

CS 4110

# Programming Languages & Logics

## Lecture 5 IMP Properties



# Command Equivalence

Intuitively, two commands are equivalent if they produce the same result under any store...

## Definition (Equivalence of commands)

Two commands  $c$  and  $c'$  are equivalent (written  $c \sim c'$ ) if, for any stores  $\sigma$  and  $\sigma'$ , we have

$$\langle \sigma, c \rangle \Downarrow \sigma' \iff \langle \sigma, c' \rangle \Downarrow \sigma'.$$

# Command Equivalence

For example, we can prove that every **while** command is equivalent to its “unrolling”:

## Theorem

For all  $b \in \mathbf{Bexp}$  and  $c \in \mathbf{Com}$ ,

$$\mathbf{while } b \mathbf{ do } c \sim \mathbf{if } b \mathbf{ then } (c; \mathbf{while } b \mathbf{ do } c) \mathbf{ else skip}$$

## Proof.

We show each implication separately...



# 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,

$$\frac{\frac{\frac{}{\langle \sigma, 6 \rangle \Downarrow 6}}{\langle \sigma, 6 \times 7 \rangle \Downarrow 42} \quad \frac{}{\langle \sigma, 7 \rangle \Downarrow 7}}{\langle \sigma, i := 6 \times 7 \rangle \Downarrow \sigma[i \mapsto 42]}$$

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



# 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 inference rule, assume that the property  $P$  holds for all immediate subderivations, and show that it holds of the conclusion.

# Large-Step Semantics

$$\text{SKIP} \frac{}{\langle \sigma, \mathbf{skip} \rangle \Downarrow \sigma} \qquad \text{ASSGN} \frac{\langle \sigma, a \rangle \Downarrow n}{\langle \sigma, x := a \rangle \Downarrow \sigma[x \mapsto n]}$$

$$\text{SEQ} \frac{\langle \sigma, c_1 \rangle \Downarrow \sigma' \quad \langle \sigma', c_2 \rangle \Downarrow \sigma''}{\langle \sigma, c_1; c_2 \rangle \Downarrow \sigma''}$$

$$\text{IF-T} \frac{\langle \sigma, b \rangle \Downarrow \mathbf{true} \quad \langle \sigma, c_1 \rangle \Downarrow \sigma'}{\langle \sigma, \mathbf{if } b \mathbf{ then } c_1 \mathbf{ else } c_2 \rangle \Downarrow \sigma'}$$

$$\text{IF-F} \frac{\langle \sigma, b \rangle \Downarrow \mathbf{false} \quad \langle \sigma, c_2 \rangle \Downarrow \sigma'}{\langle \sigma, \mathbf{if } b \mathbf{ then } c_1 \mathbf{ else } c_2 \rangle \Downarrow \sigma'}$$

$$\text{WHILE-T} \frac{\langle \sigma, b \rangle \Downarrow \mathbf{true} \quad \langle \sigma, c \rangle \Downarrow \sigma' \quad \langle \sigma', \mathbf{while } b \mathbf{ do } c \rangle \Downarrow \sigma''}{\langle \sigma, \mathbf{while } b \mathbf{ do } c \rangle \Downarrow \sigma''}$$

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