Abstract syntax of IMP

Syntactic domains:

```
C ∈ Command
E ∈ Expression
L ∈ Location
N ∈ Numeral
```

Syntactic rules:

$$
\begin{aligned}
\texttt{C} &::= \texttt{L := E} \\
&\mid \texttt{C}_1\texttt{; C}_2 \\
&\mid \textbf{if}\ \texttt{E}\ \textbf{then}\ \texttt{C}_1\ \textbf{else}\ \texttt{C}_2 \\
&\mid \textbf{while}\ \texttt{E}\ \textbf{do}\ \texttt{C} \\
&\mid \textbf{skip} \\
\texttt{E} &::= \texttt{N} \\
&\mid \texttt{@ L} \\
&\mid \texttt{E}_1 + \texttt{E}_2 \mid \ldots \\
&\mid \textbf{not}\ \texttt{E} \mid \texttt{E}_1 = \texttt{E}_2 \mid \ldots \\
\texttt{L} &::= loc_i,\ i > 0 \\
\texttt{N} &::= n,\ n \in \mathit{Integer}
\end{aligned}
$$

Type system for IMP

For `Numeral`:

$$n : int \qquad \text{if } n \in Integer$$

For `Location`:

$$loc_i : intloc$$

For `Expression`:

$$\frac{N : int}{N : intexp} \qquad \frac{L : intloc}{@L : intexp} \qquad \frac{E_1 : intexp \qquad E_2 : intexp}{E_1 + E_2 : intexp}$$

$$\frac{E : boolexp}{\text{not } E : boolexp} \qquad \frac{E_1 : \tau exp \qquad E_2 : \tau exp}{E_1 = E_2 : boolexp} \qquad \text{if} \tau \in \{int, bool\}$$

For `Command`:

$$\frac{L : intloc \qquad E : intexp}{L := E : comm} \qquad \frac{C_1 : comm \qquad C_2 : comm}{C_1 ; C_2 : comm}$$

$$\frac{E : boolexp \qquad C_1 : comm \qquad C_2 : comm}{\text{if } E \text{ then } C_1 \text{ else } C_2 : comm}$$

$$\frac{E : boolexp \qquad C : comm}{\text{while } E \text{ do } C : comm} \qquad \text{skip} : comm$$

Denotational semantics for IMP

Semantic domains:

$\texttt{Bool} = \{\texttt{true}, \texttt{false}\}$

- $\texttt{not: Bool} \to \texttt{Bool}$
- $\texttt{equalbool: Bool} \times \texttt{Bool} \to \texttt{Bool}$

$\texttt{Int} = \{\ldots, -1, 0, 1, \ldots\}$

- $\texttt{plus: Int} \times \texttt{Int} \to \texttt{Int}$
- ...
- $\texttt{equalint: Int} \times \texttt{Int} \to \texttt{Bool}$

$\texttt{Loc} = \{ \mathit{loc}_i \mid i > 0 \}$

$\texttt{Store} = \{\langle n_1, n_2, \ldots, n_m\rangle \mid n_i \in \texttt{Int},\ 1 \le i \le m,\ m \ge 0\}$

- $\texttt{lookup: Loc} \times \texttt{Store} \to \texttt{Int}$
  - $\texttt{lookup}(\mathit{loc}_j, \langle n_1, n_2, \ldots, n_m\rangle) =$
    - n_j, if $j \le m$
    - 0, otherwise
- $\texttt{update: Loc} \times \texttt{Int} \times \texttt{Store} \to \texttt{Store}$
  - $\texttt{update}(\mathit{loc}_j, n, \langle n_1, n_2, \ldots, n_j, \ldots, n_m\rangle) =$
    - $\langle n_1, n_2, \ldots, n, \ldots, n_m\rangle$, if $j \le m$
    - $\langle n_1, n_2, \ldots, n_j, \ldots, n_m\rangle$, otherwise
- $\texttt{if: Bool} \times \texttt{Store} \times \texttt{Store} \to \texttt{Store}$
  - $\texttt{if}(\texttt{true}, s_1, s_2) = s_1$
  - $\texttt{if}(\texttt{false}, s_1, s_2) = s_2$

Denotational semantics, cont.

Semantic rules:

$\mathcal{N}\texttt{: Numeral} \to \texttt{Int}$

$\mathcal{N}\ [\![n : \mathit{int}]\!] = n$

$\mathcal{L}\texttt{: Location} \to \texttt{Loc}$

$\mathcal{L}\ [\![\mathit{loc}_i : \mathit{intloc}]\!] = \mathit{loc}_i$

$\mathcal{E}\texttt{: Expression} \times \texttt{Store} \to \texttt{Int} \cup \texttt{Bool}$

$\mathcal{E}\ [\![\texttt{N} : \mathit{intexp}]\!]\ \texttt{s} = \mathcal{N}\ [\![\texttt{N} : \mathit{int}]\!]$

$\mathcal{E}\ [\![\texttt{@L} : \mathit{intexp}]\!]\ \texttt{s} = \texttt{lookup}(\mathcal{L}\ [\![\texttt{L} : \mathit{intloc}]\!],\ \texttt{s})$

$\mathcal{E}\ [\![\texttt{E}_1 + \texttt{E}_2 : \mathit{intexp}]\!]\ \texttt{s} =$
$\quad \texttt{plus}(\mathcal{E}\ [\![\texttt{E}_1 : \mathit{intexp}]\!]\ \texttt{s},\ \mathcal{E}\ [\![\texttt{E}_2 : \mathit{intexp}]\!]\ \texttt{s})$

$\mathcal{E}\ [\![\texttt{not E} : \mathit{boolexp}]\!]\ \texttt{s} =$
$\quad \texttt{not}(\mathcal{E}\ [\![\texttt{E} : \mathit{boolexp}]\!]\ \texttt{s})$

$\mathcal{E}\ [\![\texttt{E}_1 = \texttt{E}_2 : \mathit{boolexp}]\!]\ \texttt{s} =$
$\quad \texttt{equalint}(\mathcal{E}\ [\![\texttt{E}_1 : \mathit{intexp}]\!]\ \texttt{s},\ \mathcal{E}\ [\![\texttt{E}_2 : \mathit{intexp}]\!]\ \texttt{s})$

$\mathcal{E}\ [\![\texttt{E}_1 = \texttt{E}_2 : \mathit{boolexp}]\!]\ \texttt{s} =$
$\quad \texttt{equalbool}(\mathcal{E}[\![\texttt{E}_1 : \mathit{boolexp}]\!]\ \texttt{s},\ \mathcal{E}[\![\texttt{E}_2 : \mathit{boolexp}]\!]\ \texttt{s})$

Denotational semantics for commands, first try

Semantic rules, cont.:

$\mathcal{C}$: Command × Store → Store

```
𝒞 [[L := E:comm]] s =
    update(ℒ [[L:intloc]], ℰ [[E:intexp]] s, s)
𝒞 [[C1;C2:comm]] s =
    𝒞 [[C2:comm]] (𝒞 [[C1:comm]] s)
𝒞 [[if E then C1 else C2:comm]] s =
    if(ℰ [[E:boolexp]] s,
       𝒞 [[C1:comm]] s,
       𝒞 [[C2:comm]] s)
𝒞 [[while E do C:comm]] s = ...
𝒞 [[skip:comm]] s = s
```


Alternatives for while rule

Alternative 1:

```
𝒞 [[while E do C:comm]] s =
    if(ℰ [[E:boolexp]] s,
       𝒞 [[while E do C:comm]] (𝒞 [[C:comm]] s),
       s)
```

- not clear what recursion in definition of $\mathcal{C}$ means
- not compositional

Alternative 2:

```
𝒞 [[while E do C:comm]] s = w(s)
    where w(s) =
       if(ℰ [[E:boolexp]] s,
          w(𝒞 [[C:comm]] s),
          s)
```

- not clear what recursion in definition of `w` means

What is meaning of

```
while 0 = 0 do skip
```

?

Unrollings of recursive function

$\mathtt{w}_0\mathtt{(s)} = \bot$

$\mathtt{w}_1\mathtt{(s)} = \mathtt{if}(\mathcal{E}[\![\mathtt{E}:boolexp]\!]\ \mathtt{s},\ \mathtt{w}_0(\mathcal{C}[\![\mathtt{C}:comm]\!]\ \mathtt{s}),\ \mathtt{s})$

$\mathtt{w}_2\mathtt{(s)} = \mathtt{if}(\mathcal{E}[\![\mathtt{E}:boolexp]\!]\ \mathtt{s},\ \mathtt{w}_1(\mathcal{C}[\![\mathtt{C}:comm]\!]\ \mathtt{s}),\ \mathtt{s})$

...

$\mathtt{w}_i\mathtt{(s)} = \mathtt{if}(\mathcal{E}[\![\mathtt{E}:boolexp]\!]\ \mathtt{s},\ \mathtt{w}_{i-1}(\mathcal{C}[\![\mathtt{C}:comm]\!]\ \mathtt{s}),\ \mathtt{s})$

...

$$w \approx \lim_{i \to \infty} (w_i)$$

Denotational semantics for commands, revised

Revised semantic rules for `Command`:

$\mathcal{C}$: $\mathtt{Command} \times \mathtt{Store}_\bot \to \mathtt{Store}_\bot$

$\mathcal{C}[\![\mathtt{C}:comm]\!]\ \bot = \bot$

$\mathcal{C}[\![\mathtt{L := E}:comm]\!]\ \mathtt{s} =$
  $\mathtt{update}(\mathcal{L}[\![\mathtt{L}:intloc]\!],\ \mathcal{E}[\![\mathtt{E}:intexp]\!]\ \mathtt{s},\ \mathtt{s})$

$\mathcal{C}[\![\mathtt{C}_1\mathtt{;C}_2:comm]\!]\ \mathtt{s} =$
  $\mathcal{C}[\![\mathtt{C}_2:comm]\!]\ (\mathcal{C}[\![\mathtt{C}_1:comm]\!]\ \mathtt{s})$

$\mathcal{C}[\![\mathtt{if\ E\ then\ C}_1\ \mathtt{else\ C}_2:comm]\!]\ \mathtt{s} =$
  $\mathtt{if}(\mathcal{E}[\![\mathtt{E}:boolexp]\!]\ \mathtt{s},$
    $\mathcal{C}[\![\mathtt{C}_1:comm]\!]\ \mathtt{s},$
    $\mathcal{C}[\![\mathtt{C}_2:comm]\!]\ \mathtt{s})$

$\mathcal{C}[\![\mathtt{while\ E\ do\ C}:comm]\!]\ \mathtt{s} = \ldots$

$\mathcal{C}[\![\mathtt{skip}:comm]\!]\ \mathtt{s} = \mathtt{s}$

Also revised interface to `if` function:

$\mathtt{if}$: $\mathtt{Bool} \times \mathtt{Store}_\bot \times \mathtt{Store}_\bot \to \mathtt{Store}_\bot$