Formal Methods for Java
Lecture 25: What you should know for the exams

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

July 26, 2017
Operational Semantics

• What is a transition system.
• What are the states, what are the actions?
• How can we define the (infinite) transition relation?
• Do the rules show in which order something is executed?
• How can we define a rule for the Java expression/statement . . .?
JML and runtime checking

- What is JML good for?
- What are the keywords requires, ensures, modifies,...?
- How can we write a contract for a method that
  - adds two numbers?
  - finds the minimum in an array?
  - sorts an array?
- How does runtime checking work?
- How does old(...) work? How can it be checked at runtime?
- Under what conditions can quantifiers be checked at runtime?
- How can one specify when exceptions may/must occur?
What is static checking?

Difference between KeY and ESC/Java?

How do require and ensures relate to assert and assume?

How does ESC/Java check assert/assume?
What is the problem with invariants?
When should an invariant hold?
How does ownership model help with this problem?
What is the pack/unpack mechanism? How does it work?
How does the pack/unpack mechanism help with invariants?
What are the limitation of ownership?
How can friendship help?
Sequent Calculus and Dynamic Logic

- $\phi_1, \phi_2 \implies \psi_1, \psi_2$

- What are the rules of sequent calculus? Are they sound/complete? What does that mean?

- Hoare-Triples vs. $\phi \implies \langle \alpha \rangle \psi$ and $\phi \implies [\alpha] \psi$

- What is the meaning of $\langle \alpha \rangle \phi$?

- What are the rules for dynamic logic?

- How can one proof loops with KeY?
What is model checking?
Difference to static checking?
Difference to theorem proving (like KeY)?
How can we write our own listeners?
How can we use choice generators?
What should you have learned

- How to give formal semantics to Java/JML (e.g. operational semantics).
- How to give pre-/post-conditions in JML.
- What is the relation between assume, assert and ensures, requires?
- What is run-time checking? Why is it useful? What are the limits?
- What is static checking? Why useful? What are the limits?
- What are the problems of class invariants and how to solve them.
- What is soundness and completeness? How does it apply to software verification.
- How to prove with KeY-System. How can loops be checked?
- How can verification conditions be generated from a program with assumes and asserts?