

Formal Methods for Java

Lecture 14: Dynamic Logic for Java Card

Jochen Hoenicke



Software Engineering
Albert-Ludwigs-University Freiburg



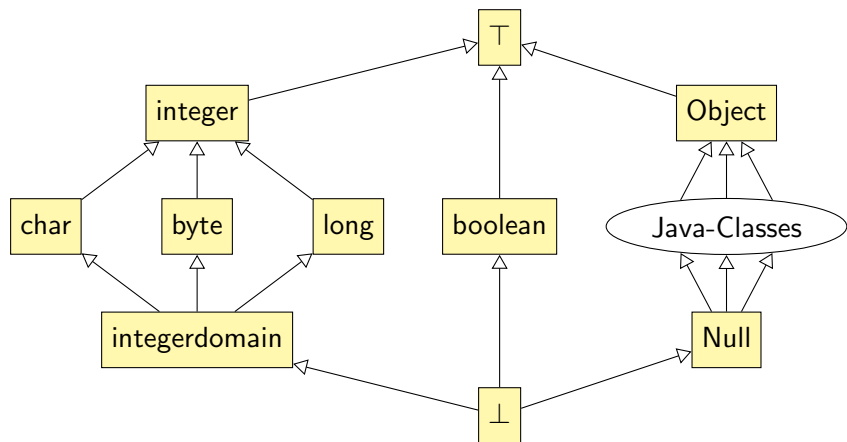
June 19, 2007

- Theorem Prover
- Developed at University of Karlsruhe
- <http://www.key-project.org/>.
- Theory specialized for JavaCard.
- Can generate proof-obligations from JML specification.
- Underlying theory: Sequent Calculus + Dynamic Logic

- $[\alpha]\phi$: after all terminating runs of program α formula ϕ holds.
- $\langle\alpha\rangle\phi$: after some terminating run of program α formula ϕ holds.

Where α is a JavaCard program fragment.

Type-Hierarchy in the KeY-System



Rigid vs. Non-Rigid Functions vs. Variables

KeY distinguishes the following symbols:

- Rigid Functions: These are functions that do not depend on the current state of the program.
 - $+, -, * : integer \times integers \rightarrow integer$ (mathematical operations)
 - $0, 1, \dots : integer, TRUE, FALSE : boolean$ (mathematical constants)
- Non-Rigid Constants: These are functions that depend on current state.
 - $.[.] : T \times int \rightarrow T$ (array access)
 - $.next : T \rightarrow T$ if `next` is a field of a class.
 - $i, j : T$ if `i, j` are program variables.
- Variables: These are logical variables that can be quantified. Variables may not appear in programs.
 - x, y, z

Example

$$\forall x. i = x \rightarrow \langle \{ \text{while}(i > 0) \{ i = i - 1; \} \} \rangle i = 0$$

- $0, 1, -$ are rigid functions.
- $>$ is a rigid relation.
- i is a non-rigid function.
- x is a logical variable.

Quantification over i is not allowed and x must not appear in a program.

Builtin Rigid Functions

- $+, -, *, /, \%, jdiv, jmod$: operations on *integer*.
- $\dots, -1, 0, 1, \dots, TRUE, FALSE, null$: constants.
- (A) for any type A : cast function.
- $A :: get$ gives the n -th object of type A .

Formula $\{i := t\}\phi$ is true, iff ϕ holds in a state, where the program variable i has the value denoted by the term t .

Here:

- i is a program variable (non-rigid function).
- t is a term (may contain logical variables).
- ϕ a formula

The meaning is the same as $\langle\{i = t;\}\rangle\phi$, except that t can contain logical variables.

Rules for Java Dynamic Logic

- $\langle\{i = j; \dots\}\rangle\phi$ is rewritten to:
 $\{i := j\}\langle\{\dots\}\rangle\phi$.
- $\langle\{i = j + k; \dots\}\rangle\phi$ is rewritten to:
 $\{i := j + k\}\langle\{\dots\}\rangle\phi$.
- $\langle\{i = j ++; \dots\}\rangle\phi$ is rewritten to:
 $\langle\{\mathbf{int} \ j_0; j_0 = j; j = j + 1; i = j_0; \dots\}\rangle\phi$.
- $\langle\{\mathbf{int} \ k; \dots\}\rangle\phi$ is rewritten to:
 $\langle\{\dots\}\rangle\phi$ and k is added as new program variable.