COMP 317: Semantics of Programming Languages

Lecture 4 Summary

This and the next two lectures present the inductive definition of the denotation of programs. Programs update states, so for any Program *P*, [[*P*]] is a function from States to States; if *S* is a State, then [[*P*]](*S*) is the state that results from running the program *P* in the starting state *S*. For example, if S is the state

'a	'b	...	'x	'y	...	'aa	'ab	...
3	0	...	7	25	...	-5	27	...

then [[`'x := 2; 'y := 'x + 'y;`]](S) is the state

'a	'b	...	'x	'y	...	'aa	'ab	...
3	0	...	2	27	...	-5	27	...

Note that this updates the table at `'x` and `'y`.

In order to give a semantics for assignments, we need a notation for table update: *S*[*V*<-*N*] represents the State *S* where the Variable *V* has the integer value *N*. For example, if S is the state

'a	'b	...	'x	'y	...	'aa	'ab	...
3	0	...	7	25	...	-5	27	...

then S[`'x`<-2] is

'a	'b	...	'x	'y	...	'aa	'ab	...
3	0	...	2	25	...	-5	27	...

and S[`'x`<-2][`'y`<-27] is the state

'a	'b	...	'x	'y	...	'aa	'ab	...
3	0	...	2	27	...	-5	27	...

The notation *S*[*V*<-*N*] is formally defined by

- *S*[*V*<-*N*](*V'*) = *N* if *V* equals *V'*; and

- $S[V\text{<-}N](V') = S(V')$ if V is different to V'.

The semantics of assignment is defined by
$[[V\texttt{:=}E\texttt{;}]](S) = S[\ V \text{<-}\ [[E]](S)\ ]$ for all Variables V, Expressions E and States S.

Key Points

- The semantics of assignment is table-update.
- When new notation, such as $S[V\text{<-}N]$, is introduced, it needs to be defined formally, and ideally the formal definition should allow calculation (this is, after all, what we're doing with the semantics of SImpL).

You should by now be thoroughly familiar with inductive definitions, and be able to anticipate how we complete the denotational semantics of SImpL. We've seen formal definitions of the syntax of numerals; of expressions, boolean expressions, and programs in SImpL; we've seen formal definitions of the semantics of numerals, expressions, and assignments. As a general principle, whenever you come across a formal definition, you should be able to think of several examples of that definition, and be able to work out those examples. For example, given the definition of the semantics of assignment, you should be able to come up with, and work out, examples such as $[[$`'x:='x+1;`$]](S)$ for any given State S.

Grant Malcolm