Lecture 7 – Trace Semantics

AAA551: Programming Language Theory

Jihyeok Park

PLRG

2026 Spring

Recall

- Partial Orders and Lattices
    - Partial Orders
    - Complete Partial Orders (CPOs)
    - Lattices
    - Posets vs. CPOs vs. Lattices
- Fixpoint Theory
    - Monotone and Continuous Functions
    - Fixpoints
    - Tarski's Fixpoint Theorem
    - Kleene's Fixpoint Theorem
    - `while` Statements in IMP
    - `while` Statements in NIMP

From Small-Step Semantics to Trace Semantics

The small-step operational semantics describes how a program evolves *one step at a time* — but it only exposes a **single execution path**, not the **full space of possible behaviors**.

$$\sigma_0 \to \sigma_1 \to \sigma_2 \to \cdots \to \sigma_n$$

From Small-Step Semantics to Trace Semantics

The small-step operational semantics describes how a program evolves *one step at a time* — but it only exposes a **single execution path**, not the **full space of possible behaviors**.

$$\sigma_0 \to \sigma_1 \to \sigma_2 \to \cdots \to \sigma_n$$

Trace semantics lifts this to a *set* of all possible execution traces, sequences of states a program may pass through.

From Small-Step Semantics to Trace Semantics

The small-step operational semantics describes how a program evolves *one step at a time* — but it only exposes a **single execution path**, not the **full space of possible behaviors**.

$$\sigma_0 \to \sigma_1 \to \sigma_2 \to \cdots \to \sigma_n$$

Trace semantics lifts this to a *set* of all possible execution traces, sequences of states a program may pass through.

Why do we need this?

- Reason about **all** possible runs, not just one.
- Foundation for **safety** and **liveness** properties.
- Bridge toward **abstract interpretation** and program analysis.

Key concept: We model program execution as a **transition system** and define trace semantics over it.

Contents

1. Trace Semantics
    Transition Systems
    Traces
    Finite Trace Semantics
    Infinite Trace Semantics
    Compositionality

2. Example
    Non-Deterministic Imperative Language – NIMP

Contents

1. Trace Semantics
    - Transition Systems
    - Traces
    - Finite Trace Semantics
    - Infinite Trace Semantics
    - Compositionality

2. Example
    - Non-Deterministic Imperative Language – NIMP

Transition Systems

Definition (Transition System)

A **transition system** is a tuple $\mathcal{T} = (\mathbb{S}, \rightarrow)$ where:

- $\mathbb{S}$ is a **set of states**
- $\rightarrow \subseteq \mathbb{S} \times \mathbb{S}$ is a **transition relation**

Note that states might be infinite.

Transition Systems

Definition (Transition System)

A **transition system** is a tuple $\mathcal{T} = (\mathbb{S}, \to)$ where:

- $\mathbb{S}$ is a **set of states**
- $\to \subseteq \mathbb{S} \times \mathbb{S}$ is a **transition relation**

Note that states might be infinite.

A transition system is **deterministic** if a state has at most one successor:

$$\forall \sigma_0, \sigma_1, \sigma_2 \in \mathbb{S}.\ (\sigma_0 \to \sigma_1 \land \sigma_0 \to \sigma_2) \implies \sigma_1 = \sigma_2$$

Transition Systems

Definition (Transition System)

A **transition system** is a tuple $\mathcal{T} = (\mathbb{S}, \rightarrow)$ where:

- $\mathbb{S}$ is a **set of states**
- $\rightarrow \subseteq \mathbb{S} \times \mathbb{S}$ is a **transition relation**

Note that states might be infinite.

A transition system is **deterministic** if a state has at most one successor:

$$\forall \sigma_0, \sigma_1, \sigma_2 \in \mathbb{S}.\ (\sigma_0 \rightarrow \sigma_1 \land \sigma_0 \rightarrow \sigma_2) \implies \sigma_1 = \sigma_2$$

It is often **non-deterministic**, allowing multiple possible next states:

$$\exists \sigma_0, \sigma_1, \sigma_2 \in \mathbb{S}.\ (\sigma_0 \rightarrow \sigma_1 \land \sigma_0 \rightarrow \sigma_2) \land \sigma_1 \neq \sigma_2$$

Initial, Final, and Blocking States

We often define transition systems with explicit initial and final states:

$$\mathcal{T} = (\mathbb{S}, \rightarrow, \mathbb{S}_{\mathcal{I}}, \mathbb{S}_{\mathcal{F}})$$

- $\mathbb{S}_{\mathcal{I}} \subseteq \mathbb{S}$ is the set of **initial states** representing program entry points
- $\mathbb{S}_{\mathcal{F}} \subseteq \mathbb{S}$ is the set of **final states** representing normal termination

We call a state σ a **blocking** state if it has no successors:

$$\nexists \sigma' \in \mathbb{S}.\ \sigma \rightarrow \sigma'$$

All final states are blocking, but not all blocking states are final (e.g., runtime errors).

Traces

Definition (Traces)

For a given transition system $\mathcal{T} = (\mathbb{S}, \rightarrow)$,

- $\tau = \langle \sigma_0, \sigma_1, \ldots, \sigma_n \rangle$ is a **finite trace**
- $\tau = \langle \sigma_0, \sigma_1, \ldots \rangle$ is an **infinite trace**

We write:

- $\mathbb{S}^*$ for the set of all **finite traces**
- $\mathbb{S}^\omega$ for the set of all **infinite traces**
- $\mathbb{S}^\infty = \mathbb{S}^* \cup \mathbb{S}^\omega$ for the set of **finite and infinite traces**

Operators on Traces

Definition (Concatenation of Traces)

The **concatenation** of traces is defined as follows:

$$\begin{array}{lcl}
\langle \sigma_0, \ldots, \sigma_n \rangle \cdot \langle \sigma'_0, \ldots, \sigma'_m \rangle & \triangleq & \langle \sigma_0, \ldots, \sigma_n, \sigma'_0, \ldots, \sigma'_m \rangle \\
\langle \sigma_0, \ldots, \sigma_n \rangle \cdot \langle \sigma'_0, \sigma'_1, \ldots \rangle & \triangleq & \langle \sigma_0, \ldots, \sigma_n, \sigma'_0, \sigma'_1, \ldots \rangle \\
\langle \sigma_0, \sigma_1, \ldots \rangle \cdot \tau' & \triangleq & \langle \sigma_0, \sigma_1, \ldots \rangle
\end{array}$$

Operators on Traces

Definition (Concatenation of Traces)

The **concatenation** of traces is defined as follows:

$$\begin{array}{lcl}
\langle \sigma_0, \ldots, \sigma_n \rangle \cdot \langle \sigma'_0, \ldots, \sigma'_m \rangle & \triangleq & \langle \sigma_0, \ldots, \sigma_n, \sigma'_0, \ldots, \sigma'_m \rangle \\
\langle \sigma_0, \ldots, \sigma_n \rangle \cdot \langle \sigma'_0, \sigma'_1, \ldots \rangle & \triangleq & \langle \sigma_0, \ldots, \sigma_n, \sigma'_0, \sigma'_1, \ldots \rangle \\
\langle \sigma_0, \sigma_1, \ldots \rangle \cdot \tau' & \triangleq & \langle \sigma_0, \sigma_1, \ldots \rangle
\end{array}$$

Definition (Length of Traces)

The **length** of a trace τ is defined as follows:

$$\begin{array}{lcl}
|\epsilon| & \triangleq & 0 \\
|\langle \sigma_0, \ldots, \sigma_n \rangle| & \triangleq & n + 1 \\
|\langle \sigma_0, \sigma_1, \ldots \rangle| & \triangleq & \omega
\end{array}$$

where ϵ is the **empty trace** and ω represents an infinite length.

Operators on Traces

Definition (Prefix Order on Traces)

The **prefix order** on traces is defined as follows:

$$\begin{array}{lcl}
\langle \sigma_0, \ldots, \sigma_n \rangle \preceq \langle \sigma'_0, \ldots, \sigma'_m \rangle & \iff & n \leq m \land \forall i \leq n.\ \sigma_i = \sigma'_i \\
\langle \sigma_0, \sigma_1, \ldots \rangle \preceq \langle \sigma'_0, \sigma'_1, \ldots \rangle & \iff & \forall i \geq 0.\ \sigma_i = \sigma'_i \\
\langle \sigma_0, \ldots, \sigma_n \rangle \preceq \langle \sigma'_0, \sigma'_1, \ldots \rangle & \iff & n \leq m \land \forall i \leq n.\ \sigma_i = \sigma'_i
\end{array}$$

Finite Trace Semantics

Definition (Finite Trace Semantics)

The **finite trace semantics** of a transition system $\mathcal{T} = (\mathbb{S}, \rightarrow)$ is the set of all finite traces:

$$[\![\mathcal{T}]\!]^* = \{\langle \sigma_0, \ldots, \sigma_n \rangle \in \mathbb{S}^* \mid \forall i < n.\ \sigma_i \rightarrow \sigma_{i+1}\}$$

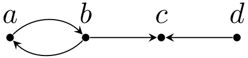
Finite Trace Semantics

Definition (Finite Trace Semantics)

The **finite trace semantics** of a transition system $\mathcal{T} = (\mathbb{S}, \to)$ is the set of all finite traces:

$$[\![\mathcal{T}]\!]^* = \{\langle \sigma_0, \ldots, \sigma_n \rangle \in \mathbb{S}^* \mid \forall i < n.\ \sigma_i \to \sigma_{i+1}\}$$

$$\mathcal{T} = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (d, c)\})$$



The finite trace semantics of $\mathcal{T}$ is:

$$\begin{array}{lll} [\![\mathcal{T}]\!]^* = \{ & & \\ & \epsilon, & \langle c \rangle, \\ & \langle a, b, \ldots, a, b \rangle, & \langle b, a, \ldots, b, a \rangle, \\ & \langle a, b, \ldots, a, b, a \rangle, & \langle b, a, \ldots, b, a, b \rangle, \\ & \langle a, b, \ldots, a, b, a, c \rangle, & \langle b, a, \ldots, b, a, b, c \rangle, \\ & \langle d \rangle & \langle d, c \rangle \\ \} & & \end{array}$$

Finite Trace Semantics

In $\mathcal{T} = (\mathbb{S}, \to, \mathbb{S}_{\mathcal{I}}, \mathbb{S}_{\mathcal{F}})$, a trace $\tau = \langle \sigma_0, \ldots, \sigma_n \rangle \in [\![\mathcal{T}]\!]^*$ is:

- a **initial trace** if it starts from an initial state:

$$\sigma_0 \in \mathbb{S}_{\mathcal{I}}$$

- a **final trace** if it ends in a final state:

$$\sigma_n \in \mathbb{S}_{\mathcal{F}}$$

- a **blocking trace** if it ends in a blocking state:

$$\forall \tau' \in \mathbb{S}^*. \ \tau \preceq \tau' \implies \tau = \tau'$$

- a **maximal trace** if it is both initial and final

Finite Trace Semantics

Let's redefine the finite trace semantics using Kleene's fixpoint theorem:

Lemma

Let $\mathcal{I} = \{\epsilon\} \cup \{\langle\sigma\rangle \mid \sigma \in \mathbb{S}\}$ *and* F_* *be the function:*

$$\begin{array}{llll} F_* : & \mathcal{P}(\mathbb{S}^*) & \rightarrow & \mathcal{P}(\mathbb{S}^*) \\ & X & \mapsto & \mathcal{I} \cup \{\langle\sigma_0, \ldots, \sigma_n, \sigma'\rangle \mid \langle\sigma_0, \ldots, \sigma_n\rangle \in X \land \sigma_n \rightarrow \sigma'\} \end{array}$$

Then, the least fixpoint of F_* *is the finite trace semantics of* $\mathcal{T}$*:*

$$[\![\mathcal{T}]\!]^* = \mathbf{lfp}(F_*) = \bigcup_{n \in \mathbb{Z}} F_*^n(\varnothing)$$

Finite Trace Semantics

Let's redefine the finite trace semantics using Kleene's fixpoint theorem:

Lemma

Let $\mathcal{I} = \{\epsilon\} \cup \{\langle\sigma\rangle \mid \sigma \in \mathbb{S}\}$ *and* F_* *be the function:*

$$\begin{array}{rccl} F_* : & \mathcal{P}(\mathbb{S}^*) & \rightarrow & \mathcal{P}(\mathbb{S}^*) \\ & X & \mapsto & \mathcal{I} \cup \{\langle\sigma_0, \ldots, \sigma_n, \sigma'\rangle \mid \langle\sigma_0, \ldots, \sigma_n\rangle \in X \land \sigma_n \rightarrow \sigma'\} \end{array}$$

Then, the least fixpoint of F_* *is the finite trace semantics of* $\mathcal{T}$*:*

$$[\![\mathcal{T}]\!]^* = \mathbf{lfp}(F_*) = \bigcup_{n \in \mathbb{Z}} F_*^n(\varnothing)$$

- $(\mathcal{P}(\mathbb{S}^*), \subseteq)$ is a **complete partial order (CPO)**.

  We already learned that the powerset poset is a CPO in the previous lecture.

Finite Trace Semantics

- F_* is a **continuous** function on $(\mathcal{P}(\mathbb{S}^*), \subseteq)$.

  Let $\mathcal{X}$ be any non-empty subset of $\mathcal{P}(\mathbb{S}^*)$. Then:

$$
\begin{array}{l}
F_*(\bigcup_{X \in \mathcal{X}} X) \\
= \mathcal{I} \cup \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \langle \sigma_0, \ldots, \sigma_n \rangle \in \bigcup_{X \in \mathcal{X}} X \land \sigma_n \to \sigma'\} \\
= \mathcal{I} \cup \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \exists X \in \mathcal{X}.\ \langle \sigma_0, \ldots, \sigma_n \rangle \in X \land \sigma_n \to \sigma'\} \\
= \mathcal{I} \cup (\bigcup_{X \in \mathcal{X}} \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \langle \sigma_0, \ldots, \sigma_n \rangle \in X \land \sigma_n \to \sigma'\}) \\
= \bigcup_{X \in \mathcal{X}} (\mathcal{I} \cup \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \langle \sigma_0, \ldots, \sigma_n \rangle \in X \land \sigma_n \to \sigma'\}) \\
= \bigcup_{X \in \mathcal{X}} F_*(X)
\end{array}
$$

  It means that this is true for any non-empty chain $\mathcal{X}$, so F_* is continuous.

Finite Trace Semantics

- F_* is a **continuous** function on $(\mathcal{P}(\mathbb{S}^*), \subseteq)$.

  Let $\mathcal{X}$ be any non-empty subset of $\mathcal{P}(\mathbb{S}^*)$. Then:

  $$\begin{array}{l} F_*(\bigcup_{X \in \mathcal{X}} X) \\ = \mathcal{I} \cup \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \langle \sigma_0, \ldots, \sigma_n \rangle \in \bigcup_{X \in \mathcal{X}} X \wedge \sigma_n \to \sigma'\} \\ = \mathcal{I} \cup \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \exists X \in \mathcal{X}.\ \langle \sigma_0, \ldots, \sigma_n \rangle \in X \wedge \sigma_n \to \sigma'\} \\ = \mathcal{I} \cup (\bigcup_{X \in \mathcal{X}} \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \langle \sigma_0, \ldots, \sigma_n \rangle \in X \wedge \sigma_n \to \sigma'\}) \\ = \bigcup_{X \in \mathcal{X}} (\mathcal{I} \cup \{\langle \sigma_0, \ldots, \sigma_n, \sigma' \rangle \mid \langle \sigma_0, \ldots, \sigma_n \rangle \in X \wedge \sigma_n \to \sigma'\}) \\ = \bigcup_{X \in \mathcal{X}} F_*(X) \end{array}$$

  It means that this is true for any non-empty chain $\mathcal{X}$, so F_* is continuous.

- By Kleene's fixpoint theorem, the **least fixpoint of F_* exists**:

  $$\mathbf{lfp}(F_*) = \bigcup_{n \in \mathbb{Z}} F_*^n(\varnothing)$$

Finite Trace Semantics

- **$\llbracket \mathcal{T} \rrbracket^*$ is equal to $\mathbf{lfp}(F_*)$.**

  We will prove it by induction on n:

  $$\forall k < n.\ \langle \sigma_0, \ldots, \sigma_k \rangle \in F_*^n(\varnothing) \iff \langle \sigma_0, \ldots, \sigma_k \rangle \in \llbracket \mathcal{T} \rrbracket^*$$

  - Base case: $n = 0$. Trivially true since:

  $$F_*^0(\varnothing) = \varnothing$$

  - Base case: $n = 1$. Trivially true since:

  $$F_*(\varnothing) = \mathcal{I} = \{\epsilon\} \cup \{\langle \sigma \rangle \mid \sigma \in \mathbb{S}\}$$

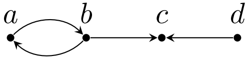
  - Inductive case: $n > 0$.

  $$\begin{array}{ll} \langle \sigma_0, \ldots, \sigma_k, \sigma' \rangle \in F_*^{n+1}(\varnothing) & \\ \iff \langle \sigma_0, \ldots, \sigma_k \rangle \in F_*^n(\varnothing) \land \sigma_k \to \sigma' & \\ \iff \langle \sigma_0, \ldots, \sigma_k \rangle \in \llbracket \mathcal{T} \rrbracket^* \land \sigma_k \to \sigma' & \text{(by I.H.)} \\ \iff \langle \sigma_0, \ldots, \sigma_k, \sigma' \rangle \in \llbracket \mathcal{T} \rrbracket^* & \end{array}$$

Finite Trace Semantics

For example, consider the transition system:

$$\mathcal{T} = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (d, c)\})$$



Then, the iteration of F_* on $\varnothing$ is:

$$\begin{array}{lcl}
F_*^0(\varnothing) & = & \varnothing \\
F_*^1(\varnothing) & = & \{\epsilon, \langle a \rangle, \langle b \rangle, \langle c \rangle, \langle d \rangle\} \\
F_*^2(\varnothing) & = & F_*^1(\varnothing) \cup \{\langle a, b \rangle, \langle b, a \rangle, \langle b, c \rangle, \langle d, c \rangle\} \\
F_*^3(\varnothing) & = & F_*^2(\varnothing) \cup \{\langle a, b, a \rangle, \langle a, b, c \rangle, \langle b, a, b \rangle\} \\
F_*^4(\varnothing) & = & F_*^3(\varnothing) \cup \{\langle a, b, a, b \rangle, \langle b, a, b, a \rangle, \langle b, a, b, c \rangle\} \\
F_*^5(\varnothing) & = & F_*^4(\varnothing) \cup \{\langle a, b, a, b, a \rangle, \langle a, b, a, b, c \rangle, \langle b, a, b, a, b \rangle\} \\
\vdots & & \vdots
\end{array}$$

The traces of $[\![\mathcal{T}]\!]^*$ of length n appear in $F_*^n(\varnothing)$.

Infinite Trace Semantics

Question

Is it possible to represent non terminating behaviors using finite trace semantics?

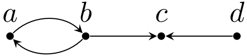
Infinite Trace Semantics

Question

Is it possible to represent non terminating behaviors using finite trace semantics?

No. For example, consider the transition system:

$$\mathcal{T} = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (d, c)\})$$



Then, the infinite trace semantics of $\mathcal{T}$ exists:

$$\langle a, b, a, b, a, b, \ldots \rangle \in \mathbb{S}^{\omega}$$

However, it cannot be represented in the finite trace semantics of $\mathcal{T}$ since it is not a finite trace.

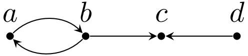
Infinite Trace Semantics

Question

Is it possible to represent non terminating behaviors using finite trace semantics?

No. For example, consider the transition system:

$$\mathcal{T} = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (d, c)\})$$



Then, the infinite trace semantics of $\mathcal{T}$ exists:

$$\langle a, b, a, b, a, b, \ldots \rangle \in \mathbb{S}^\omega$$

However, it cannot be represented in the finite trace semantics of $\mathcal{T}$ since it is not a finite trace.

Let's define the **infinite trace semantics** of $\mathcal{T}$.

Infinite Trace Semantics

Definition (Infinite Trace Semantics)

The **infinite trace semantics** of a transition system $\mathcal{T} = (\mathbb{S}, \rightarrow)$ is the set of all infinite traces:

$$[\![\mathcal{T}]\!]^\omega = \{\langle \sigma_0, \sigma_1, \ldots \rangle \in \mathbb{S}^\omega \mid \forall i \geq 0.\ \sigma_i \rightarrow \sigma_{i+1}\}$$

The **finite and infinite trace semantics** of $\mathcal{T}$ is:

$$[\![\mathcal{T}]\!]^\infty = [\![\mathcal{T}]\!]^* \uplus [\![\mathcal{T}]\!]^\omega$$

Infinite Trace Semantics

Definition (Infinite Trace Semantics)

The **infinite trace semantics** of a transition system $\mathcal{T} = (\mathbb{S}, \to)$ is the set of all infinite traces:

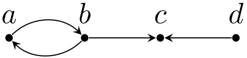
$$[\![\mathcal{T}]\!]^\omega = \{\langle \sigma_0, \sigma_1, \ldots \rangle \in \mathbb{S}^\omega \mid \forall i \geq 0.\ \sigma_i \to \sigma_{i+1}\}$$

The **finite and infinite trace semantics** of $\mathcal{T}$ is:

$$[\![\mathcal{T}]\!]^\infty = [\![\mathcal{T}]\!]^* \uplus [\![\mathcal{T}]\!]^\omega$$

For example, the infinite trace semantics of the following system $\mathcal{T}$ is:

$$\mathcal{T} = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (d, c)\})$$



$$[\![\mathcal{T}]\!]^\omega = \{ \\ \langle a, b, a, b, a, b, \ldots \rangle, \\ \langle b, a, b, a, b, a, \ldots \rangle \\ \}$$

Infinite Trace Semantics

Similar to $[\![\mathcal{T}]\!]^*$, we can also define $[\![\mathcal{T}]\!]^\omega$ using Kleene's fixpoint theorem.

Lemma

Let $\mathcal{I} = \{\langle\sigma\rangle \mid \sigma \in \mathbb{S}\}$ *and* F_ω *be the function:*

$$\begin{array}{rlcl} F_\omega : & \mathcal{P}(\mathbb{S}^\omega) & \to & \mathcal{P}(\mathbb{S}^\omega) \\ & X & \mapsto & \{\langle\sigma', \sigma_0, \sigma_1, \ldots\rangle \mid \langle\sigma_0, \sigma_1, \ldots\rangle \in X \wedge \sigma' \to \sigma_0\} \end{array}$$

Then, the greatest fixpoint of F_ω *is the infinite trace semantics of* $\mathcal{T}$*:*

$$[\![\mathcal{T}]\!]^\omega = \mathbf{gfp}(F_\omega) = \bigcap_{n \in \mathbb{Z}} F_\omega^n(\mathbb{S}^\omega)$$

Infinite Trace Semantics

Similar to $[\![\mathcal{T}]\!]^*$, we can also define $[\![\mathcal{T}]\!]^\omega$ using Kleene's fixpoint theorem.

Lemma

Let $\mathcal{I} = \{\langle\sigma\rangle \mid \sigma \in \mathbb{S}\}$ *and* F_ω *be the function:*

$$\begin{array}{rccl} F_\omega : & \mathcal{P}(\mathbb{S}^\omega) & \rightarrow & \mathcal{P}(\mathbb{S}^\omega) \\ & X & \mapsto & \{\langle\sigma', \sigma_0, \sigma_1, \ldots\rangle \mid \langle\sigma_0, \sigma_1, \ldots\rangle \in X \wedge \sigma' \rightarrow \sigma_0\} \end{array}$$

Then, the greatest fixpoint of F_ω *is the infinite trace semantics of* $\mathcal{T}$*:*

$$[\![\mathcal{T}]\!]^\omega = \mathbf{gfp}(F_\omega) = \bigcap_{n \in \mathbb{Z}} F_\omega^n(\mathbb{S}^\omega)$$

We need to use dual version with $\cap$-continuous functions and $\mathbf{gfp}$.

Infinite Trace Semantics

Similar to $[\![\mathcal{T}]\!]^*$, we can also define $[\![\mathcal{T}]\!]^\omega$ using Kleene's fixpoint theorem.

Lemma

Let $\mathcal{I} = \{\langle\sigma\rangle \mid \sigma \in \mathbb{S}\}$ and F_ω be the function:

$$\begin{array}{llll} F_\omega : & \mathcal{P}(\mathbb{S}^\omega) & \rightarrow & \mathcal{P}(\mathbb{S}^\omega) \\ & X & \mapsto & \{\langle\sigma', \sigma_0, \sigma_1, \ldots\rangle \mid \langle\sigma_0, \sigma_1, \ldots\rangle \in X \wedge \sigma' \rightarrow \sigma_0\} \end{array}$$

Then, the greatest fixpoint of F_ω is the infinite trace semantics of $\mathcal{T}$:

$$[\![\mathcal{T}]\!]^\omega = \mathbf{gfp}(F_\omega) = \bigcap_{n \in \mathbb{Z}} F_\omega^n(\mathbb{S}^\omega)$$

We need to use dual version with $\cap$-continuous functions and $\mathbf{gfp}$.

- $(\mathcal{P}(\mathbb{S}^\omega), \subseteq)$ is a **complete partial order (CPO)**.

  We already learned that the powerset poset is a CPO in the previous lecture.

Infinite Trace Semantics

- F_ω is a $\cap$-**continuous** function on $(\mathcal{P}(\mathbb{S}^\omega), \subseteq)$.

  Let $\mathcal{X}$ be any subset of $\mathcal{P}(\mathbb{S}^\omega)$. Then:

$$
\begin{aligned}
& F_\omega(\textstyle\bigcap_{X \in \mathcal{X}} X) \\
& = \{\langle \sigma', \sigma_0, \sigma_1, \ldots \rangle \mid \langle \sigma_0, \sigma_1, \ldots \rangle \in \textstyle\bigcap_{X \in \mathcal{X}} X \land \sigma' \to \sigma_0\} \\
& = \{\langle \sigma', \sigma_0, \sigma_1, \ldots \rangle \mid \forall X \in \mathcal{X}.\ \langle \sigma_0, \sigma_1, \ldots \rangle \in X \land \sigma' \to \sigma_0\} \\
& = \textstyle\bigcap_{X \in \mathcal{X}} \{\langle \sigma', \sigma_0, \sigma_1, \ldots \rangle \mid \langle \sigma_0, \sigma_1, \ldots \rangle \in X \land \sigma' \to \sigma_0\} \\
& = \textstyle\bigcap_{X \in \mathcal{X}} F_\omega(X)
\end{aligned}
$$

  It means that this is true for any non-universe chain $\mathcal{X}$, so F_ω is $\cap$-continuous.

Infinite Trace Semantics

- F_ω is a $\cap$-**continuous** function on $(\mathcal{P}(\mathbb{S}^\omega), \subseteq)$.

  Let $\mathcal{X}$ be any subset of $\mathcal{P}(\mathbb{S}^\omega)$. Then:

  $$\begin{array}{l} F_\omega(\bigcap_{X \in \mathcal{X}} X) \\ = \{\langle \sigma', \sigma_0, \sigma_1, \ldots \rangle \mid \langle \sigma_0, \sigma_1, \ldots \rangle \in \bigcap_{X \in \mathcal{X}} X \wedge \sigma' \to \sigma_0\} \\ = \{\langle \sigma', \sigma_0, \sigma_1, \ldots \rangle \mid \forall X \in \mathcal{X}.\ \langle \sigma_0, \sigma_1, \ldots \rangle \in X \wedge \sigma' \to \sigma_0\} \\ = \bigcap_{X \in \mathcal{X}} \{\langle \sigma', \sigma_0, \sigma_1, \ldots \rangle \mid \langle \sigma_0, \sigma_1, \ldots \rangle \in X \wedge \sigma' \to \sigma_0\} \\ = \bigcap_{X \in \mathcal{X}} F_\omega(X) \end{array}$$

  It means that this is true for any non-universe chain $\mathcal{X}$, so F_ω is $\cap$-continuous.

- By dual version of Kleene's fixpoint theorem, the **greatest fixpoint of F_ω exists**:

  $$\mathbf{gfp}(F_\omega) = \bigcap_{n \in \mathbb{Z}} F_\omega^n(\mathbb{S}^\omega)$$

Infinite Trace Semantics

- **$[\![\mathcal{T}]\!]^\omega$ is equal to $\mathbf{gfp}(F_\omega)$.**

  We will prove it by induction on n:

  $$\langle\sigma_0, \ldots, \sigma_n, \ldots\rangle \in F_\omega^n(\mathbb{S}^\omega) \iff \langle\sigma_0, \ldots, \sigma_n, \ldots\rangle \in [\![\mathcal{T}]\!]^\omega$$

  - Base case: $n = 0$. Trivially true since:

    $$F_\omega^0(\mathbb{S}^\omega) = \mathbb{S}^\omega$$

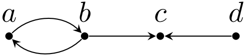
  - Inductive case: $n > 0$.

    $$\begin{array}{ll} \langle\sigma_0, \sigma_1, \ldots, \sigma_n, \ldots\rangle \in F_\omega^{n+1}(\mathbb{S}^\omega) & \\ \iff \langle\sigma_1, \ldots, \sigma_n, \ldots\rangle \in F_\omega^n(\mathbb{S}^\omega) \land \sigma_0 \to \sigma_1 & \\ \iff \langle\sigma_1, \ldots, \sigma_n, \ldots\rangle \in [\![\mathcal{T}]\!]^\omega \land \sigma_0 \to \sigma_1 & \text{(by I.H.)} \\ \iff \langle\sigma_0, \sigma_1, \ldots, \sigma_n, \ldots\rangle \in [\![\mathcal{T}]\!]^\omega & \end{array}$$

Infinite Trace Semantics

For example, consider the transition system:

$$\mathcal{T} = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (d, c)\})$$



Then, the iteration of F_ω on $\mathbb{S}^\omega$ is:

$$\begin{array}{lcl}
F_\omega^0(\mathbb{S}^\omega) & = & \mathbb{S}^\omega \\
F_\omega^1(\mathbb{S}^\omega) & = & \langle a, b\rangle \cdot \mathbb{S}^\omega \cup \langle b, a\rangle \cdot \mathbb{S}^\omega \cup \langle b, c\rangle \cdot \mathbb{S}^\omega \cup \langle d, c\rangle \cdot \mathbb{S}^\omega \\
F_\omega^2(\mathbb{S}^\omega) & = & \langle a, b, a\rangle \cdot \mathbb{S}^\omega \cup \langle a, b, c\rangle \cdot \mathbb{S}^\omega \cup \langle b, a, b\rangle \cdot \mathbb{S}^\omega \\
F_\omega^3(\mathbb{S}^\omega) & = & \langle a, b, a, b\rangle \cdot \mathbb{S}^\omega \cup \langle b, a, b, a\rangle \cdot \mathbb{S}^\omega \cup \langle b, a, b, c\rangle \cdot \mathbb{S}^\omega \\
F_\omega^4(\mathbb{S}^\omega) & = & \langle a, b, a, b, a\rangle \cdot \mathbb{S}^\omega \cup \langle a, b, a, b, c\rangle \cdot \mathbb{S}^\omega \cup \langle b, a, b, a, b\rangle \cdot \mathbb{S}^\omega \\
\vdots & & \vdots
\end{array}$$

At each iteration n, $F_\omega^n(\mathbb{S}^\omega)$ contains all infinite traces satisfying the transition relation for the first n steps ($\sigma_0 \to \sigma_1 \to \cdots \to \sigma_{n-1} \to \sigma_n$).

Compositionality

The trace semantics is **global** in the sense that it describes the behavior of the entire system.

However, we often want to reason about the behavior of individual components and how they compose together.

Compositionality is the principle that the semantics of a composite system can be derived from the semantics of its components.

Definition (Compositionality)

A semantics $[\![\cdot]\!]$ is **compositional** if is can be defined as a function of the semantics of its components:

$$\forall \pi = C[\pi_1, \ldots, \pi_n]. \qquad [\![\pi]\!] = F_C([\![\pi_1]\!], \ldots, [\![\pi_n]\!])$$

where $\pi_1, \ldots, \pi_n$ are the components of π.

Contents

1. Trace Semantics
    - Transition Systems
    - Traces
    - Finite Trace Semantics
    - Infinite Trace Semantics
    - Compositionality

2. Example
    - Non-Deterministic Imperative Language – NIMP

Non-Deterministic Imperative Language – NIMP

Let's define the **trace semantics** of NIMP.

$$
\begin{array}{lll}
\text{Expressions} & e ::= & n \mid [c_0, c_1] \mid x \mid e \;\texttt{+}\; e \mid e \;\texttt{*}\; e \mid e \;\texttt{<}\; e \mid \texttt{true} \mid \texttt{false} \\
\text{Statements} & s ::= & \ell : \texttt{skip}\ \ell \\
 & & \mid \ell : x \;\texttt{:=}\; e\ \ell \\
 & & \mid \ell : s \texttt{;} \ell : s\ \ell \\
 & & \mid \ell : \texttt{if}\ e\ \texttt{then}\ \{\ell : s\}\ \texttt{else}\ \{\ell : s\}\ \ell \\
 & & \mid \ell : \texttt{while}\ e\ \texttt{do}\ \{\ell : s\}\ \ell \\
\text{Values} & v ::= & n \mid \texttt{true} \mid \texttt{false}
\end{array}
$$

where $n \in \mathbb{Z}$, $c_0 \in \mathbb{Z} \cup \{-\infty\}$, $c_1 \in \mathbb{Z} \cup \{+\infty\}$, and $x \in \mathbb{X}$.

We use **labels** to simplify the states in the transition system.

Non-Deterministic Imperative Language – NIMP

Let's define the **trace semantics** of NIMP.

$$
\begin{array}{lll}
\text{Expressions} & e ::= & n \mid [c_0, c_1] \mid x \mid e \texttt{ + } e \mid e \texttt{ * } e \mid e \texttt{ < } e \mid \texttt{true} \mid \texttt{false} \\
\text{Statements} & s ::= & \ell : \texttt{skip}\ \ell \\
 & & \mid \ell : x \texttt{ := } e\ \ell \\
 & & \mid \ell : s \texttt{;} \ell : s\ \ell \\
 & & \mid \ell : \texttt{if}\ e\ \texttt{then}\ \{\ell : s\}\ \texttt{else}\ \{\ell : s\}\ \ell \\
 & & \mid \ell : \texttt{while}\ e\ \texttt{do}\ \{\ell : s\}\ \ell \\
\text{Values} & v ::= & n \mid \texttt{true} \mid \texttt{false}
\end{array}
$$

where $n \in \mathbb{Z}$, $c_0 \in \mathbb{Z} \cup \{-\infty\}$, $c_1 \in \mathbb{Z} \cup \{+\infty\}$, and $x \in \mathbb{X}$.

We use **labels** to simplify the states in the transition system.

We will define two forms of semantics for NIMP:

$$\boxed{E[\![e]\!] : \Sigma \to \mathcal{P}(\mathbb{V})} \qquad \boxed{\langle \ell, m \rangle \to \langle \ell, m \rangle}$$

where a state $\langle \ell, m \rangle \in \mathbb{S}$ consists of a label ℓ and a memory $m : \mathbb{X} \rightharpoonup \mathbb{V}$.

Non-Deterministic Imperative Language – NIMP

We defined the semantics of expressions $E[\![e]\!]$ in the previous lecture.

Non-Deterministic Imperative Language – NIMP

We defined the semantics of expressions $E[\![e]\!]$ in the previous lecture.

Let's define the small-step operational semantics of statements as a transition relation $\rightarrow$ on states $\langle \ell, m \rangle$:

- ℓ : `skip` ℓ'

$$\langle \ell, m \rangle \rightarrow \langle \ell', m \rangle$$

- $\ell : x$ `:=` $e\ \ell'$

$$\langle \ell, m \rangle \rightarrow \langle \ell', m[x \mapsto v] \rangle \text{ if } v \in E[\![e]\!](m)$$

- $\ell : s_1$ `;` $\ell' : s_2\ \ell''$

  The transition relations are defined in the statements s_1 and s_2.

Non-Deterministic Imperative Language – NIMP

- $\ell : \texttt{if}\ e\ \texttt{then}\ \{\ell_1 : s_1\}\ \texttt{else}\ \{\ell_2 : s_2\}\ \ell'$

$$\begin{array}{ll} \langle \ell, m\rangle \to \langle \ell_1, m\rangle & \text{if } \texttt{true} \in E[\![e]\!](m) \\ \langle \ell, m\rangle \to \langle \ell_2, m\rangle & \text{if } \texttt{false} \in E[\![e]\!](m) \end{array}$$

- $\ell : \texttt{while}\ e\ \texttt{do}\ \{\ell_1 : s\}\ \ell'$

$$\begin{array}{ll} \langle \ell, m\rangle \to \langle \ell_1, m\rangle & \text{if } \texttt{true} \in E[\![e]\!](m) \\ \langle \ell, m\rangle \to \langle \ell', m\rangle & \text{if } \texttt{false} \in E[\![e]\!](m) \end{array}$$

Non-Deterministic Imperative Language – NIMP

- $\ell : \texttt{if}\ e\ \texttt{then}\ \{\ell_1 : s_1\}\ \texttt{else}\ \{\ell_2 : s_2\}\ \ell'$

$$\begin{array}{ll} \langle \ell, m\rangle \to \langle \ell_1, m\rangle & \text{if } \texttt{true} \in E[\![e]\!](m) \\ \langle \ell, m\rangle \to \langle \ell_2, m\rangle & \text{if } \texttt{false} \in E[\![e]\!](m) \end{array}$$

- $\ell : \texttt{while}\ e\ \texttt{do}\ \{\ell_1 : s\}\ \ell'$

$$\begin{array}{ll} \langle \ell, m\rangle \to \langle \ell_1, m\rangle & \text{if } \texttt{true} \in E[\![e]\!](m) \\ \langle \ell, m\rangle \to \langle \ell', m\rangle & \text{if } \texttt{false} \in E[\![e]\!](m) \end{array}$$

We can define transition system $\mathcal{T} = (\mathbb{S}, \to)$ for NIMP using the above small-step evaluation steps as the transition relation.

Then, we can define the denotational semantics of NIMP as

$$[\![\text{NIMP}]\!] \triangleq [\![\mathcal{T}]\!]^{\infty} = [\![\mathcal{T}]\!]^{*} \uplus [\![\mathcal{T}]\!]^{\omega}$$

Non-Deterministic Imperative Language – NIMP

We can directly define the **finite trace semantics** of NIMP in a **compositional** way without defining the small-step evaluation steps →.

$$
\begin{aligned}
[\![s_1 \; ; s_2]\!]^* &\triangleq [\![s_1]\!]^* \cup [\![s_2]\!]^* \cup ([\![s_1]\!]^* \bowtie [\![s_2]\!]^*) \\
[\![\texttt{if } e \texttt{ then } s_1 \texttt{ else } s_2]\!]^* &\triangleq (\{\langle \ell, m \rangle \mid \texttt{true} \in E[\![e]\!](m)\} \bowtie [\![s_1]\!]^*) \cup \\
& (\{\langle \ell, m \rangle \mid \texttt{false} \in E[\![e]\!](m)\} \bowtie [\![s_2]\!]^*) \\
[\![\texttt{while } e \texttt{ do } s]\!]^* &\triangleq \mathbf{lfp}(F)
\end{aligned}
$$

where

$$
\begin{aligned}
F(X) &\triangleq (\{\langle \ell, m \rangle \mid \texttt{true} \in E[\![e]\!](m)\} \bowtie X) \cup \\
& (\{\langle \ell, m \rangle \mid \texttt{false} \in E[\![e]\!](m)\}) \\
\langle \sigma, \ldots, \sigma' \rangle \bowtie \langle \sigma', \ldots, \sigma'' \rangle &\triangleq \langle \sigma, \ldots, \sigma', \ldots, \sigma'' \rangle
\end{aligned}
$$

Next Lecture

- Trace Properties

Jihyeok Park
jihyeok_park@korea.ac.kr
https://plrg.korea.ac.kr