A Sound, Complete and Usable Hoare-Style Logic for a Sequential Java Subset

Progress Report

Christoph Feller

8.10.2009

Overview

1. Motivation
2. Reached goals
3. Technical details
4. Conclusion and Future Work

Motivation

Proofing a program logic sound and complete is not a new idea.

Then why do it?

- Preparation for modular program logic
- We want to have a *usable* logic

Reached goals

Definitions:

- AST of Java Subset *Java-KE*
- Notion of welltyped states and a big-step semantic
- Logic rules

Reached goals

Theorems:

- Type safety of *Java-KE*:

$$S :: pp \longrightarrow SQ \wedge pp\_in\_prog\ pp \Rightarrow wtr\ SQ\ pp$$

- Soundness of the logic rules:

$$\begin{array}{ll} & A \vdash \{P\}pp\{Q\} \wedge pp\_in\_prog\ pp \\ \Rightarrow & \forall N.((\forall tr \in A.tr\_rsem\ tr\ N \wedge \\ & \forall L\,S\,SQ.P\,L\,S \wedge S :: pp - N \rightarrow SQ) \Rightarrow Q\,L\,SQ) \end{array}$$

Definitions and theorems all in Isabelle/HOL

Old invocation rule

$$\frac{\begin{array}{c}\Gamma, A \vdash \{\text{Normal } P\} \qquad \forall a. \Gamma, A \vdash \{Q \leftarrow \text{val } a\} args \dot{\succ} \{R\ a\} \\ \forall a\ vs\ D\ l.\, \Gamma, A \vdash \{(R\ a \leftarrow \text{Vals } vs \wedge . \\ (\lambda(x, s). D = \text{target } mode\ s\ a\ md \wedge l = \text{locals } s) ; . \\ \text{init\_lvars } \Gamma D (mn, pTs) mode\ a\ vs) \wedge . \\ (\lambda \sigma. \text{ normal } \sigma \longrightarrow \Gamma \vdash mode \rightarrow D \preceq t)\} \\ \text{Methd } D(mn, pTs) - \succ \{\text{set\_lvars } l .; S\}\end{array}}{\Gamma, A \vdash \{\text{Normal } P\}\{t, md, mode\} e..mn(\{pTs\} args) - \succ \{S\}}$$

New invocation rule

$$\frac{\begin{array}{c} \mathit{stypv}([n]x := y.m(e))\ y = \mathit{Some}(\mathit{RefT}\ \mathit{rid}) \\ A \vdash \{P\}\mathit{VirMeth}\ m\ \mathit{rid}\{Q\} \end{array}}{\begin{array}{ll} A \vdash & \{(\lambda L\, S.\, S[y]_v \neq \mathit{Null} \wedge \\ & P\, L\, (S[\mathit{this} \leftarrow S[y]_v]_v[\mathit{par} \leftarrow S[e]_e]_v))\} \\ & \mathit{Stmnt}([n]x := y.m(e))\ \{Q[x/\mathit{res}]_{vv}\} \end{array}}$$

Static types

Every statement in our AST is unique.

We can get the static type of a variable just by supplying the relevant statement.

Example:

$$stypv([n]x = y.m(e))\ y = RefT\ Object$$

Conclusion and Future Work

Conclusion

- Usable logic rules are possible
- There is room for improvement

Future Work

- Completeness is missing