Wang Rui

The content of this lecture can be found in Winskel's book, Chapter 6.

How can we know that a program do the right things we want? In other words, how can we verify if a program is correct? Let's look at this simple example.

```
while x!=0 do
   if (x mod 2b = b)
     then
       x = x - b
       e = e + n
     end
   n = n * n
   b = b * 2
end
```

It is a little confused to see what this program does. In fact it computes n^x. The precondition of this program is $b = 1$, $e = 1$, $x \geq 0$. Postcondition is $e = n^x$.

Partial correctness Assertion $\{A\}c\{B\}$

The assertion $\{A\}c\{B\}$ means that starting in any state satisfying A, if we terminate after executing command c, the resulting state satisfies B. A partial correctness assertion does not require that the command terminates. If c does not terminate, then the postcondition B can be anything. For example, $\{true\}$ *while true do skip* $\{false\}$ is true.

Total correctness Assertion $[A]c[B]$

The assertion $[A]c[B]$ means that starting in state satisfying A, we will terminate after executing command c, and the resulting state satisfies B. The difference between partial correctness assertion and total correctness assertion is that we know command c will terminate in total correctness assertion. It is partial correctness plus c terminates.

Here we will not deal with total correctness much.

$\sigma \models A$ means σ satisfies A, i.e., the assertion A is $True$ in σ. The set $\{\sigma \mid \sigma \models A\}$ is called the *extension* of A, the set of states that satisfy A.

$\{A\}c\{B\}$ iff $\forall \sigma \models A$, $\langle c, \sigma\rangle \rightarrow^\star \sigma' \Rightarrow \sigma' \models B$

Using this definition , we can formally specify the meaning of a partial correctness assertion in terms of the operational semantics for the language.

Above we use the operational semantics to describe the meaning of partial assertion, we can also use denotational semantics to describe that:

$\{A\}c\{B\} \longleftrightarrow \forall \sigma \models A, \mathcal{C}[\![c]\!]\sigma \models B$.

We define that $\perp \models B$, which means any assertions are true in $\perp$.

We can define rules for proving partial correctness assertions. These rules are known as Hoare Rules. Here are 2 such rules:

$$\frac{\{A\}c_0\{C\} \quad \{C\}c_1\{B\}}{\{A\}c_0; c_1\{B\}} \qquad \frac{A \Rightarrow A' \quad \{A'\}c\{B'\} \quad B' \Rightarrow B}{\{A\}c\{B\}}$$

$\models \{A\}c\{B\}$ means $\{A\}c\{B\}$ is true statement. $\vdash \{A\}c\{B\}$ means $\{A\}c\{B\}$ can be proved true using these rules.

The problem is, we have $\vdash \{A\}c\{B\} \Rightarrow \models \{A\}c\{B\}$(soundness), can we also have $\models \{A\}c\{B\} \Rightarrow \vdash \{A\}c\{B\}$(completeness)?

Now define the Assertion language: first we extend arithmetic expression in IMP to include integer variables.

$a ::= a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1 \mid x \mid n \mid i$

$A ::= \neg A \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid a_0 = a_1 \mid a_0 \geq a_1 \mid \forall i \,.\, A \mid \exists i \,.\, A \mid A_0 \Rightarrow A_1$

The rules are:

$\sigma \models true \quad \bot \models false$

$\sigma \models A \wedge B$ if $\sigma \models A$ and $\sigma \models B$

$\sigma \models A \vee B$ if $\sigma \models A$ or $\sigma \models B$

$\sigma \models a_0 = a_1$, if $\mathcal{A}[\![a_0]\!]\sigma = \mathcal{A}[\![a_1]\!]\sigma$

$\sigma \models \forall i \,.\, A$ if $\sigma \models A[n/i]$ for all $n \in Z$

$\sigma \models \exists i \,.\, A$ if $\sigma \models A[n/i]$ for some $n \in Z$

We'd like partial correctness assertions to be universially quantified on any free variables appearing in the assertion. For example, the PCA $\{x > i\}x = x + 1\{x > i\}$ is implicitlly universially quantified over i. To express this notion, we introduce interpretation: $I = Var \rightarrow Z$,

$\mathcal{A}[\![n]\!]I\sigma = n \quad \mathcal{A}[\![x]\!]I\sigma = \sigma(x) \quad \mathcal{A}[\![i]\!]I\sigma = I(i)$

then we write $\sigma \models^I A$ instead of $\sigma \models A$ to indicate that σ satisfies A in interpretation I.

When we write $\sigma \models A$, we will mean that $\sigma \models^I A$ for all interpretations I.
Now we can extend our notion of satisfaction:

$\sigma \models^I a_0 = a_1$, if $\mathcal{A}[\![a_0]\!]\sigma = \mathcal{A}[\![a_1]\!]\sigma$

$\sigma \models^I \forall i \,.\, A$ if $\sigma \models^{I[i \rightarrow n]} A$ for all $n \in Z$

$\sigma \models^I \exists i \,.\, A$ if $\sigma \models^{I[i \rightarrow n]} A$ for some $n \in Z$

$\models^I \{A\}c\{B\} \overset{def}{\Longleftrightarrow} \forall \sigma \in \Sigma_\bot,\ \sigma \models^I A \Rightarrow \mathcal{C}[\![c]\!]\sigma \models^I B$

$\models \{A\}c\{B\} \overset{def}{\Longleftrightarrow} \forall \sigma, I, \sigma \models^I A \Rightarrow \mathcal{C}[\![c]\!]\sigma \models^I B$

$\sigma \models^I \{A\}c\{B\}$ means if any computation of c starts in a state σ satisfying A in interpretation I, it terminates in a state σ' satisfying B in I.

denotational semantics version is: $\sigma \models^I A \Longrightarrow \mathcal{C}[\![c]\!]\sigma \models^I B$