CS 476 – Programming Language Design

William Mansky

Class Project

- Work alone or in group of 2-3
- Project ideas:
  - Write syntax, typing rules, and/or operational semantics for some language or feature we won't cover in class
  - Design and implement a small domain-specific language
  - Find an existing language definition (CompCert C semantics, WebAssembly interpreter, etc.) and extend it/use it for something
- Project proposal first submission due 10/5
- If you're not sure what to do, come by office hours and we can discuss!

Questions

Nobody has responded yet.

Hang tight! Responses are coming in.

Expressions and Commands: Syntax

E ::= <#> | <ident>
| *E* + *E* | *E* – *E* | *E* * *E*
| <bool>
| *E* and *E* | *E* or *E*
| not *E*
| *E* = *E*
| if *E* then *E* else *E*

C ::= <ident> := *E*
| *C*; *C*
| skip
| if(*E*){ *C* } else { *C* }
| while(*E*) { *C* }

Adding Control Flow

- With variables, assignment, and sequencing, we can write lists of computations:

```
x := 0; x := x + 1; y := x; x := x + 1
```

- Control flow lets us write more complicated patterns:
  - Branching gives us commands that only happen sometimes
  - Loops let us write variable-length programs

```
x := if 1+2=3 then 4 else 5
```
(had before)

```
if(1+2=3) { x := 4 } else { y := 5 }
```
(need to add)

IMP: Syntax

E ::= <#> | <ident>
| *E* + *E* | *E* – *E* | *E* * *E*
| <bool>
| *E* and *E* | *E* or *E*
| not *E*
| *E* = *E*

C ::= <ident> := *E*
| *C*; *C*
| skip
| if(*E*){ *C* } else { *C* }
| while(*E*) { *C* }

IMP: Syntactic Sugar

```
if(e){ c } =>
if(e){ c } else { skip }

do{ c } while(e) =>
c; while(e){ c }

for(i = e; cond; incr) c =>
i := e; while(cond){
  c; incr }
```

$$
\begin{aligned}
C ::= & \ \langle \text{ident} \rangle := E \\
& \mid C; C \\
& \mid \texttt{skip} \\
& \mid \texttt{if}(E)\{ C \}\texttt{else}\{ C \} \\
& \mid \texttt{while}(E)\{ C \}
\end{aligned}
$$

IMP: Types

$$\frac{\Gamma \vdash e : \tau \quad \Gamma(x) = \tau}{\Gamma \vdash x := e : \mathrm{ok}} \qquad \frac{\Gamma \vdash c_1 : \mathrm{ok} \quad \Gamma \vdash c_2 : \mathrm{ok}}{\Gamma \vdash c_1 ; c_2 : \mathrm{ok}}$$

$$\frac{\Gamma \vdash e : \mathrm{bool} \quad \Gamma \vdash c_1 : \mathrm{ok} \quad \Gamma \vdash c_2 : \mathrm{ok}}{\Gamma \vdash \texttt{if}(e)\{c_1\}\ \texttt{else}\ \{c_2\} : \mathrm{ok}}$$

$$\frac{\Gamma \vdash e : \mathrm{bool} \quad \Gamma \vdash c : \mathrm{ok}}{\Gamma \vdash \texttt{while}(e)\{c\} : \mathrm{ok}}$$

Questions

Nobody has responded yet.

Hang tight! Responses are coming in.

Expressions: Small-Step Semantics

- Structural rule:

$$\frac{e \to e'}{\texttt{if}\ e\ \texttt{then}\ e_1\ \texttt{else}\ e_2 \to \texttt{if}\ e'\ \texttt{then}\ e_1\ \texttt{else}\ e_2}$$

- Computation rules:

$$\frac{}{\texttt{if true then}\ e_1\ \texttt{else}\ e_2 \to e_1}$$

$$\frac{}{\texttt{if false then}\ e_1\ \texttt{else}\ e_2 \to e_2}$$

IMP: Small-Step Semantics

- Structural rule:

$$\frac{(e,\rho) \to (e',\rho)}{(\texttt{if(}e\texttt{)\{ }c_1\texttt{ \} else \{ }c_2\texttt{ \}},\rho) \to (\texttt{if(}e'\texttt{)\{ }c_1\texttt{ \} else \{ }c_2\texttt{ \}},\rho)}$$

- Computation rules:

$$\frac{}{(\texttt{if(true)\{ }c_1\texttt{ \} else \{ }c_2\texttt{ \}},\rho) \to (c_1,\rho)}$$

$$\frac{}{(\texttt{if(false)\{ }c_1\texttt{ \} else \{ }c_2\texttt{ \}},\rho) \to (c_2,\rho)}$$

IMP: Small-Step Semantics of Loops

$$\frac{}{(\texttt{repeat}\ c, \rho) \rightarrow ?}$$

- Exercise: Try writing small-step semantic rules for the repeat command, an infinite loop.

IMP: Small-Step Semantics of Loops

$$\frac{}{(\texttt{repeat}\ c, \rho) \to (\texttt{repeat}\ c, \rho)}$$

IMP: Small-Step Semantics of Loops

$$\frac{(c, \rho) \to (c', \rho')}{(\texttt{repeat}\ c, \rho) \to (\texttt{repeat}\ c', \rho')}$$

$$\frac{}{(\texttt{repeat skip}, \rho) \to \texttt{repeat}\ c}$$

But where do we get the c?

IMP: Small-Step Semantics of Loops

$$\frac{}{(\texttt{repeat}\ c, \rho) \to (c\texttt{;}\ \texttt{repeat}\ c, \rho)}$$

$$\frac{(c_1, \rho) \to (c_1', \rho')}{(c_1\texttt{;}\ c_2, \rho) \to (c_1'\texttt{;}\ c_2, \rho')} \qquad \frac{}{(\texttt{skip}\texttt{;}\ c_2, \rho) \to (c_2, \rho)}$$

Questions

Nobody has responded yet.

Hang tight! Responses are coming in.

IMP: Small-Step Semantics of While

- Start with `repeat`:

$$\frac{}{(\texttt{while}(e)\{c\}, \rho) \rightarrow (c; \texttt{while}(e)\{c\}, \rho)}$$

- Need to check the condition before each iteration

IMP: Small-Step Semantics of While

$$\frac{}{(\texttt{while}(e)\{c\}, \rho) \rightarrow (\texttt{if}(e)\{\ c; \texttt{while}(e)\{c\}\ \}\ \texttt{else skip}, \rho)}$$

(while(not(x = 0)){ x := x – 1 }, {x = 2}) →
(if(not(x = 0)){ x := x – 1; while … } else skip, {x = 2}) → … →
(x := x – 1; while …, {x = 2}) → … →
(while(not(x = 0)){ x := x – 1; while … }, {x = 1}) →
(if(not(x = 0)){ x := x – 1; while … } else skip, {x = 1}) → … →
(while(not(x = 0)){ … }, {x = 0}) → … → (skip, {x = 0})

IMP: Big-Step Semantics

$$\frac{(e,\rho) \Downarrow \text{true} \quad (c_1,\rho) \Downarrow \rho'}{(\texttt{if(}e\texttt{)\{}c_1\texttt{\} else \{}c_2\texttt{\}},\rho) \Downarrow \rho'} \qquad \frac{(e,\rho) \Downarrow \text{false} \quad (c_2,\rho) \Downarrow \rho'}{(\texttt{if(}e\texttt{)\{}c_1\texttt{\} else \{}c_2\texttt{\}},\rho) \Downarrow \rho'}$$

$$\frac{(e,\rho) \Downarrow \text{true} \quad (c,\rho) \Downarrow \rho' \quad (\texttt{while(}e\texttt{)\{}c\texttt{\}},\rho') \Downarrow \rho''}{(\texttt{while(}e\texttt{)\{}c\texttt{\}},\rho) \Downarrow \rho''}$$

$$\frac{(e,\rho) \Downarrow \text{false}}{(\texttt{while(}e\texttt{)\{}c\texttt{\}},\rho) \Downarrow \rho}$$

IMP: Interpreter

```
let rec eval_cmd (c : cmd) (s : state) =
  match c with
  | While (e, c) ->
      (match eval_exp e s with
       | Some (BoolVal true) ->
          (match eval_cmd c s with
           | Some s' -> eval_cmd (While (e, c)) s'
```

$$\frac{(e,\rho) \Downarrow \mathrm{true} \quad (c,\rho) \Downarrow \rho' \quad (\texttt{while}(e)\{c\},\rho') \Downarrow \rho''}{(\texttt{while}(e)\{c\},\rho) \Downarrow \rho'}$$

IMP: Infinite Loops

$$\frac{(e,\rho) \Downarrow \text{true} \quad (c,\rho) \Downarrow \rho' \quad (\texttt{while}(e)\{c\},\rho') \Downarrow \rho''}{(\texttt{while}(e)\{c\},\rho) \Downarrow \rho''}$$

$$\frac{?}{(\texttt{while(true)\{ skip \}},\{\}) \Downarrow ?}$$

IMP: Infinite Loops

$$\frac{(e,\rho) \Downarrow \text{true} \quad (c,\rho) \Downarrow \rho' \quad (\texttt{while}(e)\{c\},\rho') \Downarrow \rho''}{(\texttt{while}(e)\{c\},\rho) \Downarrow \rho''}$$

$$\frac{(\texttt{true},\{\}) \Downarrow \text{true} \quad (\texttt{skip},\{\}) \Downarrow \{\} \quad (\texttt{while(true)\{ skip \}},\{\}) \Downarrow ?}{(\texttt{while(true)\{ skip \}},\{\}) \Downarrow ?}$$

IMP: Infinite Loops

$$\frac{(e,\rho) \Downarrow \text{true} \quad (c,\rho) \Downarrow \rho' \quad (\texttt{while}(e)\{c\},\rho') \Downarrow \rho''}{(\texttt{while}(e)\{c\},\rho) \Downarrow \rho''}$$

$$\frac{(\texttt{true},\{\}) \Downarrow \text{true} \quad (\texttt{skip},\{\}) \Downarrow \{\} \quad \frac{\cdots}{(\texttt{while(true)\{ skip \}},\{\}) \Downarrow ?}}{(\texttt{while(true)\{ skip \}},\{\}) \Downarrow ?}$$

- So we can never prove it evaluates to anything!

Questions

Nobody has responded yet.

Hang tight! Responses are coming in.