

Formal Methods for Java

Lecture 2: Operational Semantics

Jochen Hoenicke



Software Engineering
Albert-Ludwigs-University Freiburg

October 28, 2011

The Java Language Specification (JLS) 3rd edition gives semantics for Java

- The document has 684 pages.
- 118 pages to define semantics of expression.
- 42 pages to define semantics of method invocation.

Semantics are only defined by prosa text.

How can we give the semantics formally?

Need a mathematical model for computations.

Idea: define transition system for Java

Definition (Transition System)

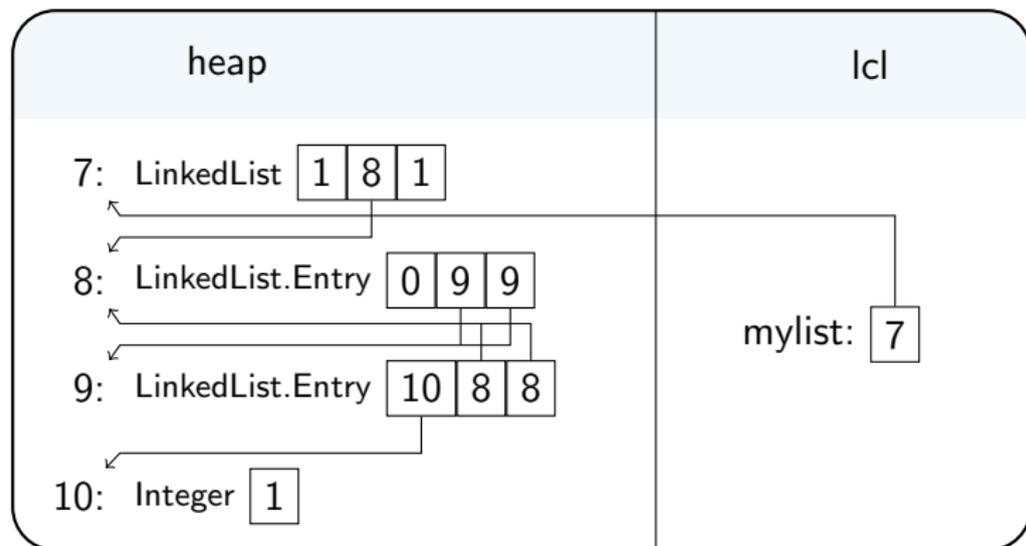
A transition system (TS) is a structure $TS = (Q, Act, \rightarrow)$, where

- Q is a set of states,
 - Act a set of actions,
 - $\rightarrow \subseteq Q \times Act \times Q$ the transition relation.
-
- Q reflects the current dynamic state (heap and local variables).
 - Act is the executed code.
 - Idea from: D. v. Oheimb, T. Nipkow, [Machine-checking the Java specification: Proving type-safety](#), 1999

Example: State of a Java Program

What is the state after executing this code?

```
List mylist = new LinkedList();  
mylist.add(new Integer(1));
```



State of a Java Program

The state of a Java program gives valuations local and global (heap) variables.

- $Q = \text{Heap} \times \text{Local}$
- $\text{Heap} = \text{Address} \rightarrow \text{Class} \times \text{seq Value}$
- $\text{Local} = \text{Identifier} \rightarrow \text{Value}$
- $\text{Value} = \mathbb{Z}, \text{Address} \subseteq \mathbb{Z}$

A state is denoted as $(\text{heap}, \text{lcl})$, where $\text{heap} : \text{Heap}$ and $\text{lcl} : \text{Local}$.

Actions of a Java Program

An action of a Java Program is either

- the evaluation of an expression e to a value v , denoted as $e \triangleright v$, or
- a Java statement, or
- a Java code block.

Note that expressions with side-effects can modify the current state

Example: Actions of a Java Program

Post-increment expression:

$$(heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{x++\triangleright 5} (heap, lcl \cup \{x \mapsto 6\})$$

Pre-increment expression:

$$(heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{++x\triangleright 6} (heap, lcl \cup \{x \mapsto 6\})$$

Assignment expression:

$$(heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{x=x*2\triangleright 10} (heap, lcl \cup \{x \mapsto 10\})$$

Assignment statement:

$$(heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{x=x*2;} (heap, lcl \cup \{x \mapsto 10\})$$

The last slide listed some examples for transitions.

Define rules when a transition is valid.

Definition (Inference Rules)

A rule of inference

$$\frac{F_1 \dots F_n}{G}, \text{ where } \dots$$

is a **decidable** relation between formulae. The formulae F_1, \dots, F_n are called the **premises** of the rule and G is called the conclusion.

If $n = 0$ the rule is called an **axiom schema**. In this case the bar may be omitted.

The intuition of a rule is that if all premises hold, the conclusion also holds.

Rules for Java expressions (1)

axiom for evaluating local variables:

$$(heap, lcl) \xrightarrow{x \triangleright lcl(x)} (heap, lcl)$$

rule for field access:

$$\frac{(heap, lcl) \xrightarrow{e \triangleright v} (heap', lcl')}{(heap, lcl) \xrightarrow{e.fld \triangleright heap'(v)(idx)} (heap', lcl')}, \text{ where } idx \text{ is the index of the field } fld \text{ in the object } heap'(v)$$

rule for assignment to local:

$$\frac{(heap, lcl) \xrightarrow{e \triangleright v} (heap', lcl')}{(heap, lcl) \xrightarrow{x = e \triangleright v} (heap', lcl' \oplus \{x \mapsto v\})}$$

Rules for Java expressions (2)

axiom for evaluating a constant expression c :

$$(heap, lcl) \xrightarrow{c \triangleright c} (heap, lcl)$$

rule for multiplication (similar for other binary operators)

$$\frac{\begin{array}{l} (heap_1, lcl_1) \xrightarrow{e_1 \triangleright v_1} (heap_2, lcl_2) \\ (heap_2, lcl_2) \xrightarrow{e_2 \triangleright v_2} (heap_3, lcl_3) \end{array}}{(heap_1, lcl_1) \xrightarrow{e_1 * e_2 \triangleright (v_1 \cdot v_2) \bmod 2^{32}} (heap_3, lcl_3)}$$

A derivation for $x = x * 2$

$$\frac{\begin{array}{c} (heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{x \triangleright 5} (heap, lcl \cup \{x \mapsto 5\}) \\ (heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{2 \triangleright 2} (heap, lcl \cup \{x \mapsto 5\}) \end{array}}{\frac{(heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{x * 2 \triangleright 10} (heap, lcl \cup \{x \mapsto 5\})}{(heap, lcl \cup \{x \mapsto 5\}) \xrightarrow{x = x * 2 \triangleright 10} (heap, lcl \cup \{x \mapsto 10\})}}$$

Rules for Java Statements

expression statement (assignment or method call):

$$\frac{(heap, lcl) \xrightarrow{e \triangleright v} (heap', lcl')}{(heap, lcl) \xrightarrow{e_i} (heap', lcl')}$$

sequence of statements:

$$\frac{(heap_1, lcl_1) \xrightarrow{s_1} (heap_2, lcl_2) \quad (heap_2, lcl_2) \xrightarrow{s_2} (heap_3, lcl_3)}{(heap_1, lcl_1) \xrightarrow{s_1 s_2} (heap_3, lcl_3)}$$

Rules for Java Statements

if statement:

$$\frac{(heap_1, lcl_1) \xrightarrow{e \triangleright v} (heap_2, lcl_2) \quad (heap_2, lcl_2) \xrightarrow{b_1} (heap_3, lcl_3)}{(heap_1, lcl_1) \xrightarrow{\text{if}(e) b_1 \text{ else } b_2} (heap_3, lcl_3)}, \text{ where } v \neq 0$$

$$\frac{(heap_1, lcl_1) \xrightarrow{e \triangleright v} (heap_2, lcl_2) \quad (heap_2, lcl_2) \xrightarrow{b_2} (heap_3, lcl_3)}{(heap_1, lcl_1) \xrightarrow{\text{if}(e) b_1 \text{ else } b_2} (heap_3, lcl_3)}, \text{ where } v = 0$$