Axiomatic Semantics

Computer Science and Artificial Intelligence Laboratory
MIT

October 27, 2011
Example

\[\vdash \{A[x \to e] \} x := e \{A\}\]

\[\vdash \{A \land b\} c_1 \{B\} \quad \vdash \{A \land \neg b\} c_2 \{B\}\]

\[\vdash \{A\} \text{if } b \text{ then } c_1 \text{ else } c_2 \{B\}\]

\[\vdash A' \Rightarrow A \quad \vdash \{A\} c \{B\} \quad \vdash B \Rightarrow B'\]

\[\vdash \{A'\} c \{B'\}\]

\[\vdash \{A \land b\} c \{A\}\]

\[\vdash \{A\} \text{while } b \text{ do } c \{A \land \neg b\}\]

\[\vdash \{A\} c_1 \{C\} \quad \vdash \{C\} c_2 \{B\}\]

\[\vdash \{A\} c_1 ; c_2 \{B\}\]

\{ x=x_0 \text{ and } y=y_0 \}
if(x > y){
    t = x - y;
    while(t > 0){
        x = x - 1;
        y = y + 1;
        t = t - 1;
    }
}
\{ x_0 > y_0 \Rightarrow y=x_0 \text{ and } x=y_0 \}
From partial to total correctness

Total correctness judgment
- \( \vdash [A] c [B] \)
- Just like before, but must also prove termination

\[
\begin{align*}
\vdash [A \land b] c_1 [B] & \quad \vdash [A \land \neg b] c_2 [B] \\
\vdash [A] \text{if } b \text{ then } c_1 \text{ else } c_2 [B] & \quad \vdash [A[x \rightarrow e]] x := e [A]
\end{align*}
\]

\[
\begin{align*}
\vdash [A] c_1 [C] & \quad \vdash [C] c_2 [B] \\
\vdash [A] c_1 ; c_2 [B] & \quad \vdash [A] c_1 ; c_2 [B]
\end{align*}
\]

What about loops
Rank function

Function F of the state that
- a) Maps state to an integer
- b) Decreases with every iteration of the loop
- c) Is guaranteed to stay greater than zero
- Also called variant function

\[
\begin{align*}
\vdash & \ [A \land b \land F = z] c \ [A \land F < z] \quad \vdash \ A \land b \Rightarrow F \geq 0 \ \\
\vdash & \ [A] \text{while } b \text{ do } c \ [A \land \neg b]
\end{align*}
\]
Example

Can we prove this?

\[
\begin{array}{l}
[ x=x_0 \text{ and } y=y_0 ] \\
\text{if}(x > y)\{ \\
\quad t = x - y; \\
\quad \text{while}(t > 0)\{ \\
\quad \quad x = x - 1; \\
\quad \quad y = y + 1; \\
\quad \quad t = t - 1; \\
\quad \} \\
\} \\
[ x_0 > y_0 \Rightarrow y=x_0 \text{ and } x=y_0 ]
\end{array}
\]
Denotational semantics

Another notation to describe semantics of programs

- Program semantics as a function

\[ s \in S : \text{Id} \to \text{Val} \]

\[ [[C]] : S \to S \cup \{\bot\} \]

\[ [[\text{skip}]]s = s \quad [[x = e]]s = s\{x \to [[e]]s\} \quad [[C_1; C_2]]s = [[C_2]]([[C_1]]s) \]

\[ [[\text{if } B \text{ then } C_1 \text{ else } C_2]]s = \text{if } [[B]]s \text{ then } [[C_1]]s \text{ else } [[C_2]]s \]

\[ [[\text{while } B \text{ do } C]]s = \text{if not } [[B]]s \text{ then } s \text{ else } [[\text{while } B \text{ do } C]]([[C]]s) \]

\[ [[\text{while } B \text{ do } C]]s = \text{fix}(\lambda f(\lambda s \text{ if not } [[B]]s \text{ then } s \text{ else } f([[C]]s))) \]
Denotational semantics

\[ [E] : S \rightarrow Val \]

\[ [n]s = n \]

\[ [v]s = s(v) \]

\[ [[E_1 + E_2]]s = [[E_1]]s + [[E_2]]s \]

\[ [[E_1 \times E_2]]s = [[E_1]]s \times [[E_2]]s \]
Verification

What does $\models \{ A \} c \{ B \}$ mean?

$$\forall \sigma \forall \sigma' (\sigma \models A \land \langle c, \sigma \rangle \rightarrow \sigma') \Rightarrow \sigma' \models B$$

$$\forall s \left[ A \right] s \Rightarrow \left[ B \right] [c] s$$

What about $\vdash \{ A \} c \{ B \}$?

Completeness: Given $\models \{ A \} c \{ B \}$, show $\vdash \{ A \} c \{ B \}$
Weakest Preconditions

Weakest predicate $P$ such that $\models \{P\} c \{A\}$
- $P$ weaker than $Q$ iff $Q \Rightarrow P$

\[
\begin{align*}
\text{wpc}(\text{skip } \{Q\}) &= Q \\
\text{wpc}(x = e\{Q\}) &= Q[e/x] \\
\text{wpc}(C1; C2\{Q\}) &= \text{wpc}(C1\{\text{wpc}(C2\{Q\})\}) \\
\text{wpc}(\text{if } B \text{ then } C1 \text{ else } C2\{Q\}) &= (B \text{ and } \text{wpc}(C1\{Q\})) \text{ or } (\text{not } B \text{ and } \text{wpc}(C2\{Q\}))
\end{align*}
\]
Weakest Precondition

While-loop is tricky
- Let \( W = \text{wpc}(\text{while } e \text{ do } c, B) \)
- then,

\[
W = e \Rightarrow \text{wpc}(c, W) \land \neg e \Rightarrow B
\]
Verification Condition

Stronger than the weakest precondition

Can be computed by using an invariant

\[ VC(\text{while}_I \ e \ do \ c, B) = \]
\[ I \land \ \forall x_1, \ldots, x_n \ I \Rightarrow (e \Rightarrow VC(c, I) \land \neg e \Rightarrow B) \]

- Where \( x_i \) are variables modified in \( c \).
Completeness

Assume $\models \{P\}C1; C2\{Q\}$

Want $\vdash \{P\}C1; C2\{Q\}$

Must find $R$ such that

- $\vdash \{P\}C1\{R\}$
- $\vdash \{R\}C2\{Q\}$
Completeness

Assume $\vdash \{P\}$ while $B$ do $C\{Q\}$

Want $\vdash \{P\}$ while $B$ do $C\{Q\}$

Must find $R$ such that

- $\vdash \{R\}$ while $B$ do $C\{R \text{ and not } B\}$
- $P$ implies $R$
- $R$ and not $B$ implies $Q$