The Basic Principles of Operational Semantics

§1. Abstract Syntax – Bims

§2. Transition Systems

§3. Operational Semantics for Aexp

- Big-step Semantics
- Small-step Semantics

§4. Proving properties of SOS,

Slides Courtesy of Radu Mardare

The presentation of this chapter is centered on the language Bims, which forms the core of many programming languages involving arithmetic and Boolean expressions.

Bims – basic imperative statements

– it is Turing-complete – all computable functions can be represented in Bims

§1. Abstract Syntax

In program semantics we are not interested in syntax analysis – which is part of the theory of parsing.

<u>Abstract syntax</u> – allows to describe the essential structure of a program
– It is defined as follows:

- assume a collection of syntactic categories
- for each syntactic category assume a finite set of BNF context-free rules – <u>formation rules</u>
- the formation rules define how the elements of the category can be constructed

Bims

Abstract Syntax

Syntactic categories

$n \in \mathbb{N}\mathrm{um}$ – Numerals
$x \in \mathbb{V}\mathrm{ar}$ – Variables
} basic syntactic categories

$a \in \mathbb{A}\mathrm{exp}$ – Arithmetic expressions

$b \in \mathbb{B}\mathrm{exp}$ – Boolean expressions

$S \in \mathbb{S}\mathrm{tm}$ – Statements

n, x, a, b, S – metavariables

Formation rules

$$S ::= \text{"}x = a\text{"} \mid \mathrm{skip} \mid S_1 ; S_2 \mid \mathrm{if}\ b\ \mathrm{then}\ S_1\ \mathrm{else}\ S_2 \mid \mathrm{while}\ b\ \mathrm{do}\ S$$

$$b ::= \text{"}a_1 = a_2\text{"} \mid \text{"}a_1 < a_2\text{"} \mid \neg b_1 \mid b_1 \wedge b_2 \mid (b_1)$$

$$a ::= n \mid x \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1)$$

We assume that $\mathbb{N}\mathrm{um}$ are numerals in <u>decimal notation</u>.

$\mathbb{V}\mathrm{ar}$ are <u>strings</u> of the Latin alphabet letters.

An arithmetic expression $a ::= exp$ has the right-hand side that can be

- a <u>simple element</u>, such as n or x
- a <u>composite element</u> built up from simpler entities called <u>immediat constituents</u>

Which are the immediat constituents of the expression

$$(\underline{3} + \underline{4}) * (\underline{14} + \underline{9}) \,?$$

They are $(\underline{3} + \underline{4})$ and $(\underline{14} + \underline{9})$ and not $\underline{3}$, $\underline{4}$, $\underline{14}$, $\underline{9}$.

<u>Observation</u>: In what follows we distingush between numbers such as 3, 4, 5 and numerals in decimal notation such as $\underline{3}$, $\underline{4}$, $\underline{5}$. Similarly we distinguish between operations on numbers and operations on numerals.

Derived Boolean expressions:

$$a_1 \neq a_2 \stackrel{df}{=} \neg(a_1 = a_2)$$

$$a_1 \leq a_2 \stackrel{df}{=} (a_1 = a_2) \vee (a_1 < a_2)$$

$$b_1 \vee b_2 \stackrel{df}{=} \neg((\neg b_1) \wedge (\neg b_2))$$

$$b_1 \rightarrow b_2 \stackrel{df}{=} (\neg b_1) \vee b_2$$

$$b_1 \leftrightarrow b_2 \stackrel{df}{=} (b_1 \rightarrow b_2) \wedge (b_2 \rightarrow b_1)$$

$$\mathit{tt} \stackrel{df}{=} (\neg b_1) \vee b_1$$

$$\mathit{ff} \stackrel{df}{=} \neg \mathit{tt}$$

The order of the Boolean operations

$$\neg, [\wedge, \vee], [\rightarrow, \leftrightarrow]$$

The order of the Arithmetic operations:

$$*, [+, -]$$

Statements: the semicolon is left-associative:

$$S_1; S_2; S_3 = S_1; (S_2; S_3)$$

§2. Transition Systems

A structural operational semantics (SOS) defines a transition system.

Definition [Transition System]:

A transition system (TS) is a tuple $(\Gamma, \rightarrow, T)$ where

- Γ is a set of states (configurations) – the vertices
- $\rightarrow \subseteq \Gamma \times \Gamma$ is the transition relation – the edges
- $T \subseteq \Gamma$ is the set of terminal (final) states

Example: ①

$$\begin{tikzcd} \gamma_1 \arrow[r] \arrow[d] & \gamma_2 \arrow[d] \\ \gamma_3 & \gamma_4 \end{tikzcd}$$

$\Gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$

$\rightarrow = \{(\gamma_1, \gamma_2), (\gamma_1, \gamma_3), (\gamma_2, \gamma_4)\}$

$T = \{\gamma_3, \gamma_4\}$

②

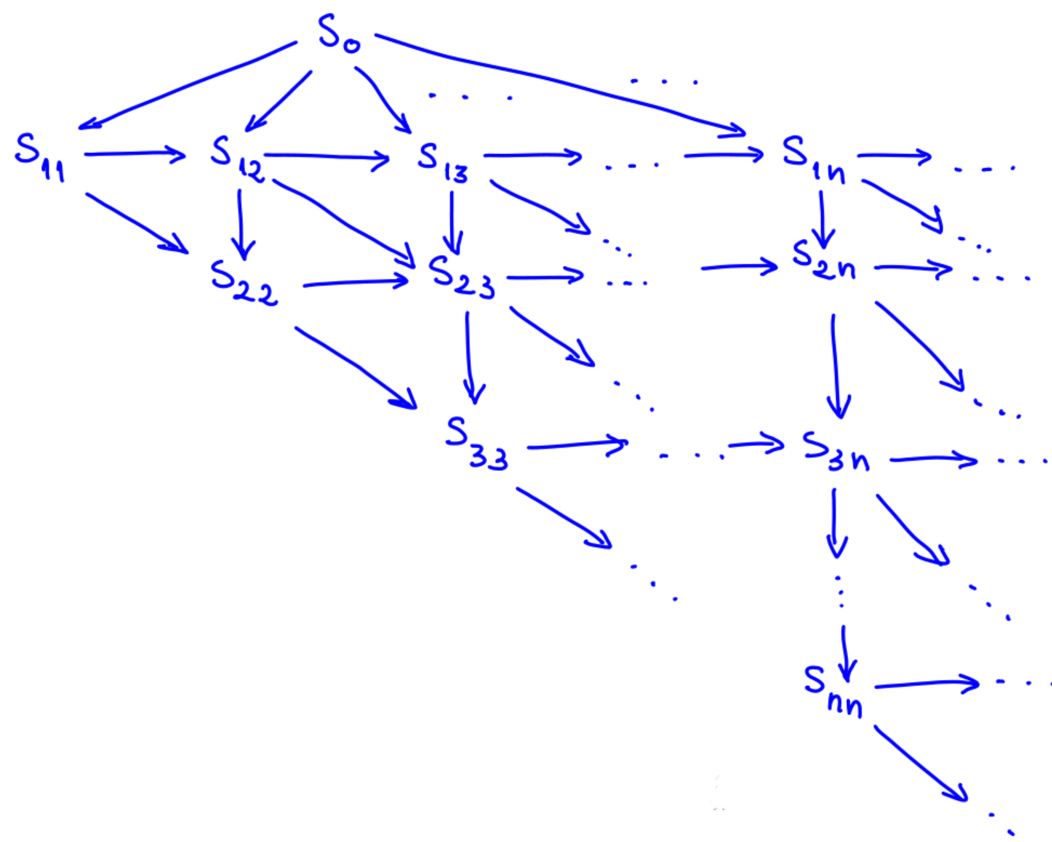
$$\begin{tikzcd} \gamma_1 \arrow[r, shift left] & \gamma_2 \arrow[l, shift left] \\ & \gamma_3 \end{tikzcd}$$

$\Gamma = \{\gamma_1, \gamma_2, \gamma_3\}$

$\rightarrow = \{(\gamma_1, \gamma_2), (\gamma_2, \gamma_1)\}$

$T = \{\gamma_1, \gamma_2, \gamma_3\}$

Define the transition system drawn below.



There exist two types of SOS:

- <u>big-step semantics</u> (BS): each transition $\gamma \to \gamma'$ expresses an entire computation from γ_1 to $\gamma_2 \in T$.

- <u>small-step semantics</u> (SS): each transition $\gamma \to \gamma'$ expresses a single step of a larger computation.

§3. Operational Semantics for $\mathbb{A}$exp in $\mathbb{B}$ims

We consider $\mathbb{A}$exp without variables.

$$a ::= n \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1)$$

Big-step semantics for $\mathbb{A}$exp

We will have transitions of type

$$a \longrightarrow v \quad , a \in \mathbb{A}\text{exp} \text{ and } v \in \mathbb{Z}$$

it means "expression a evaluates to value v".

Example: To evaluate $(\underline{2}+\underline{3})*(\underline{4}+\underline{9})$ we need to evaluate $(\underline{2}+\underline{3})$ and $(\underline{4}+\underline{9})$, for which we need to evaluate $\underline{2}$, $\underline{3}$, $\underline{4}$ and $\underline{9}$

Big-step semantics for Aexp

$$a ::= n \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1)$$

Big-step Transition Rules

$$[\text{PLUS}_{BS}] \quad \frac{a_1 \to v_1 \quad a_2 \to v_2}{a_1 + a_2 \to v} \; v = v_1 + v_2 \qquad [\text{MINUS}_{BS}] \quad \frac{a_1 \to v_1 \quad a_2 \to v_2}{a_1 - a_2 \to v} \; v = v_1 - v_2$$

$$[\text{MULT}_{BS}] \quad \frac{a_1 \to v_1 \quad a_2 \to v_2}{a_1 * a_2 \to v} \; v = v_1 \cdot v_2 \qquad [\text{PARENT}_{BS}] \quad \frac{a_1 \to v_1}{(a_1) \to v_1}$$

$$[\text{NUM}_{BS}] \quad \frac{}{n \to v} \; \mathcal{N}[\![n]\!] = v$$

The semantics of the basic syntactic categories is given by <u>semantic functions</u> such as $\mathcal{N}$ that maps numerals to integers: $\mathcal{N}[\![\underline{v}]\!] = v$

$$\mathcal{N}[\![\underline{2}]\!] = 2 \qquad \mathcal{N}[\![\underline{5}]\!] = 5 \qquad \mathcal{N}[\![\underline{0}]\!] = 0$$

Big-step semantics for Aexp

$$a ::= n \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1)$$

Big-step Transition Rules

$$[\mathrm{PLUS}_{BS}]\ \frac{a_1 \to v_1 \quad a_2 \to v_2}{a_1 + a_2 \to v}\ v = v_1 + v_2 \qquad [\mathrm{MINUS}_{BS}]\ \frac{a_1 \to v_1 \quad a_2 \to v_2}{a_1 - a_2 \to v}\ v = v_1 - v_2$$

$$[\mathrm{MULT}_{BS}]\ \frac{a_1 \to v_1 \quad a_2 \to v_2}{a_1 * a_2 \to v}\ v = v_1 \cdot v_2 \qquad [\mathrm{PARENT}_{BS}]\ \frac{a_1 \to v_1}{(a_1) \to v_1}$$

$$[\mathrm{NUM}_{BS}]\ \frac{}{n \to v}\ \mathcal{N}[\![n]\!] = v$$

A transition rule is form by <u>premisses</u>, <u>conclusion</u> and <u>side conditions</u>.
Some rule might have no side conditions.
A rule can also have no premisses — <u>axiom</u>

The transition system induced by the BS of $\mathbb{A}\mathrm{exp}$ is defined as follows

$(\Gamma, \rightarrow, T)$ where

- $\Gamma = \mathbb{A}\mathrm{exp} \cup \mathbb{Z}$
- $\rightarrow \subseteq \mathbb{A}\mathrm{exp} \times \mathbb{Z}$ as defined by the transition rules
- $T = \mathbb{Z}$

Questions:
1. Why not $\Gamma = \mathbb{A}\mathrm{exp}$?
2. Why not $\Gamma = \mathbb{A}\mathrm{exp} \cup \mathbb{N}\mathrm{um}$?
3. Why not $\Gamma = \mathbb{A}\mathrm{exp} \cup \mathbb{N}\mathrm{um} \cup \mathbb{Z}$?

To define an SOS means to have an abstract syntax for which to define a transition system $(\Gamma, \rightarrow, T)$ as follows:

1. determine the format of the transitions (e.g., big-step or small-step). All the transitions must have this format.

2. define the set Γ of states and identify the subset $T \subseteq \Gamma$ of terminal configurations

3. define the transition relation "$\longrightarrow$": a transition is legal iff it can be proven from the axioms using the transition rules.

<u>Example</u>: We prove, using the BS of Aexp that

$$(\underline{2}+\underline{3})*(\underline{4}+\underline{9}) \to 65$$

$$[\mathrm{MULT}_{BS}]\ \dfrac{[\mathrm{PARENT}_{BS}]\ \dfrac{[\mathrm{PLUS}_{BS}]\ \dfrac{[\mathrm{NUM}_{BS}]\ \underline{2}\to 2 \qquad \underline{3}\to 3}{\underline{2}+\underline{3}\to 5}}{(\underline{2}+\underline{3})\to 5} \qquad [\mathrm{PARENT}_{BS}]\ \dfrac{[\mathrm{PLUS}_{BS}]\ \dfrac{[\mathrm{NUM}_{BS}]\ \underline{4}\to 4 \qquad \underline{9}\to 9}{\underline{4}+\underline{9}\to 13}}{(\underline{4}+\underline{9})\to 13}}{(\underline{2}+\underline{3})*(\underline{4}+\underline{9})\to 65}$$

<u>Exercise</u>: Prove that

$$\underline{2}*(\underline{3}+(\underline{2}+\underline{5}))\to 20$$

Definition [Derivation tree]:

Given a set $\mathcal{R}$ of transition rules and a subset $\mathcal{A} \subseteq \mathcal{R}$ of axioms, a derivation tree is a finite tree whose nodes are labelled by transitions as follows:

① - all the leaves are labelled by elements of $\mathcal{A}$ (axioms)

② - a node labelled by C has descendants $P_1, \dots P_n$ if there exists a transition rule in $\mathcal{R}$ of the form

$$\frac{P_1, \dots P_n}{C}$$

We want to find an evaluation of the expression

$$(\underline{2}+\underline{3})*(\underline{4}+\underline{9}).$$

- this means to find whether there is a transition

$$(\underline{2}+\underline{3})*(\underline{4}+\underline{9}) \longrightarrow v \quad \text{for some } v \in \mathbb{Z}$$

- it is equivalent to asking whether there is a derivation tree whose root is labelled by

$$(\underline{2}+\underline{3})*(\underline{4}+\underline{9}) \longrightarrow v$$

- we construct such a tree (bottom-up):

$$\dfrac{\dfrac{\dfrac{\underline{2} \longrightarrow v_{11} \qquad \underline{3} \longrightarrow v_{12}}{\underline{2}+\underline{3} \longrightarrow v_1}\; v_1 = v_{11}\cdot v_{12}}{(\underline{2}+\underline{3}) \longrightarrow v_1} \qquad \dfrac{\dfrac{\underline{4} \longrightarrow v_{21} \qquad \underline{9} \longrightarrow v_{22}}{\underline{4}+\underline{9} \longrightarrow v_2}\; v_2 = v_{21}\cdot v_{22}}{(\underline{4}+\underline{9}) \longrightarrow v_2}}{(\underline{2}+\underline{3})*(\underline{4}+\underline{9}) \longrightarrow v}\; v = v_1 \cdot v_2$$

The tree can be constructed if we take

$$v_{11} = 2, \quad v_{12} = 3, \quad v_{21} = 4 \text{ and } v_{22} = 9$$

To construct a derivation tree we apply the following recursive strategy:

1 - find a transition rule whose syntactic construct in the conclusion matches that of the root and whose side conditions are true

2 - construct the derivation tree for each of the premisses of the rule found at step 2; if the rule has no premisses, terminate this branch

3 - if at step 2 you can chose more than one rule, use the procedure for each, until one succeeds.

A transition rule is <u>compositional</u> if the premises make use only of syntactic entities that are imediate constituents of the syntactic construct found in the conclusion.

<u>Examples:</u>

1. An axiom is a compositional rule
2. The following rule is not compositional

$$\frac{a_1 + a_2 \longrightarrow v}{a_2 + a_1 \longrightarrow v}$$

Small-Step Semantics for Aexp

In a SS-semantics a transition represents a single step of the computation.
They have the form

$$a \Rightarrow a'$$

a' is called an <u>intermediate configuration</u> and it can be either a terminal or an arithmetic expression.

Small-Step Semantics for Aexp

$$a ::= n \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1) \mid v$$

$[\text{PLUS-L}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{a_1 + a_2 \Rightarrow a_1 + a_2'}$ $\qquad$ $[\text{PLUS-R}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{a_2 + a_1 \Rightarrow a_2 + a_1'}$

$[\text{MULT-L}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{a_1 * a_2 \Rightarrow a_1' * a_2}$ $\qquad$ $[\text{MULT-R}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{a_2 * a_1 \Rightarrow a_2 * a_1'}$

$[\text{SUB-L}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{a_1 - a_2 \Rightarrow a_1' - a_2}$ $\qquad$ $[\text{SUB-R}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{a_2 - a_1 \Rightarrow a_2 - a_1'}$

$[\text{PARENT}_{ss}]\ \dfrac{a_1 \Rightarrow a_1'}{(a_1) \Rightarrow (a_1')}$ $\qquad$ $[\text{NUM}_{ss}]\ n \Rightarrow v\ ,\ \mathcal{N}[\![n]\!] = v$

$[\text{PLUS-V}_{ss}]\ v_1 + v_2 \Rightarrow v\ ,\ v_1 + v_2 = v$ $\qquad$ $[\text{MULT-V}_{ss}]\ v_1 * v_2 \Rightarrow v\ ,\ v_1 \cdot v_2 = v$

$[\text{SUB-V}_{ss}]\ v_1 - v_2 \Rightarrow v,\ v_1 - v_2 = v$ $\qquad$ $[\text{PARENT-V}_{ss}]\ (v) \Rightarrow v$

The SS-semantics induces a TS $(\Gamma, \Rightarrow, T)$, where

- $\Gamma = \mathbb{A}_{exp} \cup \mathbb{Z}$
- $\Rightarrow\ \subseteq (\mathbb{A}_{exp} \cup \mathbb{Z}) \times (\mathbb{A}_{exp} \cup \mathbb{Z})$
- $T = \mathbb{Z}$

Observe that

$$(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) => (3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8}))$$

Because $\underline{3} => 3$

implying $\underline{3} + \underline{12} => 3 + \underline{12}$

implying $(\underline{3} + \underline{12}) => (3 + \underline{12})$

implying $(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) => (3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8}))$

Similarly, $(3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) => (3 + 12) * (\underline{4} * (\underline{5} * \underline{8}))$

$$(3 + 12) * (\underline{4} * (\underline{5} * \underline{8})) => (15) * (\underline{4} * (\underline{5} * \underline{8}))$$

$$(15) * (\underline{4} * (\underline{5} * \underline{8})) => 15 * (\underline{4} * (\underline{5} * \underline{8}))$$

The difference between BS-semantics and SS-semantics is ilustrated below:

$$(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow (3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8}))$$

$$(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) \not\rightarrow (3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8}))$$

$$(3 + \underline{2}) * (\underline{5} + \underline{1}) \Rightarrow 5 * (\underline{5} + \underline{1})$$

$$(3 + \underline{2}) * (\underline{5} + \underline{1}) \not\rightarrow 5 * (\underline{5} + \underline{1})$$

Let $(\Gamma, \Rightarrow, T)$ be a TS (generated by a SS-semantics).

The <u>k-step transition closure</u> for $k \geq 0$ is defined inductively as follows:

(i) $\gamma \Rightarrow^0 \gamma$ for all $\gamma \in \Gamma$

(ii) $\gamma \Rightarrow^{k+1} \gamma'$ if for some $\gamma'' \in \Gamma$ we have:

- $\gamma \Rightarrow \gamma''$
- $\gamma'' \Rightarrow^k \gamma'$

Observe that for each $k \geq 0$, $\Rightarrow^k \subseteq \Gamma \times \Gamma$

Let $\Rightarrow^* = \bigcup_{k \geq 0} \Rightarrow^k$

We have seen that

$$(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow (3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8}))$$

$$(3 + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow (3 + 12) * (\underline{4} * (\underline{5} * \underline{8}))$$

$$(3 + 12) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow (15) * (\underline{4} * (\underline{5} * \underline{8}))$$

$$(15) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow 15 * (\underline{4} * (\underline{5} * \underline{8}))$$

Hence,

$$(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow^{4} 15 * (\underline{4} * (\underline{5} * \underline{8}))$$

And going further,

$$(\underline{3} + \underline{12}) * (\underline{4} * (\underline{5} * \underline{8})) \Rightarrow^{*} 2400$$

<u>Exercise</u>: Find $v \in \mathbb{Z}$ such that $(\underline{2} + \underline{3}) * (\underline{4} + \underline{9}) \Rightarrow^{*} v$.

§4. Proving properties of SOS

Determinacy: A Big-Step semantics is deterministic if

$$a \longrightarrow v_1 \text{ and } a \longrightarrow v_2 \text{ implies } v_1 = v_2$$

If a semantics is deterministic, then each expression has a unique evaluation.

Problem: Prove that the BS-semantics of Aexp is deterministic.

Since the operational semantics presented before are structural, i.e., it depends of the structure of the terms, we can use proof techniques based on induction to prove properties of a given SOS.

Use induction on the structure of algebraic expressions to solve the aforementioned problem.