Some decidability questions in abstract program semantics

Luca Zaninotto

Master degree in Computer Science

Università degli studi di Padova

03 Jul 2024

The cost of software failures

Figure: Ariane 5 crash, circa 370mln$ in damages

- Testing and careful design might not be enough.
- Formal methods can help, by providing strong guarantees.

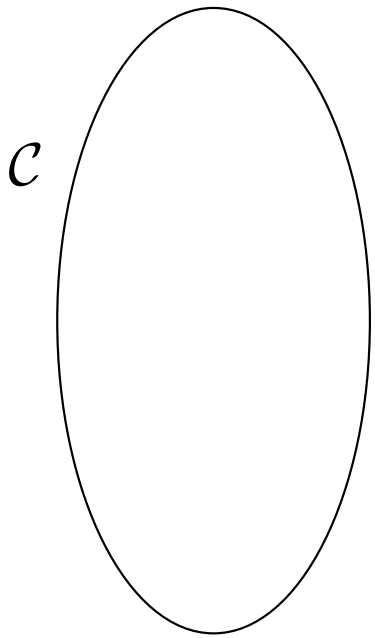
The cost of software failures

Figure: Ariane 5 crash, circa 370mln$ in damages

- Testing and careful design might not be enough.
- Formal methods can help, by providing strong guarantees.
- We focus in particular on Abstract interpretation.

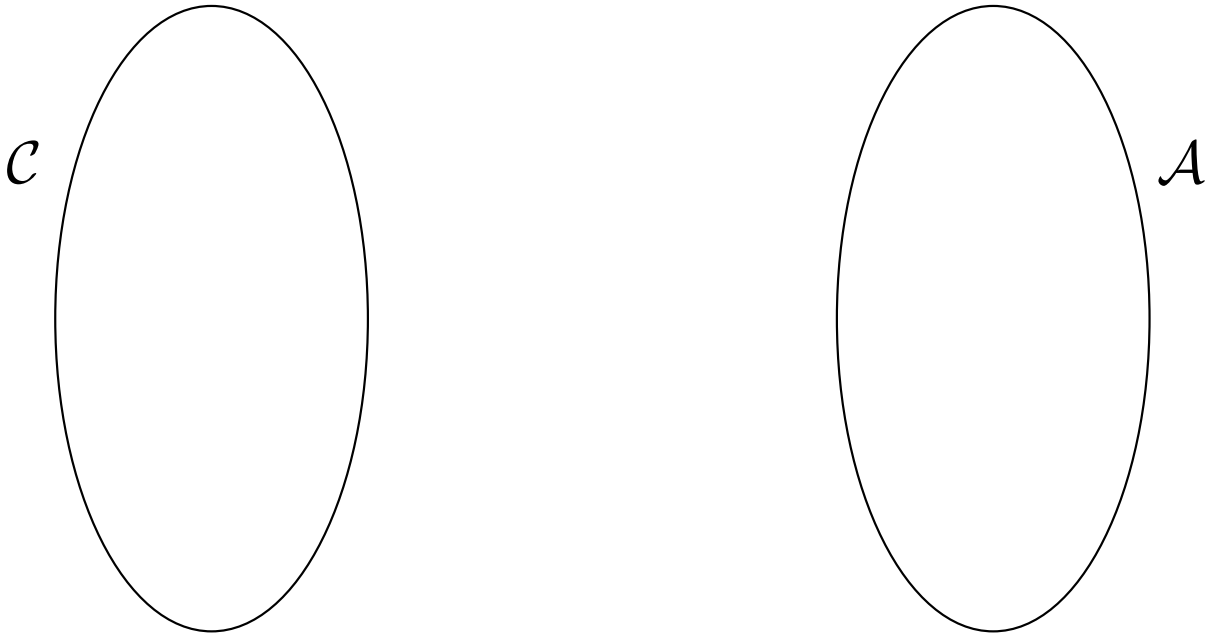
Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



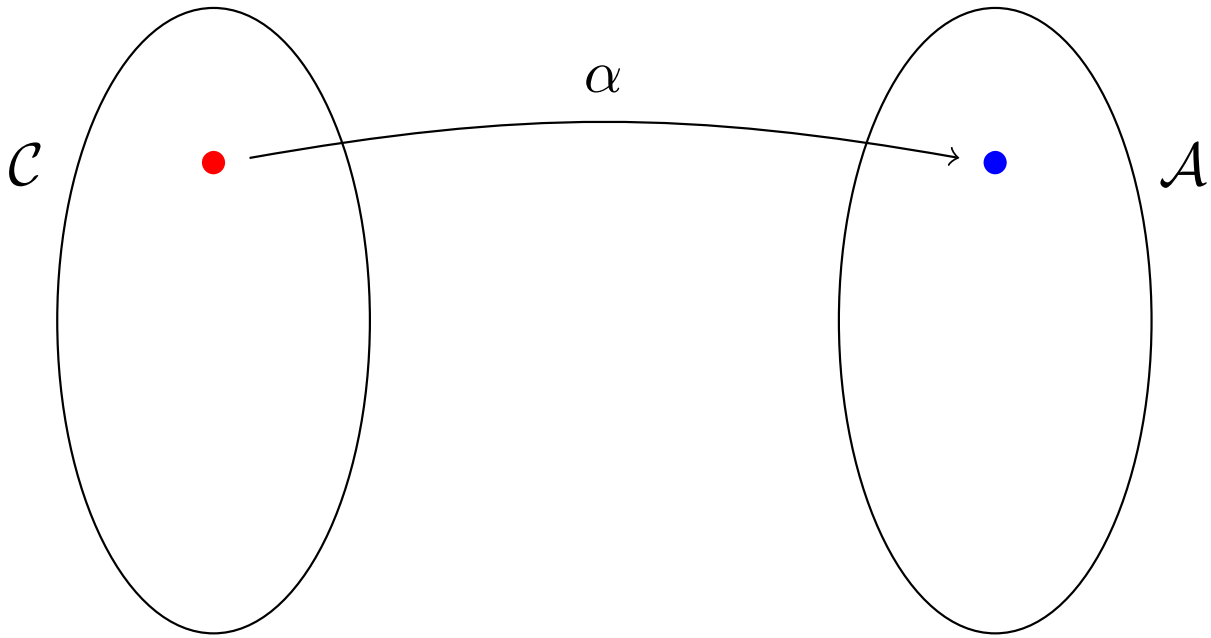
Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



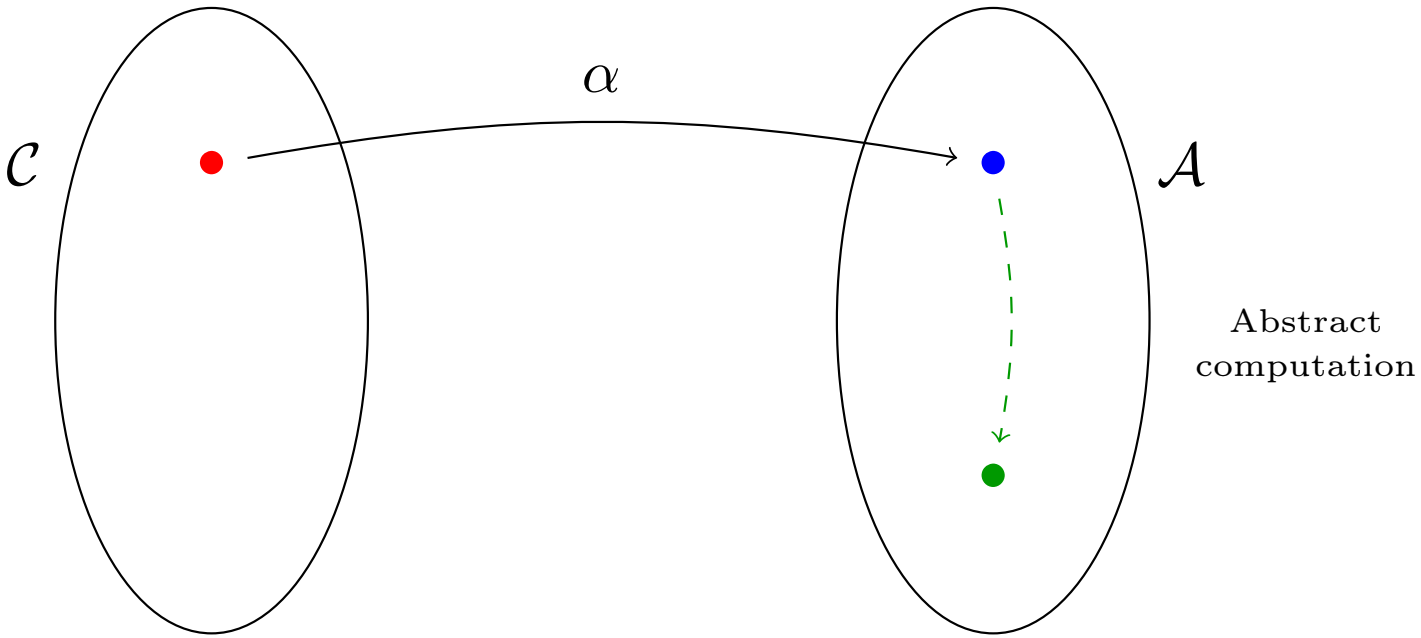
Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



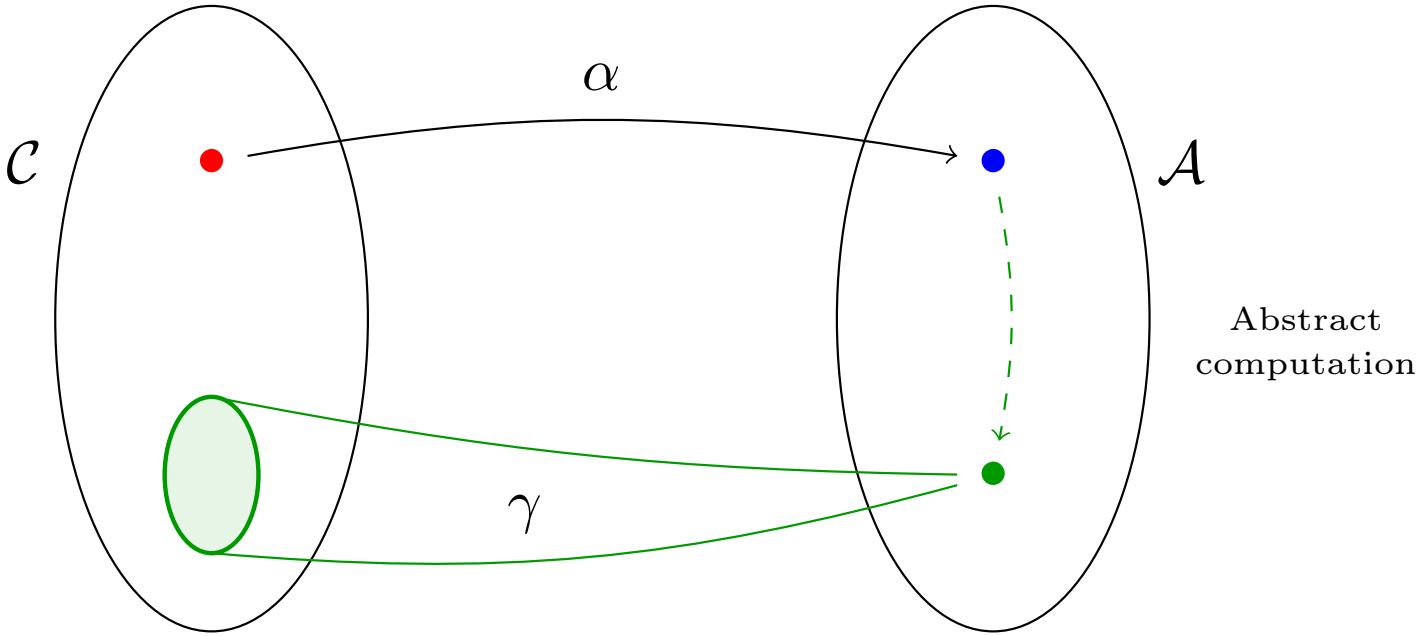
Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



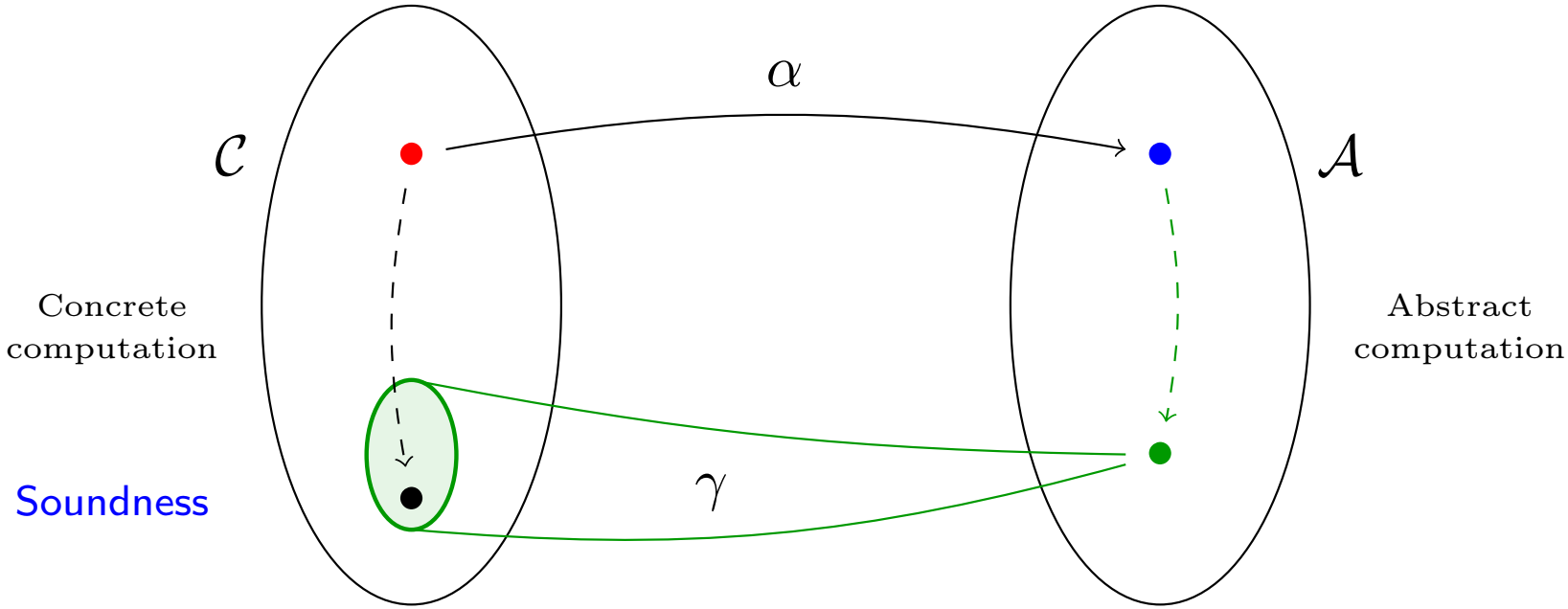
Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



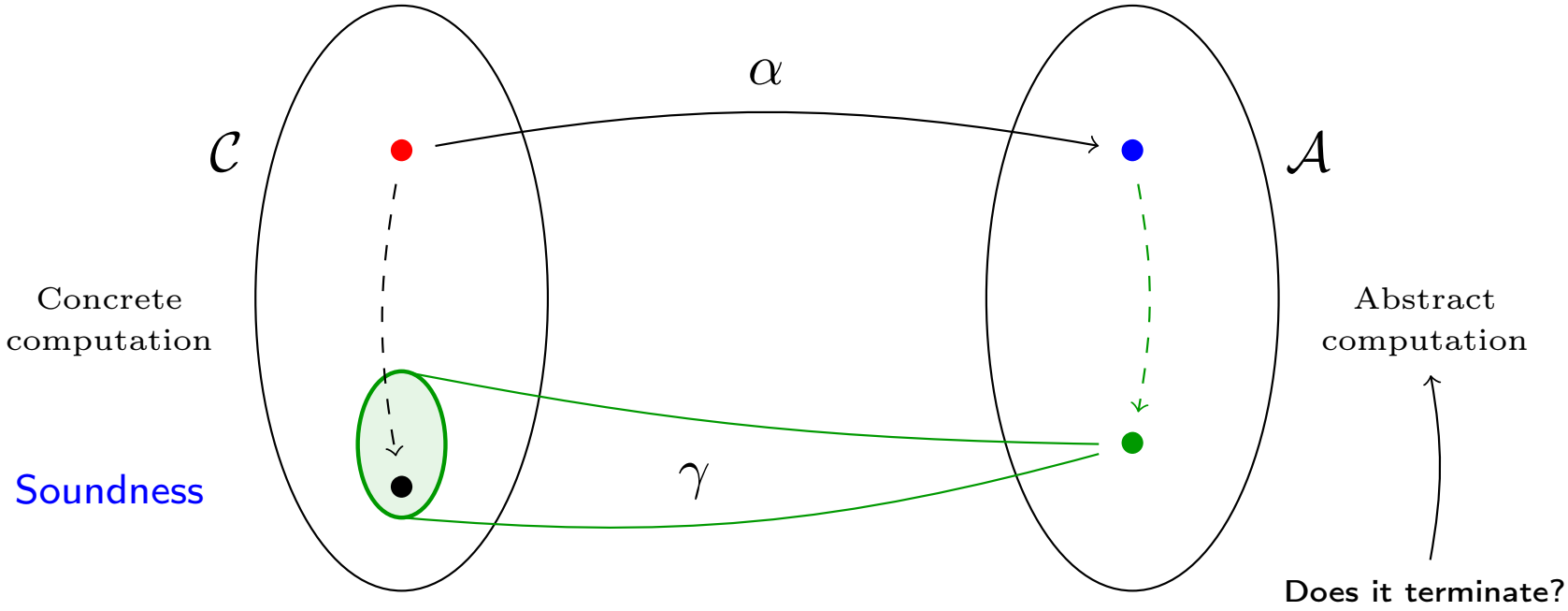
Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



Abstract interpretation

Given a program semantics, abstracts its behaviour and provide an over-approximation of the program semantics



Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[x \mapsto 0]$$

Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[x \mapsto 0] \rightarrow \{[x \mapsto 0], [x \mapsto 1]\}$$

Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[\mathrm{x} \mapsto 0] \to \{[\mathrm{x} \mapsto 0], [\mathrm{x} \mapsto 1]\} \to^* \{[\mathrm{x} \mapsto n] \mid 0 \leqslant n \leqslant k, k \in \mathbb{N}\}$$

Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[\mathrm{x} \mapsto 0] \to \{[\mathrm{x} \mapsto 0], [\mathrm{x} \mapsto 1]\} \to^* \{[\mathrm{x} \mapsto n] \mid 0 \leqslant n \leqslant k, k \in \mathbb{N}\} \to \dots$$

Intervals would also diverge

$$[\mathrm{x} \mapsto [0, 0]]$$

Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[\mathrm{x} \mapsto 0] \to \{[\mathrm{x} \mapsto 0], [\mathrm{x} \mapsto 1]\} \to^* \{[\mathrm{x} \mapsto n] \mid 0 \leqslant n \leqslant k, k \in \mathbb{N}\} \to \dots$$

Intervals would also diverge

$$[\mathrm{x} \mapsto [0, 0]] \to [\mathrm{x} \mapsto [0, 1]]$$

Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[\mathrm{x} \mapsto 0] \to \{[\mathrm{x} \mapsto 0], [\mathrm{x} \mapsto 1]\} \to^* \{[\mathrm{x} \mapsto n] \mid 0 \leqslant n \leqslant k, k \in \mathbb{N}\} \to \dots$$

Intervals would also diverge

$$[\mathrm{x} \mapsto [0,0]] \to [\mathrm{x} \mapsto [0,1]] \to^* [\mathrm{x} \mapsto [0,k]] \text{ with } k \in \mathbb{N}$$

Analyzer termination is not guaranteed

Consider the C-like program

```
int x = 0;
while(true) {
  x++;
}
```

A concrete semantics that collects variables values diverges

$$[\mathrm{x} \mapsto 0] \to \{[\mathrm{x} \mapsto 0], [\mathrm{x} \mapsto 1]\} \to^* \{[\mathrm{x} \mapsto n] \mid 0 \leqslant n \leqslant k, k \in \mathbb{N}\} \to \dots$$

Intervals would also diverge

$$[\mathrm{x} \mapsto [0,0]] \to [\mathrm{x} \mapsto [0,1]] \to^* [\mathrm{x} \mapsto [0,k]] \text{ with } k \in \mathbb{N} \to \dots$$

Goal

- Establish if some abstract semantics are computable.
- Focus on non-relational domains:
  1. Interval domain $\dot{\mathbb{I}} \triangleq (\mathit{Var} \mapsto \mathbb{I})$: $\mathtt{x} \mapsto$range where $\mathtt{x}$ can vary
  2. Non-relational collecting domain $\mathbb{C} \triangleq (\mathit{Var} \mapsto \wp(\mathbb{Z}))$: $\mathtt{x} \mapsto$ set of possible values of $\mathtt{x}$.

Goal

- Establish if some abstract semantics are computable.
- Focus on non-relational domains:
  1. Interval domain $\dot{\mathbb{I}} \triangleq (\mathit{Var} \mapsto \mathbb{I})$: $\mathtt{x} \mapsto$range where $\mathtt{x}$ can vary Computable
  2. Non-relational collecting domain $\mathbb{C} \triangleq (\mathit{Var} \mapsto \wp(\mathbb{Z}))$: $\mathtt{x} \mapsto$ set of possible values of $\mathtt{x}$. Partial results

Outline

Grammar

- Minimal core of an imperative language;
- Turing complete;
- Based on Kleene algebras with tests.

$$\mathsf{Exp} \ni \mathrm{e} ::= \mathrm{x} \in I \mid \mathrm{x} := k \mid \mathrm{x} := \mathrm{y} + k$$
$$\mathsf{Imp} \ni \mathrm{C} ::= \mathrm{e} \mid \mathrm{C} + \mathrm{C} \mid \mathrm{C}; \mathrm{C} \mid \mathrm{C}^* \mid \mathrm{fix}(\mathrm{C})$$

- while b do C $\implies$ $\mathrm{fix}(b; \mathrm{C}); \neg b$
- if b then C_1 else C_2 $\implies$ $(b; \mathrm{C}_1) + (\neg b; \mathrm{C}_2)$

Concrete semantics

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

- Collecting semantics.
- Finiteness and termination are undecidable because of Rice.

Undecidability of collecting semantics

Given an initial state $\rho \in \wp(\text{Env})$ and a program C

- Does the computation of $\langle \text{C} \rangle \rho$ terminate?
- Is $\langle \text{C} \rangle \rho$ finite?

Undecidability of collecting semantics

Given an initial state $\rho \in \wp(\text{Env})$ and a program C

- Does the computation of $\langle \text{C} \rangle \rho$ terminate?
- Is $\langle \text{C} \rangle \rho$ finite?

Both undecidable because of Rice's Theorem.

Interval domain

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

- Variables map to an interval $[\mathrm{x} \mapsto [-1, 1], \mathrm{y} \mapsto [0, 0], \dots]$
- Non-relational: relations beween variables (e.g., $\mathrm{x} = 3 * \mathrm{y}$) are not modelled
- Computation of fixpoints non trivial

Infinite chains

```
x := 0; fix(true; x++)
```

- Computation does not halt

$$[\mathtt{x} \mapsto 0] \to \{[\mathtt{x} \mapsto 0], [\mathtt{x} \mapsto 1]\} \to \cdots \to \{[\mathtt{x} \mapsto n] \mid n \in \mathbb{N}\}$$

- Analysis does not halt either

$$[\mathtt{x} \mapsto [0,0]] \to [\mathtt{x} \mapsto [0,1]] \to \cdots \to [\mathtt{x} \mapsto [0,\infty]]$$

- Problem: iterating over an infinite chain in the domain

$$[0,0] \sqsubseteq [0,1] \sqsubseteq \cdots \sqsubseteq [0,\infty]$$

Widening and narrowing

- Common approach: widening ∇
- Widening over-approximates a result. Example

```
x := 0; fix(x < 10; x++);
```

  - Precise analysis (not guaranteed to halt): $[\mathrm{x} \mapsto [0, 10]]$
  - Analysis with widening (halts): $[\mathrm{x} \mapsto [0, \infty]])$

The problem

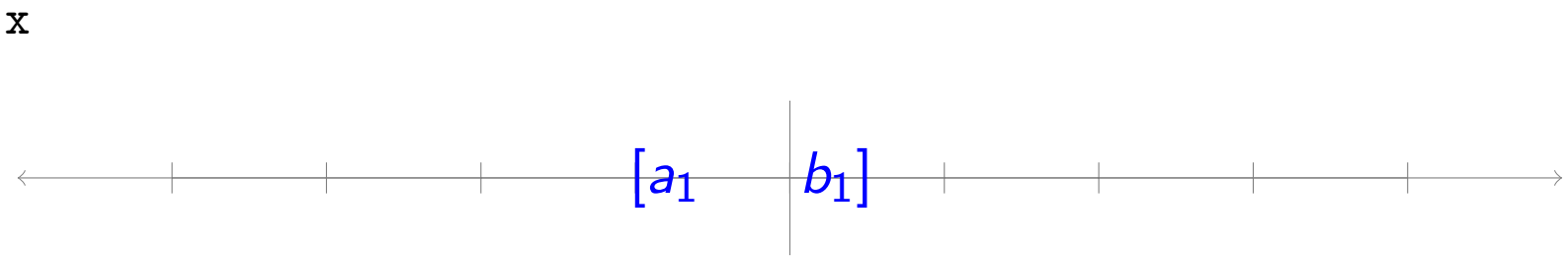
Problem

Can we compute the precise interval semantics while ensuring the termination of the analyzer?

Bounding the interval domain

Consider the behavior of some variable x while computing

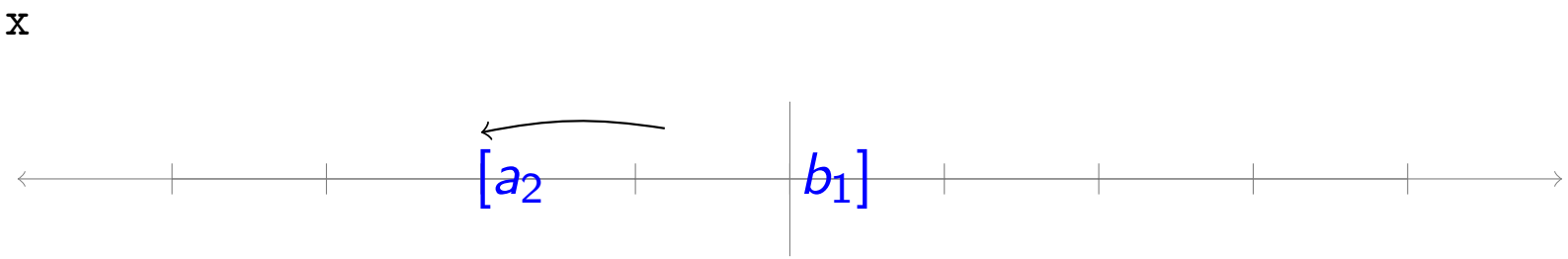
$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathsf{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu))$$



Bounding the interval domain

Consider the behavior of some variable x while computing

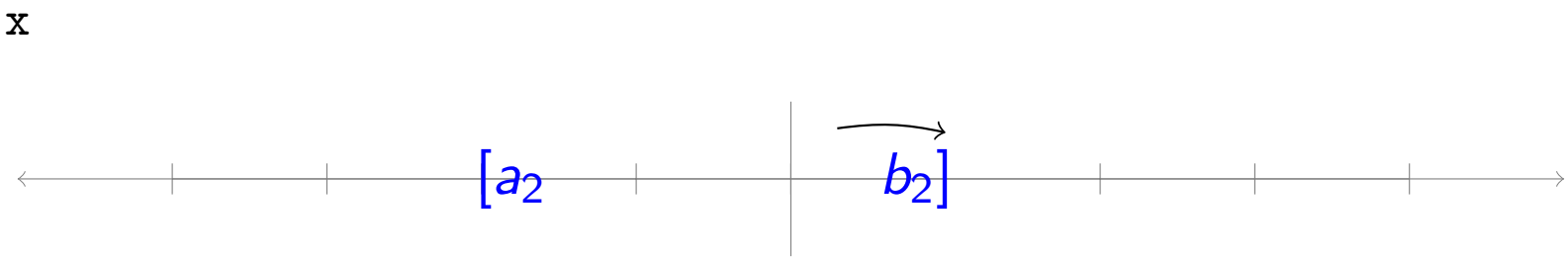
$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathsf{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu))$$



Bounding the interval domain

Consider the behavior of some variable x while computing

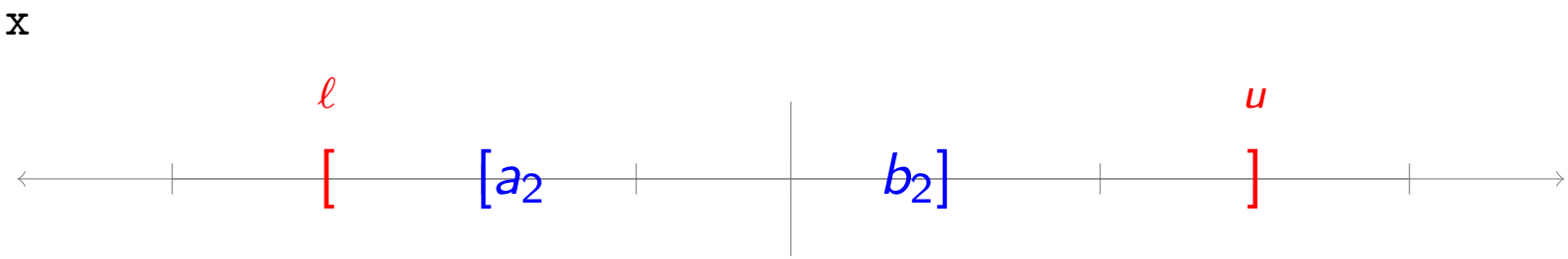
$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathsf{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu))$$



Bounding the interval domain

Consider the behavior of some variable x while computing

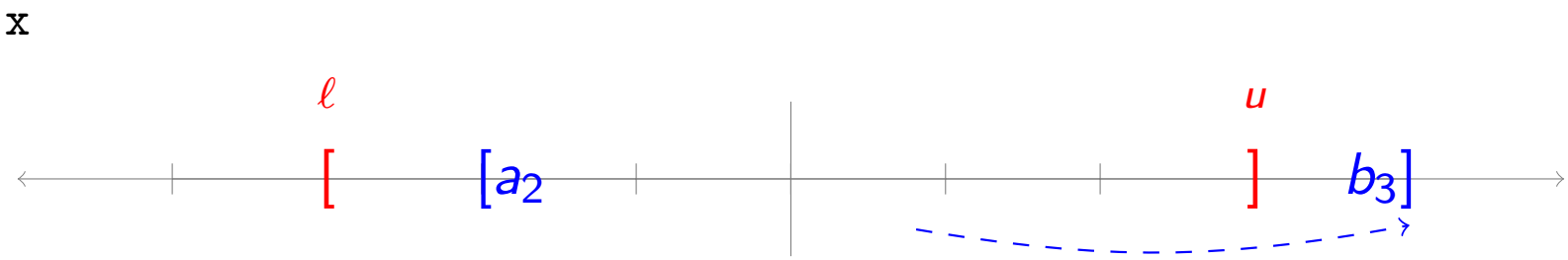
$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathsf{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu))$$



Bounding the interval domain

Consider the behavior of some variable x while computing

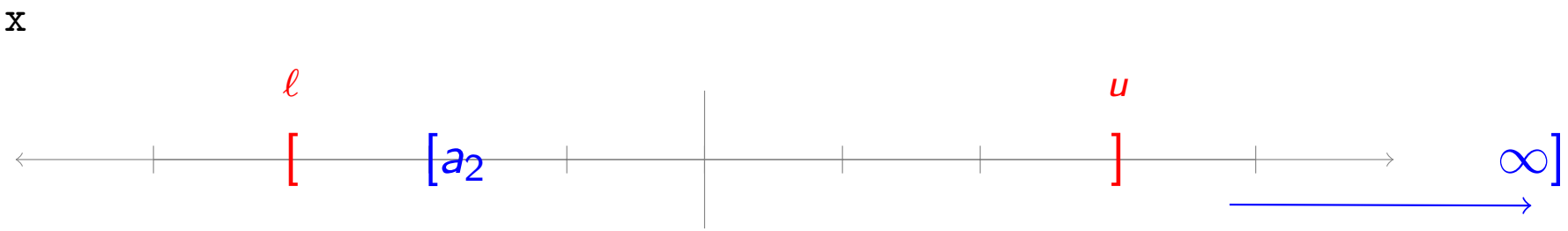
$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathsf{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu))$$



Bounding the interval domain

Consider the behavior of some variable x while computing

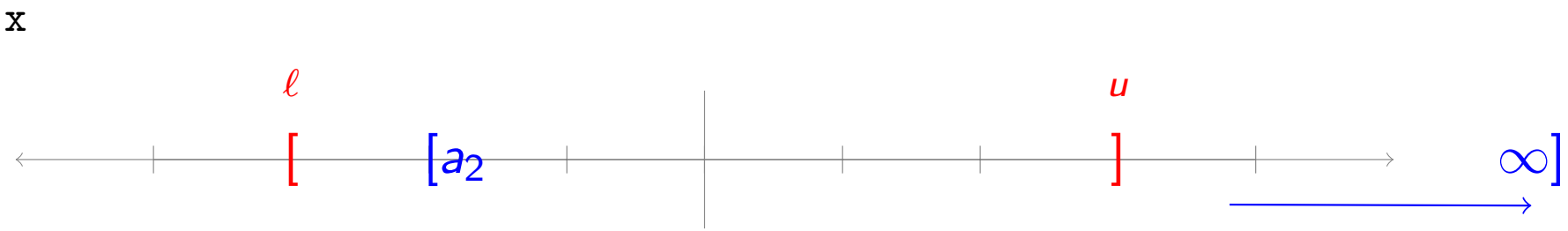
$$[\![\mathsf{fix}(\mathsf{C})]\!]\eta = \mathsf{lfp}(\lambda\mu.(\eta \sqcup [\![\mathsf{C}]\!]\mu))$$



Bounding the interval domain

Consider the behavior of some variable x while computing

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



- Bounds are determined by the program C and the initial environment
- If a variable exceeds a bound the corresponding side of the interval is pushed to infinity

Bounding the interval domain

By choosing ℓ, u appropriately

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

it holds that

$$[\![\mathsf{C}]\!]\eta = [\![\mathsf{C}]\!]_\ell^u \eta$$

Since $\dot{\mathbb{I}}_\ell^u$ does not contain infinite chains, the termination trivializes.

Non-relational collecting domain

$$\mathbb{C} \triangleq (\mathit{Var} \to \wp(\mathbb{Z})) \cup \{\bot\}$$

- Variables mapped to a generic subset of integers.
- Variable images are no longer convex.
- We could only prove some partial results.

Bounding the non-relational collecting domain

$$\wp(\mathbb{Z})_\ell^u \triangleq \{S \subseteq \mathbb{Z} \mid S \neq \varnothing \wedge \forall x \in S \quad \ell \leqslant x \leqslant u\}$$

$$\overline{\mathbb{C}}_\ell^u \triangleq (\mathit{Var} \to \wp(\mathbb{Z})_\ell^u) \cup \{\bot, \top\}$$

- Variables mapped to bounded subsets of $\mathbb{Z}$.
- If some variable exceeds the bound than the whole analysis results in the smashed $\top$ element.
- We can decide analysis termination. If the analysis halts then we can provide the most precise abstract invariant (not in general).

Results

- Interval analysis can be computed precisely in finite time

$$[\![\mathsf{C}]\!]^{\dot{\mathbb{I}}}\eta = [\![\mathsf{C}]\!]^{\dot{\mathbb{I}}^{\boldsymbol{u}}_{\ell}}\eta$$

- For non-relational collecting semantics we can decide termination of the analyzer.

Future work

- Generalize to more expressive languages (e.g. include non-linear expressions).
- Investigate the computability of non-relational collecting semantics.
- Generalize to other abstract semantics
  - Non-relational
  - With a controlled amount of relationality