

Augmenting a denotational definition

Sample Answer

The syntax extensions could be:

To declare a pointer variable:

$D ::= \text{ptr } I \mid \dots$

To create a dynamic variable :

$C ::= I := \text{new} \mid \dots$

To assign a value using a pointer variable:

$C ::= \textcircled{I} := E \mid \dots$

To find a variable's location:

$E ::= \textcircled{I} \mid \dots$

To dereference a pointer:

$E ::= \textcircled{I} \mid \dots$

Example program

```
begin
  var x;
  var y;
  ptr p;
  p = @x;
  x := 1;
  ⓐp = 2;
  p := new;
  ⓐp = 3;
  y = ⓐp
end.
```

We need to extend *Denotable-value* as indicated.

$\text{Denotable-value} = \text{NatLocn} + \text{PtrLocn} + \text{Nat} + \text{Errvalue}$

where $\text{Natlocn} = \text{Location}$ (locations that hold numbers)

$\text{Ptrlocn} = \text{Location}$ (locations that hold pointers)

$\text{Errvalue} = \text{Unit}$

We also need to extend *Storable-value* so that pointer variables can hold location values:

$\text{Storable-value} = \text{Nat} + \text{Locnvalue}$,

where $\text{Locnvalue} = \text{Location}$.

The valuation function for declaring a number variable:

$D[\text{ptr } I] = \lambda e. \text{let } (l', e') = (\text{reserve} - \text{locn } e) \text{ in}$

$\text{updateenv}[\textcircled{I}] \text{ in } \text{NatLocn}(l') e'$

The valuation function for declaring a pointer variable:

$$D[\text{ptr } I] = \lambda e. \text{let } (l', e') = (\text{reserve} - \text{locn } e) \text{ in} \\ \text{updateenv}[\![I]\!] \text{inPtrLocn}(l') e'$$

A dynamic variable is created with new:

$$C[\![I := \text{new}]\!] = \lambda e. \lambda s. \text{cases } (\text{accessenv}[\![I]\!] e) \\ \text{isPtrLocn}(l) \rightarrow \text{let } (l', e') = (\text{reserve} - \text{locn } e) \text{ in} \\ \text{inOK}(\text{update } l \text{ inLocnvalue}(l') s) \\ \square \text{isNatLocn}(l'') \rightarrow \text{inErr}(s) \\ \square \text{isNat}(n) \rightarrow \text{inErr}(s) \\ \text{end}$$

Finding the address of a number variable:

$$E[\![@ I]\!] = \lambda e. \lambda s. \text{cases } (\text{accessenv}[\![I]\!] e) \text{ of} \\ \text{isPtrlocn}(l) \rightarrow \text{inErrorvalue}() \\ \square \text{isNatlocn}(l') \rightarrow \text{inLocnvalue}(l') \\ \square \text{isNat}(n) \rightarrow \text{inErrorvalue}() \\ \text{end}$$

Retrieve the contents of a pointer:

$$E[\![\textcircled{I}]\!] = \lambda e. \lambda s. \text{cases } (\text{accessenv}[\![I]\!] e) \text{ of} \\ \text{isPtrlocn}(l) \rightarrow \text{inLocnvalue}(\text{access } l s) \\ \square \text{isNatlocn}(l') \rightarrow \text{inErrorvalue}() \\ \square \text{isNat}(n) \rightarrow \text{inErrorvalue}() \\ \text{end}$$