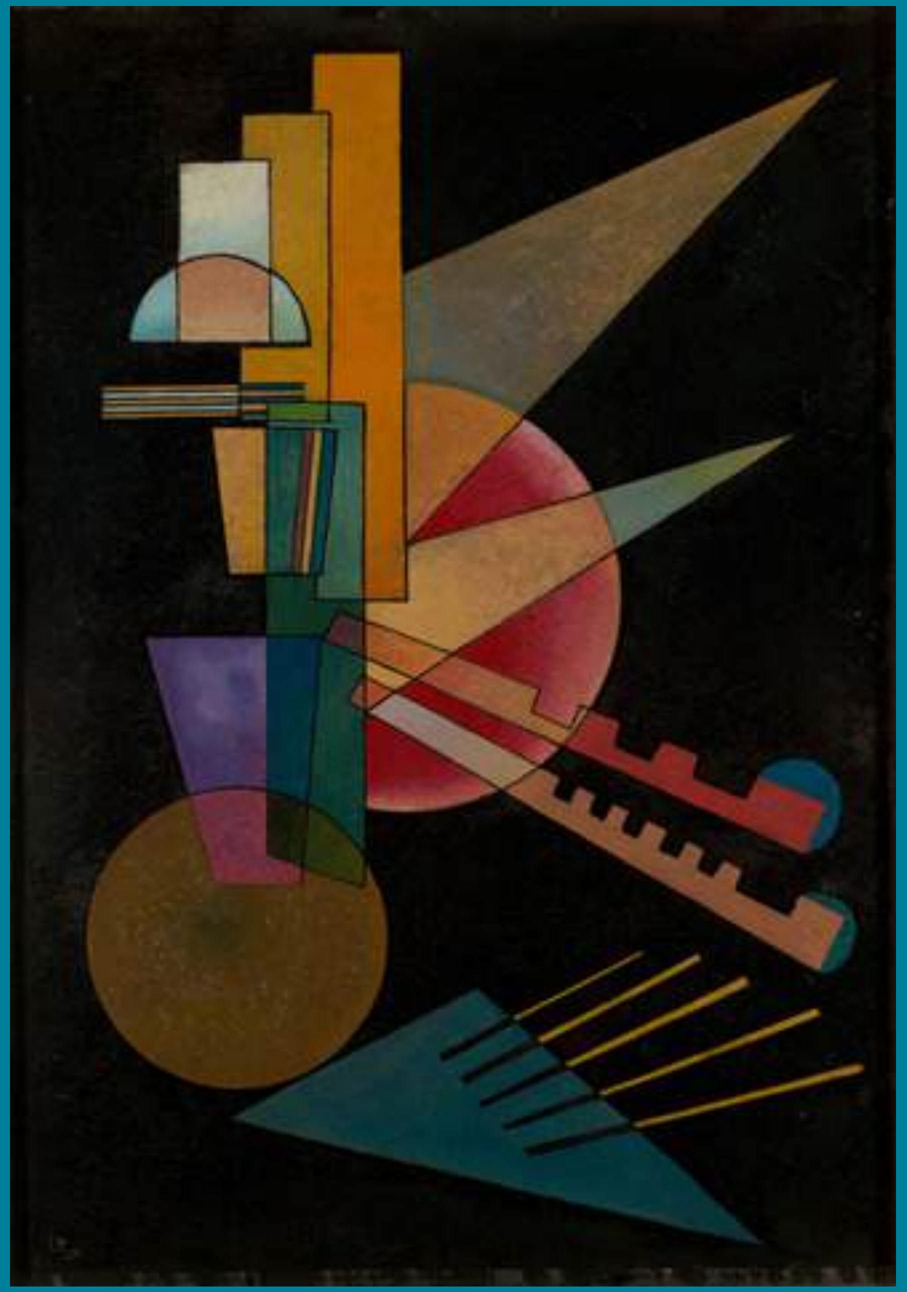

Kandinsky - Abstract Interpretation, 1925

Abstract Interpretation

and Applications in Security, Data Science, and Machine Learning

OPLSS 2025

Caterina Urban
Inria & École Normale Supérieure | Université PSL

Static Analysis of Liveness Properties

The Art of Losing Precision

No Surprises, Please

What Could
Possibly Go Right?

It's Complicated

Trace Properties

$T \in \mathscr{P}(\Sigma^\infty)$

Liveness Properties = "Something Good Eventually Happens"

Example

- Termination: $T \stackrel{\text{def}}{=} \Sigma^*$

Liveness Property Verification

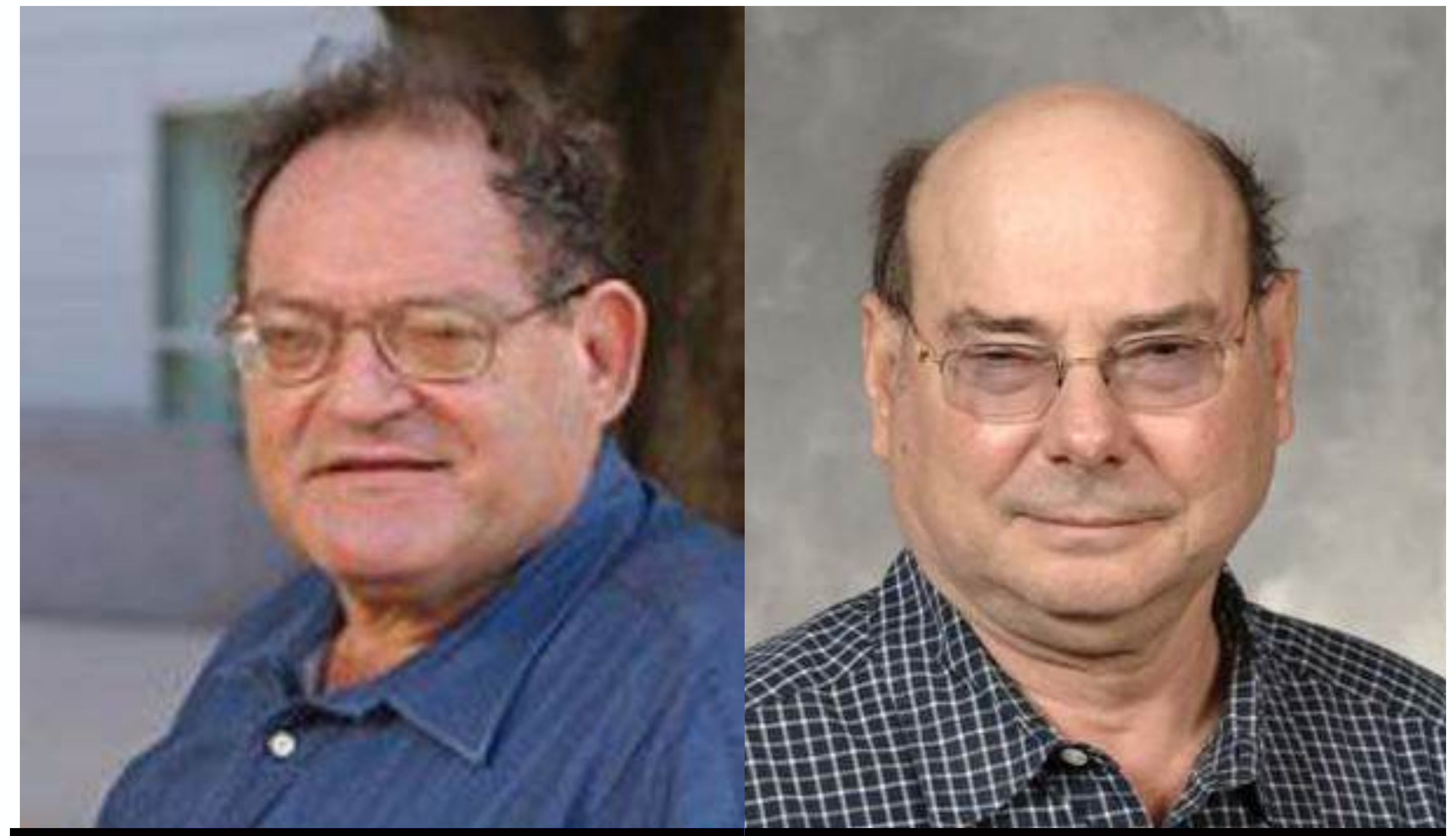
- T cannot be **verified** by **testing**

$\mathscr{M} \subseteq T$

- falsifying T requires finding **an infinite execution not in** T

Liveness Properties

- **Guarantee Properties**
  “something good eventually happens at least once”
  - Example: Program Termination
- **Recurrence Properties**
  “something good eventually happens infinitely often”
  - Example: Starvation Freedom



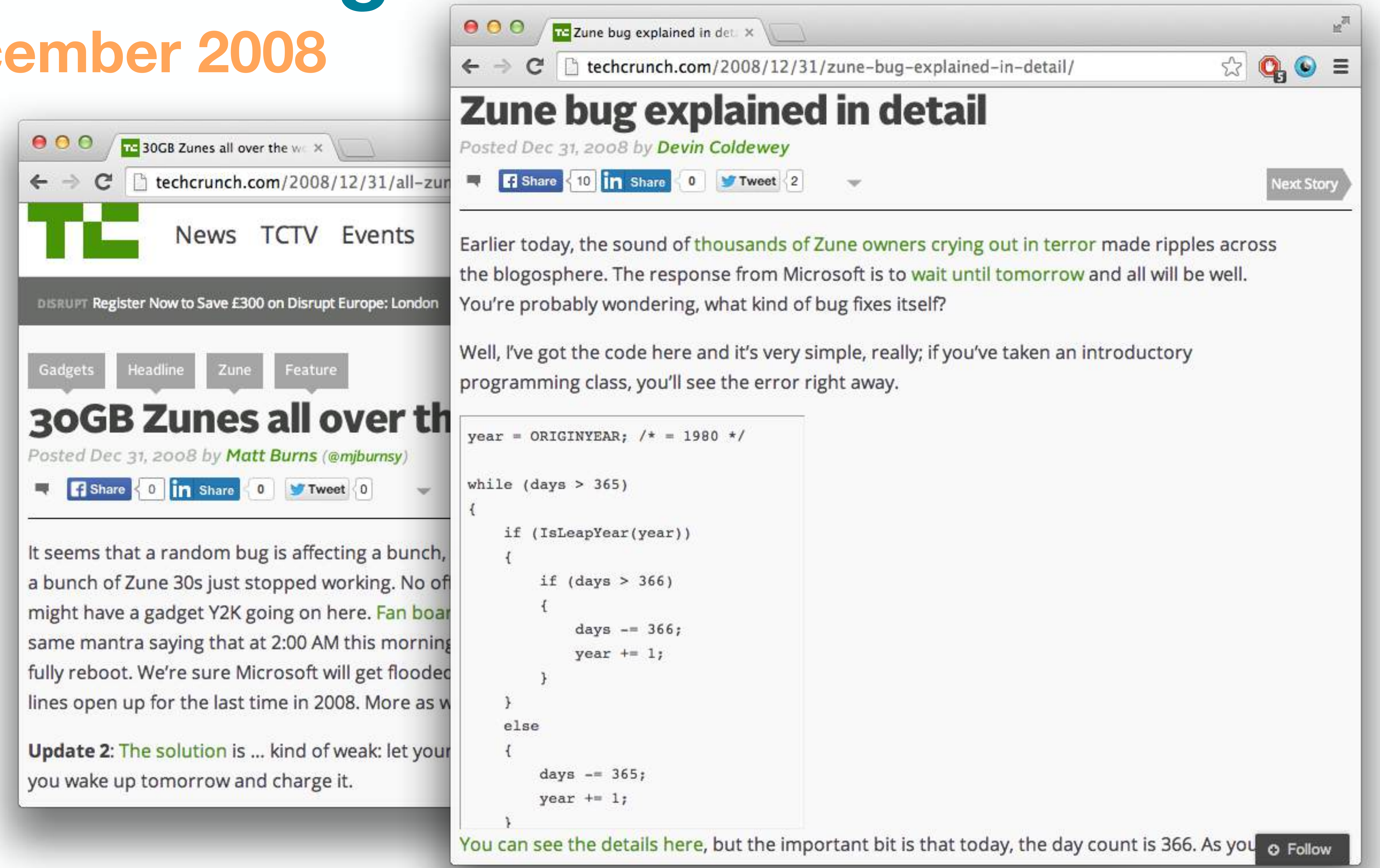
Zohar Manna Amir Pnueli

Program Termination

The Zune Bug

31 December 2008

unresponsive systems



30GB Zunes all over the w...

techcrunch.com/2008/12/31/all-zun...

News TCTV Events

DISRUPT Register Now to Save £300 on Disrupt Europe: London

Gadgets Headline Zune Feature

30GB Zunes all over th...

Posted Dec 31, 2008 by Matt Burns (@mjburnsy)

Share 0 Share 0 Tweet 0

It seems that a random bug is affecting a bunch, a bunch of Zune 30s just stopped working. No of... might have a gadget Y2K going on here. Fan boar... same mantra saying that at 2:00 AM this morning... fully reboot. We're sure Microsoft will get flooded... lines open up for the last time in 2008. More as w...

Update 2: The solution is ... kind of weak: let your... you wake up tomorrow and charge it.

Zune bug explained in det...

techcrunch.com/2008/12/31/zune-bug-explained-in-detail/

Zune bug explained in detail

Posted Dec 31, 2008 by Devin Coldewey

Share 10 Share 0 Tweet 2 Next Story

Earlier today, the sound of thousands of Zune owners crying out in terror made ripples across the blogosphere. The response from Microsoft is to wait until tomorrow and all will be well. You're probably wondering, what kind of bug fixes itself?

Well, I've got the code here and it's very simple, really; if you've taken an introductory programming class, you'll see the error right away.

```
year = ORIGINYEAR; /* = 1980 */

while (days > 365)
{
    if (IsLeapYear(year))
    {
        if (days > 366)
        {
            days -= 366;
            year += 1;
        }
    }
    else
    {
        days -= 365;
        year += 1;
    }
```

You can see the details here, but the important bit is that today, the day count is 366. As you...

Follow

Apache HTTP Server

Versions <2.3.3

denial-of-service attacks

Azure Storage Service

19 November 2014

service interruptions

Update on Azure Storage Serv ×

Secure | https://azure.microsoft.com/en-us/blog/update-on-azure-st...

Microsoft Azure

Blog > Announcements

Update on Azure Storage Service Interruption

Posted on November 19, 2014

Jason Zander
Corporate Vice President, Microsoft Azure Team

Update: 11/22/2014, 12:41 PM PST Since Wednesday, we have been working to help a subset of customers take final steps to fully recover from Tuesday's storage service interruption. The incident has now been resolved and we are seeing normal activity in the system. You can find updates on the status dashboard: https://azure.microsoft.com/en-us/status. If you feel you are still having issues due to the incident, please contact azcommsm@microsoft.com, and we will be happy to assist, whether you have a support contract or not. Thank you all again for your feedback regarding communications around this incident. We are actively working to incorporate that feedback into our planning going forward. **Wednesday, November, 19, 2014** As part of a performance update to Azure Storage, an issue was discovered that resulted in reduced capacity across services utilizing Azure Storage, including Virtual Machines, Visual Studio Online, Websites, Search and other Microsoft services. Prior to applying the performance update, it had been tested over several weeks in a subset of our customer-facing storage service for Azure Tables. We typically call this "flighting," as we work to identify issues before we broadly deploy any updates. The flighting test demonstrated a notable performance improvement and we proceeded to deploy the update across the storage service. During the rollout we discovered an issue that resulted in storage blob front ends going into an infinite loop, which had gone undetected during flighting. The net result was an inability for the front ends to take on further traffic, which

Potential and Definite Termination

Potential Termination

Definition

A program with trace semantics $\mathscr{M} \in \mathscr{P}(\Sigma^\infty)$ **may terminate** if and only if $\mathscr{M} \cap \Sigma^* \neq \emptyset$

Definite Termination

Definition

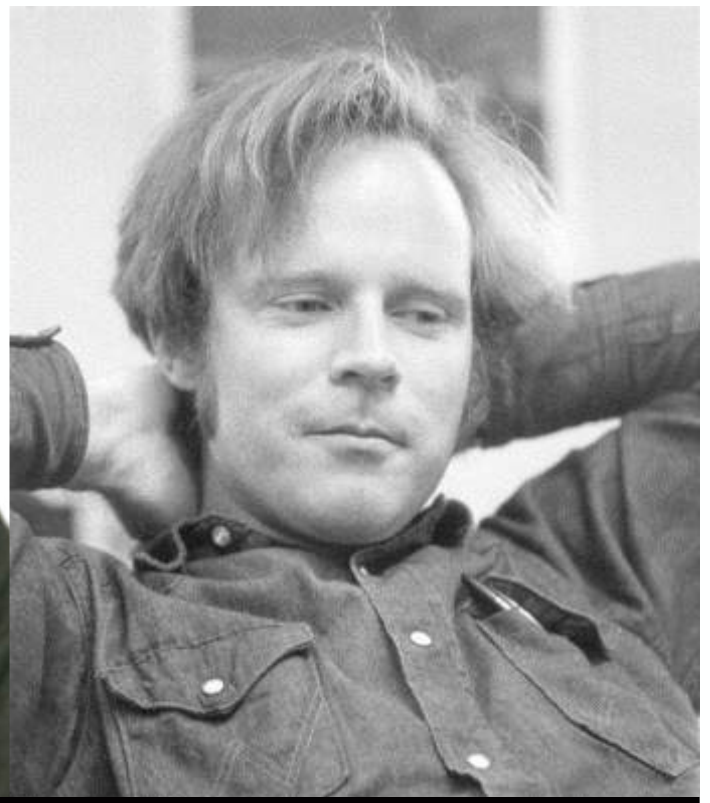
A program with trace semantics $\mathscr{M} \in \mathscr{P}(\Sigma^\infty)$ **must terminate** if and only if $\mathscr{M} \subseteq \Sigma^*$

In *absence of non-determinism*, potential and definite termination coincide

Definite Termination

Ranking Functions


Alan Turing


Robert W. Floyd

Definition

Given a transition system $\langle \Sigma, \tau \rangle$, a **ranking function** is a partial function $f \colon \Sigma \rightharpoonup \mathscr{W}$ from the set of program states Σ into a well-ordered set $\langle \mathscr{W}, \leq \rangle$ whose value *strictly decreases* through transitions between states, that is, $\forall \sigma, \sigma' \in \text{dom}(f) \colon (\sigma, \sigma') \in \tau \Rightarrow f(\sigma') < f(\sigma)$

The best known *well-ordered sets* are **naturals** $\langle \mathbb{N}, \leq \rangle$ and **ordinals** $\langle \mathbb{O}, \leq \rangle$

Ranking Functions

Example

```
¹x ← [-∞, +∞]
while ²(1 - x < 0) do
    ³x ← x - 1
done⁴
```

$$\Sigma \overset{\text{def}}{=} \{1,2,3,4\} \times \mathscr{E}$$

$$\begin{aligned}\tau \overset{\text{def}}{=} & \{((1,\rho),(2,\rho[X \mapsto v])) \mid \rho \in \mathscr{E}, v \in \mathbb{Z}\} \\ \cup & \{((2,\rho),(3,\rho)) \mid \mid \rho \in \mathscr{E}, \exists v \in E[\![1-x]\!]\rho : v < 0\} \\ \cup & \{((3,\rho),(2,\rho[X \mapsto v])) \mid \rho \in \mathscr{E}, v \in E[\![x-1]\!]\rho\} \\ \cup & \{((2,\rho),(4,\rho)) \mid \mid \rho \in \mathscr{E}, \exists v \in E[\![1-x]\!]\rho : v \nless 0\}\end{aligned}$$

Ranking Functions

Example

^{1}x ← [-∞, +∞]
while ^{2}(1 - x < 0) **do**
    ^{3}x ← x - 1
done^{4}

Most obvious ranking function:
a mapping $f\colon \Sigma \rightharpoonup \mathbb{O}$ **from each program state to**
(an upper bound on) **the number of steps until termination**

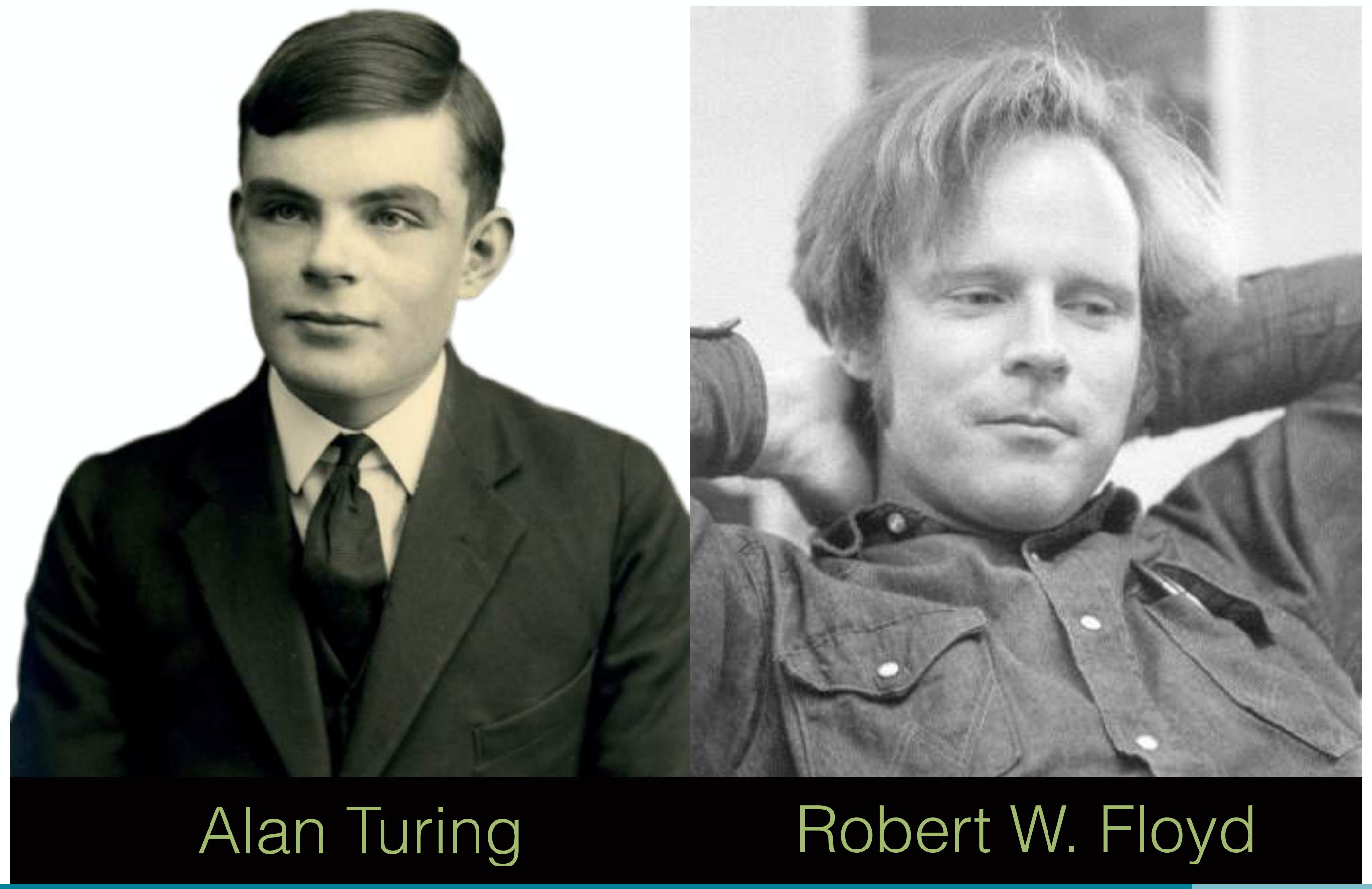
We define $f\colon \Sigma \rightharpoonup \mathbb{O}$ by partitioning with respect to the program control points, i.e., $f\colon \mathscr{L} \to (\mathscr{E} \rightharpoonup \mathbb{O})$

$$f(\mathbf{4}) \stackrel{\text{def}}{=} \lambda\rho.0$$

$$f(\mathbf{2}) \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} 1 & 1-\rho(x) \not< 0 \\ 2\rho(x)-1 & 1-\rho(x) < 0 \end{cases}$$

$$f(\mathbf{3}) \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} 2 & 2-\rho(x) \not< 0 \\ 2\rho(x)-2 & 2-\rho(x) < 0 \end{cases}$$

$$f(\mathbf{1}) \stackrel{\text{def}}{=} \lambda\rho\,.\,\omega$$

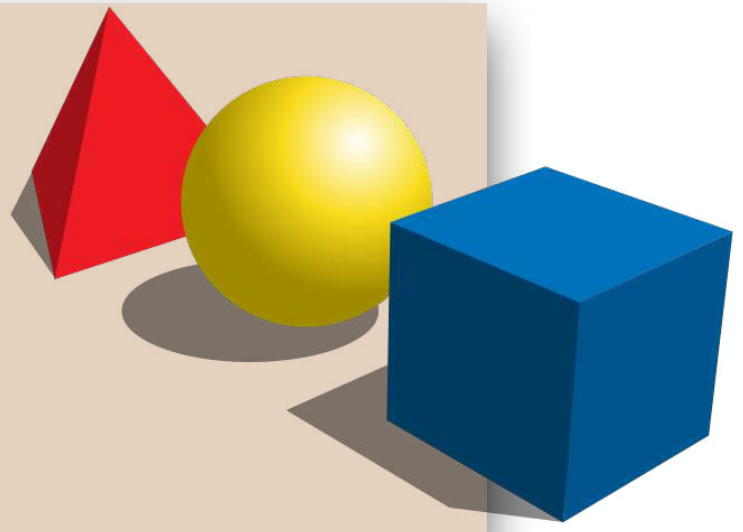


Static Termination Analysis

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Static Termination Analysis

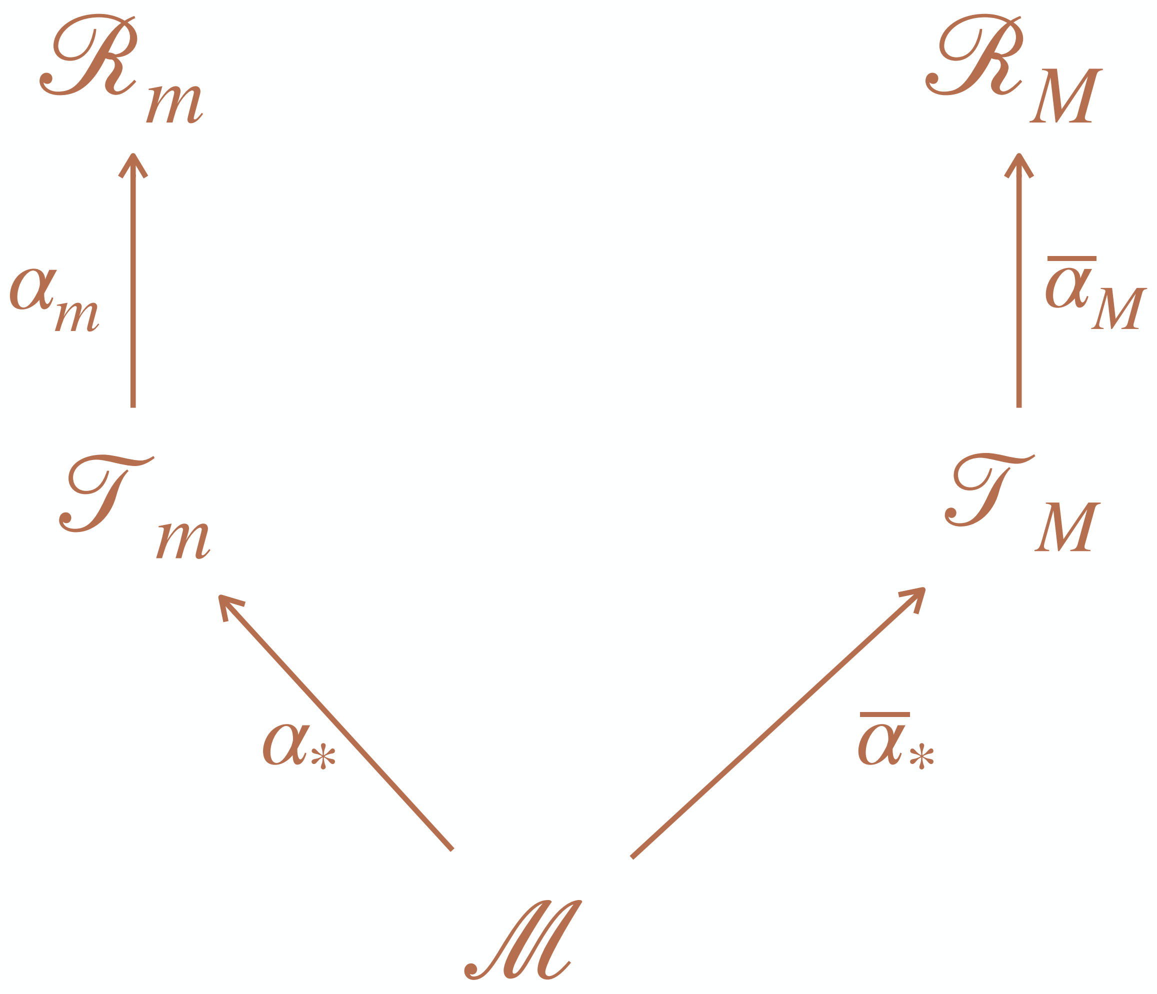
Program Termination Semantics

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

concrete semantics
mathematical models of the program behavior

(Yet Another) Hierarchy of Semantics

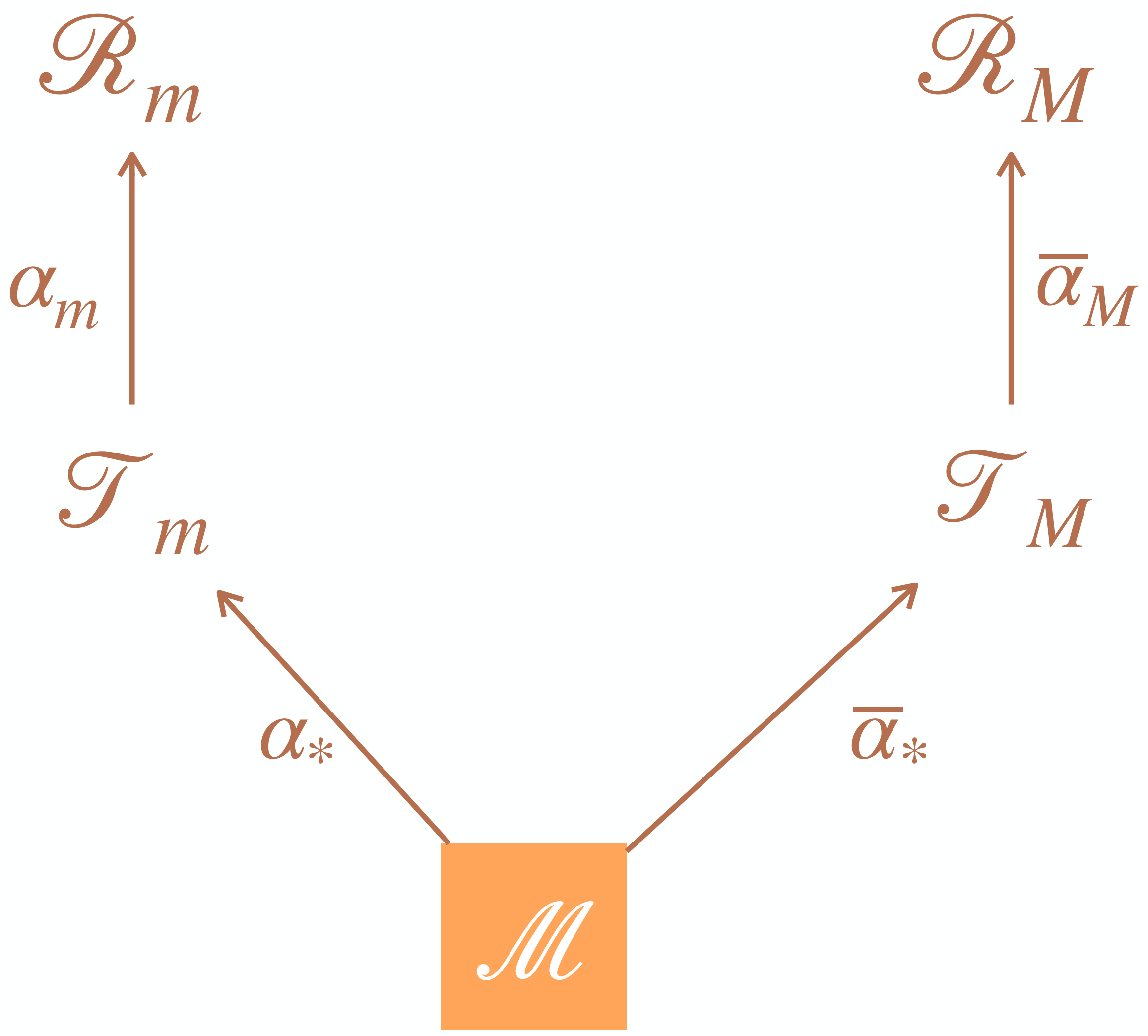


termination semantics

termination trace semantics

maximal trace semantics

(Yet Another) Hierarchy of Semantics



termination semantics

termination trace semantics

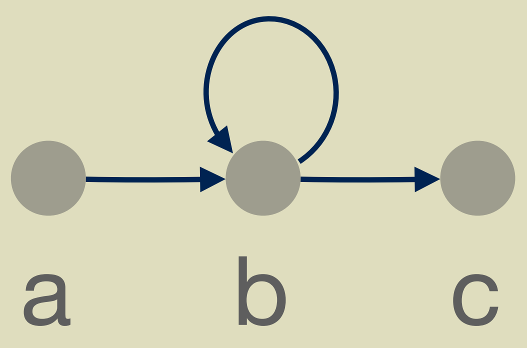
maximal trace semantics

Maximal Trace Semantics

Least Fixpoint Formulation

$$\mathscr{M} = \mathsf{lfp}^{\sqsubseteq}_{\Sigma^\omega} F$$

$$F(T) \stackrel{\text{def}}{=} \mathscr{B} \cup \tau ; T$$



- $F^0(\emptyset) = \Sigma^\omega$
- $F^1(F^0) = \{c\} \cup \{ab\Sigma^\omega, bb\Sigma^\omega, bc\Sigma^\omega\}$
- $F^2(F^1) = \{bc, c\} \cup \{abb\Sigma^\omega, bbb\Sigma^\omega, abc\Sigma^\omega, bbc\Sigma^\omega\}$
- $F^3_p(F^2_p) = \{abc, bbc, bc, c\} \cup \{abbb\Sigma^\omega, bbbb\Sigma^\omega, abbc\Sigma^\omega, bbbc\Sigma^\omega\}$

$$\mathscr{M} = \{ab^ic, b^ic, c \mid i \geq 1\} \cup \{ab^\omega, b^\omega\}$$

Maximal Trace Semantics

Example

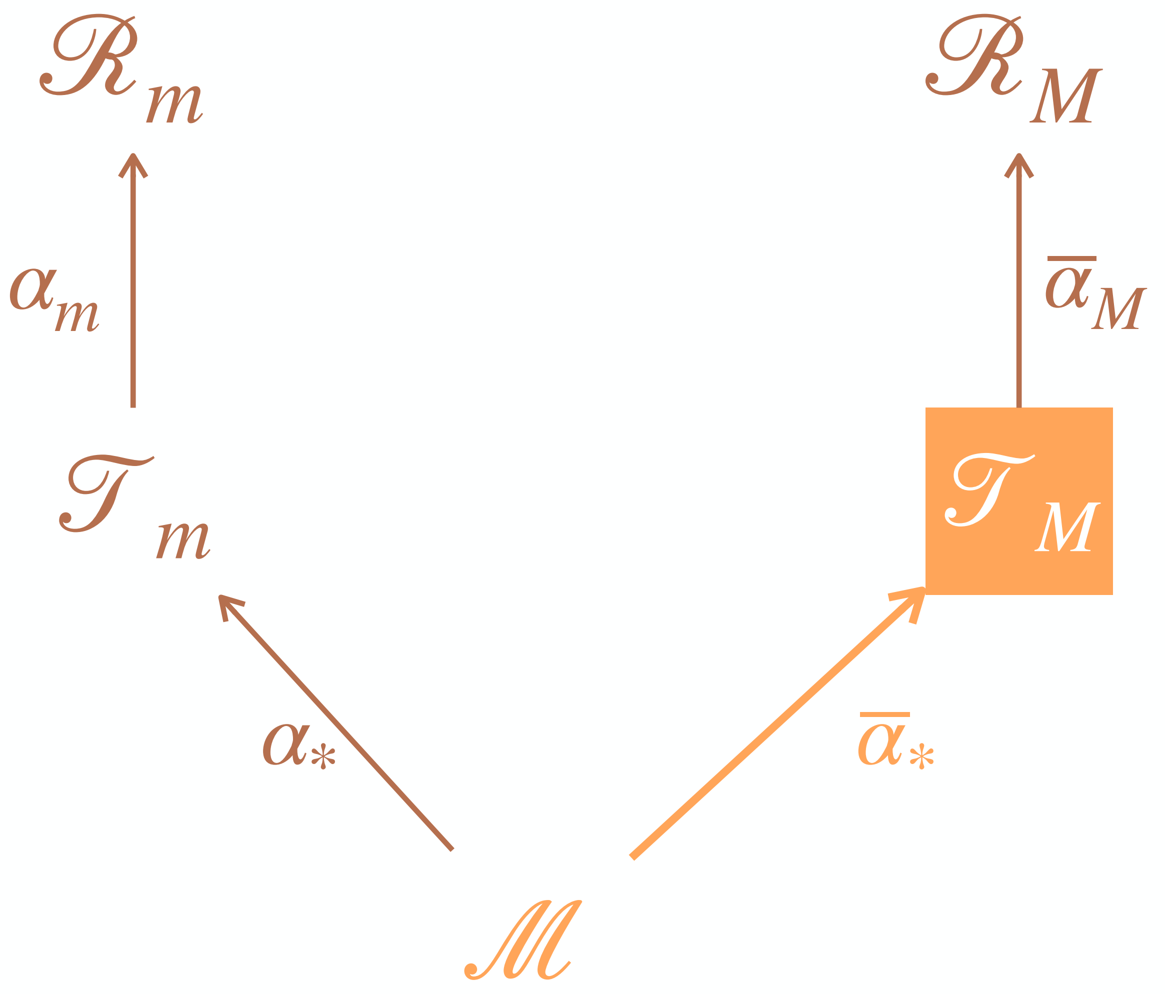
while ^{1}([-∞, +∞] ≠ 0) **do**
    ^{2}**skip**
done^3

$$\Sigma \stackrel{\text{def}}{=} \{1,2,3\} \times \mathscr{E}$$

$$\tau \stackrel{\text{def}}{=} \{((1,\rho),(2,\rho)) \mid \rho \in \mathscr{E}\} \\ \cup \{((2,\rho),(1,\rho)) \mid \rho \in \mathscr{E}\} \\ \cup \{((1,\rho),(3,\rho)) \mid \rho \in \mathscr{E}\}$$

$$\mathscr{M} \stackrel{\text{def}}{=} \{(1,\rho)(2,\rho)^*(3,\rho) \mid \rho \in \mathscr{E}\} \cup \{(1,\rho)(2,\rho)^\omega \mid \rho \in \mathscr{E}\}$$

(Yet Another) Hierarchy of Semantics



termination semantics

definite termination trace semantics

maximal trace semantics

Definite Termination Trace Semantics

Definite Termination Abstraction

$\langle \mathscr{P}(\Sigma^\infty), \sqsubseteq \rangle \xrightarrow{\overline{\alpha}_*} \langle \mathscr{P}(\Sigma^*), \subseteq \rangle$

$$\overline{\alpha}_*(T) \stackrel{\text{def}}{=} \{t \in T \cap \Sigma^* \mid \text{nhdb}(t, T \cap \Sigma^\omega) = \varnothing\}$$

where $\text{nhdb}(t, T) \stackrel{\text{def}}{=} \{t' \in T \mid \text{pf}(t) \cap \text{pf}(t') \neq \varnothing\}$

$$\text{pf}(t) \stackrel{\text{def}}{=} \{t' \in \Sigma^\infty \backslash \{\epsilon\} \mid \exists t'' \in \Sigma^\infty \colon t = t' \cdot t''\}$$

Example:
$\alpha_*(\{ab, aba, bb, ba^\omega\}) = \{ab, aba\}$ since $\text{pf}(bb) \cap \text{pf}(ba^\omega) = \{b\} \neq \varnothing$

Order Theory

Tarskian Fixpoint Transfer

Theorem

Let $\langle C, \leq, \vee, \wedge, \bot, \top \rangle$ and $\langle A, \sqsubseteq, \sqcup, \sqcap, \bot^\#, \top^\# \rangle$ be **complete lattices**, let $f\colon C \to C$ and $f^\#\colon A \to A$ be **monotonic functions**, and let $\alpha\colon C \to A$ be **an abstraction function** that

- is a **complete $\wedge$-morphism** ($\forall S \subseteq C\colon f(\wedge S) = \sqcap \{f(s) \mid s \in S\}$),
- **satisfies** $f^\# \circ \alpha \sqsubseteq \alpha \circ f$,
- **satisfies the post-fixpoint correspondence** $\forall a^\# \in A : f^\#(a^\#) \sqsubseteq a^\# \Rightarrow \exists a \in C : f(a) \leq d \wedge \alpha(a) = a^\#$ (i.e., each abstract post-fixpoint of $f^\#$ is the abstraction by α of some concrete post-fixpoint of f).

Then, we have the fixpoint abstraction $\alpha(\mathsf{lfp}_c^{\leq} f) = \mathsf{lpf}_{\alpha(c)}^{\sqsubseteq} f^\#$

Maximal to Definite Termination Trace Semantics

Tarskian Fixpoint Transfer

Maximal Trace Semantics

$$\mathscr{M} = \mathsf{lfp}^{\sqsubseteq}_{\Sigma^\omega} F$$

$$F(T) \stackrel{\text{def}}{=} \mathscr{B} \cup \tau ; T$$

monotonic

Definite Termination Abstraction

$\langle \mathscr{P}(\Sigma^\infty), \sqsubseteq \rangle$ complete lattice — $\overline{\alpha}_*$ — $\langle \mathscr{P}(\Sigma^*), \subseteq \rangle$ complete lattice

$$\overline{\alpha}_*(T) \stackrel{\text{def}}{=} \{t \in T \cap \Sigma^* \mid \mathsf{nhdb}(t, T \cap \Sigma^\omega) = \varnothing\}$$

complete $\wedge$-morphism

$\overline{F}_* \circ \overline{\alpha}_* \sqsubseteq \overline{\alpha}_* \circ F$

$\forall \overline{T} \in \mathscr{P}(\Sigma^*) : \overline{F}_*(\overline{T}) \subseteq \overline{T} \Rightarrow \exists T \in \mathscr{P}(\Sigma^\infty) : F(T) \sqsubseteq T \wedge \overline{\alpha}_*(T) = \overline{T}$

Exercise: prove this 🙂

Definite Termination Trace Semantics

$$\mathscr{T}_M = \mathsf{lfp}^{\subseteq}_{\varnothing} \overline{F}_*$$

$$\overline{F}_*(T) \stackrel{\text{def}}{=} \mathscr{B} \cup ((\tau ; T) \cap (\Sigma^+ \setminus (\tau ; (\Sigma^+ \setminus T))))$$

monotonic

$$\overline{\alpha}_*(\mathscr{M}) = \overline{\alpha}_*(\mathsf{lfp}^{\sqsubseteq}_{\Sigma^\omega} F) = \mathsf{lfp}^{\subseteq}_{\varnothing} \overline{F}_* = \mathscr{T}_M$$

Definite Termination Trace Semantics

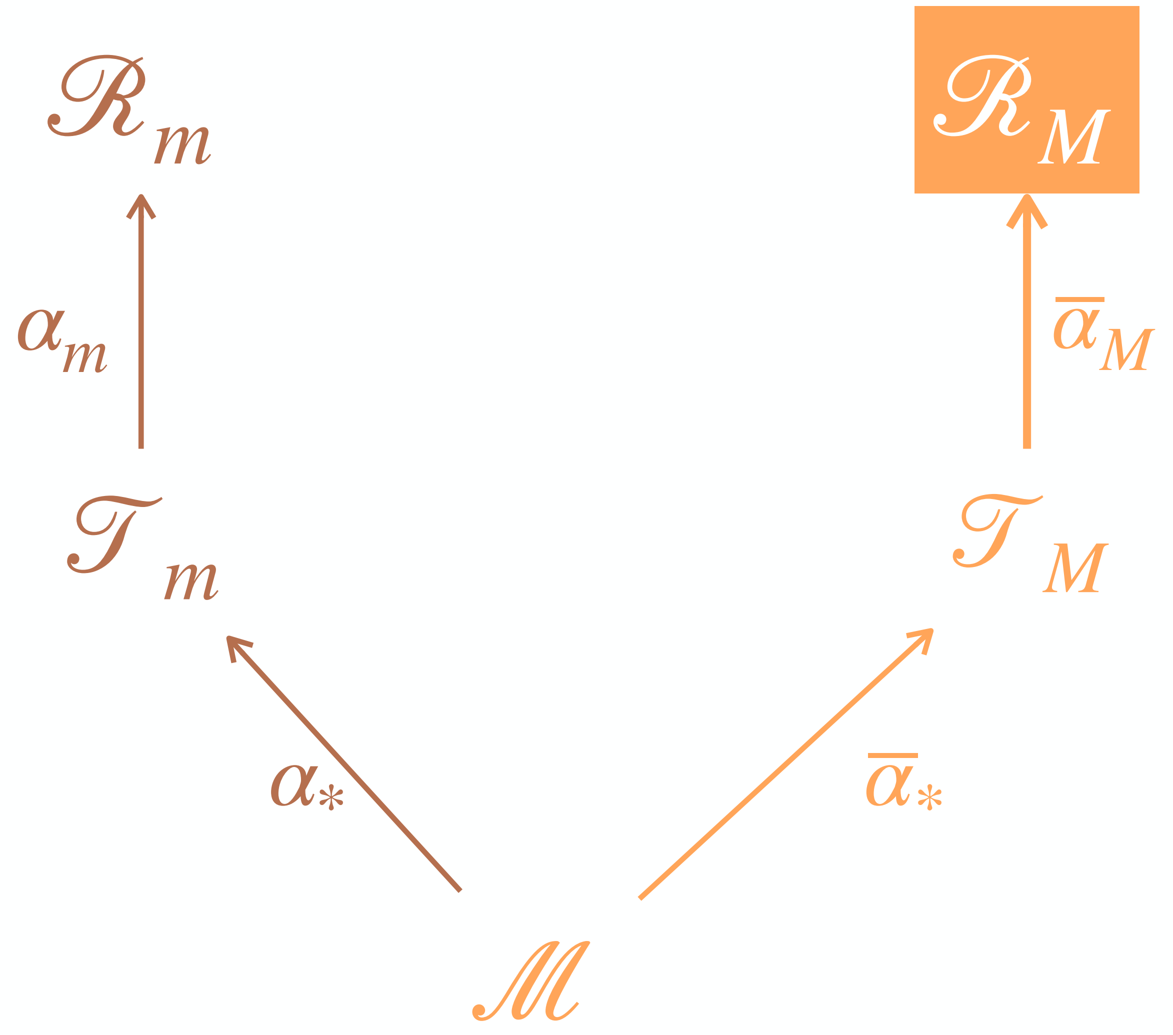
Example

while [1]([-∞, +∞] ≠ 0) **do**
[2]**skip**
done[3]

$$\mathscr{M} \stackrel{\text{def}}{=} \{(\mathbf{1}, \rho)(\mathbf{2}, \rho)^*(\mathbf{3}, \rho) \mid \rho \in \mathscr{E}\} \cup \{(\mathbf{1}, \rho)(\mathbf{2}, \rho)^\omega \mid \rho \in \mathscr{E}\}$$

$$\mathscr{T}_M \stackrel{\text{def}}{=} \varnothing$$

(Yet Another) Hierarchy of Semantics



definite termination semantics

termination trace semantics

maximal trace semantics

Definite Termination Semantics

Definite Ranking Abstraction

count execution steps backwards

$\langle \mathscr{P}(\Sigma^*), \subseteq \rangle$ $\quad \xrightarrow{\alpha_M} \quad$ $\langle \Sigma \rightharpoonup \mathbb{O}, \preceq \rangle$

$f_2 \preceq f_1 \stackrel{\text{def}}{=} \text{dom}(f_1) \subseteq \text{dom}(f_2) \wedge \forall x \in \text{dom}(f_1) \colon f_1(x) \leq f_2(x)$

$$\overline{\alpha}_M(T) \stackrel{\text{def}}{=} \overline{\alpha}_V(\overrightarrow{\alpha}(T))$$

where $\overline{\alpha}_V(\emptyset) \stackrel{\text{def}}{=} \dot{\emptyset}$

$$\overline{\alpha}_V(r)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \forall \sigma' \in \Sigma \colon (\sigma, \sigma') \notin r \\ \sup\{\overline{\alpha}_V(r)\sigma' + 1 \mid \sigma' \in \text{dom}(\overline{\alpha}_V(r)) \wedge (\sigma, \sigma') \in r\} & \text{otherwise} \end{cases}$$

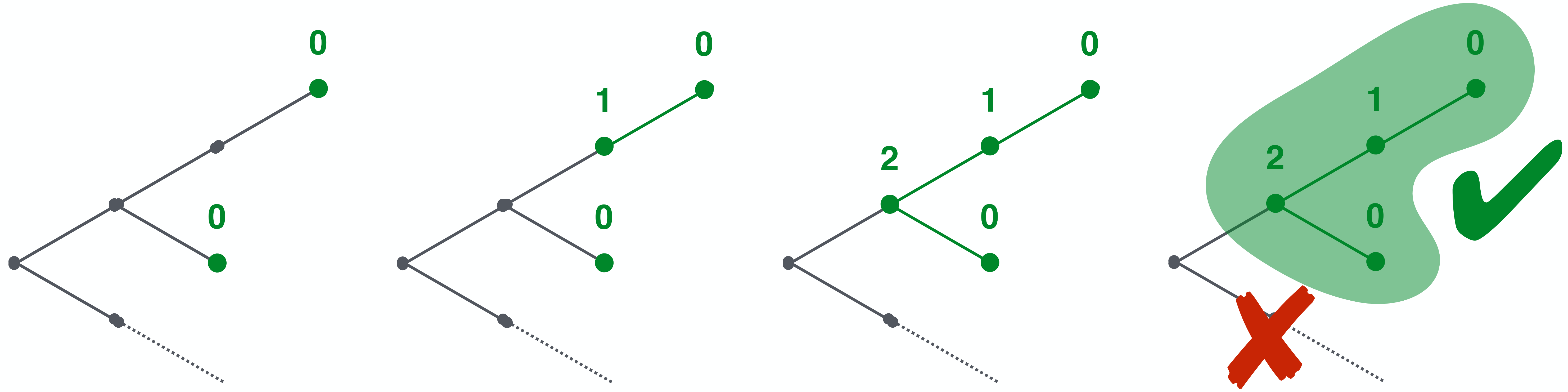
$$\overrightarrow{\alpha}(T) \stackrel{\text{def}}{=} \{(\sigma, \sigma') \in \Sigma \times \Sigma \mid \exists t \in \Sigma^*, t' \in \Sigma^\infty \colon t\sigma\sigma't' \in T\}$$

Definite Termination Semantics

Least Fixpoint Formulation

$$\mathcal{R}_M \stackrel{\text{def}}{=} \overline{\alpha}_M(\mathcal{T}_M) = \text{lfp}^{\preceq}\, \overline{F}_M$$

$$\overline{F}_M(f)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \sigma \in \mathcal{B} \\ \text{sup}\{f(\sigma') + 1 \mid (\sigma, \sigma') \in \tau\} & \sigma \in \widetilde{\text{pre}}_\tau(\text{dom}(f)) \\ \text{undefined} & \text{otherwise} \end{cases}$$



Theorem

A program **must terminate** for traces starting from a set of initial state I if and only if $I \subseteq \text{dom}(\mathcal{R}_M)$

Definite Termination Semantics

Denotational Formulation

We define the $\mathscr{R}_M \colon \Sigma \rightharpoonup \mathbb{O}$ by partitioning with respect to $\mathscr{L}$, i.e., $\mathscr{R}_M \colon \mathscr{L} \to (\mathscr{E} \rightharpoonup \mathbb{O})$.

Thus, for each program instruction stmt, we define a transformer $\mathscr{R}_M[\![\mathsf{stmt}]\!] \colon (\mathscr{E} \rightharpoonup \mathbb{O}) \to (\mathscr{E} \rightharpoonup \mathbb{O})$:

- $\mathscr{R}_M[\![X \leftarrow e]\!]$
- $\mathscr{R}_M[\![\mathbf{if}\ \ e \bowtie 0\ \mathbf{then}\ s\ \mathbf{end}]\!]$
- $\mathscr{R}_M[\![\mathbf{while}\ \ e \bowtie 0\ \mathbf{do}\ s\ \mathbf{done}]\!]$
- $\mathscr{R}_M[\![s_1; s_2]\!]$

$$
\begin{aligned}
stmt ::= &\ {}^{\ell}X \leftarrow expr^{\ell} \\
\mid &\ \textsf{if}\ {}^{\ell}expr \bowtie 0\ \textsf{then}\ stmt\ \textsf{end}^{\ell} \\
\mid &\ \textsf{while}\ {}^{\ell}expr \bowtie 0\ \textsf{do}\ stmt\ \textsf{done}^{\ell} \\
\mid &\ stmt; stmt
\end{aligned}
$$

Definite Termination Semantics

$\mathscr{R}_M[\![X \leftarrow e]\!]$

$$\mathscr{R}_M[\![X \leftarrow e]\!]f \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} \sup\{f(\rho[X \mapsto v]) + 1 \mid v \in E[\![e]\!]\rho\} & \exists \bar{v} \in E[\![e]\!]\rho \wedge \forall v \in E[\![e]\!]\rho\colon \rho[X \mapsto v] \in \text{dom}(f) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Example:

Let $\mathbb{V} = \{x\}$ and $f\colon \mathscr{E} \rightharpoonup \mathbb{O}$ defined as follows:

$$f(\rho) \stackrel{\text{def}}{=} \begin{cases} 2 & \rho(x) = 1 \\ 3 & \rho(x) = 2 \\ \text{undefined} & \text{otherwise} \end{cases}$$

We have

$$\mathscr{R}_M[\![x \leftarrow x + [1,2]]\!]f \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} 4 & \rho(x) = 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

Definite Termination Semantics

$\mathscr{R}_M[\![\textbf{if } e \bowtie 0 \textbf{ then } s \textbf{ end}]\!]$

$$\mathscr{R}_M[\![\textbf{if } e \bowtie 0 \textbf{ then } s \textbf{ end}]\!]f \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} ① \\ ② \\ ③ \\ \text{undefined} & \text{otherwise} \end{cases}$$

① $\sup\{\mathscr{R}_M[\![s]\!]f(\rho)+1,\ f(\rho)+1\}$ $\quad \rho \in \text{dom}(\mathscr{R}_M[\![s]\!]f) \cap \text{dom}(f) \wedge \exists v_1, v_2 \in E[\![e]\!]\rho : v_1 \bowtie 0 \wedge v_2 \not\bowtie 0$

② $\mathscr{R}_M[\![s]\!]f(\rho)+1$ $\quad \rho \in \text{dom}(\mathscr{R}_M[\![s]\!]f) \wedge \forall v \in E[\![e]\!]\rho : v \bowtie 0$

③ $f(\rho)+1$ $\quad \rho \in \text{dom}(f) \wedge \forall v \in E[\![e]\!]\rho : v \not\bowtie 0$

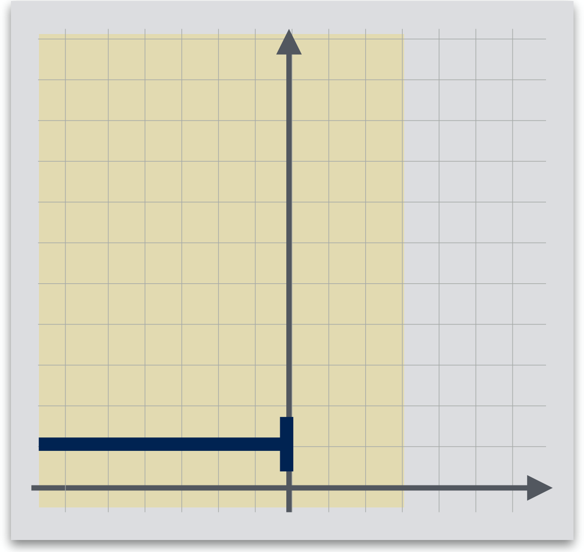
Definite Termination Semantics

$\mathscr{R}_M[\![\mathbf{if}\ \ e \bowtie 0\ \mathbf{then}\ s\ \mathbf{end}]\!]$ (continue)

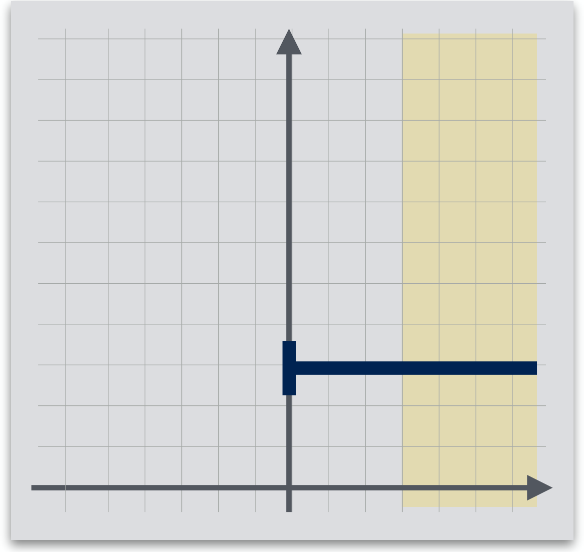
Example:

Let $\mathbb{V} = \{x\}$ and $f\colon \mathscr{E} \rightharpoonup \mathbb{O}$, and $\mathscr{R}_M[\![s]\!]f$ defined as follows:

$$f \overset{\text{def}}{=} \lambda\rho\,.\begin{cases} 1 & \rho(x) \leq 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

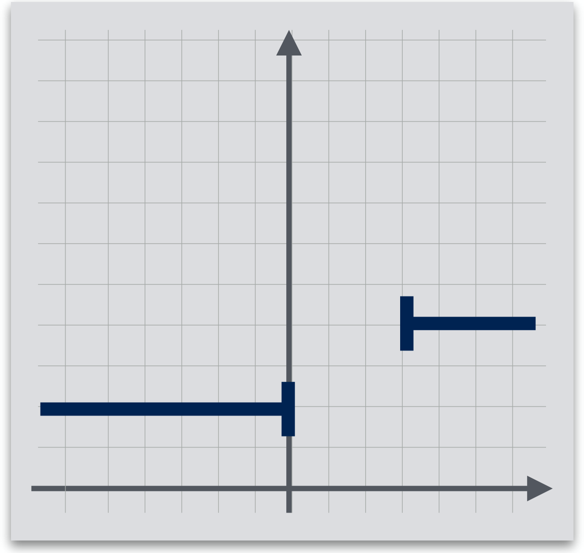


$$\mathscr{R}_M[\![s]\!]f \overset{\text{def}}{=} \lambda\rho\,.\begin{cases} 3 & 0 \leq \rho(x) \\ \text{undefined} & \text{otherwise} \end{cases}$$



We have

$$\mathscr{R}_M[\![\mathbf{if}\ 3 - x < 0\ \mathbf{then}\ s]\!]f \overset{\text{def}}{=} \lambda\rho\,.\begin{cases} 2 & \rho(x) \leq 0 \\ 4 & 3 < \rho(x) \\ \text{undefined} & \text{otherwise} \end{cases}$$



and $\mathscr{R}_M[\![\mathbf{if}\ [-\infty, +\infty] \neq 0\ \mathbf{then}\ s]\!]f \overset{\text{def}}{=} \lambda\rho\,.\begin{cases} 4 & \rho(x) = 0 \\ \text{undefined} & \text{otherwise} \end{cases}$

Definite Termination Semantics

$\mathcal{R}_M[\![\textbf{while}\ \ e \bowtie 0\ \textbf{do}\ s\ \textbf{done}]\!]$

$$\mathcal{R}_M[\![\textbf{while}\ \ e \bowtie 0\ \textbf{do}\ s\ \textbf{done}]\!]f \stackrel{\text{def}}{=} \text{lfp}^{\preccurlyeq}_{\dot\emptyset}\ \overline{F}_M$$

$$\text{where } F_M(x) \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} \text{①} \\ \text{②} \\ \text{③} \\ \text{undefined} & \text{otherwise} \end{cases}$$

① $\sup\{\mathcal{R}_M[\![s]\!]x(\rho) + 1,\ f(\rho) + 1\}$ $\quad\rho \in \text{dom}(\mathcal{R}_M[\![s]\!]x) \cap \text{dom}(f) \wedge \exists v_1, v_2 \in E[\![e]\!]\rho : v_1 \bowtie 0 \wedge v_2 \not\bowtie 0$

② $\mathcal{R}_M[\![s]\!]x(\rho) + 1$ $\quad\rho \in \text{dom}(\mathcal{R}_M[\![s]\!]x) \wedge \forall v \in E[\![e]\!]\rho : v \bowtie 0$

③ $f(\rho) + 1$ $\quad\rho \in \text{dom}(f) \wedge \forall v \in E[\![e]\!]\rho : v \not\bowtie 0$

Definite Termination Semantics

$\mathcal{R}_M[\![s_1; s_2]\!]$

$$\mathcal{R}_M[\![s_1; s_2]\!]f \stackrel{\text{def}}{=} \mathcal{R}_M[\![s_1]\!](\mathcal{R}_M[\![s_2]\!]f)$$

Definite Termination Semantics

Denotational Formulation

Definition

The **definite termination semantics** $\mathcal{R}_M[\![\mathsf{s}^\ell]\!] : \mathcal{E} \rightharpoonup \mathbb{O}$ of a program s^ℓ is:

$$\mathcal{R}_M[\![\mathsf{s}^\ell]\!] \stackrel{\text{def}}{=} \mathcal{R}_M[\![\mathsf{s}]\!](\lambda\rho.0)$$

where $\mathcal{R}_M[\![\mathsf{s}]\!] : (\mathcal{E} \rightharpoonup \mathbb{O}) \to (\mathcal{E} \rightharpoonup \mathbb{O})$ is the definite termination semantics of each instruction s

Theorem

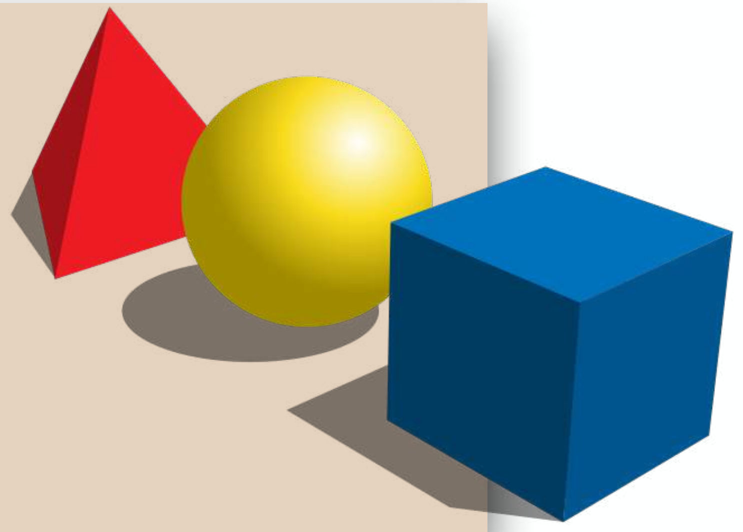
A program s^ℓ **must terminate** starting from a set of initial states I if and only if $I \subseteq \text{dom}(\mathcal{R}_M[\![\mathsf{s}^\ell]\!])$

Termination Static Analysis

Abstract Program Termination Semantics

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



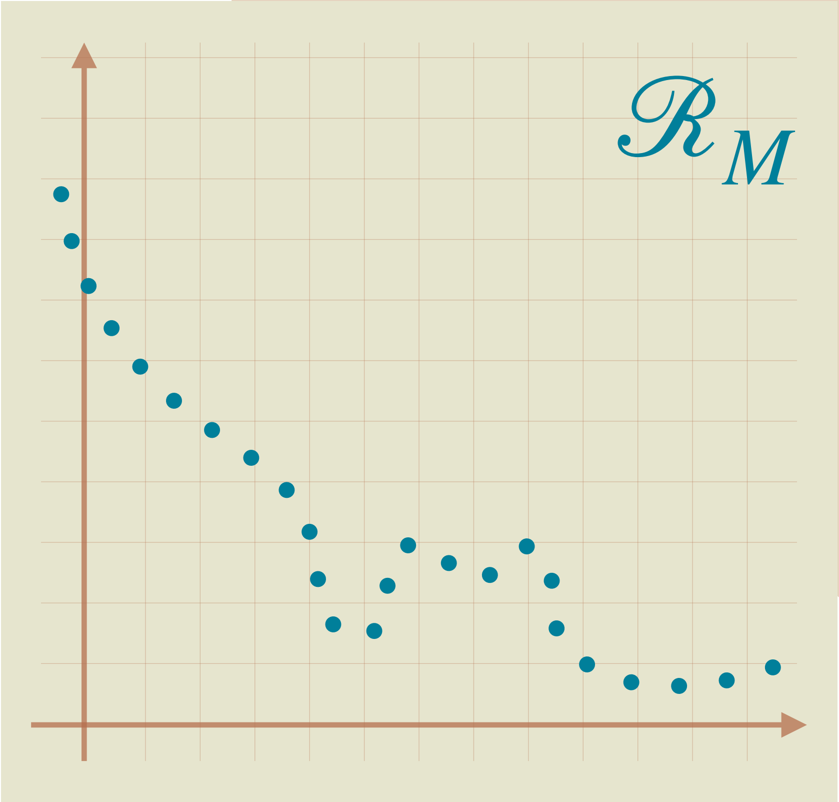
concrete semantics
mathematical models of the program behavior

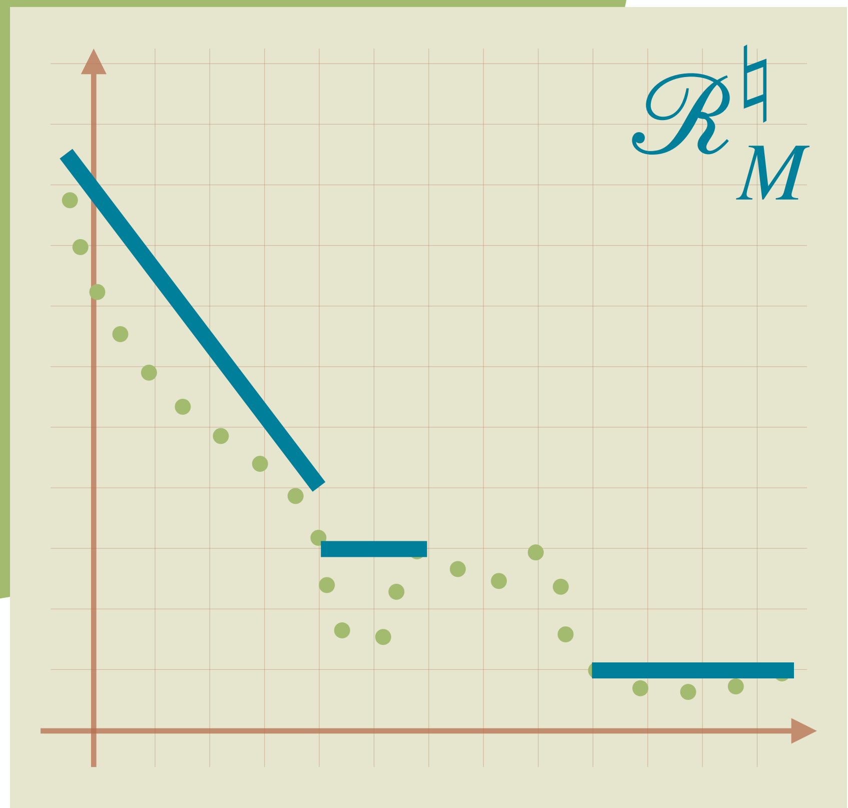
Piecewise-Defined Ranking Functions

γ

Definite Termination Semantics

Abstract Definite Termination Semantics



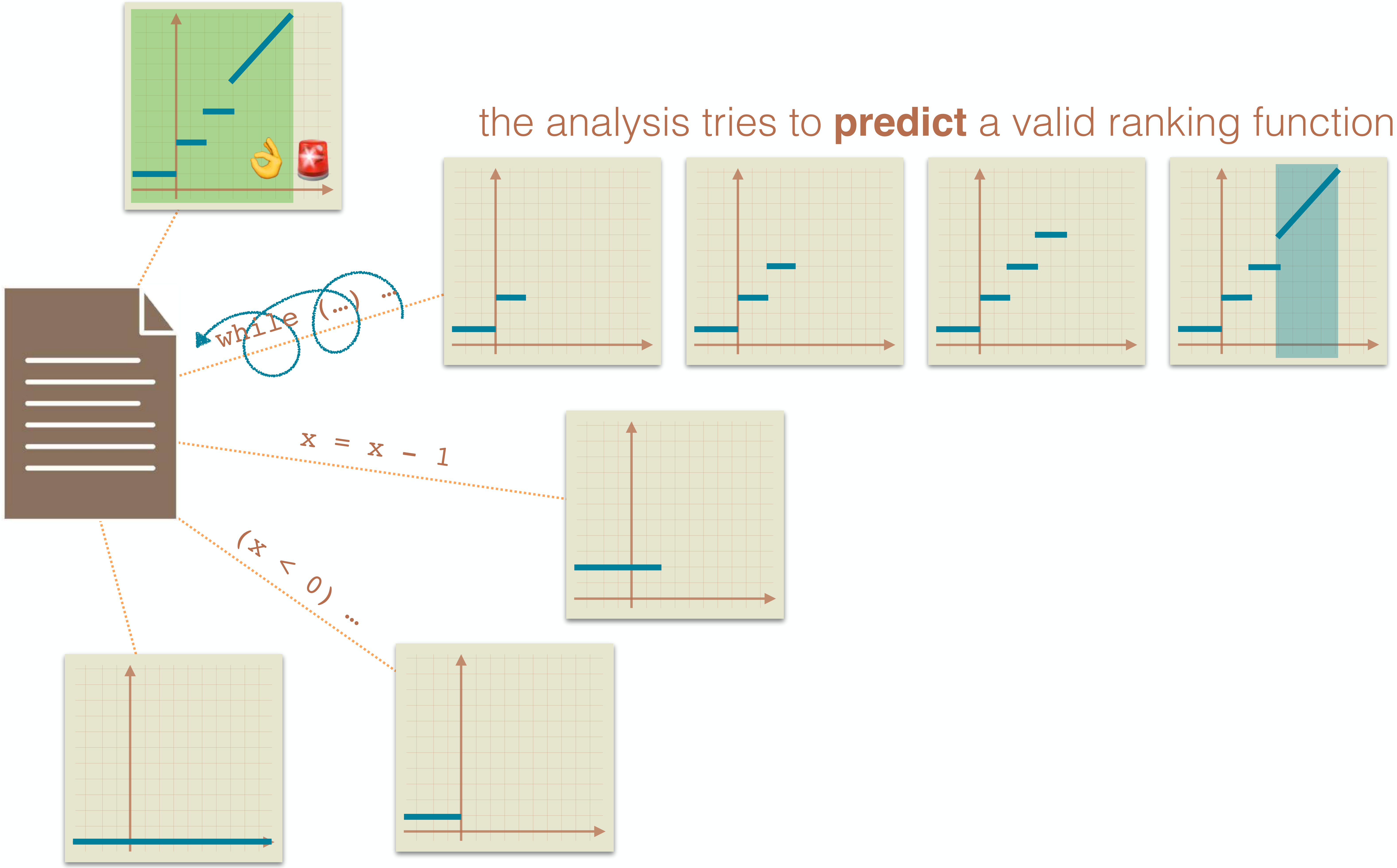


$$\mathrm{dom}(\gamma(\mathcal{R}^\natural_M)) \subseteq \mathrm{dom}(\mathcal{R}_M)$$

$$\forall x \in \mathrm{dom}(\gamma(\mathcal{R}^\natural_M)) : \mathcal{R}_M(x) \leq \gamma(\mathcal{R}^\natural_M)(x)$$

Termination Static Analysis

the analysis tries to **predict** a valid ranking function



Piecewise-Defined Function Abstraction

γ_A

$\mathcal{R}^{\#}_M[\![\text{stmt}^\ell]\!] \in \mathcal{A}$

$\langle \mathcal{E} \rightharpoonup \mathbb{O}, \preccurlyeq \rangle$ $\qquad$ $\langle \mathcal{A}, \preccurlyeq_A \rangle$

$\mathcal{R}_M[\![\text{stmt}^\ell]\!] \colon \mathcal{E} \rightharpoonup \mathbb{O}$

$f_1 \preccurlyeq f_2 \overset{\text{def}}{=} \text{dom}(f_1) \supseteq \text{dom}(f_2) \wedge \forall x \in \text{dom}(f_1) \colon f_1(x) \leq f_2(x)$

approximation order

By *pointwise lifiting* we obtain an abstraction $\mathcal{R}^{\#}_M$ of $\mathcal{R}_M$:

$\dot{\gamma}_A$

$\langle \mathcal{L} \rightarrow (\mathcal{E} \rightharpoonup \mathbb{O}), \dot{\preccurlyeq} \rangle$ $\qquad$ $\langle \mathcal{L} \rightarrow \mathcal{A}, \dot{\preccurlyeq}_A \rangle$

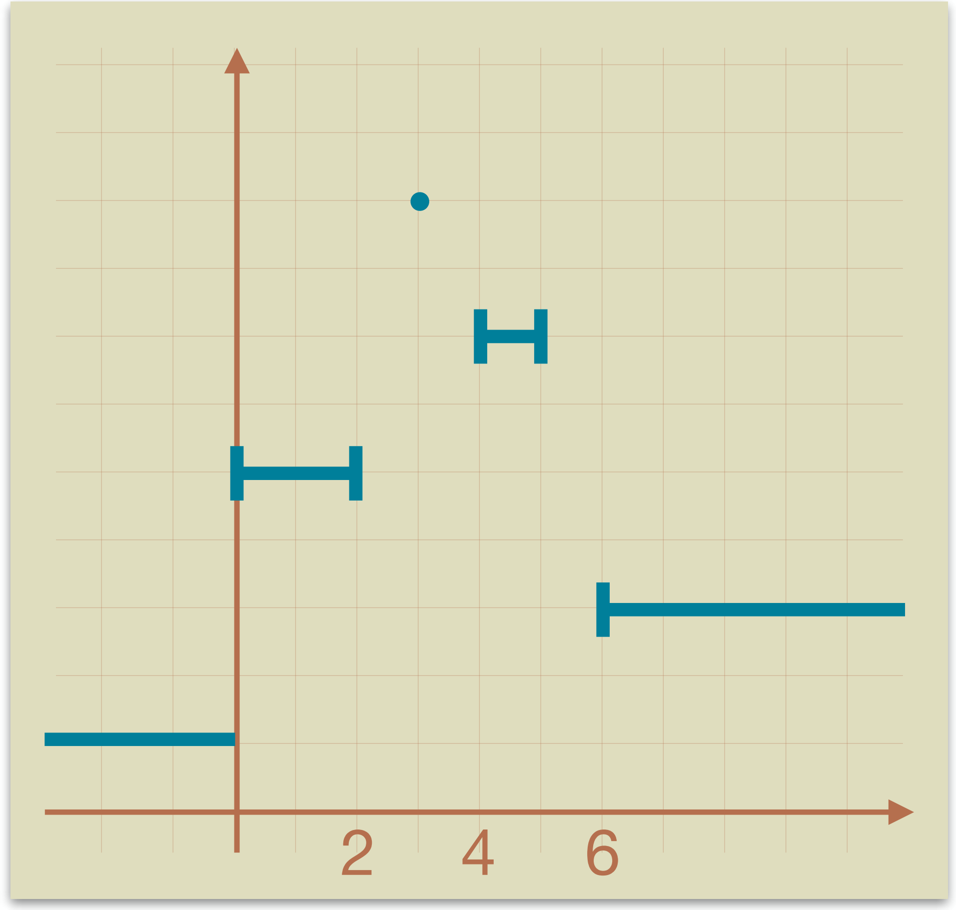
$\mathcal{R}_M \colon \mathcal{L} \rightarrow (\mathcal{E} \rightharpoonup \mathbb{O})$ $\qquad$ $\mathcal{R}^{\#}_M \colon \mathcal{L} \rightarrow \mathcal{A}$

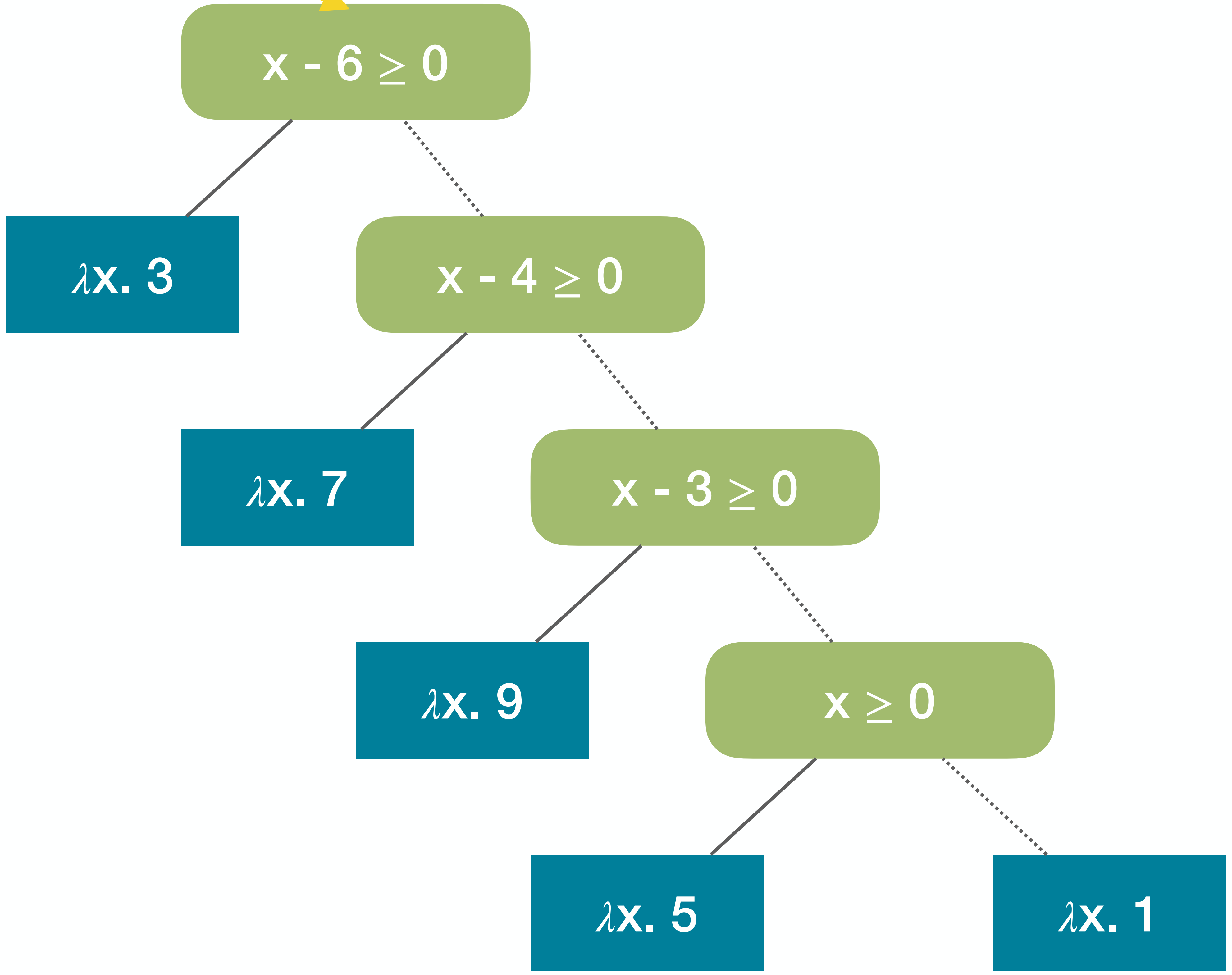
Piecewise-Defined Function Domain

$\langle \mathcal{A}, \preccurlyeq_A \rangle$

Example

^{1}x ← [-∞, +∞]
while ^{2}(x ≥ 0) **do**
    ^{3}x ← - 2 · x + 10
done^4

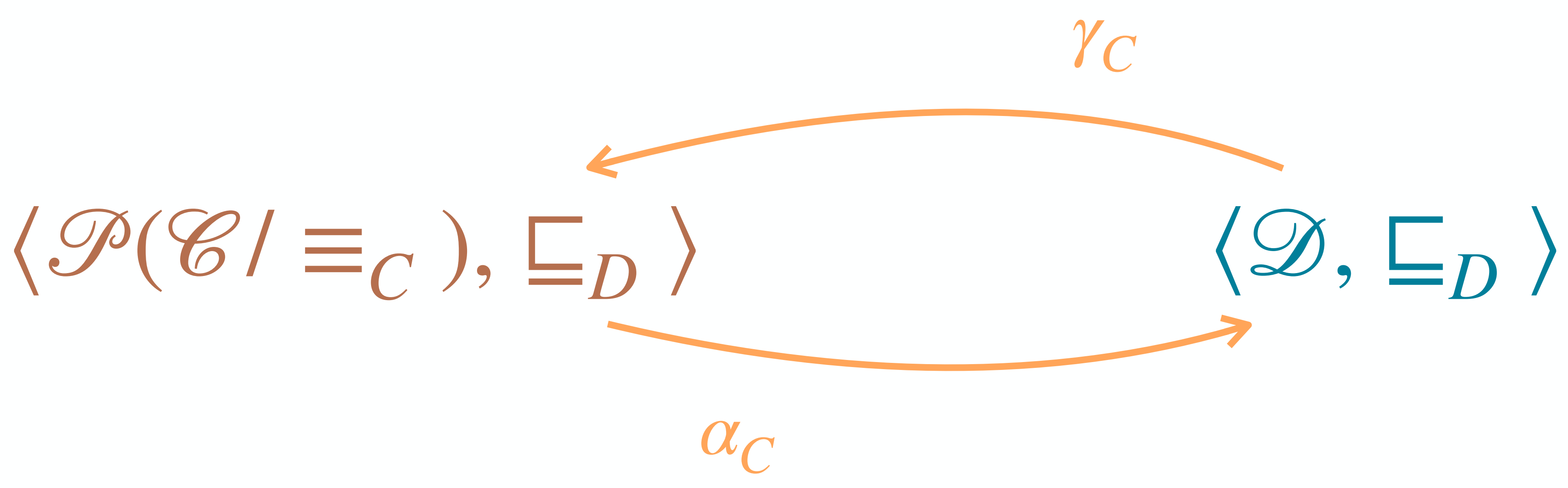


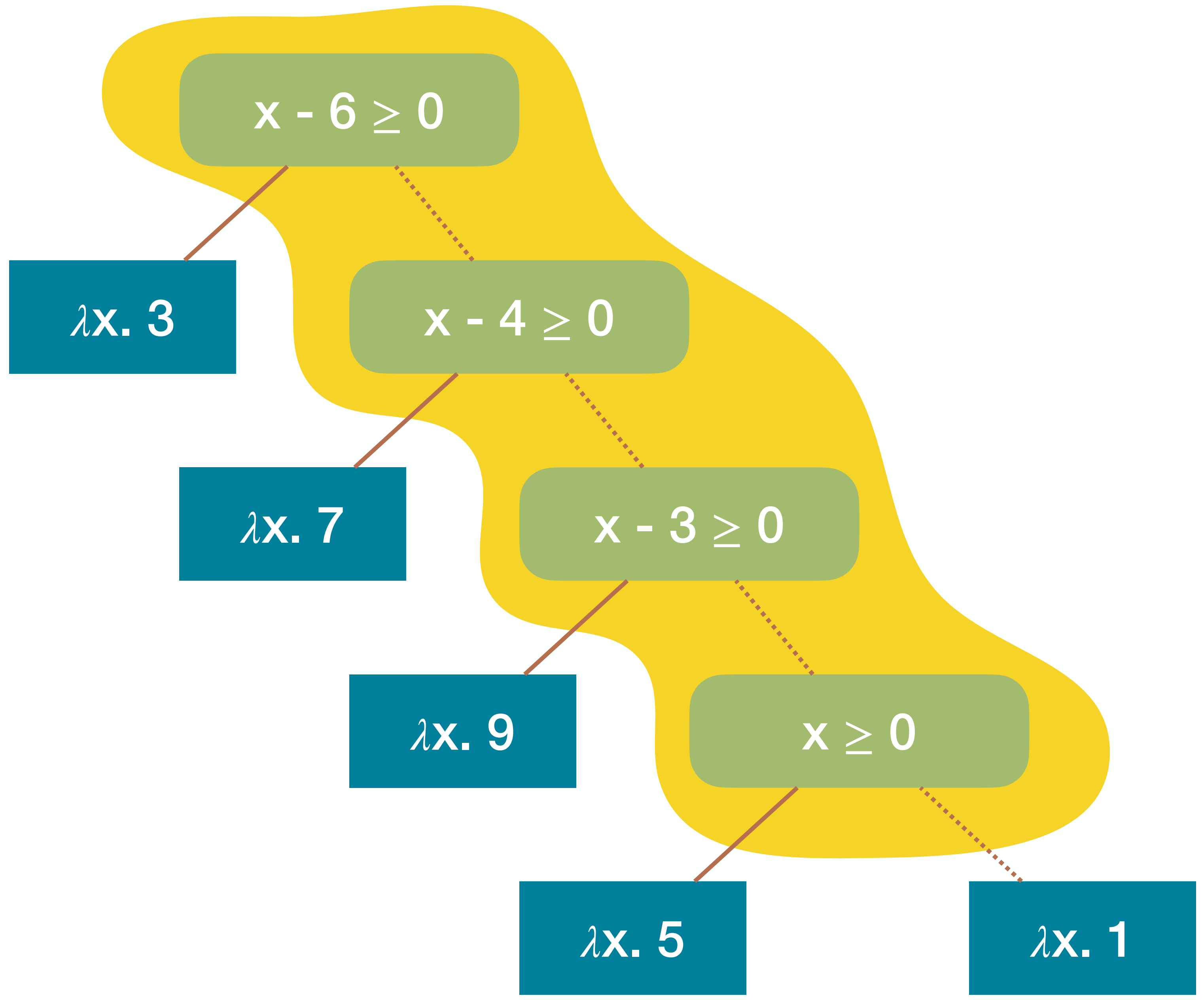


Piecewise-Defined Function Domain

Linear Constraints Auxiliary Abstract Domain

- Parameterized by an ***underlying numerical abstract domain*** $\langle \mathcal{D}, \sqsubseteq_D \rangle$ (e.g., intervals, polyhedra):





Example:

$$X \rightarrow [-\infty, 3],\ Y \rightarrow [0, \infty] \quad \xrightarrow{\gamma_C} \quad \{3 - X \geq 0,\ Y \geq 0\}$$

- $\mathcal{C}$ is a set of linear constraints *in canonical form*, equipped with a total order $\leq_C$:

$$\mathcal{C} \stackrel{\text{def}}{=} \{c_1 \cdot X_1 + c_k \cdot X_k + c_{k+1} \geq 0 \mid X_1, \ldots, X_k \in \mathbb{V} \wedge c_1, \ldots, c_{k+1} \in \mathbb{Z} \wedge \text{gcd}(|c_1|, \ldots, |c_{k+1}|) = 1\}$$

Natural-Valued Ranking Functions

Piecewise-Defined Function Domain

Functions Auxiliary Abstract Domain

- Parameterized by an ***underlying numerical abstract domain*** $\langle \mathcal{D}, \sqsubseteq_D \rangle$

- $\mathcal{F} \stackrel{\text{def}}{=} \{ \bot_F \} \cup (\mathbb{Z}^{|\mathbb{V}|} \to \mathbb{N}) \cup \{ \top_F \}$

  We consider **affine functions**:

  $$\mathcal{F}_A \stackrel{\text{def}}{=} \{ \bot_F \} \cup \{f\colon \mathbb{Z}^{|\mathbb{V}|} \to \mathbb{N} \mid f(X_1, \ldots, X_k) = \sum_{i=1}^{k} m_i \cdot X_i + q \} \cup \{ \top_F \}$$