CS 4110

Programming Languages & Logics

Lecture 5
IMP Properties

10 September 2014

Announcements

Office Hours

- Fran: Wednesday at 11-12pm

Homework #1

- Due: Today

Homework #2

- Out: Today

Review

Last time we defined the IMP programming language...

$$
\begin{aligned}
a &::= x \mid n \mid a_1 + a_2 \mid a_1 \times a_2 \\
b &::= \textbf{true} \mid \textbf{false} \mid a_1 < a_2 \\
c &::= \textbf{skip} \\
&\quad \mid x := a \\
&\quad \mid c_1 ; c_2 \\
&\quad \mid \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2 \\
&\quad \mid \textbf{while } b \textbf{ do } c
\end{aligned}
$$

Quiz: What does this program do?

```
x1 := n1;  x2 := n2;  x3 := n3;  tmp := 0;
if x2 <= x1 then
        tmp := x2;
        x2 := x1;
        x1 := tmp
else skip;
if x3 <= x2 then
        tmp := x3;
        x3 := x2;
        x2 := tmp
else skip
```

How about this one?

```
x1 := n1;   x2 := n2;   x3 := n3;   tmp := 0;   swaps := 1;
while 0 < swaps do
    swaps := 0;
    if x2 <= x1 then
        tmp := x2;
        x2 := x1;
        x1 := tmp;
        swaps := swaps + 1
    else skip;
    if x3 <= x2 then
        tmp := x3;
        x3 := x2;
        x2 := tmp;
        swaps := swaps + 1
    else skip
```

Does this program terminate?

```
x1 := n1;   x2 := n2;   x3 := n3;   tmp := 0;   swaps := 1;
while 0 < swaps do
        swaps := 0;
        if x2 <= x1 then
                tmp := x2;
                x2 := x1;
                x1 := tmp;
                swaps := swaps + 1
        else skip;
        if x3 <= x2 then
                tmp := x3;
                x3 := x2;
                x2 := tmp;
                swaps := swaps + 1
        else skip
```

IMP Questions

- Q: Can you write a program that doesn’t terminate?

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A: **while true do skip**

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A: **while true do skip**
- Q: Does this mean that IMP is Turing complete?

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A: **while true do skip**
- Q: Does this mean that IMP is Turing complete?
- A: Not quite... we also need to check the language is not finite state... but IMP has real mathematical integers.

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A: **while true do skip**
- Q: Does this mean that IMP is Turing complete?
- A: Not quite... we also need to check the language is not finite state... but IMP has real mathematical integers.
- Q: What if we replace **Int** with **Int64**?

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A: `while true do skip`
- Q: Does this mean that IMP is Turing complete?
- A: Not quite... we also need to check the language is not finite state... but IMP has real mathematical integers.
- Q: What if we replace **Int** with **Int64**?
- A: Then we would lose Turing completeness.

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A: **while true do skip**
- Q: Does this mean that IMP is Turing complete?
- A: Not quite... we also need to check the language is not finite state... but IMP has real mathematical integers.
- Q: What if we replace **Int** with **Int64**?
- A: Then we would lose Turing completeness.
- Q: How much space do we need to represent configurations during execution of an IMP program?

IMP Questions

- Q: Can you write a program that doesn't terminate?
- A:
```
while true do skip
```
- Q: Does this mean that IMP is Turing complete?
- A: Not quite... we also need to check the language is not finite state... but IMP has real mathematical integers.
- Q: What if we replace **Int** with **Int64**?
- A: Then we would lose Turing completeness.
- Q: How much space do we need to represent configurations during execution of an IMP program?
- A: Can calculate a fixed bound!

Determinism

Theorem

$\forall c \in \mathbf{Com}, \sigma, \sigma' \sigma'' \in \mathbf{Store}.$
if $\langle \sigma, c \rangle \Downarrow \sigma'$ *and* $\langle \sigma, c \rangle \Downarrow \sigma''$ *then* $\sigma' = \sigma''$.

Determinism

Theorem

$\forall c \in \mathbf{Com}, \sigma, \sigma'\ \sigma'' \in \mathbf{Store}.$
if $\langle \sigma, c \rangle \Downarrow \sigma'$ *and* $\langle \sigma, c \rangle \Downarrow \sigma''$ *then* $\sigma' = \sigma''$.

Proof.

By structural induction on *c*... □

Determinism

Theorem

$\forall c \in \mathbf{Com}, \sigma, \sigma'\ \sigma'' \in \mathbf{Store}.$
if $\langle \sigma, c \rangle \Downarrow \sigma'$ *and* $\langle \sigma, c \rangle \Downarrow \sigma''$ *then* $\sigma' = \sigma''$.

Proof.

By structural induction on *c*... □

Proof.

By induction on the derivation of $\langle \sigma, c \rangle \Downarrow \sigma'$... □

Derivations

Write $\mathcal{D} \Vdash y$ if the conclusion of derivation $\mathcal{D}$ is y.

Derivations

Write $\mathcal{D} \Vdash y$ if the conclusion of derivation $\mathcal{D}$ is y.

Example:

Given the derivation,

$$\dfrac{\dfrac{\dfrac{}{\langle \sigma, 6 \rangle \Downarrow 6} \qquad \dfrac{}{\langle \sigma, 7 \rangle \Downarrow 7}}{\langle \sigma, 6 \times 7 \rangle \Downarrow 42}}{\langle \sigma, \mathrm{i} := 6 \times 7 \rangle \Downarrow \sigma[\mathrm{i} \mapsto 42]}$$

we would write: $\mathcal{D} \Vdash \langle \sigma, \mathrm{i} := 42 \rangle \Downarrow \sigma[\mathrm{i} \mapsto 42]$

Induction on Derivations

Given a set of axioms and inference rules, the set of derivations is itself an inductively defined set!

Induction on Derivations

Given a set of axioms and inference rules, the set of derivations is itself an inductively defined set!

This means we can prove properties by induction on derivations!

Induction on Derivations

Given a set of axioms and inference rules, the set of derivations is itself an inductively defined set!

This means we can prove properties by induction on derivations!

A derivation $\mathcal{D}'$ is an immediate subderivation of $\mathcal{D}$ if $\mathcal{D}' \Vdash z$ where z is one of the premises used of the final rule of derivation $\mathcal{D}$.

Induction on Derivations

Given a set of axioms and inference rules, the set of derivations is itself an inductively defined set!

This means we can prove properties by induction on derivations!

A derivation $\mathcal{D}'$ is an immediate subderivation of $\mathcal{D}$ if $\mathcal{D}' \Vdash z$ where z is one of the premises used of the final rule of derivation $\mathcal{D}$.

In a proof by induction on derivations, for every axiom and inference rule, assume that the property P holds for all immediate subderivations, and show that it holds of the conclusion.

Large-Step Semantics

$$\text{Skip}\ \frac{}{\langle \sigma, \textbf{skip} \rangle \Downarrow \sigma} \qquad \text{Assgn}\ \frac{\langle \sigma, e \rangle \Downarrow n}{\langle \sigma, x := e \rangle \Downarrow \sigma[x \mapsto n]}$$

$$\text{Seq}\ \frac{\langle \sigma, c_1 \rangle \Downarrow \sigma' \qquad \langle \sigma', c_2 \rangle \Downarrow \sigma''}{\langle \sigma, c_1; c_2 \rangle \Downarrow \sigma''}$$

$$\text{If-T}\ \frac{\langle \sigma, b \rangle \Downarrow \textbf{true} \qquad \langle \sigma, c_1 \rangle \Downarrow \sigma'}{\langle \sigma, \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2 \rangle \Downarrow \sigma'}$$

$$\text{If-F}\ \frac{\langle \sigma, b \rangle \Downarrow \textbf{false} \qquad \langle \sigma, c_2 \rangle \Downarrow \sigma'}{\langle \sigma, \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2 \rangle \Downarrow \sigma'}$$

$$\text{While-T}\ \frac{\langle \sigma, b \rangle \Downarrow \textbf{true} \qquad \langle \sigma, c \rangle \Downarrow \sigma' \qquad \langle \sigma', \textbf{while}\ b\ \textbf{do}\ c \rangle \Downarrow \sigma''}{\langle \sigma, \textbf{while}\ b\ \textbf{do}\ c \rangle \Downarrow \sigma''}$$

$$\text{While-F}\ \frac{\langle \sigma, b \rangle \Downarrow \textbf{false}}{\langle \sigma, \textbf{while}\ b\ \textbf{do}\ c \rangle \Downarrow \sigma}$$