



**PSC 2023/24** (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni

<http://www.di.unipi.it/~bruni/>

<http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/start>

10 - Consistency and congruence

# Operational equivalence

# Operational equivalence

$$a_1 \sim_{\text{op}} a_2 \quad \text{iff} \quad \forall \sigma, n. ( \langle a_1, \sigma \rangle \rightarrow n \Leftrightarrow \langle a_2, \sigma \rangle \rightarrow n )$$

$$b_1 \sim_{\text{op}} b_2 \quad \text{iff} \quad \forall \sigma, v. ( \langle b_1, \sigma \rangle \rightarrow v \Leftrightarrow \langle b_2, \sigma \rangle \rightarrow v )$$

$$c_1 \sim_{\text{op}} c_2 \quad \text{iff} \quad \forall \sigma, \sigma'. ( \langle c_1, \sigma \rangle \rightarrow \sigma' \Leftrightarrow \langle c_2, \sigma \rangle \rightarrow \sigma' )$$

termination and determinacy does not matter:  
operational equivalence is always well-defined

# Congruence

$$a_1 \sim_{\text{op}} a_2 \quad \text{iff} \quad \forall \sigma, n. ( \langle a_1, \sigma \rangle \rightarrow n \Leftrightarrow \langle a_2, \sigma \rangle \rightarrow n )$$

take any context  $\mathbb{A}[\cdot]$  e.g.  $2 \times ([\cdot] + 5)$

is it the case that  $a_1 \sim_{\text{op}} a_2 \Rightarrow \mathbb{A}[a_1] \sim_{\text{op}} \mathbb{A}[a_2]$  ?

that is: can we replace a subexpressions with any equivalent one without changing the outcome?



# Contexts

what are the possible contexts for arithmetic expressions?

$$[\cdot] + 5$$

$$2 \times ([\cdot] + 5)$$

$$2 \times ([\cdot] + 5) \leq 50$$

$$(2 \times ([\cdot] + 5) \leq 50) \wedge x = y$$

$$x := 2 \times ([\cdot] + 5)$$

**while**  $x \leq 100$  **do**  $x := 2 \times ([\cdot] + 5)$

# Contexts

what are the possible contexts for arithmetic expressions?

$\mathbb{A}[\cdot]$	$::=$	$[\cdot]$		$\mathbb{C}[\cdot]$	$::=$	$x := \mathbb{A}[\cdot]$
		$\mathbb{A}[\cdot] \text{ op } a$				$\mathbb{C}[\cdot]; c$
		$a \text{ op } \mathbb{A}[\cdot]$				$c; \mathbb{C}[\cdot]$
						<b>if</b> $\mathbb{B}[\cdot]$ <b>then</b> $c$ <b>else</b> $c$
$\mathbb{B}[\cdot]$	$::=$	$\mathbb{A}[\cdot] \text{ cmp } a$				<b>if</b> $b$ <b>then</b> $\mathbb{C}[\cdot]$ <b>else</b> $c$
		$a \text{ cmp } \mathbb{A}[\cdot]$				<b>if</b> $b$ <b>then</b> $c$ <b>else</b> $\mathbb{C}[\cdot]$
		$\neg \mathbb{B}[\cdot]$				<b>while</b> $\mathbb{B}[\cdot]$ <b>do</b> $c$
		$\mathbb{B}[\cdot] \text{ bop } b$				<b>while</b> $b$ <b>do</b> $\mathbb{C}[\cdot]$
		$b \text{ bop } \mathbb{B}[\cdot]$				

# Proof obligations

many proof obligations to deal with:

$$\forall a, a_1, a_2. ( a_1 \sim_{\text{op}} a_2 \Rightarrow a_1 \text{ op } a \sim_{\text{op}} a_2 \text{ op } a )$$

$$\forall a, a_1, a_2. ( a_1 \sim_{\text{op}} a_2 \Rightarrow a \text{ op } a_1 \sim_{\text{op}} a \text{ op } a_2 )$$

$$\forall a, a_1, a_2. ( a_1 \sim_{\text{op}} a_2 \Rightarrow a \text{ cmp } a_1 \sim_{\text{op}} a \text{ cmp } a_2 )$$

$$\forall a, a_1, a_2. ( a_1 \sim_{\text{op}} a_2 \Rightarrow a_1 \text{ cmp } a \sim_{\text{op}} a_2 \text{ cmp } a )$$

$$\forall x, a_1, a_2. ( a_1 \sim_{\text{op}} a_2 \Rightarrow x := a_1 \sim_{\text{op}} x := a_2 )$$

similarly for boolean expressions and commands

# Denotational equivalence

# Denotational equivalence

$$a_1 \sim_{\text{den}} a_2 \quad \text{iff} \quad \mathcal{A}[a_1] = \mathcal{A}[a_2]$$

$$b_1 \sim_{\text{den}} b_2 \quad \text{iff} \quad \mathcal{B}[b_1] = \mathcal{B}[b_2]$$

$$c_1 \sim_{\text{den}} c_2 \quad \text{iff} \quad \mathcal{C}[c_1] = \mathcal{C}[c_2]$$

(two functions are the same  
if they coincide on all arguments)

# Compositionality principle

$$a_1 \sim_{\text{den}} a_2 \quad \text{iff} \quad \mathcal{A}[[a_1]] = \mathcal{A}[[a_2]]$$

take any context  $\mathbb{A}[\cdot]$

is it the case that  $a_1 \sim_{\text{den}} a_2 \Rightarrow \mathbb{A}[a_1] \sim_{\text{den}} \mathbb{A}[a_2]$ ?

YES! it is guaranteed by the compositionally principle of denotational semantics:

*the meaning of a compound expression is solely determined by the meaning of its constituents*

# Consistency

if we guarantee the consistency between  
the operational semantics and  
the denotational semantics  
then the congruence property is guaranteed  
for the operational semantics too

$$\forall a_1, a_2. ( a_1 \sim_{\text{op}} a_2 \stackrel{?}{\Leftrightarrow} a_1 \sim_{\text{den}} a_2 )$$

$$\forall b_1, b_2. ( b_1 \sim_{\text{op}} b_2 \stackrel{?}{\Leftrightarrow} b_1 \sim_{\text{den}} b_2 )$$

$$\forall c_1, c_2. ( c_1 \sim_{\text{op}} c_2 \stackrel{?}{\Leftrightarrow} c_1 \sim_{\text{den}} c_2 )$$

# Consistency: expressions

$$\forall a \in Aexp \ \forall \sigma \in \Sigma. \langle a, \sigma \rangle \rightarrow \mathcal{A} \llbracket a \rrbracket \sigma$$

$$P(a) \stackrel{\text{def}}{=} \forall \sigma \in \Sigma. \langle a, \sigma \rangle \rightarrow \mathcal{A} \llbracket a \rrbracket \sigma$$

by structural induction

$$\forall b \in Bexp \ \forall \sigma \in \Sigma. \langle b, \sigma \rangle \rightarrow \mathcal{B} \llbracket b \rrbracket \sigma$$

$$P(b) \stackrel{\text{def}}{=} \forall \sigma \in \Sigma. \langle b, \sigma \rangle \rightarrow \mathcal{B} \llbracket b \rrbracket \sigma$$

by structural induction



# Consistency: commands

$$\forall c \in Com. \forall \sigma, \sigma' \in \Sigma. \quad \langle c, \sigma \rangle \rightarrow \sigma' \quad \Leftrightarrow \quad \mathcal{C} [c] \sigma = \sigma'$$

can we write it as

$$\forall c \in Com. \forall \sigma \in \Sigma. \quad \langle c, \sigma \rangle \rightarrow \mathcal{C} [c] \sigma \quad ?$$

no, because there is no such formula as

$$\langle c, \sigma \rangle \rightarrow \perp$$

# Consistency: commands

$$\forall c \in Com. \forall \sigma, \sigma' \in \Sigma. \quad \langle c, \sigma \rangle \rightarrow \sigma' \quad \Leftrightarrow \quad \mathcal{C} \llbracket c \rrbracket \sigma = \sigma'$$

$$\forall c \in Com. \forall \sigma, \sigma' \in \Sigma.$$

Correctness

$$P(\langle c, \sigma \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket c \rrbracket \sigma = \sigma' \quad \text{by rule induction}$$

$$\forall c \in Com.$$

Completeness

$$P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma' \in \Sigma. \quad \mathcal{C} \llbracket c \rrbracket \sigma = \sigma' \quad \Rightarrow \quad \langle c, \sigma \rangle \rightarrow \sigma'$$

by structural induction

# Correctness

$$\forall c \in Com, \forall \sigma, \sigma' \in \Sigma$$

$$P(\langle c, \sigma \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} [c] \sigma = \sigma'$$

by rule induction

$$\frac{}{\langle \mathbf{skip}, \sigma \rangle \rightarrow \sigma}$$

We want to prove

$$P(\langle \mathbf{skip}, \sigma \rangle \rightarrow \sigma) \stackrel{\text{def}}{=} \mathcal{C} \llbracket \mathbf{skip} \rrbracket \sigma = \sigma$$

Obviously the proposition is true by the definition of the denotational semantics.

$$\frac{\langle a, \sigma \rangle \rightarrow m}{\langle x := a, \sigma \rangle \rightarrow \sigma [^m / x]}$$

We assume  $\langle a, \sigma \rangle \rightarrow m$  and hence  $\mathcal{A} \llbracket a \rrbracket \sigma = m$  by the equivalence of the operational and denotational semantics of arithmetic expressions.

We want to prove

$$P(\langle x := a, \sigma \rangle \rightarrow \sigma [^m / x]) \stackrel{\text{def}}{=} \mathcal{C} \llbracket x := a \rrbracket \sigma = \sigma [^m / x]$$

By the definition of the denotational semantics

$$\mathcal{C} \llbracket x := a \rrbracket \sigma = \sigma [\mathcal{A} \llbracket a \rrbracket \sigma / x] = \sigma [^m / x]$$

$$\frac{\langle c_0, \sigma \rangle \rightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \rightarrow \sigma'}{\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'}$$

We assume

$$P(\langle c_0, \sigma \rangle \rightarrow \sigma'') \stackrel{\text{def}}{=} \mathcal{C} \llbracket c_0 \rrbracket \sigma = \sigma''$$

$$P(\langle c_1, \sigma'' \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket c_1 \rrbracket \sigma'' = \sigma'$$

We want to prove

$$P(\langle c_0; c_1, \sigma \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket c_0; c_1 \rrbracket \sigma = \sigma'$$

By the denotational semantics definition and the inductive hypotheses

$$\mathcal{C} \llbracket c_0; c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_1 \rrbracket^* (\mathcal{C} \llbracket c_0 \rrbracket \sigma) = \mathcal{C} \llbracket c_1 \rrbracket^* \sigma'' = \mathcal{C} \llbracket c_1 \rrbracket \sigma'' = \sigma'$$

Note that the lifting operator can be removed because  $\sigma'' \neq \perp$  by the inductive hypothesis.

$$\frac{\langle b, \sigma \rangle \rightarrow \mathbf{true} \quad \langle c_0, \sigma \rangle \rightarrow \sigma'}{\langle \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1, \sigma \rangle \rightarrow \sigma'}$$

We assume

- $\langle b, \sigma \rangle \rightarrow \mathbf{true}$  and therefore  $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{true}$  by the correspondence between the operational and denotational semantics for boolean expressions;
- $P(\langle c_0, \sigma \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket c_0 \rrbracket \sigma = \sigma'$

We want to prove

$$P(\langle \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1, \sigma \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1 \rrbracket \sigma = \sigma'$$

In fact, we have

$$\begin{aligned} \mathcal{C} \llbracket \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1 \rrbracket \sigma &= \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket c_0 \rrbracket \sigma, \mathcal{C} \llbracket c_1 \rrbracket \sigma \\ &= \mathbf{true} \rightarrow \sigma', \mathcal{C} \llbracket c_1 \rrbracket \sigma \\ &= \sigma' \end{aligned}$$

$$\frac{\langle b, \sigma \rangle \rightarrow \mathbf{false}}{\langle \mathbf{while } b \mathbf{ do } c, \sigma \rangle \rightarrow \sigma}$$

We assume  $\langle b, \sigma \rangle \rightarrow \mathbf{false}$  and therefore  $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{false}$ .

We want to prove

$$P(\langle \mathbf{while } b \mathbf{ do } c, \sigma \rangle \rightarrow \sigma) \stackrel{\text{def}}{=} \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket \sigma = \sigma$$

By the fixpoint property of the denotational semantics

$$\begin{aligned} \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket \sigma &= \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket^* (\mathcal{C} \llbracket c \rrbracket \sigma), \sigma \\ &= \mathbf{false} \rightarrow \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket^* (\mathcal{C} \llbracket c \rrbracket \sigma), \sigma \\ &= \sigma \end{aligned}$$



$$\frac{\langle b, \sigma \rangle \rightarrow \mathbf{true} \quad \langle c, \sigma \rangle \rightarrow \sigma'' \quad \langle \mathbf{while } b \mathbf{ do } c, \sigma'' \rangle \rightarrow \sigma'}{\langle \mathbf{while } b \mathbf{ do } c, \sigma \rangle \rightarrow \sigma'}$$

We assume

- $\langle b, \sigma \rangle \rightarrow \mathbf{true}$  and therefore  $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{true}$
- $P(\langle c, \sigma \rangle \rightarrow \sigma'') \stackrel{\text{def}}{=} \mathcal{C} \llbracket c \rrbracket \sigma = \sigma''$
- $P(\langle \mathbf{while } b \mathbf{ do } c, \sigma'' \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket \sigma'' = \sigma'$

We want to prove

$$P(\langle \mathbf{while } b \mathbf{ do } c, \sigma \rangle \rightarrow \sigma') \stackrel{\text{def}}{=} \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket \sigma = \sigma'$$

By the definition of the denotational semantics and the inductive hypotheses

$$\begin{aligned} \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket \sigma &= \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket^* (\mathcal{C} \llbracket c \rrbracket \sigma), \sigma \\ &= \mathbf{true} \rightarrow \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket^* \sigma'', \sigma \\ &= \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket^* \sigma'' \\ &= \mathcal{C} \llbracket \mathbf{while } b \mathbf{ do } c \rrbracket \sigma'' \\ &= \sigma' \end{aligned}$$

Note that the lifting operator can be removed since  $\sigma'' \neq \perp$ .

# Completeness

$$\forall c \in Com$$

$$P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma' \in \Sigma. \quad \mathcal{C} [c] \sigma = \sigma' \quad \Rightarrow \quad \langle c, \sigma \rangle \rightarrow \sigma'$$

by structural induction

We prove  $P(\mathbf{skip}) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket \mathbf{skip} \rrbracket \sigma = \sigma' \Rightarrow \langle \mathbf{skip}, \sigma \rangle \rightarrow \sigma'$

Assume  $\mathcal{C} \llbracket \mathbf{skip} \rrbracket \sigma = \sigma'$

Then  $\sigma' = \sigma$

By rule (skip)  $\langle \mathbf{skip}, \sigma \rangle \rightarrow \sigma = \sigma'$

We prove  $P(x := a) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket x := a \rrbracket \sigma = \sigma' \Rightarrow \langle x := a, \sigma \rangle \rightarrow \sigma'$

Assume  $\mathcal{C} \llbracket x := a \rrbracket \sigma = \sigma'$

Then  $\sigma' = \sigma[\mathcal{A} \llbracket a \rrbracket \sigma / x]$

By consistency for expressions  $\langle a, \sigma \rangle \rightarrow \mathcal{A} \llbracket a \rrbracket \sigma$

By rule (asgn)  $\langle x := a, \sigma \rangle \rightarrow \sigma[\mathcal{A} \llbracket a \rrbracket \sigma / x] = \sigma'$

Assume  $P(c_0) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathcal{C} \llbracket c_0 \rrbracket \sigma = \sigma'' \Rightarrow \langle c_0, \sigma \rangle \rightarrow \sigma''$   
 $P(c_1) \stackrel{\text{def}}{=} \forall \sigma'', \sigma'. \mathcal{C} \llbracket c_1 \rrbracket \sigma'' = \sigma' \Rightarrow \langle c_1, \sigma'' \rangle \rightarrow \sigma'$

We want to prove  $P(c_0; c_1) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket c_0; c_1 \rrbracket \sigma = \sigma' \Rightarrow \langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$

Assume  $\mathcal{C} \llbracket c_0; c_1 \rrbracket \sigma = \sigma'$

we have  $\mathcal{C} \llbracket c_0; c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_1 \rrbracket^* (\mathcal{C} \llbracket c_0 \rrbracket \sigma) = \sigma' \neq \perp$

thus  $\mathcal{C} \llbracket c_0 \rrbracket \sigma = \sigma''$  for some  $\sigma'' \neq \perp$

and  $\mathcal{C} \llbracket c_1 \rrbracket \sigma'' = \sigma'$

by inductive hypotheses  $\langle c_0, \sigma \rangle \rightarrow \sigma''$   $\langle c_1, \sigma'' \rangle \rightarrow \sigma'$

By rule (seq)  $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$

Assume

$$P(c_0) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket c_0 \rrbracket \sigma = \sigma' \Rightarrow \langle c_0, \sigma \rangle \rightarrow \sigma'$$

$$P(c_1) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma' \Rightarrow \langle c_1, \sigma \rangle \rightarrow \sigma'$$

We prove  $P(\text{if } b \text{ then } c_0 \text{ else } c_1) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \sigma' \Rightarrow \langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$

Assume  $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \sigma'$

we have  $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket c_0 \rrbracket \sigma, \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma'$

either  $\mathcal{B} \llbracket b \rrbracket \sigma = \text{false}$  or  $\mathcal{B} \llbracket b \rrbracket \sigma = \text{true}$ .

if  $\mathcal{B} \llbracket b \rrbracket \sigma = \text{false}$   $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma'$   
 $\langle b, \sigma \rangle \rightarrow \text{false}$  by inductive hypotheses  $\langle c_1, \sigma \rangle \rightarrow \sigma'$   
By rule (iff)  $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$

if  $\mathcal{B} \llbracket b \rrbracket \sigma = \text{true}$   $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_0 \rrbracket \sigma = \sigma'$   
 $\langle b, \sigma \rangle \rightarrow \text{true}$  by inductive hypotheses  $\langle c_0, \sigma \rangle \rightarrow \sigma'$   
By rule (iftt)  $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$

Assume  $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathcal{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$

We prove  $P(\text{while } b \text{ do } c) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} \llbracket \text{while } b \text{ do } c \rrbracket \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$

we have  $\mathcal{C} \llbracket \text{while } b \text{ do } c \rrbracket \sigma = \text{fix } \Gamma_{b,c} \sigma = \left( \bigsqcup_{n \in \mathbb{N}} \Gamma_{b,c}^n \perp \right) \sigma$

$\mathcal{C} \llbracket \text{while } b \text{ do } c \rrbracket \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$

iff  $\left( \bigsqcup_{n \in \mathbb{N}} \Gamma_{b,c}^n \perp \right) \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$

iff  $\left( \exists n \in \mathbb{N}. (\Gamma_{b,c}^n \perp) \sigma = \sigma' \right) \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$

iff  $\forall n \in \mathbb{N}. \left( \Gamma_{b,c}^n \perp \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma' \right)$

let  $A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \Gamma_{b,c}^n \perp \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$

we prove  $\forall n \in \mathbb{N}. A(n)$  by mathematical induction

Assume  $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathcal{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$

we prove  $\forall n \in \mathbb{N}. A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \Gamma_{b,c}^n \perp \sigma = \sigma' \Rightarrow \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma'$

$A(0) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \Gamma_{b,c}^0 \perp \sigma = \sigma' \Rightarrow \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma'$

$$\Gamma_{b,c}^0 \perp \sigma = \perp \sigma = \perp$$

the premise  $\Gamma_{b,c}^0 \perp \sigma = \sigma'$  is false  $\sigma' \neq \perp$

$A(0)$  is true



Assume  $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathcal{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$

we prove  $\forall n \in \mathbb{N}. A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \Gamma_{b,c}^n \perp \sigma = \sigma' \Rightarrow \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma'$

assume  $A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \Gamma_{b,c}^n \perp \sigma = \sigma' \Rightarrow \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma'$

we prove  $A(n+1) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \Gamma_{b,c}^{n+1} \perp \sigma = \sigma' \Rightarrow \langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma'$

assume  $\Gamma_{b,c}^{n+1} \perp \sigma = \Gamma_{b,c} \left( \Gamma_{b,c}^n \perp \right) \sigma = \sigma' \neq \perp$

by def  $\mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \left( \Gamma_{b,c}^n \perp \right)^* \left( \mathcal{C} \llbracket c \rrbracket \sigma \right), \sigma = \sigma'$

if  $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{false}$   $\langle b, \sigma \rangle \rightarrow \mathbf{false}$   $\sigma = \sigma'$

by rule (whff)  
 $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma = \sigma'$

if  $\mathcal{B} \llbracket b \rrbracket \sigma = \mathbf{true}$   $\langle b, \sigma \rangle \rightarrow \mathbf{true}$   $\left( \Gamma_{b,c}^n \perp \right)^* \left( \mathcal{C} \llbracket c \rrbracket \sigma \right) = \sigma' \neq \perp$

$\left( \Gamma_{b,c}^n \perp \right) \sigma'' = \sigma'$

$\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma'' \rangle \rightarrow \sigma'$

thus  $\mathcal{C} \llbracket c \rrbracket \sigma = \sigma''$  for some  $\sigma'' \neq \perp$   
 $\langle c, \sigma \rangle \rightarrow \sigma''$

By rule (whff)  
 $\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma'$

# Final remarks

## Commands

Big-step operational semantics

Denotational semantics

Termination 

(partial functions)

Determinacy 

Operational equivalence

Denotational equivalence  
is a congruence

Consistency  
(correctness + completeness)

Operational equivalence = Denotational equivalence  
they are congruences

Well-founded induction

Kleene's fixpoint theorem