



# 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

<b>1</b>	<b>Framework</b>	<b>1</b>
1.1	The Imp language . . . . .	1
1.2	Semantics . . . . .	1
<b>2</b>	<b>Intervals</b>	<b>5</b>
<b>3</b>	<b>Non relational collecting</b>	<b>6</b>

# Chapter 1

## Framework

### 1.1 The Imp language

We'll denote by  $\mathbb{Z}$  the set of integers with the usual order plus two bonus elements  $-\infty$  and  $+\infty$ , s.t.  $-\infty \leq z \leq +\infty \quad \forall z \in \mathbb{Z}$ . We also extend addition and subtraction by letting, for  $z \in \mathbb{Z}$   $+\infty + z = +\infty$   $-\infty + z = -\infty$  and  $-\infty - z = -\infty$ .

We'll focus on the following non-deterministic language.

$$\begin{aligned} \text{Exp} \ni e ::= & \quad x \in S \mid x \in [a, b] \mid x \leq k \mid x > k \mid \mathbf{true} \mid \mathbf{false} \mid \\ & \quad x := k \mid x := y + k \mid x := y - k \\ \text{Imp} \ni C ::= & \quad e \mid C + C \mid C; C \mid C* \end{aligned}$$

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

### 1.2 Semantics

In the following section we'll provide the semantics of the language we're working on and we'll make some observations; we'll also prove some properties of the language we'll use in the next sections.

The first building block is that of environments. We'll use environments to model the most precise invariant our semantic can describe for a program.

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

$$\text{Env} \triangleq \{\rho \mid \rho : \text{Var} \rightarrow \mathbb{Z}\}$$

Note: the set of notable variables is assumed to be finite.

**Definition 1.2** (Semantics of Basic Expressions). For basic expressions  $e \in \text{Exp}$  the concrete

semantics  $\llbracket \cdot \rrbracket : \text{Exp} \rightarrow \text{Env} \rightarrow \text{Env} \cup \{\perp\}$  is recursively defined as follows:

$$\begin{aligned}
\llbracket x \in S \rrbracket \rho &\triangleq \begin{cases} \rho & \rho(x) \in S \\ \perp & \text{otherwise} \end{cases} \\
\llbracket x \in [a, b] \rrbracket \rho &\triangleq \begin{cases} \rho & \rho(x) \in [a, b] \\ \perp & \text{otherwise} \end{cases} \\
\llbracket x \leq k \rrbracket \rho &\triangleq \begin{cases} \rho & \rho(x) \leq k \\ \perp & \text{otherwise} \end{cases} \\
\llbracket x > k \rrbracket \rho &\triangleq \begin{cases} \rho & \rho(x) > k \\ \perp & \text{otherwise} \end{cases} \\
\llbracket \text{true} \rrbracket \rho &\triangleq \rho \\
\llbracket \text{false} \rrbracket \rho &\triangleq \perp \\
\llbracket x := k \rrbracket \rho &\triangleq \rho[x \mapsto k] \\
\llbracket x := y + k \rrbracket \rho &\triangleq \rho[x \mapsto \rho(y) + k] \\
\llbracket x := y - k \rrbracket \rho &\triangleq \rho[x \mapsto \rho(y) - k]
\end{aligned}$$

The next building block is the concrete collecting semantics for the language, maps each program in  $\text{Imp}$  to a function on the  $\mathbb{C}$  complete lattice.

**Definition 1.3** (Concrete collecting domain). The concrete collecting domain for the  $\text{Imp}$  language concrete collecting semantics is the complete lattice

$$\mathbb{C} \triangleq \langle 2^{\text{Env}}, \subseteq \rangle$$

We can therefore define the concrete collecting semantics for our language:

**Definition 1.4** (Concrete collecting semantics). The concrete collecting semantics for  $\text{Imp}$  is given by the total mapping

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

which maps each program  $C \in \text{Imp}$  to its total mapping

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

on the complete lattice  $\mathbb{C}$ . The semantics is recursively defined as follows: given  $X \in 2^{\text{Env}}$

$$\begin{aligned}
\langle e \rangle X &\triangleq \{ \llbracket e \rrbracket \rho \mid \rho \in X, \llbracket e \rrbracket \rho \neq \perp \} \\
\langle C_1 + C_2 \rangle X &\triangleq \langle C_1 \rangle X \cup \langle C_2 \rangle X \\
\langle C_1; C_2 \rangle X &\triangleq \langle C_2 \rangle (\langle C_1 \rangle X) \\
\langle C^* \rangle X &\triangleq \bigcup_{i \in \mathbb{N}} \langle C \rangle^i X
\end{aligned}$$

Along with the collecting semantics we're also defining a one step transition relation. This will be useful to prove that finiteness is not decidable, as it helps to reason about program execution in a recursive way. The relation is defined on program states:

**Definition 1.5** (Program State). Program states are tuples of programs and program environments:

$$\text{State} \triangleq \text{Imp} \times \text{Env}$$

**Definition 1.6** (Small step semantics). The small step transition relation  $\rightarrow: \text{State} \times (\text{State} \cup \text{Env})$  is a small step semantics for the Imp language. It is defined based on the following rules

$$\begin{array}{c} \frac{\langle e \rangle \rho \neq \perp}{\langle e, \rho \rangle \rightarrow \langle e \rangle \rho} \text{expr} \\[10pt] \frac{}{\langle C_1 + C_2, \rho \rangle \rightarrow \langle C_1, \rho \rangle} \text{sum}_1 \quad \frac{}{\langle C_1 + C_2, \rho \rangle \rightarrow \langle C_2, \rho \rangle} \text{sum}_2 \\[10pt] \frac{\langle C_1, \rho \rangle \rightarrow \langle C'_1, \rho' \rangle}{\langle C_1; C_2, \rho \rangle \rightarrow \langle C'_1; C_2, \rho' \rangle} \text{comp}_1 \quad \frac{\langle C_1, \rho \rangle \rightarrow \rho'}{\langle C_1; C_2, \rho \rangle \rightarrow \langle C_2, \rho' \rangle} \text{comp}_2 \\[10pt] \frac{}{\langle C^*, \rho \rangle \rightarrow \langle C; C^*, \rho \rangle} \text{star} \quad \frac{}{\langle C^*, \rho \rangle \rightarrow \rho} \text{star}_{\text{fix}} \end{array}$$

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

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

Therefore  $\langle C \rangle X = \emptyset \iff \forall \rho \in X \langle C, \rho \rangle$  does not reach a final environment  $\rho_t$ .

*Proof.* by induction on  $C$ :

**Base case**  $C \equiv e$ :

$\langle e \rangle X = \{ \langle e \rangle \rho \mid \rho \in X \wedge \langle e \rangle \rho \neq \perp \}$ ,  $\forall \rho \in X. \langle e, \rho \rangle \rightarrow \langle e \rangle \rho$  if  $\langle e \rangle \rho \neq \perp$  because of the expr rule

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

**Inductive cases:**

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

$$\langle C_1 \rangle X = \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho_t \} \quad \langle C_2 \rangle X = \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X, \langle C_2, \rho \rangle \rightarrow^* \rho_t \}$$

Therefore

$$\begin{aligned} \langle C_1 + C_2 \rangle X &= \langle C_1 \rangle X \cup \langle C_2 \rangle X && \text{(by definition)} \\ &= \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho_t \} \cup \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X, \langle C_2, \rho \rangle \rightarrow^* \rho_t \} && \text{(by ind. hp)} \\ &= \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho_t \vee \langle C_2, \rho \rangle \rightarrow^* \rho_t \} \\ &= \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X, \langle C_1 + C_2, \rho \rangle \rightarrow^* \rho_t \} \end{aligned}$$

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

$$\begin{aligned} \langle C_1; C_2 \rangle X &= \langle C_2 \rangle (\langle C_1 \rangle X) && \text{(by definition)} \\ &= \{ \rho_t \in 2^{\text{Env}} \mid \rho_x \in \{ \rho_x \mid \rho \in X, \langle C_1, \rho \rangle \rightarrow^* \rho_x \}, \langle C_2, \rho_x \rangle \rightarrow^* \rho_t \} && \text{(by ind. hp)} \\ &= \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X \langle C_1, \rho \rangle \rightarrow^* \rho_x \wedge \langle C_2, \rho_x \rangle \rightarrow^* \rho_t \} && \text{(by definition)} \\ &= \{ \rho_t \in 2^{\text{Env}} \mid \rho \in X. \langle C_1; C_2, \rho \rangle \rightarrow^* \rho_t \} \end{aligned}$$

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

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

□

We can notice that  $\langle C \rangle X = \emptyset \iff \nexists \rho_t \in \text{Env}, \rho \in X \mid \langle C, \rho \rangle \rightarrow^* \rho_t$ .



## Chapter 2

# Intervals

## Chapter 3

# Non relational collecting