Z} \mid x \in \eta(\mathtt{x})\} \subseteq \{x \in \mathbb{Z} \mid x \in \vartheta(\mathtt{x})\} \\
&\iff \{x \in \mathbb{Z} \mid x \in \eta(\mathtt{x})\} \cap \{x \in \mathbb{Z} \mid x \in \mathit{Int}(S)\} \subseteq \{x \in \mathbb{Z} \mid x \in \vartheta(\mathtt{x})\} \cap \{x \in \mathbb{Z} \mid x \in \mathit{Int}(S)\} \\
&\iff \{x \in \mathbb{Z} \mid x \in \eta(\mathtt{x}) \wedge x \in \mathit{Int}(S)\} \subseteq \{x \in \mathbb{Z} \mid x \in \vartheta(\mathtt{x}) \wedge x \in \mathit{Int}(S)\} \\
&\iff \{x \in \mathbb{Z} \mid x \in \eta(\mathtt{x}) \sqcap \mathit{Int}(S)\} \subseteq \{x \in \mathbb{Z} \mid x \in \vartheta(\mathtt{x}) \sqcap \mathit{Int}(S)\} \\
&\iff \gamma_{\mathit{Int}}(\eta[x \mapsto \eta(\mathtt{x}) \sqcap \mathit{Int}(S)](\mathtt{x})) \subseteq \gamma_{\mathit{Int}}(\vartheta[x \mapsto \vartheta(\mathtt{x}) \sqcap \mathit{Int}(S)](\mathtt{x})) \\
&\iff [\![\mathtt{x} \in S]\!]\eta \sqsubseteq [\![\mathtt{x} \in S]\!]\vartheta
\end{aligned}
$$

- for the base cases $\mathtt{x} \in [a,b], \mathtt{x} \leqslant k, \mathtt{x} > k$ we can use the same proceedings;
- $\mathsf{C} \equiv \mathsf{true}$. Then $[\![\mathsf{true}]\!]\eta = \eta \sqsubseteq \vartheta = [\![\mathsf{true}]\!]\vartheta$;

- $\mathsf{C} \equiv \mathsf{false}$. Then $[\![\mathsf{false}]\!]\eta = \bot \sqsubseteq \bot = [\![\mathsf{false}]\!]\vartheta$;
- $\mathsf{C} \equiv \mathtt{x} := k$. Then

$$\begin{aligned}\eta \sqsubseteq \vartheta &\iff \gamma_{Int}(\eta) \subseteq \gamma_{Int}(\vartheta)\\ &\iff \{\rho \in \mathsf{Env} \mid \forall \mathtt{x} \in \mathit{Var} \rho(\mathtt{x}) \in \gamma(\eta(\mathtt{x}))\} \subseteq \{\rho \in \mathsf{Env} \mid \forall \mathtt{x} \in \mathit{Var} \rho(\mathtt{x}) \in \gamma(\vartheta(\mathtt{x}))\}\\ &\iff \forall \mathtt{x} \in \mathit{Var}, \rho \in \mathsf{Env} \quad \rho(\mathtt{x}) \in \gamma(\eta(\mathtt{x})) \Rightarrow \rho(\mathtt{x}) \in \gamma(\vartheta(\mathtt{x}))\end{aligned} \tag{2.1}$$

Notice that

$$[\![\mathtt{x} := k]\!]\eta = \eta[\mathtt{x} \mapsto [k,k]]$$

$$[\![\mathtt{x} := k]\!]\vartheta = \vartheta[\mathtt{x} \mapsto [k,k]]$$

because of equation 2.1 in this case we know that $\forall \mathtt{y} \in \mathit{Var},\ \mathtt{y} \neq \mathtt{x}\ \rho(\mathtt{y}) \in \gamma(\eta(\mathtt{y})) \Rightarrow \rho(\mathtt{y}) \in \gamma(\vartheta(\mathtt{y}))$. For $\mathtt{x}$ it holds that $\rho(\mathtt{x}) \in \gamma([k,k]) \Rightarrow \rho(\mathtt{x}) \in \gamma([k,k])$ and therefore

$$\begin{aligned}&\forall \mathtt{y} \in \mathit{Var}, \rho \in \mathsf{Env} \quad \rho(\mathtt{y}) \in \gamma(\eta[\mathtt{x} \mapsto [k,k]](\mathtt{y})) \Rightarrow \rho(\mathtt{y}) \in \gamma(\vartheta[\mathtt{x} \mapsto [k,k]](\mathtt{y}))\\ &\iff \gamma_{Int}([\![\mathtt{x} := k]\!]\eta) \subseteq \gamma_{Int}([\![\mathtt{x} := k]\!]\vartheta)\\ &\iff [\![\mathtt{x} := k]\!]\eta \sqsubseteq [\![\mathtt{x} := k]\!]\vartheta\end{aligned}$$

- For $\mathsf{C} \equiv \mathtt{x} := \mathtt{y} + k, \mathtt{x} := \mathtt{y} - k$ the procedure is the same.

Recusrsive cases:

- $\mathsf{C} \equiv \mathsf{C}_1 + \mathsf{C}_2$. Then

$$\begin{aligned}[\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta &= [\![\mathsf{C}_1]\!]\eta \sqcup [\![\mathsf{C}_2]\!]\eta\\ &\sqsubseteq [\![\mathsf{C}_1]\!]\vartheta \sqcup [\![\mathsf{C}_2]\!]\vartheta && \text{by inductive hp.}\\ &= [\![\mathsf{C}_1 + \mathsf{C}_2]\!]\vartheta\end{aligned}$$

- $\mathsf{C} \equiv \mathsf{C}_1; \mathsf{C}_2$. Then

$$\begin{aligned}[\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta = {}& [\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta)\\ & \alpha = [\![\mathsf{C}_1]\!]\eta \sqsubseteq [\![\mathsf{C}_1]\!]\vartheta = \beta && \text{by inductive hp.}\\ & [\![\mathsf{C}_2]\!]\alpha \sqsubseteq [\![\mathsf{C}_2]\!]\beta && \text{by inductive hp.}\\ & [\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta) \sqsubseteq [\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\vartheta) && \text{by substitution}\end{aligned}$$

- C^*. Then by inductive hypothesis $\forall i \in \mathbb{N}. [\![\mathsf{C}]\!]^i\eta \sqsubseteq [\![\mathsf{C}]\!]^i\vartheta$, which means

$$[\![\mathsf{C}^*]\!]\eta = \bigsqcup_{i\in\mathbb{N}} [\![\mathsf{C}]\!]^i\eta \sqsubseteq \bigsqcup_{i\in\mathbb{N}} [\![\mathsf{C}]\!]^i\vartheta = [\![\mathsf{C}^*]\!]\vartheta.$$

□

Example 2.1. This is the case, for instance, the following program P rapresents the difference between the Kleene Star and the Fix operator:

```
while x < 8 do
if x = 2 then x := x+6;
x := x-3
if x <= 0 then x:=0
```

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

$$\begin{aligned}\text{Kleene: } &\sqcup\{[3,4],[0,1],[0,0],[0,0],\ldots\} = [0,4]\\ \text{Fix: } &\sqcup\{\perp,[3,4],[0,4],[0,5],[0,5],\ldots\} = [0,5]\end{aligned}$$

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

Lemma 2.2 ($\mathsf{fix(C)}$ **is syntactic sugar**)**.** *For all* η, $[\![\mathsf{fix(C)}]\!]\eta = [\![(\mathsf{true} + \mathsf{C})^*]\!]\eta$.

Proof. Let us first show by induction that

$$\forall i \geq 0.\ (\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!])^{i+1}\perp = (\mathsf{true} \sqcup [\![\mathsf{C}]\!])^{i}\eta \tag{$\sharp$}$$

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

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

Let us also show that:

$$\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu) = \mathrm{lfp}\lambda\mu.(\eta \sqcup \mu \sqcup [\![\mathsf{C}]\!]\mu) \tag{$\diamond$}$$

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

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

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

Then, we have that:

$$\begin{aligned}
[\![\mathsf{fix(C)}]\!]\eta &= \\
\mathrm{lfp}\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu) &= && [\text{By } (\diamond)]\\
\mathrm{lfp}\lambda\mu.(\eta \sqcup \mu \sqcup [\![\mathsf{C}]\!]\mu) &= && [\text{By Knaster-Tarski Theorem}]\\
\bigsqcup_{i\in\mathbb{N}}(\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!])^{i}\perp &= \\
\perp \sqcup \bigsqcup_{i\in\mathbb{N}}(\eta \sqcup \mathsf{true} \sqcup [\![\mathsf{C}]\!])^{i+1}\perp &= && [\text{By } (2.3)]\\
\bigsqcup_{i\in\mathbb{N}}(\mathsf{true} \sqcup [\![\mathsf{C}]\!])^{i}\eta &= \\
[\![(\mathsf{true} + \mathsf{C})^*]\!]\eta. &
\end{aligned}$$

□

Theorem 1 (**Correctness**)**.** *For all* $\mathsf{C} \in$ *Imp and* $\eta \in \mathbb{A}$, $\langle \mathsf{C}\rangle\gamma(\eta) \subseteq \gamma([\![\mathsf{C}]\!]\eta)$ *holds.*

Proof. by induction on $\mathsf{C} \in$ Imp:

Base cases:

- $\mathsf{C} \equiv \mathtt{x} \in S$:
  - $\langle \mathtt{x} \in \mathsf{S}\rangle \gamma_{Int}(\eta) = \{\rho \in \mathsf{Env} \mid \forall y \in \mathit{Var}\ \rho(y) \in \gamma(\eta(y))\} \cap \{\rho \in \mathsf{Env} \mid \rho(\mathtt{x}) \in S\}$
  - $\gamma_{Int}([\![\mathtt{x} \in S]\!]\eta) = \{\rho \in \mathsf{Env} \mid \forall y \in \mathit{Var}\ \rho(y) \in \gamma(\eta(y))\} \cap \{\rho \in \mathsf{Env} \mid \rho(\mathtt{x}) \in \mathit{Int}(S)\}$

  S is just decidable, not directly an interval, therefore in general $S \subseteq \mathit{Int}(S)$, and therefore

  $$\langle \mathtt{x} \in \mathsf{S}\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathtt{x} \in S]\!]\eta);$$

- $\mathsf{C} \equiv \mathtt{x} \in [a, b], \mathtt{x} \leqslant k, \mathtt{x} > k$: is the same as the latter case;

- $\mathsf{C} \equiv \mathsf{true}$: $\langle \mathsf{true}\rangle \gamma_{Int}(\eta) = \gamma_{Int}(\eta)$, $\gamma_{Int}([\![\mathsf{true}]\!]\eta) = \gamma_{Int}(\eta)$, and since $\gamma_{Int}(\eta) \subseteq \gamma_{Int}(\eta)$

  $$\langle \mathsf{true}\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathsf{true}]\!]\eta);$$

- $\mathsf{C} \equiv \mathsf{false}$: $\langle \mathsf{false}\rangle \gamma_{Int}(\eta) = \varnothing$, $\gamma_{Int}([\![\mathsf{false}]\!]\eta) = \varnothing$ and therefore

  $$\langle \mathsf{false}\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathsf{false}]\!]\eta);$$

- $\mathsf{C} \equiv \mathtt{x} := k$ therefore $\langle \mathtt{x} := \mathsf{k}\rangle \gamma_{Int}(\eta) = \{\rho \in \mathsf{Env} \mid \forall y \in \mathit{Var}.y \neq \mathtt{x} \Rightarrow \rho(y) \in \gamma(\eta(y)), \rho(\mathtt{x}) \in \gamma(\eta(\mathtt{x}) + k)\} = \gamma_{Int}([\![\mathtt{x} := k]\!]\eta)$ therefore

  $$\langle \mathtt{x} := \mathsf{k}\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathtt{x} := k]\!]\eta);$$

- $\mathsf{C} \equiv \mathtt{x} := \mathtt{y} + k, \mathtt{x} := \mathtt{y} - k$ is the same as the latter case.

Inductive cases:

- $\mathsf{C} \equiv \mathsf{C}_1 + \mathsf{C}_2$, therefore

  $$\langle \mathsf{C}_1 + \mathsf{C}_2\rangle \gamma_{Int}(\eta) = \langle \mathsf{C}_1\rangle \gamma_{Int}(\eta) \cup \langle \mathsf{C}_2\rangle \gamma_{Int}(\eta)$$

  and

  $$\gamma_{Int}([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta) = \gamma_{Int}([\![\mathsf{C}_1]\!]\eta \sqcup [\![\mathsf{C}_2]\!]\eta) = \gamma_{Int}([\![\mathsf{C}_1]\!]\eta) \cup \gamma_{Int}([\![\mathsf{C}_2]\!]\eta).$$

  By inductive hypothesis both $\langle \mathsf{C}_1\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathsf{C}_1]\!]\eta)$ and $\langle \mathsf{C}_2\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathsf{C}_2]\!]\eta)$, therefore

  $$\langle \mathsf{C}_1 + \mathsf{C}_2\rangle \gamma_{Int}(\eta) \subseteq \gamma_{Int}([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta);$$

- $\mathsf{C} \equiv \mathsf{C}_1; \mathsf{C}_2$, therefore $\langle \mathsf{C}_1; \mathsf{C}_2\rangle \gamma_{Int}(\eta) = \langle \mathsf{C}_2\rangle(\langle \mathsf{C}_1\rangle \gamma_{Int}(\eta))$, while

  $$\gamma_{Int}([\![\mathsf{C}_1; \mathsf{C}_2]\!]\eta) = \gamma_{Int}([\![\mathsf{C}_2]\!]([\![\mathsf{C}_1]\!]\eta)).$$

- $\mathsf{C} \equiv \mathsf{C}^*$, therefore $\langle \mathsf{C}^*\rangle \gamma_{Int}(\eta) = \bigcup_{i \in \mathbb{N}} \langle \mathsf{C}\rangle^i \gamma_{Int}(\eta)$, while $\gamma_{Int}([\![\mathsf{C}^*]\!]\eta) = \gamma_{Int}(\bigsqcup_{i \in \mathbb{N}} [\![\mathsf{C}^*]\!]\eta)$

$\square$

Remark 2.1. 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 nontermination 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, $[\![\mathsf{C}]\!]\eta$ might be infinite on some variable when C includes a either Kleene or fix iteration which does not convernge in finitely many steps.

2.2 Computing the interval semantics

In this section we argue that for the language Imp the interval abstract semantics is computable in finite time without widening.

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 below the semantics maps x and y to $[0, 2]$ and $[6, 8]$ resp., while widening/narrowing to $[0, \infty]$ and $[6, \infty]$

```
x:=0;
y:=0;
while (x<=5) do
   if (y=0) then
      y=y+1;
   endif;
   if (x==0) then
      x:=y+7;
   endif;
done;
end
```

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've seen in section 1.5.

Problem 2.1 (Termination of interval analysis)**.** Given $\mathsf{C} \in$ Imp, $\eta \in \mathbb{A}$, decide: $[\![\mathsf{C}]\!]\eta =^? \top$

First, given a program, we associate each variable with a *single bound*, which captures both both an *upper bound*, for which the rough idea is that, whenever a variable is beyond that bound, the behaviour of the program with respect to that variable becomes stable and an *increment bound* which captures the largest increment or decrement that can affect a variable.

Definition 2.9 (**Program bound**)**.** The *bound* associated with a command $\mathsf{C} \in$ Imp is a natural number, denoted $(\mathsf{C})^{\mathsf{b}} \in \mathbb{N}$, defined inductively as follows:

$$\begin{aligned}
(\mathtt{x} \in S)^{\mathsf{b}} &\triangleq \begin{cases} \min(S) & \text{if } \max(S) = \infty \\ \max(S) & \text{if } \max(S) \in \mathbb{N} \end{cases} \\
(\mathtt{x} \in [a, b])^{\mathsf{b}} &\triangleq \begin{cases} a & \text{if } b = \infty \\ b & \text{if } b \in \mathbb{N} \end{cases} \\
(\mathtt{x} \leqslant k)^{\mathsf{b}} &\triangleq k \\
(\mathtt{x} > k)^{\mathsf{b}} &\triangleq k \\
(\mathsf{true})^{\mathsf{b}} &\triangleq 0 \\
(\mathsf{false})^{\mathsf{b}} &\triangleq 0 \\
(\mathtt{x} := k)^{\mathsf{b}} &\triangleq k \\
(\mathtt{x} := \mathtt{y} + k)^{\mathsf{b}} &\triangleq k \\
(\mathtt{x} := \mathtt{y} - k)^{\mathsf{b}} &\triangleq k \\
(\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}} &\triangleq (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} \\
(\mathsf{C}_1 ; \mathsf{C}_2)^{\mathsf{b}} &\triangleq (\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} \\
(\mathsf{C}^*)^{\mathsf{b}} &\triangleq (|\mathit{vars}(\mathsf{C})| + 1)(\mathsf{C})^{\mathsf{b}}
\end{aligned}$$

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

Definition 2.10 (Bound Environment). A bound environment (benv for short) is a total function $b : \mathit{Var} \to \mathbb{N}$. We define $\mathsf{bEnv} \triangleq \{b \mid b : \mathit{Var} \to \mathbb{N}\}$. Each command $\mathsf{C} \in$ Imp induces a benv transformer $[C]^{\mathsf{b}} : \mathsf{bEnv} \to \mathsf{bEnv}$, which is defined inductively as follows:

$$\begin{aligned}
[\mathtt{x} \in S]^{\mathsf{b}} b &\triangleq \begin{cases} b[\mathtt{x} \mapsto b(\mathtt{x}) + \min(S)] & \text{if } \max(S) = \infty \\ b[\mathtt{x} \mapsto b(\mathtt{x}) + \max(S)] & \text{if } \max(S) \in \mathbb{N} \end{cases} \\
[\mathtt{x} := k]^{\mathsf{b}} b &\triangleq b[\mathtt{x} \mapsto b(\mathtt{x}) + k] \\
[\mathtt{x} := \mathtt{y} + k]^{\mathsf{b}} b &\triangleq b[\mathtt{x} \mapsto b(\mathtt{x}) + b(\mathtt{y}) + k] \\
[\mathtt{x} := \mathtt{y} - k]^{\mathsf{b}} b &\triangleq b[\mathtt{x} \mapsto b(\mathtt{x}) + b(\mathtt{y}) - k] \\
[\mathsf{C}_1 + \mathsf{C}_2]^{\mathsf{b}} b &\triangleq \lambda \mathtt{x}.([\mathsf{C}_1]^{\mathsf{b}} b)(\mathtt{x}) + ([\mathsf{C}_2]^{\mathsf{b}} b)(\mathtt{x}) \\
[\mathsf{C}_1 ; \mathsf{C}_2]^{\mathsf{b}} b &\triangleq \lambda \mathtt{x}.([\mathsf{C}_1]^{\mathsf{b}} b)(\mathtt{x}) + ([\mathsf{C}_2]^{\mathsf{b}} b)(\mathtt{x}) \\
[\mathsf{C}^*]^{\mathsf{b}} b &\triangleq \lambda \mathtt{x}.(|\mathit{vars}(\mathsf{C})| + 1)([\mathsf{C}]^{\mathsf{b}} b)(\mathtt{x})
\end{aligned}$$

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

Lemma 2.3. *For all* $\mathsf{C} \in \mathit{Imp}$, $(\mathsf{C})^{\mathsf{b}} = \sum_{\mathtt{x} \in \mathit{vars}(\mathsf{C})} ([\mathsf{C}]^{\mathsf{b}} b_0)(\mathtt{x})$, *with* $b_0 \triangleq \lambda x.0$.

Proof. By induction on $\mathsf{C} \in$ Imp.

Base cases:

$(\mathtt{x} \in S)$:

$$\begin{aligned}
(\mathtt{x} \in S)^{\mathsf{b}} &= \begin{cases} \min(S) & \text{if } \max(S) = \infty \\ \max(S) & \text{otherwise} \end{cases} \\
[\mathtt{x} \in S]^{\mathsf{b}} b_0 &= \begin{cases} b_0[\mathtt{x} \mapsto 0 + \min(S)] & \text{if } \max(S) = \infty \\ b_0[\mathtt{x} \mapsto 0 + \max(S)] & \text{if } \max(S) \in \mathbb{N} \end{cases}
\end{aligned}$$

and since $\mathtt{x}$ is the only variable in $\mathit{vars}(\mathtt{x} \in S)$, $(\mathtt{x} \in S)^{\mathsf{b}} = [\mathtt{x} \in S]^{\mathsf{b}} b_0(\mathtt{x})$

$(\mathtt{x} \in [a, b]), (\mathtt{x} \leqslant k), (\mathtt{x} > k)$ are the same as the latter case;

$(\mathsf{true}), (\mathsf{false})$: notice that $\mathit{vars}(\mathsf{true}) = \mathit{vars}(\mathsf{false}) = \varnothing$;

$(\mathtt{x} := k)$: just notice that $(\mathtt{x} := k)^{\mathsf{b}} = k = b_0(\mathtt{x}) + k = b_0[\mathtt{x} \mapsto b_0 + k] = [\mathtt{x} := k]^{\mathsf{b}} b_0$ and $\mathtt{x}$ is the only variable in $\mathtt{x} := k$.

$(\mathtt{x} := \mathtt{x} + k), (\mathtt{x} := \mathtt{x} - k)$ are analogous to the latter case.

Inductive cases:

$(\mathsf{C}_1 + \mathsf{C}_2)$

$$
\begin{aligned}
(\mathsf{C}_1 + \mathsf{C}_2)^{\mathsf{b}} &= \\
(\mathsf{C}_1)^{\mathsf{b}} + (\mathsf{C}_2)^{\mathsf{b}} &= \qquad \text{by inductive hypothesis} \\
\sum_{\mathtt{x} \in vars(\mathsf{C}_1)} ([\mathsf{C}]^{\mathsf{b}} b_0)(\mathtt{x}) + \sum_{\mathtt{x} \in vars(\mathsf{C}_2)} ([\mathsf{C}]^{\mathsf{b}} b_0)(\mathtt{x}) &= \\
\sum_{\mathtt{x} \in vars(\mathsf{C}_1) \cap vars(\mathsf{C}_2)} ([\mathsf{C}_1]^{\mathsf{b}} b_0)(\mathtt{x}) + ([\mathsf{C}_2]^{\mathsf{b}} b_0)(\mathtt{x}) &+ \\
\sum_{\mathtt{x} \in vars(\mathsf{C}_1) \smallsetminus vars(\mathsf{C}_2)} ([\mathsf{C}_1]^{\mathsf{b}} b_0)(\mathtt{x}) &+ \\
\sum_{\mathtt{x} \in vars(\mathsf{C}_2) \smallsetminus vars(\mathsf{C}_1)} ([\mathsf{C}_2]^{\mathsf{b}} b_0)(\mathtt{x}) &= \\
[\mathsf{C}_1 + \mathsf{C}_2]^{\mathsf{b}} b_0
\end{aligned}
$$

$(\mathsf{C}_1; \mathsf{C}_2)$ identical to $(\mathsf{C}_1 + \mathsf{C}_2)$;

(C^*)

$$
\begin{aligned}
(\mathsf{C}^*)^{\mathsf{b}} &= \\
|vars(C) + 1| (\mathsf{C})^{\mathsf{b}} &= \qquad \text{by inductive hypothesis} \\
|vars(C) + 1| \sum_{\mathtt{x} \in vars(\mathsf{C})} ([\mathsf{C}]^{\mathsf{b}} b_0)(\mathtt{x}) &= \\
\sum_{\mathtt{x} \in vars(\mathsf{C})} |vars(C) + 1| ([\mathsf{C}]^{\mathsf{b}} b_0)(\mathtt{x}) &= \\
[\mathsf{fix}(\mathsf{C})]^{\mathsf{b}} b_0
\end{aligned}
$$

□

We next 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). $\ell \geq 1$

(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 2.4 (**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 $|p| = \ell \geq |X|$. Then, we show our claim by induction on $|p| = \ell \geq |X|$.
($|p| = |X|$): Since the path p includes exactly $|X|+1 = \ell+1$ nodes, there exist indices $0 \leqslant i < j \leqslant \ell$ 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

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

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

Lemma 2.5. *Let* $\mathsf{C} \in \mathit{Imp}$ *and* $\mathtt{y} \in \mathit{Var}$.
For all $\eta \in \mathbb{A}$ *and* $\mathtt{y} \in \mathit{Var}$, *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* $\mathtt{z} \in \mathit{Var}$ *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 $\eta' \in \mathbb{A}$, *if* $\eta' \sqsupseteq \eta$ *then* $\max([\![\mathsf{C}]\!]\eta'\mathtt{y}) \geq \max(\eta'\mathtt{z}) + h$.

Proof. The proof is by structural induction on the command $\mathsf{C} \in \mathrm{Imp}$. We preliminarly observe that we can safely assume $\eta \neq \bot$. In fact, if $\eta = \bot$ then $[\![\mathsf{C}]\!]\bot = \bot$ and thus $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) = 0 \leqslant (\mathsf{C})^{\mathsf{b}}$, against the hypothesis $\max([\![\mathsf{C}]\!]\eta\mathtt{y}) > (\mathsf{C})^{\mathsf{b}}$. Moreover, when quantifying over η' such that $\eta' \sqsupseteq \eta$ in (i), if $\max([\![\mathsf{C}]\!]\eta'\mathtt{y}) = \infty$ holds, then $\max([\![\mathsf{C}]\!]\eta'\mathtt{y}) \geq \max(\eta'\mathtt{z}) + h$ trivially holds, hence we will sometimes silently omit to consider this case.

Case ($\mathtt{x} \in S$)
Take $\eta \in \mathbb{A}$ 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 $\max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) = 0 \leqslant (\mathtt{x} \in S)^{\mathsf{b}}$.
We distinguish two cases:

- If $\mathtt{y} \neq \mathtt{x}$, then for all $\eta' \in \mathbb{A}$ such that $\eta \sqsubseteq \eta'$ it holds $\bot \neq [\![\mathtt{x} \in S]\!]\eta' = \eta'[\mathtt{x} \mapsto \eta(\mathtt{x}) \sqcap \mathit{Int}(S)]$ 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 $\eta(\mathtt{x}) \in \mathit{Int}_*$ and
$$\max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) = \max(\eta(\mathtt{x}) \sqcap \mathit{Int}(S))$$
Note that it cannot be $\max(S) \in \mathbb{N}$. Otherwise, by Definition 2.9, $\max(\eta(\mathtt{x}) \sqcap \mathit{Int}(S)) \leqslant \max(S) = (\mathtt{x} \in S)^{\mathsf{b}}$, violating the assumption $\max([\![\mathtt{x} \in S]\!]\eta\mathtt{y}) > (\mathtt{x} \in S)^{\mathsf{b}}$. Hence, $\max(S) =$

∞ must hold and therefore $\max(\eta(\mathtt{x}) \sqcap \mathit{Int}(S)) = \max(\eta(\mathtt{x})) = \max(\eta(\mathtt{x}))+0$. It is immediate to check that the same holds for all $\eta' \sqsupseteq \eta$, i.e.,

$$\max(\eta'(\mathtt{x}) \sqcap \mathit{Int}(S)) = \max(\eta'(\mathtt{x})) = \max(\eta'(\mathtt{x})) + 0$$

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

Case (true) A consequence of the fact that $\mathsf{true} \equiv x \in \mathbb{N}$.

Case (false) A consequence of the fact that $\mathsf{false} \equiv x \in \varnothing$.

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

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

$$\max(\llbracket \mathtt{x} := k \rrbracket \eta \mathtt{y}) = k = (\mathtt{x} := k)^{\mathsf{b}}.$$

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

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

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 $\eta \in \mathbb{A}$ and assume $\max(\llbracket \mathtt{x} := \mathtt{w} + k \rrbracket \eta \mathtt{y}) > (\mathtt{x} := \mathtt{w} + k)^{\mathsf{b}} = k$. Recall that $\llbracket \mathtt{x} := \mathtt{w} + k \rrbracket \eta = \eta[\mathtt{x} \mapsto \eta \mathtt{w} + k]$.

We distinguish two cases:

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

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

  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 $\llbracket \mathtt{x} := \mathtt{w} + k \rrbracket \eta' \mathtt{y} = \eta' \mathtt{w} + k$ and thus

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

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

Case ($\mathtt{x} := \mathtt{w} - k$) Take $\eta \in \mathbb{A}$ and assume $\max(\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta \mathtt{y}) > (\mathtt{x} := \mathtt{w} - k)^{\mathsf{b}} = k$. Recall that $\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta = \eta[\mathtt{x} \mapsto \eta \mathtt{w} - k]$.

We distinguish two cases:

- If $\mathtt{y} \neq \mathtt{x}$, then for all $\eta' \in \mathbb{A}$ such that $\eta \sqsubseteq \eta'$, we have $\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta' \mathtt{y} = \eta' \mathtt{y}$ and thus

  $$\max(\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta' \mathtt{y}) = \max(\eta' \mathtt{y}).$$

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

- If $\mathtt{x} = \mathtt{y}$ then for all $\eta' \in \mathbb{A}$ such that $\eta \sqsubseteq \eta'$, we have $\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta' \mathtt{y} = \eta' \mathtt{w} - k$ and thus

  $$\max(\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta' \mathtt{y}) = \max(\eta' \mathtt{w}) - k.$$

  Note that the assumption $\max(\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta \mathtt{y}) > k$ and thus $\max(\llbracket \mathtt{x} := \mathtt{w} - k \rrbracket \eta' \mathtt{y}) > k$ ensures that subtraction is not truncated on the maximum.

  Hence the thesis holds, with $h = -k$, hence $|h| = (\mathtt{x} := \mathtt{w} - k)^{\mathsf{b}}$, and $\mathtt{z} = \mathtt{w}$.

Case $(\mathsf{C}_1 + \mathsf{C}_2)$ Take $\eta \in \mathbb{A}$ 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 $[\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y} \neq \infty$, we have that $[\![\mathsf{C}_1]\!]\eta\mathtt{y} \neq \infty \neq [\![\mathsf{C}_2]\!]\eta\mathtt{y}$.

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 realised by the first component, i.e., $\max([\![\mathsf{C}_1 + \mathsf{C}_2]\!]\eta\mathtt{y}) = \max([\![\mathsf{C}_1]\!]\eta\mathtt{y})$. Hence, by inductive hypothesis on C_1, we have 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 $\eta' \in \mathbb{A}$, $\eta \sqsubseteq \eta'$,

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

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 $\eta' \in \mathbb{A}$, $\eta \sqsubseteq \eta'$,

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

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

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

since $\max(\mathsf{C}_2\eta_1\mathtt{y}) \neq \infty$ and $\max(\mathsf{C}_2\eta_1\mathtt{y}) > (\mathsf{C}_1; \mathsf{C}_2)^{\mathsf{b}} \geq (\mathsf{C}_2)^{\mathsf{b}}$, 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 $\eta_1' \in \mathbb{A}$ with $\eta_1 \sqsubseteq \eta_1'$

$$\max([\![\mathsf{C}_2]\!]\eta_1'\mathtt{y}) \geq \max(\eta_1'\mathtt{w}) + h_2 \tag{†}$$

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, $[\![\mathsf{C}_1]\!]\eta\mathtt{w} \neq \infty$, 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 $\eta' \in \mathbb{A}$ with $\eta \sqsubseteq \eta'$

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

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

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

and

$$\begin{aligned}
\max(\llbracket \mathsf{C}_1; \mathsf{C}_2 \rrbracket \eta' \mathtt{y}) = & \\
\max(\llbracket \mathsf{C}_2 \rrbracket (\llbracket \mathsf{C}_1 \rrbracket \eta') \mathtt{w}) \geq & \\
\max(\llbracket \mathsf{C}_1 \rrbracket \eta' \mathtt{w}') + h_2 \geq & \qquad \text{by } (\dagger)\text{, since } \eta_1 = \llbracket \mathsf{C}_1 \rrbracket \eta \sqsubseteq \llbracket \mathsf{C}_1 \rrbracket \eta' \text{, by monotonicity} \\
(\max(\eta' \mathtt{y}) + h_1) + h_2 & \qquad \text{by } (\ddagger)
\end{aligned}$$

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 ($\mathsf{fix(C)}$) Let $\eta \in \mathbb{A}$ such that $\llbracket \mathsf{fix(C)} \rrbracket \eta \mathtt{y} \neq \infty$. Recall that $\llbracket \mathsf{fix(C)} \rrbracket \eta = \mathrm{lfp} \lambda\mu.(\llbracket \mathsf{C} \rrbracket \mu \sqcup \eta)$. 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

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

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

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

Since $\llbracket \mathsf{fix(C)} \rrbracket \eta \mathtt{y} \neq \infty$, we have that for all $i \in \mathbb{N}$, $\eta_i \mathtt{y} \neq \infty$. Moreover, the lub $\bigsqcup_{i \in \mathbb{N}} \eta_i$ 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 $i \geq m+1$

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

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) $\forall \eta' \sqsupseteq \eta_j.\, \max(\llbracket \mathsf{C} + \mathsf{true} \rrbracket \eta' \mathtt{x}) \geq \max(\eta' \mathtt{z}) + h$.

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 $\mathtt{y} \in \mathit{Var}$

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

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 $|h| \leqslant (\mathsf{fix(C)})^{\mathsf{b}}$ such that:

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

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

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

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

$$\sigma \triangleq (\mathtt{y}_0, 0) \to_{h_0} (\mathtt{y}_1, 1) \to_{h_1} \ldots \to_{h_m} (\mathtt{y}_{m+1}, m+1) \tag{2.2}$$

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

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

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 2.4 to the projection π_p of the nodes of this path π to the variable component to deduce that π_p has a subpath which is a cycle with a strictly positive weight. More precisely, there exist $i \leqslant k_1 < k_2 \leqslant m+1$ 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,

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

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

$$\max([\![\mathsf{C} + \mathsf{true}]\!]^{k_2-k_1} \eta'\mathtt{w}) \geq \max(\eta'\mathtt{w}) + h \tag{2.3}$$

This property (2.3) can be shown by induction on $k_2 - k_1 \geq 1$.
Then, an inductive argument allows us to show that for all $r \in \mathbb{N}$:

$$\max([\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)} \eta_{k_1}\mathtt{w}) \geq \max(\eta_{k_1}\mathtt{w}) + rh \tag{2.4}$$

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

$$\begin{aligned}
&\max([\![\mathsf{C} + \mathsf{true}]\!]^{(r+1)(k_2-k_1)} \eta_{k_1}\mathtt{w}) = \\
&\max([\![\mathsf{C} + \mathsf{true}]\!]^{k_2-k_1} ([\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)} \eta_{k_1})\mathtt{w}) \geq \quad [\text{by (2.3) as } \eta_{k_1} \sqsubseteq [\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)} \eta_{k_1}\ ] \\
&\max([\![\mathsf{C} + \mathsf{true}]\!]^{r(k_2-k_1)} \eta_{k_1}\mathtt{w}) + h \geq \quad [\text{by inductive hypothesis}] \\
&\max(\eta_{k_1}\mathtt{w}) + rh + h \geq \max(\eta_{k_1}\mathtt{w}) + (r+1)h
\end{aligned}$$

However, This would contradict the hypothesis $[\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y} \neq \infty$. In fact the inequality (2.4) would imply

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

Now, from (2.2) 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 $\max(\mu_{j+1}\mathtt{y}_{j+1}) \geq \max(\mu_{j+1}\mathtt{y}_j) + h_j$ and thus

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

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

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

contradicting the assumption.
Therefore, the path σ of (2.2) 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 $p[\![\mathsf{fix}(\mathsf{C})]\!]\eta\mathtt{y} = \infty$.

Let us now show (ii). Given $\eta' \sqsupseteq \eta$ from (2.2) 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

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

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:

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

as desired. □

Lemma 2.5 provides an effective algorithm for computing the interval semantics of commands. More precisely, given a command C, the corresponding finite set of variables $\mathit{Var}_{\mathsf{C}} \triangleq \mathit{vars}(\mathsf{C})$, and an interval environment $\rho : \mathit{Var}_{\mathsf{C}} \to \mathit{Int}$, we define

$$\max(\rho) \triangleq \max\{\max(\rho(\mathtt{x})) \mid \mathtt{x} \in \mathit{Var}_{\mathsf{C}}\}.$$

Then, when computing $\langle \mathsf{C}^* \rangle\rho$ on such ρ having a finite domain, we can restrict to a bounded interval domain $\mathbb{A}_{\mathsf{C},\rho} \triangleq (\mathit{Var}_{\mathsf{C}} \to \mathit{Int}_{\mathsf{C},\rho}) \cup \{\top, \bot\}$ where

$$\mathit{Int}_{\mathsf{C},\rho} \triangleq \{[a, b] \mid a, b \in \mathbb{N} \ \wedge \ a \leqslant b \leqslant \max\{\max(\rho), 2(\mathsf{C})^{\mathsf{b}}\}\}.$$

Lemma 2.6. *Let $\mathsf{C} \in \mathit{Imp}$ be a command. Then, for all finitely supported $\rho : \mathit{Var} \to \mathit{Int}$, the abstract semantics $\langle \mathsf{C}^* \rangle\rho = \bigsqcup_{i\in\mathbb{N}} \langle \mathsf{C} \rangle^i(\rho)$ computed in $\mathbb{A}$ and in $\mathbb{A}_{\mathsf{C},\rho}$ coincide.*

Proof. Todo: consequence of Lemma 2.5. □

Chapter 3

Non relational collecting

Let
$$\mathsf{Env}^{\mathsf{c}} \triangleq \{\eta \mid \eta : \mathit{Var} \to 2^{\mathbb{Z}} \setminus \{\varnothing\}\} \cup \{\bot\}.$$

The nonrelational collecting domain is the complete lattice $\mathbb{C}^{\mathsf{c}} \triangleq \langle \mathsf{Env}^{\mathsf{c}}, \dot{\subseteq} \rangle$ where for all $\eta, \eta' : \mathit{Var} \to \wp^{\neq\varnothing}(\mathbb{Z})$

$$\begin{aligned}
&\bot \mathrel{\dot{\subseteq}} \eta \\
&\eta \mathrel{\dot{\subseteq}} \eta' \quad \text{if} \quad \forall \mathtt{x} \in \mathit{Var}. \eta(\mathtt{x}) \subseteq \eta'(\mathtt{x})
\end{aligned}$$

The nonrelation abstraction $\alpha : \langle 2^{\mathsf{Env}}, \subseteq \rangle \to \langle \mathsf{Env}^{\mathsf{c}}, \dot{\subseteq} \rangle$ is defined as follows:

$$\alpha(X) \triangleq \begin{cases} \bot & \text{if } X = \varnothing \\ \lambda \mathtt{x}. \{\rho(\mathtt{x}) \in \mathbb{Z} \mid \rho \in X\} & \text{if } X \neq \varnothing \end{cases}$$

while the concretization $\gamma : \langle \mathsf{Env}^{\mathsf{c}}, \dot{\subseteq} \rangle \to \langle 2^{\mathsf{Env}}, \subseteq \rangle$ is defined as follows:

$$\begin{aligned}
&\gamma(\bot) \triangleq \varnothing \\
&\gamma(\eta) \triangleq \{\rho : \mathit{Var} \to \mathbb{Z} \mid \forall \mathtt{x} \in \mathit{Var}. \rho(\mathtt{x}) \in \eta(\mathtt{x})\}
\end{aligned}$$

Bibliography

[Cut80] Nigel Cutland. *Computability: An introduction to recursive function theory.* Cambridge university press, 1980.

[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.