

Consistenza e congruenza-6.3

Equivalenza operazionale

Equivalenza operazionale

$$
a_1 \sim_{\text{op}} a_2
$$
 sse $\forall \sigma, n. (\langle a_1, \sigma \rangle \to n \Leftrightarrow \langle a_2, \sigma \rangle \to n)$
\n $b_1 \sim_{\text{op}} b_2$ sse $\forall \sigma, v. (\langle b_1, \sigma \rangle \to v \Leftrightarrow \langle b_2, \sigma \rangle \to v)$
\n $c_1 \sim_{\text{op}} c_2$ sse $\forall \sigma, \sigma'. (\langle c_1, \sigma \rangle \to \sigma' \Leftrightarrow \langle c_2, \sigma \rangle \to \sigma')$

terminazione and determinismo non hanno importanza: l'equivalenza operazionale e' sempre ben definita

Congruenza

$$
a_1 \sim_{\text{op}} a_2
$$
 sse $\forall \sigma, n. (\langle a_1, \sigma \rangle \to n \Leftrightarrow \langle a_2, \sigma \rangle \to n)$

prendiamo un qls contesto A[*·*] **p.e.** $2 \times (|\cdot| + 5)$

e' vero che
$$
a_1 \sim_{\text{op}} a_2 \Rightarrow \mathbb{A}[a_1] \sim_{\text{op}} \mathbb{A}[a_2]
$$
 ?

ovvero: possiamo rimpiazzare una sottoespressione con una equivalente senza cambiare il risultato?

Contesti

quali sono i contesti possibili per le espressioni aritmetiche?

 $[\cdot]+5$ $2 \times ([\cdot] + 5)$ $2 \times ([\cdot] + 5) \leq 50$ $(2 \times (\lceil \cdot \rceil + 5) \le 50) \wedge x = y$ $x := 2 \times ([\cdot] + 5)$ while $x \le 100$ do $x := 2 \times (1 + 5)$

Contesti

quali sono i contesti possibili per le espressioni aritmetiche?

$A[\cdot]$::= []	$\begin{array}{c}\n A[\cdot] \text{ op } a \\ \mid \quad A[\cdot] \text{ op } a \\ \mid \quad a \text{ op } A[\cdot]\n \end{array}$ \n	$\mathbb{C}[\cdot]$::= $x := A[\cdot]$ \n
$\mathbb{C}[\cdot]; c$	$\mathbb{C}[\cdot]; c$	
$\mathbb{B}[\cdot]$ then c else c		
$\mathbb{B}[\cdot]$::= $A[\cdot]$ comp a	$\mathbb{I}^{\mathsf{f}} b$ then $\mathbb{C}[\cdot]$ else c	
$\mathbb{B}[\cdot]$ = $\mathbb{E}[\cdot]$	$\mathbb{I}^{\mathsf{f}} b$ then c else $\mathbb{C}[\cdot]$	
$\mathbb{B}[\cdot]$ hop b	$\mathbb{I}^{\mathsf{f}} b$ then c else $\mathbb{C}[\cdot]$	
$\mathbb{B}[\cdot]$ hop b	$\mathbb{I}^{\mathsf{f}} b$ then c else $\mathbb{C}[\cdot]$	
$\mathbb{B}[\cdot]$ hop b	$\mathbb{I}^{\mathsf{f}} b$ then c else $\mathbb{I}^{\mathsf{f}} \mathsf{f} b$ do $\mathbb{I}^{\mathsf{f}} \mathsf{f} b$	

Proof obligation

dobbiamo trattare molte proof obligation:

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

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

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

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

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

la stessa cosa per espressioni booleane e comandi

Equivalenza denotazionale

Equivalenza denotazionale

- $a_1 \sim_{\text{den}} a_2$ sse $\mathcal{A}[[a_1]] = \mathcal{A}[[a_2]]$
- $b_1 \sim_{\text{den}} b_2$ sse $\mathcal{B}[[b_1]] = \mathcal{B}[[b_2]]$
- $c_1 \sim_{\text{den}} c_2$ sse $\mathcal{C}[[c_1]] = \mathcal{C}[[c_2]]$

(due funzioni sono la stessa se coincidono su tutti gli argomenti)

Principio di Composizionalita'

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

prendiamo un qls contesto A[*·*]

 e' vero che $a_1 \sim_{\text{den}} a_2 \Rightarrow \mathbb{A}[a_1] \sim_{\text{den}} \mathbb{A}[a_2]$?

SI, è garantito dal principio di composizionalita' della semantica denotazionale:

il significato di un'espressione composta è unicamente determinato dal significato dei suoi costituenti

Consistenza

se garantiamo la coerenza tra la semantica operazionale e la semantica denotazionale allora la proprietà di congruenza è garantita anche per la semantica operazionale

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

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

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

Consistenza: espressioni regard the operational semantics as an interpreter and the denotational semantics CONSISTENZA: ESDPES compiled version starting from the same memory leads to the same memory leads to the same result. The same res
The same result is the same result of the same result in the same result in the same result. The same result i ^h*a*⁰ *a*1*,*sⁱ ! *^A* ^J*a*0K^s *· ^A* ^J*a*1K^s Finally, by definition of *A* ^h*a*⁰ *a*1*,*sⁱ ! *^A* ^J*a*0K^s *· ^A* ^J*a*1K^s Finally, by definition of *A* 142 6 Denotational Semantics of IMP Finally, by definition of *A ^A* ^J*a*0K^s *· ^A* ^J*a*1K^s ⁼ *^A* ^J*a*⁰ *a*1K^s

$$
\forall a \in A \exp \forall \sigma \in \Sigma. \ \langle a, \sigma \rangle \to \mathscr{A}[[a]] \sigma
$$

$$
P(a) \stackrel{\text{def}}{=} \forall \sigma \in \Sigma. \ \langle a, \sigma \rangle \to \mathscr{A}[[a]] \sigma
$$

per induzione strutturale per induzione strutturale

per induzione strutturale

$$
\forall b \in Bexp \ \forall \sigma \in \Sigma. \ \langle b, \sigma \rangle \to \mathscr{B}[[b]] \sigma
$$

$$
P(b) \stackrel{\text{def}}{=} \forall \sigma \in \Sigma. \ \langle b, \sigma \rangle \to \mathscr{B}[[b]] \sigma
$$
per induzione strutturale

From now on we will assume the equivalence between denotational and opera-

From now on we will assume the equivalence between denotational and opera-

Consistenza: comandi \mathcal{L} allows us to calculate the execution of \mathcal{L} CONSISTENZA. CONTANTAL

 $\forall c \in Com. \ \forall \sigma, \sigma' \in \Sigma$. $\langle c, \sigma \rangle \rightarrow \sigma' \Leftrightarrow \mathscr{C}[[c]] \sigma = \sigma'$ $v_{\rm eff}$ different formalisms: on the one hand we have an inference rule system which we have an inference rule system which we have a system ϵ $\forall C \in \mathcal{C} \text{ on. } \forall \sigma, \sigma' \in \mathcal{\Sigma}.$ $\langle C, \sigma \rangle \rightarrow \sigma' \Leftrightarrow C \in \mathcal{C}$ $v_{\rm eff}$ different formalisms: on the one hand we have an inference rule system which we have an inference rule system which we have a system ϵ $\forall C \in \mathcal{C} \text{ on. } \forall \sigma, \sigma' \in \mathcal{\Sigma}.$ $\langle C, \sigma \rangle \rightarrow \sigma' \Leftrightarrow C \in \mathcal{C}$ $v_{\rm eff}$ different formalisms: on the one hand we have an inference rule system which we have an inference rule system which we have a system ϵ $\forall C \in \mathcal{C} \text{ on. } \forall \sigma, \sigma' \in \mathcal{\Sigma}.$ $\langle C, \sigma \rangle \rightarrow \sigma' \Leftrightarrow C \in \mathcal{C}$

function which associates with each command its functional meaning. So to show

6.3 Equivalence Between Operational and Denotational Semantics 143

function which associates with each command its functional meaning. So to show

possiamo scriverio come possiamo scriverlo come entrant formalisms: one hand we have an inference rule system which we have an inference rule system which we have a system which the equivalence between the two semantics we will prove the following property. $\forall c \in Com. \ \forall \sigma \in \Sigma. \quad \langle c, \sigma \rangle \rightarrow \mathscr{C}[[c]] \sigma$? the equivalence between the two semantics we will prove the following property. the equivalence between the two semantics we will prove the following property.

⁼ *^C* ^J*c*K^s ⁼ ^s⁰ Completeness: 8*c* 2 *Com* we prove $\langle c, \sigma \rangle \rightarrow$ $\langle C \cap \mathcal{L} \rangle$ is a sequence of $\langle C \cap \mathcal{L} \rangle$ Correctness: 8*c* 2 *Com,* 8s*,*s⁰ 2 S we prove no, non c'e' una formula del tipo no pope's' upo formulo dol tipo $\langle c,\sigma\rangle\rightarrow\bot$ \perp

 A usual we divide the proof into two parts: A usual we divide the proof into two parts: A usual we divide two parts: A

function which associates with each command its functional meaning. So to show

 A usual we divide the proof into two parts: A usual we divide the proof into two parts: A usual we divide two parts: A

P(*c*) def

As usual we divide the proof into two parts:

Consistenza: comandi the equivalence between the two semantics we will prove the following property. allows us to calculate the execution of each command; on the other hand we have a foncicton za: comandi the equivalence between the two semantics we will prove the following property. \mathcal{L} allows us to calculate the execution of \mathcal{L} CONSISTENZU. CONTUNIUI very different formalisms: on the one hand we have an inference rule system which allows us to calculate the execution of each command; on the other hand we have a

 $H_0 \subset C$ om $H_0 \subset T' \subset \Gamma$ $I_0 \subset T'$ $\subset T'$ $\overline{}$ $\forall c \in Com. \ \forall \sigma, \sigma' \in \Sigma$. $\langle c, \sigma \rangle \rightarrow \sigma' \iff \mathscr{C}[[c]] \sigma$

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

vcccom, vo, o $\begin{array}{ccc} \begin{array}{ccc} \begin{array}{ccc} \end{array} & \begin{array}{ccc} \end{array} & \end{array} & \begin{array}{ccc} \end{array} & \begin{array}{$ $\forall c \in Com. \ \forall \sigma, \sigma' \in \Sigma$. $\forall c \in \mathcal{C}$ and $\forall \sigma \sigma' \in \Sigma$ allows us to calculate the execution of each community of each community of each community of each community of \mathcal{C}

 $P(\langle c,\sigma\rangle\rightarrow\sigma')$ $\frac{\text{def}}{\sqrt{2\pi}}$ σ' = \mathscr{C} $[c]$ $\sigma = \sigma'$ **Correttezza** *Correttezza* $\dot{ }$ = $\dot{ }$ $= 0$ *sulle regole* per induzione **Correttezza**

Both Barrel we divide the proof into the p) def ⁼ *^C* ^J*c*K^s ⁼ ^s⁰ Correctness: 8*c* 2 *Com,* 8s*,*s⁰ 2 S we prove function which associates with each communication with each communication of the so to show with $P(\langle c,\sigma\rangle\to\sigma')\stackrel{\text{def}}{=} \mathscr{C}\llbracket c\rrbracket\,\sigma=\sigma'$ will property.

 $\forall c \in C$

 $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma' \in \Sigma. \quad \mathscr{C}\llbracket c \rrbracket \, \sigma = \sigma' \quad \Rightarrow$ Completezza vcccom.
Comp $\mathcal{L} = \forall \sigma, \sigma' \in \Sigma$. $\mathscr{C}[[c]] \sigma = \sigma' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma'$ $P(c) = \nabla \sigma, \sigma \in \mathcal{L}.$ ⁼ ⁸s*,*s⁰ ² ^S*. ^C* ^J*c*K^s ⁼ ^s⁰) ^h*c,*sⁱ ! ^s⁰ $\forall c \in Com.$ **Completezza**
As two parts: two part
See two parts: two p Correctness: 8*c* 2 *Com,* 8s*,*s⁰ 2 S we prove

per induzione strutturale cases are also handled for the unit per in the under structure cases are also handled for the equivalence: for *P*(*c*) def per induzione strutturale

Correttezza As usual we divide the proof into two parts:

$\forall c \in Com$, $P(\langle c,\sigma\rangle\rightarrow\sigma')$ $\frac{\text{def}}{\sqrt{2\pi}}$ $\stackrel{\text{def}}{=} \mathscr{C}\llbracket c \rrbracket \sigma = \sigma'$ $\forall c \in Com, \ \forall \sigma, \sigma' \in \Sigma$ $\lbrack c \rbrack$ $\lbrack c - c \rbrack$

Complete Supersy: 8 Provence in the provence in the second series in the second series in the series of the per induzione sulle regole

$$
\overline{\langle skip, \sigma\rangle \rightarrow \sigma}
$$

Wooliamo provare Vogliamo provare

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

Ovviamente la preposizione e' vera per definizione della semantica denotazionale

N.B. Possiamo assumere solo che la semantica operazionale delle espressioni aritmetiche mi dia m: non abbiamo nessuna ipotesi induttiva sulle espressioni aritmetiche!

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

Assumiamo $\langle a, \sigma \rangle \to m$ e quindi $\mathscr{A}\llbracket a \rrbracket \sigma = m$ per equivalenza della semantica operazionale e denotazionale delle espressioni aritmetiche. Vogliamo provare che

$$
P(\langle x := a, \sigma \rangle \to \sigma[^m/_{x}]) \stackrel{\text{def}}{=} \mathscr{C}[[x := a]] \sigma = \sigma[^m/_{x}]
$$

Per definizione della semantica denotazionale abbiamo che

$$
\mathscr{C}\llbracket x := a \rrbracket \sigma = \sigma \llbracket \mathscr{A}\llbracket a \rrbracket \sigma /_{x} = \sigma \llbracket \mathscr{A} \llbracket a \rrbracket
$$

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

Assumiamo

$$
P(\langle c_0, \sigma \rangle \to \sigma'') \stackrel{\text{def}}{=} \mathscr{C}[[c_0]] \sigma = \sigma''
$$

$$
P(\langle c_1, \sigma'' \rangle \to \sigma') \stackrel{\text{def}}{=} \mathscr{C}[[c_1]] \sigma'' = \sigma'
$$

Vogliamo provare

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

Per la definizione di semantica denotazionale e per ipotesi induttiva

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

Notare che l'operatore di lifting puo' essere rimosso perche'

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

Assumiamo

• $\langle b, \sigma \rangle \rightarrow$ **true** e percio' $\mathscr{B}\llbracket b \rrbracket \sigma =$ **true** per la corrispondenza e percio' \mathscr{B} $\llbracket b \rrbracket$ $\sigma = \textbf{true}$ per la corrispondenza

between the operational and denotational semantics for books for books for books for books for books for books Senianuca tra semantica denotazionale e operazionale per le espressioni booleane

 \bullet $P(\langle c_0, \sigma \rangle \to \sigma')$ $\bullet \quad P(\langle c_0, \sigma \rangle \to \sigma') \stackrel{\text{def}}{=} \mathscr{C}\llbracket c_0 \rrbracket \, \sigma = \sigma'$

vogliamo provare

$$
P(\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \to \sigma') \stackrel{\text{def}}{=} \mathscr{C}[\text{if } b \text{ then } c_0 \text{ else } c_1] \sigma = \sigma'
$$

infatti abbiamo

$$
\mathscr{C}[\![\text{if } b \text{ then } c_0 \text{ else } c_1]\!] \sigma = \mathscr{B}[\![b]\!] \sigma \to \mathscr{C}[\![c_0]\!] \sigma, \mathscr{C}[\![c_1]\!] \sigma
$$

$$
= \text{true} \to \sigma', \mathscr{C}[\![c_1]\!] \sigma
$$

$$
= \sigma'
$$

$$
P(\langle \text{while } b \text{ do } c, \sigma \rangle \to \sigma) \stackrel{\text{def}}{=} \mathscr{C}[\text{while } b \text{ do } c] \sigma = \sigma
$$

Per la proprieta' della semantica denotazionale

$$
\mathscr{C}[\![\text{while } b \text{ do } c]\!] \sigma = \mathscr{B}[\![b]\!] \sigma \to \mathscr{C}[\![\text{while } b \text{ do } c]\!]^* (\mathscr{C}[\![c]\!] \sigma), \sigma
$$

$$
= \text{false} \to \mathscr{C}[\![\text{while } b \text{ do } c]\!]^* (\mathscr{C}[\![c]\!] \sigma), \sigma
$$

$$
= \sigma
$$

 $\langle b, \sigma \rangle \rightarrow \textbf{true}$ $\langle c, \sigma \rangle \rightarrow \sigma''$ $\langle \textbf{while } b \textbf{ do } c, \sigma'' \rangle \rightarrow \sigma''$ \langle while *b* do $c, \sigma \rangle \rightarrow \sigma'$

Assumiamo

• $\langle b, \sigma \rangle \rightarrow$ true e percio' $\mathscr{B}\llbracket b \rrbracket \sigma =$ true

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

• $P(\langle \textbf{while } b \textbf{ do } c, \sigma'' \rangle \rightarrow \sigma'$ $\frac{\text{def}}{\sqrt{2\pi}}$ $=$ *C* [while *b* do *c*] $\sigma'' = \sigma'$

 $\frac{1}{2}$ Vogliamo provare

$$
P(\langle \text{while } b \text{ do } c, \sigma \rangle \to \sigma') \stackrel{\text{def}}{=} \mathscr{C}[\text{while } b \text{ do } c] \sigma = \sigma'
$$

$$
\mathscr{C}[\text{while } b \text{ do } c] \sigma = \mathscr{B}[b] \sigma \rightarrow \mathscr{C}[\text{while } b \text{ do } c]^{*} (\mathscr{C}[c] \sigma), \sigma
$$

= true $\rightarrow \mathscr{C}[\text{while } b \text{ do } c]^{*} \sigma'', \sigma$
= $\mathscr{C}[\text{while } b \text{ do } c]^{*} \sigma''$
= $\mathscr{C}[\text{while } b \text{ do } c] \sigma''$
= σ'

L'operatore di lifting puo' essere rimosso $\sigma'' \neq \bot.$

per inquzione strutturale per induzione strutturale

 $P(\textbf{skip}) \overset{\text{def}}{=}$ \textsf{V} ogliamo provare $\;P(\textbf{skip}) \overset{\text{def}}{=} \forall \sigma, \sigma'.\; \mathscr{C}\left[\textbf{skip} \right]\sigma = \sigma' \Rightarrow \langle \textbf{skip}, \sigma \rangle \to \sigma'$ Vogliamo provare

 $\textsf{Assumiamo} \quad \mathscr{C}\llbracket \textsf{skip} \rrbracket \, \sigma = \sigma'$ $\textsf{gamma} \quad \mathscr{C}\llbracket \textsf{skip} \rrbracket \, \sigma = \sigma'$ (*^C* ^J*c*Ks)*,*^s ⁼ ^s⁰ $\overline{\mathsf{A}}$ $\overline{\mathsf{A}}$ \mathbf{A}

 $\overline{}$ $\sigma = \sigma$ Allora $\sigma' = \sigma$ $= 8.8$ $\sigma' = \sigma$ $\prime = \sigma$

per la regola (skip) per la regola (skip) \langle skip, $\sigma \rangle \rightarrow \sigma = \sigma'$

Provi assign: We need to prove *P*(*x* := *a*) def $\frac{p}{p}$ $\frac{p}{p}$ $\frac{p}{p}$ $\frac{p}{p}$ $\frac{p}{p}$ *. ^C* ^J*^x* :⁼ *^a*K^s ⁼ ^s⁰) ^h*^x* :⁼ *^a,*sⁱ ! ^s⁰ Proviamo $P(x := a) \stackrel{\text{def}}{=}$ \mathcal{L}^{ucl} $\forall \sigma, \sigma'. \mathscr{C}$ $\llbracket x := a \rrbracket \sigma = \sigma' \Rightarrow \langle x := a, \sigma \rangle \rightarrow \sigma'$

Assumiamo
$$
\mathscr{C}[x := a] \sigma = \sigma'
$$

 λ λ λ λ λ λ λ λ

Allora
$$
\sigma' = \sigma[^{\mathscr{A}}[a]\sigma/_{x}]
$$

 $\mathsf{D}\mathsf{a}\mathsf{r}$ concidence between operational and denote between $\mathsf{A}\mathsf{A}\mathsf{A}$ expressions we have the copicosion $\langle x, \circ \rangle$, \sim rule \mathbb{P}^1 $\overline{}$ by the equivalence between operational and denotational semantics for expressions we have ^h*a,*sⁱ ! *^A* ^J*a*Ks, thus we can apply the rule (assign) Per consistenza delle espressioni $\langle a, \sigma \rangle \rightarrow \mathscr{A} \llbracket a \rrbracket \sigma$ \mathbf{t} $\left\langle \mathcal{L} \right| \mathcal{L} \left| \mathcal{L} \right| \leq \sigma'$ Per la regola (asgn) \mathcal{L}_c to conclude the concluded state \mathcal{L}_c $\langle x := a, \sigma \rangle \rightarrow \sigma[^{\mathscr{A}}[a]\sigma/_{x}] = \sigma'$ Fer consistenza delle espressioni $\langle a, \sigma \rangle \rightarrow \mathscr{A} \llbracket a \rrbracket \sigma$ ancictonz \mathcal{P} se la regula (e

Assumiamo
$$
P(c_0) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathcal{C} [c_0] \sigma = \sigma'' \Rightarrow \langle c_0, \sigma \rangle \to \sigma''
$$

Assumiamo
$$
P(c_1) \stackrel{\text{def}}{=} \forall \sigma'', \sigma'. \mathcal{C} [c_1] \sigma'' = \sigma' \Rightarrow \langle c_1, \sigma'' \rangle \to \sigma'
$$

 \mathbf{R} \mathbf{R} \mathbf{R} Vogliamo provare $P(c_0;c_1) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathcal{C} [c_0;c_1] \sigma = \sigma' \Rightarrow \langle c_0;c_1, \sigma \rangle \rightarrow \sigma'$ C^1 C^1 C^1 C^1 C^1 C^1 C^1 C^1 Since ^s⁰ ⁶⁼ ?, it must be that *^C* ^J*c*0K^s ⁶⁼ ?, i.e., we can assume the *C C*₀, *C*₁ D **C** D **C** α *C*₀, *C*₁ D **C** α *C*₀, *C*₁, α \rightarrow O *C*₁, α _{*/*} \rightarrow O *C* termination of *c*⁰ and thus omit the lifting operator:

Assumiamo $\mathscr{C}[\![c_0;c_1]\!]\sigma=\sigma'$ the conclusion h*c*0; *c*1*,*si ! s⁰ Since ^s⁰ ⁶⁼ ?, it must be that *^C* ^J*c*0K^s ⁶⁼ ?, i.e., we can assume the σ' $-\sigma'$ $\text{Asquilibrium} \quad \text{of} \quad \text{[C0, C1]} \quad \text{O} \quad \text{O}$ *^C* ^J*c*0; *^c*1K^s ⁼ *^C* ^J*c*1^K ⇤ (*^C* ^J*c*0Ks) = ^s⁰ $-\sigma'$

6.3 Equivalence Between Operational and Denotational Semantics 147

 $\mathbf{O} \ \mathscr{C} \llbracket c_0; c_1 \rrbracket \, \sigma = \mathscr{C} \llbracket c_1 \rrbracket^* \left(\mathscr{C} \llbracket c_0 \rrbracket \, \sigma \right) = \sigma' \! \neq \bot$ Abbiamo $\mathscr{C}[[c_0;c_1]]\sigma = \mathscr{C}[[c_1]]^*(\mathscr{C}[[c_0]]\sigma) = \sigma' \neq \bot$ $\sigma' \neq \bot$ Δ hhinmo $\mathscr{L}[c_0; c_1]$ $\sigma = \mathscr{L}[c_1]$ Since $\frac{d}{dx}$ (1949)
 $\frac{d}{dx}$ (2005)
 $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ $\frac{d}{dx}$ Abbiamo \mathscr{C} $[c_0; c_1]$ c termination of *c*⁰ and thus omit the lifting operator: \mathscr{C} $\llbracket c_0 : c_1 \rrbracket$ $\sigma = \mathscr{C} \llbracket c_1 \rrbracket^* (\mathscr{C} \llbracket c_0 \rrbracket \sigma) = \sigma' \neq 0$ \mathscr{C} [*c*₀; *c*₁] $\sigma = \mathscr{C}$ [*c*₁], $(\mathscr{C}$ [*c*₀] $\sigma) = \sigma \neq \bot$

 marain' $\mathscr{L}\llbracket c_0 \rrbracket \boldsymbol{\sigma} - \boldsymbol{\sigma}''$ perccio \mathscr{C} $\llbracket c_0 \rrbracket$ $\sigma = \sigma'$ \quad per qualche $\sigma'' \neq \bot$ $tanh \alpha$ τ'' \pm 1 percio' $\mathscr{C}[\![c_0]\!]\sigma=\sigma''$ per qualche $\sigma''\neq\bot$ $\textsf{percio'} \qquad \mathscr{C}\llbracket c_0 \rrbracket\, \sigma = \sigma'' \qquad \textsf{per c}$ β *C C*₀ $\sigma = \sigma$ ^{*n*} $\mathscr{C}\llbracket c_0\rrbracket\, \sigma = \sigma''$ per qualche $\sigma'' \neq \bot$

$$
e \quad \mathscr{C}\llbracket c_1 \rrbracket \, \sigma'' = \sigma'
$$

 $\langle c_0, \sigma \rangle \rightarrow \sigma''$ $\langle c_0, \sigma \rangle \rightarrow \sigma'' \quad \langle c_1, \sigma'' \rangle \rightarrow \sigma'$ and h*c*1*,*s00i ! s⁰ $\rightarrow \sigma'$ ponens to the inductive and $\langle c_0, \sigma \rangle \rightarrow \sigma''$ $\langle c_1, \sigma \rangle$ and h*c*1*,*s00i ! s⁰ J ules muuttiva $\langle v_0, v_1 \rangle$ ρ er ipotesi induttiva $\langle c_0, \sigma \rangle \rightarrow \sigma''$ $\langle c_1, \sigma'' \rangle \rightarrow \sigma''$ Let *^C* ^J*c*0K^s ⁼ ^s00. We have *^C* ^J*c*1Ks⁰⁰ ponens to the inductive assumptions *P*(*c*0) and *P*(*c*1), to get h*c*0*,*si ! s⁰⁰

Per la regola (seq) $\langle c_0; c_1, \sigma \rangle \rightarrow$ σ' ponens to the inductive assumptions *P*(*c*0) and *P*(*c*1), to get h*c*0*,*si ! s⁰⁰ p' and h*c*1*,*s00i ! s⁰ ϕ **r** la regola (seq) $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$ σ' . Thus we can apply the inference rule: $\mathcal{L} = \mathcal{L} \mathcal{L}$ the inference rule: $\mathcal{L} = \mathcal{L} \mathcal{L}$ ^h*c*0*,*sⁱ ! ^s⁰⁰ ⌦ **c**₁, **c**₁, **c**₁ Per la regola (seq) $\langle c_0; c_1, \sigma \rangle \rightarrow \sigma'$

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

\n $P(c_1) \stackrel{\text{def}}{=} \forall \sigma, \sigma', \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma' \Rightarrow \langle c_1, \sigma \rangle \rightarrow \sigma'$
\nproviamo $P(\text{if } b \text{ then } c_0 \text{ else } c_1) \stackrel{\text{def}}{=} \forall \sigma, \sigma', \mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \sigma'$
\n $\Rightarrow \langle \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \sigma'$
\nabsumiamo $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \sigma'$
\nabslimo $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket c_0 \rrbracket \sigma, \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma'$
\ne $\mathcal{B} \llbracket b \rrbracket \sigma = \text{false}$ $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma'$
\n $\langle b, \sigma \rangle \rightarrow \text{false}$ $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_1 \rrbracket \sigma = \sigma'$
\nPer la regola (ifff) $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$
\nSee $\mathcal{B} \llbracket b \rrbracket \sigma = \text{true}$ $\mathcal{C} \llbracket \text{if } b \text{ then } c_0 \text{ else } c_1 \rrbracket \sigma = \mathcal{C} \llbracket c_0 \rrbracket \sigma =$

 $\mathscr{B}[[b]]$ $\sigma = \textbf{true}.$ $\mathscr{C}[[\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]]$ $\sigma = \mathscr{C}[[c_0]]$ $\sigma = \sigma'$ *C*₀, $\langle c_0, c_1 \rangle$ = $\langle c_0, c_2 \rangle$ $B = \text{gcd}(B \cup B)$ is the inductive hypothesis inductive $C_1, O \rightarrow O$ $\langle b, \sigma \rangle \rightarrow \textbf{true}$ per ip **then** co else $c_1, \sigma \rangle \rightarrow \sigma'$ $\mathcal{L}[\mathbf{c} \mathbf{r}] = \mathbf{c} \mathbf{r} \mathbf{u}$ $\mathcal{L}_{\mathcal{A}}$ we can apply the rule the $\boldsymbol{\varepsilon}$ id regola (fill) (if *b* then *c*₀ else *c*₁, **o**_/ \rightarrow $\mathscr{B}[[b]]$ $\sigma =$ true. \rightarrow **true peripotesi induttiva** (iftt) \langle if *b* then c_0 else c_1 tt) \langle if b then c_0 else $c_1,\sigma \rangle$ ponens in the induction c_0 else c_1 $\sigma = \mathscr{C}$ $[c_0]$ $\sigma = \sigma'$ and the south of the south \overline{a} . Thus we can apply the inference rule: **h**_c $c_1, \sigma \rangle \rightarrow \sigma'$ se $\mathscr{B}[b]$ $\sigma =$ true. $\mathscr{C}[if \; b \; then \; c_0 \; else \; c_1 \;]$ $\sigma = \mathscr{C}[c_0]$ σ λ (iftt) \langle if b then c_0 else $c_1,$ ϕ *c*₀, σ $\left\{ \text{m} \right\}$ $\left\{ \text{n} \right\}$ $\left\{ \text{n} \right\}$ $\left\{ \text{n} \right\}$ $\left\{ \text{n} \right\}$ $\mathbf{u} \mathbf{e}$ **example 1** \mathscr{C} if *b* then c_0 else c_1 **hig idea by the** *c* electron per ipotesi induttiva **Per la regola (Itt)** \langle **if** b **then** c_0 **else** $\mathscr{B} [b] \sigma = \text{true}.$ *P*(*c*) def $\lim c_0 \text{ else } c_1, \sigma \rangle \rightarrow \sigma'$ se Per la regola (iftt) \langle if *b* then c_0 else $c_1, \sigma \rangle \rightarrow \sigma'$ \mathscr{C} [if *b* then c_0 else $c_1 \cdot \cdot \cdot$ $\sigma = \mathscr{C}$ [$c_0 \cdot \cdot \cdot \cdot$ $\sigma = \sigma'$ $\begin{array}{ccc} \text{if} & \text{$ *Phen* c_0 else c_1] $\sigma = \mathscr{C}$ [c_0] $\sigma = \sigma'$ per ipotesi induttiva $\langle c_0, \sigma \rangle \rightarrow \sigma'$ $\mathbb{I} \sigma = \text{true}$ $\mathscr{C} \mathbb{I}$ $\langle b, \sigma \rangle \rightarrow$ **true per ipotesi induttiva** ϕ *b* then *contention* ϕ *c*₀, **c** By modus ponens on the inductive hypothesis *P*(*c*1) we have h*c*1*,*si ! s⁰ . **b Per la regola** (IIII) (II *D* f *b* then c_0 else c_1, σ

The case where *^B* ^J*b*K^s ⁼ true is completely analogous and thus omitted.

The case where *^B* ^J*b*K^s ⁼ true is completely analogous and thus omitted.

Assumiamo $P(c) \stackrel{\text{def}}{=}$ $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathscr{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$ Λ oou miqma Λ *P*(while *b* do *c*) def m iamo $P(c)$ $\mathcal{L}(C) = \nabla \sigma, \sigma \cdot \mathcal{L}$ [*C*] $\sigma = \sigma \Rightarrow \langle C, C \rangle$ $\frac{1}{2}$ **b** $P(x)$ de $\stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathscr{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle$) hwhile *b* do *c,*si ! s⁰ *. ^C* ^Jwhile *^b* do *^c*K^s ⁼ ^s⁰ $\Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$ P *b* ssumiamo define P $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathscr{C}$ [c] $\sigma = \sigma'' \Rightarrow$

= 8s*,*s⁰

Dimostriamo $P(\textbf{while } b \textbf{ do } c) \stackrel{\text{def}}{=}$ $\mathcal{L}^{\text{ref}} = \forall \sigma, \sigma'. \mathscr{C}$ while *b* do $c \mathbb{J} \sigma = \sigma'$ \Rightarrow \langle while *b* do *c*, σ $\rangle \rightarrow \sigma'$ *mostriamo P*(while ϕ do *c*) $\stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathscr{C}$ while *b* do *P*(while *b*) *b* do *c*) $\stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathscr{C}$ while *b* do *c* $\mathbf{d} \mathbf{a}$ c) $\stackrel{\text{def}}{=} \forall \sigma \; \sigma' \; \mathscr{C}$ while *b* do $c \mathbb{I}$ ∂ *b* do *c* \int $\sigma = \sigma'$ \int ile *b* do $c \cdot \sigma$ σ = σ ' By definition *^C* ^Jwhile *^b* do *^c*K^s ⁼ fix ^G*b,^c* ^s ⁼ \rightarrow **w** e \sim \sim By definition *^C* ^Jwhile *^b* do *^c*K^s ⁼ fix ^G*b,^c* ^s ⁼ triamo $P(\textbf{while } b \textbf{ do } c) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \mathscr{C}[\textbf{while } b \textbf{ do } c]$ \Rightarrow \langle while *i* \mathbf{z} $\overline{\textbf{do}}\ \overline{c}$ \overline{c} $\sigma = \sigma'$) hwhile *b* do *c,*si ! s⁰ *n* ⇒ \langle $\bf W$ \mathbf{n} ile l

$$
\text{abbiano} \qquad \mathscr{C} \llbracket \text{while } b \text{ do } c \rrbracket \sigma = \text{fix } \Gamma_{b,c} \sigma = \left(\sqcup_{n \in \mathbb{N}} \Gamma_{b,c}^n \bot \right) \sigma
$$

$$
\mathscr{C}[\![\text{while }b\text{ do }c]\!] \sigma = \sigma' \Rightarrow \langle \text{while }b\text{ do }c,\sigma\rangle \rightarrow \sigma'
$$

\n
$$
\text{SSe} \quad\n \left(\bigcup_{n \in \mathbb{N}} \Gamma_{b,c}^n \perp \right) \sigma = \sigma' \Rightarrow\n \langle \text{while } b \text{ do } c, \sigma \rangle \to \sigma'
$$
\n

\n\n $\text{Sse} \quad\n \left(\bigcup_{n \in \mathbb{N}} \Gamma_{b,c}^n \perp \right) \sigma = \sigma' \Rightarrow\n \langle \text{while } b \text{ do } c, \sigma \rangle \to \sigma'$ \n

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

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

$$
\text{definiamo } A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma'. \ \Gamma^n_{b,c} \bot \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'
$$

 $V/I \subseteq I\mathcal{A}$. A *sh* $C = 19.21(n)$ por madricile matematica Let *A*(*n*) def vialilo _V proviamo $\forall n$ $\nexists n \in \mathbb{N}$. $A(n)$ per induzione matematica Let *A*(*n*) def $\forall n \in \mathbb{N}$ $\Lambda(n)$ har induziona matamatica wand $\forall n \in \mathbb{N}$. $A(n)$ por mudzione matematical views $\forall n \in \mathbb{N}$. $A(n)$ per induzione matematica We prove that 8*n* 2 N*. A*(*n*) by mathematical induction. proviamo $\forall n \in \mathbb{N}$. $A(n)$ per induzione matematica

σ'' . C $\parallel \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$ Let *A*(*n*) def α , α Let *A*(*n*) def $\mathbf{S}\mathbf{S}\mathbf{u}\mathbf{m}$ $\mathbf{i}\mathbf{a}\mathbf{m}\mathbf{o}$ $P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''.\ \mathscr{C}$ $\llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c \rangle$ We prove that 8*n* 2 N*. A*(*n*) by mathematical induction. ASSUMI<mark>a</mark> $\lim_{x \to a} \frac{f(x)}{g(x)}$. $\lim_{x \to a} \frac{f(x)}{g(x)}$. $\lim_{x \to a} \frac{f(x)}{g(x)}$ \overline{P} ⇣ $\sigma \stackrel{{\mathrm{def}}}{=} \forall \sigma, \sigma''.\ \mathscr{C} \llbracket c \rrbracket\, \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \to \sigma'$ \mathbf{a} 8s*,*s⁰ *^b,c*?s = s⁰) hwhile *b* do *c,*si ! s⁰ $\frac{1}{2}$ Let *A*(*n*) def $\mathbb{E}[\mathbf{E} \mathbf{B} \mathbf{B} \mathbf{B}]$ Assumiamo $P(c) \stackrel{\text{def}}{=}$ $\mathcal{L}^{\text{ucl}} = \forall \sigma, \sigma''. \mathscr{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''$ $sumia$ ⇣ $\mathbf{O} \quad P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma''. \mathscr{C} \llbracket c \rrbracket \sigma = \sigma'' \Rightarrow \langle c \rangle$ $P(c) \stackrel{\text{def}}{=} \forall \sigma \; \sigma'' \; \mathscr{C} \mathbb{L} \mathbb{I} \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle$

$$
proviamo √n ∈ ℕ. A(n) \stackrel{\text{def}}{=} ∀σ, σ'. Γb,cn ⊥ σ = σ' ⇒ \langle while b do c, σ \rangle → σ'
$$

^b,c?s = s⁰) hwhile *b* do *c,*si ! s⁰

^b,c?s = s⁰) hwhile *b* do *c,*si ! s⁰

^b,c?s = s⁰) hwhile *b* do *c,*si ! s⁰

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

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

la premessa $\Gamma_{b,c}^{0} \bot \sigma = \sigma'$ e' falsa $\sigma' \neq \bot$
A(0) e' vero

8s*,*s⁰

We prove that 8*n* 2 N*. A*(*n*) by mathematical induction.

Base case: We have to prove *A*(0), namely

8s*,*s⁰

8s*,*s⁰

Assumiamo
$$
P(c) \stackrel{\text{def}}{=} \forall \sigma, \sigma'', \mathscr{C} \parallel c \parallel \sigma = \sigma'' \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma''
$$

\nproviamo $\forall n \in \mathbb{N}. A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma', \frac{P^n_{b,c} \perp \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'}$
\nassumiamo $A(n) \stackrel{\text{def}}{=} \forall \sigma, \sigma', \frac{P^n_{b,c} \perp \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$
\nproviamo $A(n+1) \stackrel{\text{def}}{=} \forall \sigma, \sigma', \frac{P^{n+1}_{b,c} \perp \sigma = \sigma' \Rightarrow \langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$
\nassumiamo $\frac{P^{n+1}_{b,c} \perp \sigma = \Gamma_{b,c}}{\Gamma_{b,c}^n \perp} \left(\frac{P^n_{b,c} \perp}{P^n_{b,c} \perp}\right) \sigma = \sigma' \neq \perp$
\nby def $\mathscr{B} \llbracket b \rrbracket \sigma \rightarrow (\Gamma^n_{b,c} \perp)^* (\mathscr{C} \llbracket c \rrbracket \sigma), \sigma = \sigma' \quad \text{per la regola (whff)}$
\nif $\mathscr{B} \llbracket b \rrbracket \sigma = \text{false} \quad \langle b, \sigma \rangle \rightarrow \text{false} \quad \sigma = \sigma' \quad \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma \neq \sigma'$
\nif $\mathscr{B} \llbracket b \rrbracket \sigma = \text{true} \quad \langle b, \sigma \rangle \rightarrow \text{true} \quad (\Gamma_{b,c}^n \perp)^* (\mathscr{C} \llbracket c \rrbracket \sigma) = \sigma' \neq \perp$
\n $\langle \text{while } b \text{ do } c, \sigma'' \rangle \rightarrow \sigma' \quad \text{period} \quad \mathscr{C} \llbracket c \rrbracket \sigma = \sigma'' \quad \text{per qualche } \sigma'' \neq \perp$
\nwhile $b \text{ do } c, \sigma'' \rangle \rightarrow \sigma'$
\n $\langle \text{while } b \$

we conclude hwhile α do α is the concluded has been concluded to α

 \langle while *b* do $c, \sigma \rangle \rightarrow \sigma'$ hwhile *b* do *c,*si ! s \langle while *b* do $c, \sigma \rangle \rightarrow \sigma'$ ⇣ G *n b,c*? $\overline{}$ $c, \sigma \rangle \rightarrow \sigma'$ (*^C* ^J*c*Ks)*,*^s ⁼ ^s⁰ per la regula per
Per \mathbf{b} $\frac{1}{2}$ $\binom{n}{2}$ while *b* do $c, \sigma \rangle \rightarrow \sigma'$ σ' ber ia regola(whit) ⌦ while *b* do *c,*s00↵ \langle while *b* do $c, \sigma \rangle \rightarrow \sigma'$

ponens to the inductive assumptions *P*(*c*0) and *P*(*c*1), to get h*c*0*,*si ! s⁰⁰

ponens to the inductive assumptions *P*(*c*0) and *P*(*c*1), to get h*c*0*,*si ! s⁰⁰

Considerazioni finali Comandi Semantica operazionale Big-step Semantica denotazionale Terminazione **X** Determinismo <> Equivalenza operazionale Equivalenza denotazionale e' una congruenza **Consistenza** (correttezza+ completezza) E quivalenza operazionale E Equivalenza denotazionale sono congruenze (funzioni parziali) induzione ben fondata teorema di punto fisso di Kleene