

PSC 2020/21 (375AA, 9CFU)

Principles for Software Composition

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

[http://didawiki.di.unipi.it/doku.php/](http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/) [magistraleinformatica/psc/](http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/)

09 - Denotational semantics of commands

Lambda notation

Lambda notation

Key ingredients

anonymous functions

 $\lambda x. e$ *x* serves as a formal parameter in e

denotes a function that waits for one value to be substituted for x and then evaluates e

application

 e_1 e_2 e_2 is the argument passed to the function e_1 denotes the application of the function e_1 to e_2 reduces the need of parentheses *e*1(*e*2)

Function definition

$$
f(x) \triangleq x^2 - 2 \cdot x + 5
$$

$$
f \triangleq \lambda x. \ (x^2 - 2 \cdot x + 5)
$$

unnecessary parentheses added for clarity

Associative rules

 $e_1 e_2 e_3$ is read $(e_1 e_2) e_3$ application is

left-associative

 $\lambda x. \lambda y. \lambda z. e$ is read $\lambda x. (\lambda y. (\lambda z. e))$ abstraction is

right-associative

Scoping

 $\lambda x. e$

the scope of x is e

x not visible outside *e*

like a local variable

Alpha-conversion

$$
\lambda x.\ (x^2 - 2 \cdot x + 5)
$$

 $\lambda y. (y^2 - 2 \cdot y + 5)$ the same function names of formal parameters are inessential: the two expressions denote

 $\lambda x. e \equiv \lambda y. (e^{y}/x)$ (under suitable conditions on e, y) capture-avoiding substitution (to be formalised later)

Application (beta rule)

$$
\lambda x. \ (x^2 - 2 \cdot x + 5) \qquad \text{a function}
$$

$$
(\lambda x. (x2 - 2 \cdot x + 5)) 2
$$
 its application

$$
\equiv 22 - 2 \cdot 2 + 5 = 5
$$
 its evaluation

$$
\lambda x. \lambda y. (x^2 - 2 \cdot y + 5)
$$
 a function

 $(\lambda x. \lambda y. (x^2 - 2 \cdot y + 5))$ 2 its application

$$
\lambda y. (2^2 - 2 \cdot y + 5)
$$

 \equiv

its evaluation

it is still a function!

a function its application $\lambda x. (x^2 + (\lambda y. (2 \cdot y)) 1)$ its evaluation $\lambda f. \lambda x. (x^2 + f 1)$ $(\lambda f. \lambda x. (x^2 + f 1)) (\lambda y. (2 \cdot y))$ ⌘ (the argument is a function!)

higher-order: functions as arguments/results

$$
\lambda f. \lambda x. (x^{2} + f 1)
$$
 a function
\n
$$
(\lambda f. \lambda x. (x^{2} + f 1)) (\lambda y. (2 \cdot y)) 3
$$
 its application
\n
$$
=
$$

\n
$$
\lambda x. (x^{2} + (\lambda y. (2 \cdot y)) 1)
$$
 3 its evaluation
\n
$$
=
$$

\n
$$
3^{2} + (\lambda y. (2 \cdot y)) 1
$$
 its evaluation
\nits evaluation
\nits application
\nits application
\nits application
\nits evaluation
\nits evaluation
\nits evaluation

Conditional

$$
e \rightarrow e_1, e_2
$$

if e *then* e_1 else e_2

example
$$
\min \triangleq \lambda x. \lambda y. x < y \rightarrow x, y
$$

Denotational semantics of commands

From your forms

(over 14 answers)

Denotational semantics **Communism is manufacture** *a* via *A* and then modifies the memory by assigning the corresponding value to the Let ional ional sequentias *a* via *A* and then modifies the memory by assigning the corresponding value to the location *x*. TATIONAL SEMANTICS **c c** α is the state α interpretation memory and the state produced memory and α *a* via *A* and then modifies the memory by assigning the corresponding value to the location *x*. Let us now consider the sequential composition of two commands. In interpreting The denotational semantics of the assignment evaluates the assignment evaluation \mathbf{r} *a* via *A* and then modifies the memory by assigning the corresponding value to the WE STRUCTURALLY START SIMPLES S., not not not not not not not need the cannot apply \mathbf{A} . To work the cannot apply \mathbf{A} . To work the canonical the canonical the canonical the canonical term of \mathbf{A} introduce a *lifting* operator (*·*)⇤: it takes a function in S ! S? and returns a function in S? ! S?, i.e., its type is (S ! S?) ! (S? ! S?).

$$
\mathscr{C}: Com \to (\Sigma \to \Sigma) \qquad \mathscr{C}: Com \to (\Sigma \to \Sigma_{\perp})
$$

 $\mathscr{C}: Com \to (\Sigma \to \Sigma_{\perp})$ (Γ, Γ, Γ) φ is Γ and (Γ, Γ) \rightarrow $(2 - 2)$ o. COM \rightarrow $(2 - 2)$ φ : C_{2m} i(Γ if) φ : C_{2m} i(Γ if) $\rightarrow (\Sigma \rightarrow \Sigma) \qquad \qquad \mathscr{C}: Com \rightarrow (\Sigma \rightarrow \Sigma_{\perp})$ ^S?, not necessarily in ^S, so we cannot apply *^C* ^J*c*1K. To work this problem out we by *^c*0. The problem is that from the first application of *^C* ^J*c*0^K we obtain a value in *c*0; *c*¹ we first interpret *c*⁰ in the starting memory and then *c*¹ in the state produced

 \mathscr{C} [skip] $\sigma \stackrel{\text{def}}{=} \sigma$ $f \cdot \sum \rightarrow Y$ We shall define α and α and α over the syntax of α or α or α \mathbb{L}^2 is extensive on definition. We have \mathbb{L}^2 the form \mathscr{C} defining its defining the separate equation of $f: \mathcal{E} \to \mathcal{E}_{\perp}$ and $f^* : \mathcal{E}_{\perp} \to \mathcal{E}_{\perp}$ $\mathcal{I}[x := a \mathbf{0} \boldsymbol{\sigma} = \boldsymbol{\sigma} \mathbf{e}^{\mathcal{A} \mathbf{u} \mathbf{u} \mathbf{v}}/x]$ $f^*(x) = \begin{cases} \frac{1}{f(x)} & \text{if } x \neq 0 \\ f(x) & \text{otherwise} \end{cases}$
 f $f(x) = \begin{cases} \frac{1}{f(x)} & \text{if } x = 1 \\ f(x) & \text{otherwise} \end{cases}$ \mathcal{L} **i** \mathbf{L} **f** \mathbf{L} **def** \mathbf{L} $[\mathscr{A}][a]\sigma / 1$ \mathbf{L} $\$ $\epsilon \otimes \mathbb{F}$, \mathbb{F} , \mathbb{F} and \mathbb{F} of \mathbb{F} . *f*(*x*) otherwise α *y* othe $\int_{a}^{f} \sigma^{a}$ \mathscr{C} $\lceil \mathbf{skin} \rceil$ $\sigma \stackrel{\text{def}}{=} \sigma$ $\qquad \qquad (\cdot)^*: (\Sigma \to \Sigma_\bot) \to (\Sigma_\bot \to \Sigma_\bot)$ σ $x := a \circ \sigma$ σ is the integration. $\stackrel{\text{def}}{=} \mathscr{C}\llbracket c_1 \rrbracket$ $\mathscr{C}\llbracket \textbf{skip} \rrbracket$ σ def $\frac{dE}{dt}$ σ \mathscr{L} if *h* then coeke cid $\sigma \stackrel{\text{def}}{=} \mathscr{R}$ $h \mathbb{I}$ $\sigma \rightarrow \mathscr{L}$ col $\sigma \mathscr{L}$ \det $(\cdot)^*, (\Sigma \to \Sigma)$ $\mathscr{C}\llbracket x:=a\rrbracket$ σ def $\mathscr{C}\llbracket x := a \rrbracket$ $\sigma \stackrel{\text{def}}{=} \sigma \left[\frac{\mathscr{A}\llbracket a \rrbracket \sigma}{x} \right]$ \mathscr{C} [*c*₀; *c*₁] $\sigma \stackrel{\text{def}}{=} \mathscr{C}$ [*c*₁] * (\mathscr{C} [*c*₀] σ) (*f*($\boldsymbol{\delta}$ [SKIP] $\boldsymbol{\delta} = \boldsymbol{\delta}$ def $\stackrel{\text{def}}{=} \mathscr{C}\llbracket c_1 \rrbracket$ $\mathscr{E}\left[\mathcal{C}_0\right]\sigma$ (f(x) and \mathbf{r} and \mathbf{r} is the set of \mathbf{r} def \mathscr{C} solition $\sigma \stackrel{\text{def}}{=} \sigma$ (i):(2) $\sigma \stackrel{\text{def}}{=} \mathscr{C}\left[\hspace{-0.5mm}\left[c_1\right]\hspace{-0.5mm}\right]^* \left(\mathscr{C}\left[\hspace{-0.5mm}\left[c_0\right]\hspace{-0.5mm}\right]\sigma\right)$ T definition of the denotation of the while community of the while community of the while community $\frac{1}{\sqrt{2}}$

 $\mathbf{I} \cdot \mathbf{I} = \mathbf{I} \cdot \math$ ^S?, not necessarily in ^S, so we cannot apply *^C* ^J*c*1K. To work this problem out we interview a lifting operator (*·*) **introduce a function in S ! S** $f^*(x) = \begin{cases} \frac{1}{f(x)} & \text{if } x = 0 \end{cases}$ $\int f(x) dx$ ^S?, not necessarily in ^S, so we cannot apply *^C* ^J*c*1K. To work this problem out we interview a lifting operator (*·*) **introduce a function in S ! S** introduce a *lifting* operator (*·*)⇤: it takes a function in S ! S? and returns a function $\begin{array}{c} \text{Limiting} \\ (1,1) \rightarrow \text{Matrix} \end{array}$ σ $f : \Sigma \to \Sigma$. $f^* : \Sigma \to \Sigma$. We define a function f $f^*(x) = \begin{cases} \perp & \text{if } x = \perp \\ f(x) & \text{otherwise} \end{cases}$ *f*(*x*) otherwise ^S?, not necessarily in ^S, so we cannot apply *^C* ^J*c*1K. To work this problem out we introduce a *lifting* operator (*·*)⇤: it takes a function in S ! S? and returns a function $\mathcal{A}[[a]]\sigma_{\binom{n}{k}}$ $\qquad \qquad$ $\mathcal{I}: \mathcal{L} \to \mathcal{L}_{\perp}$ $\qquad \qquad \mathcal{I}: \mathcal{L}_{\perp} \to \mathcal{L}_{\perp}$ $f(x) = \begin{cases} f(x) & \text{otherwise} \end{cases}$ by *^c*0. The problem is that from the first application of *^C* ^J*c*0^K we obtain a value in ^S?, not necessarily in ^S, so we cannot apply *^C* ^J*c*1K. To work this problem out we in S? ! S?, i.e., its type is (S ! S?) ! (S? ! S?). $f^*(x) = \begin{cases} \perp & \text{if } x = \perp \\ \frac{f^*}{x} & \text{if } x = \perp \end{cases}$ *f* $\left(\begin{array}{ccc} 0 & \cdots \end{array}\right)$ cancer \cdots **Lifting** Let us now consider the consider the consider the conditional community $\mathcal{L}(\mathcal{L})$ $f^*(x) = \begin{cases} \frac{1}{f(x)} & \text{if } x \neq 0 \\ f(x) & \text{otherwise} \end{cases}$

 \mathscr{C} [if b then c_0 else c_1] $\sigma \stackrel{\text{def}}{=} \mathscr{B}$ [b] $\sigma \to \mathscr{C}$ [c₀] σ, \mathscr{C} [c₁] σ $\begin{bmatrix} \mathcal{S} & \mathcal{$ $c_1 \mathbf{0} = \mathcal{B} \left[\mathbf{0} \right]$ $\mathbf{\sigma} \to \mathcal{C} \left[\mathbf{c}_0 \right]$ $\mathbf{\sigma}, \mathcal{C} \left[\mathbf{c}_1 \right]$ $\mathbf{\sigma}$ \mathscr{C} [if *b* then c_0 else c_1] $\sigma \stackrel{\text{def}}{=} \mathscr{B}$ [*b*] $\sigma \to \mathscr{C}$ [c_0] σ, \mathscr{C} [c_1] σ \mathscr{C} [if *b* then c_0 else c_1] $\sigma \stackrel{\text{def}}{=} \mathscr{B}$ [*b*] $\sigma \to \mathscr{C}$ [c_0] σ, \mathscr{C} [c_1] σ \mathcal{B} $\llbracket b \rrbracket$ $\sigma \rightarrow \mathscr{C}$ $\llbracket c_0 \rrbracket$ σ, \mathscr{C} $\llbracket c_1 \rrbracket$ σ \mathscr{C} | into then c_0 else c_1 | $\sigma = \mathscr{B}$ | b | $\sigma \rightarrow \mathscr{C}$ | c_0 | σ, \mathscr{C} | c_1 | σ

^A ^J*n*^K def *C* \overline{a} *c*₁ \overline{b} \mathscr{C} while *b* do c $\sigma = ?$ So the definition of the interpretation function for *c*0; *c*¹ is I_{ref} semantics of the assignment evaluation \mathcal{I}_{ref} \mathbf{L} \mathscr{C} subjected by $\mathbf{d} \cdot \mathbf{d}$ of $\mathbf{d} \cdot \mathbf{d}$ $\mathbf{d} \cdot$ $\sum_{i=1}^n \mathbb{I}_{\{i,j\}}$ to define the interpretation simply as \mathscr{C} [while *b* do *c*] σ def \equiv ?

Denotational sem. (ctd) \blacksquare indemotational sem. $\sum_{n=1}^{\infty}$ **C** Deno la flondi sem. (CIO) $\partial_{\alpha} f$ $\overline{}$ UITUTT $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ $\sum_{i=1}^{n}$ The definition of the denotational semantics of the while command is more *C*if*b*then*c*else*c*s=*Bb*s!*Cc*s*,Cc*sUCITUITUITUI JUITI. (CIU) of the defining equation. In defining equation, i.e., \mathbf{r} Denotational sem. (ctd) reduce the problem of defining the semantics of iteration to a fixpoint calculation. $\mathbf{A} \times \mathbf{B}$ $\mathbf{A} \times \mathbf{B}$ $\mathbf{A} \times \mathbf{C}$ $\mathbf{A} \times \mathbf{C}$ The definition of the definition of the while community of the whole community of the whole community of the wh
The which community of the while community of the whole community of the whole community of the whole communit *Deno* def f ational sem. (c1

 \mathscr{C} [while *b* do *c*] σ def $\mathcal{B} \equiv \mathcal{B} [b] \sigma \rightarrow \mathcal{C} [$ while *b* do c ^{[*} ($\mathcal{C} [c] \sigma$), σ \mathscr{C} while b do $c \mathbf{0}$ $\sigma = \mathscr{B}$ $\llbracket b \rrbracket$ $\sigma \to \mathscr{C}$ while b do $c \mathbf{0}$ ^{*} (\mathscr{C} $\llbracket c \rrbracket$ σ), σ \mathscr{C} [while b do c] $\sigma \stackrel{\text{def}}{=} \mathscr{B}$ [b] $\sigma \rightarrow \mathscr{C}$ [while b do c] $^*(\mathscr{C})$ $\mathscr{C}[\![\text{while }b\text{ do }c]\!] \sigma \stackrel{\text{def}}{=} \mathscr{B}[\![b]\!] \sigma \rightarrow \mathscr{C}[\![\text{while }b\text{ do }c]\!]^*\left(\mathscr{C}[\![c]\!]\!\right)\sigma, \sigma$ *C* While *b* do $c \cdot \sigma = \mathcal{B} \cdot d \sigma \rightarrow \mathcal{C}$ while *b* do $c \cdot \sigma$ (*C* $c \cdot \sigma$), σ

 $\mathscr{C}[\![\text{while }b\text{ do }c]\!] = \lambda \sigma.\mathscr{B}[\![b]\!] \sigma \rightarrow \mathscr{C}[\![\text{while }b\text{ do }c]\!]^*(\mathscr{C}[\![c]\!] \sigma), \sigma$ *^C* ^Jwhile *^b* do *^c*^K whose meaning we want to define appears in the right-hand side \mathscr{C} while h do $c \mathbb{I}$ def $c \in \mathbb{R}$ of \mathbb{R} of \mathbb{R} iteration to a \mathbb{R}^* (\mathscr{O} \mathbb{R}) τ $\mathbb{L} \mathbb{L} \longrightarrow \mathbb{L}$ \mathscr{C} [while *b* do *c*] $\stackrel{\text{dcl}}{=} \lambda \sigma \cdot \mathscr{B}$ [*b*] $\sigma \rightarrow$ def \mathscr{C} [while b do c] $\stackrel{\text{def}}{=} \lambda \sigma \mathscr{B} [b] \sigma \rightarrow \mathscr{C}$ [while b do c]* ($\mathscr{C} [c]$ do \equiv \equiv $\overline{\Omega}$ α hile *b* do *c* $\parallel \cong \lambda \sigma$. $\mathscr{B}[[b]]\sigma \rightarrow \mathscr{C}[[\text{while } b \text{ do}]$ def σ . $\mathscr{B}\llbracket b\rrbracket$ $\sigma \to \mathscr{C}\llbracket \textbf{while }b\textbf{ do }c\rrbracket^*\left(\mathscr{C}\llbracket c\rrbracket\sigma\right),\sigma$ \equiv $\mathbf{L} \cdot \mathbf{d} \cdot \mathbf{e}^{\mathbb{R}^* \cdot (\mathcal{O} \mathbb{R}^n)}$ intricate. We consider the interpretation of \mathbb{F}_p of while b do $c \mathbb{I} = \lambda \sigma \otimes \mathbb{I}$ $h \mathbb{I} \sigma \rightarrow \mathscr{C}$ while b do $c \mathbb{I}^* (\mathscr{C} \mathbb{I} \sigma)$ of α \equiv reduce the problem of defining the semantics of the semantics of iteration to a fixpoint calculation. The semantics of iteration to a fixpoint calculation to a fixpoint calculation. The semantic calculation to a fixpoint \equiv

 $(1 - \rho_0)$ the defining extending \mathbb{R}^* on \mathbb{R}^* allows only for the presence of th $(\lambda \psi, \lambda \mathbf{0}, \mathcal{D} \psi) \circ \psi$ *(0* | C | C | O $\mathbf{0}, \mathbf{0}$ ψ will $\mathbf{0}$ we c def $\partial_t \left[\partial_t \phi \partial_t \partial_t \mathcal{L} \right] = \partial_t \left[\partial_t \phi \partial_t \phi \partial_t \mathcal{L} \right] = \partial_t \left[\partial_t \phi \partial_t \phi \partial_t \phi \partial_t \phi \right]$ $\alpha(\lambda \varphi, \lambda \sigma, \mathscr{B}$ $[b]$ $\sigma \rightarrow \varphi^*(\mathscr{C}$ $[c]$ $\sigma)$, σ β σ while b $(\lambda \psi, \lambda \mathbf{0}, \mathscr{B}[\![\mathbf{0}]\!] \mathbf{0} \rightarrow \psi$ (*C* $[\![\mathbf{C}]\!] \mathbf{0}$), $\mathbf{0}$) of $[\![\mathbf{W}\!]$ and $\mathbf{0}$ $(\lambda \varphi, \lambda \sigma, \mathscr{B}[[b]]\sigma \to \varphi^*(\mathscr{C}[[c]]\sigma), \sigma) \mathscr{C}[[\mathbf{while } b \mathbf{do } c]]$
 $\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda \varphi, \lambda \sigma, \mathscr{B}[[b]]\sigma \to \varphi^*(\mathscr{C}[[c]]\sigma), \sigma$ of the defining equation. In defining equation. In defining equation allows only for the presence α $(\lambda \varphi, \ \lambda \sigma, \ \mathscr{B}\llbracket b \rrbracket \, \sigma \rightarrow \varphi^*(\mathscr{C}\llbracket c \rrbracket \, \sigma), \sigma) \ \ \mathscr{C}\llbracket \text{while } b \ \textbf{do } c \rrbracket$ $(\lambda \varphi, \lambda \sigma, \mathscr{B}[[b]] \sigma \rightarrow \varphi^*$

$$
\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda \varphi \ldotp \lambda \sigma \ldotp \mathscr{B}\llbracket b \rrbracket \sigma \to \varphi^*(\mathscr{C}\llbracket c \rrbracket \sigma), \sigma
$$

G*b,^c* \overline{a} $\boldsymbol{\sigma}$ $c \parallel = \Gamma_{b,c} \mathcal{C} \parallel$ while b do $c \parallel$ \mathscr{O} $\mathbb{F}_{\mathbf{v}}$ subjects to ϕ \mathbf{r} takes a function \mathbf{r} is set to \mathbf{r} . The function \mathbf{r} is set to \mathbf{r} and \mathbf{r} is set to \mathbf{r} and \mathbf{r} is set to \mathbf{r} and \mathbf{r} and \mathbf{r} is set to \mathbf{r} and \mathbf{r} an \mathscr{C} [while *b* do c] = $\Gamma_{b,c}$ \mathscr{C} [while *b* do c] \mathscr{C} while b do $c = \Gamma_b$, \mathscr{C} while b do c $c \llbracket \mathbf{F} \rrbracket = \Gamma_{b,c} \mathscr{C} \llbracket \mathbf{while} \; b \; \mathbf{do} \; c \rrbracket$ \mathscr{C} while b do $c = \Gamma_{b,c} \mathscr{C}$ while b do c \mathbf{m} e t

 $p = J(p)$ a lixpoint eq $p = f(p)$ a fixpoint equation! $p = J(p)$ a fixpoint equation: $p = f(p)$ a fixpoint equation! $p = f(p)$ a fixpoint equation! S? $p = f(p)$ a fixpoint equation! $p = f(p)$ a fixpoint equation! α α β α β β β β β Let ion define a function \mathbf{S} $\binom{S}{\infty}$ of \mathbf{f} $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$

Denotational sem. (ctd) ⁼ *^B* ^J*b*K^s ! *^C* ^J*c*0Ks*,^C* ^J*c*1K^s (6.4) The definition of the definition of the definition of the denotation of the while community \mathcal{L}_max **C** Denotational sem (ct of the defining equation. Indeed structural recursion allows only for the presence outcome returns either ^j⇤(*^C* ^J*c*Ks) or ^s. Note that the definition of ^G*b,^c* refers only to subterms of the command while *^b* do *^c*. Clearly we require that *^C* ^Jwhile *^b* do *^c*^K is a fixpoint of G*b,c*, i.e., that a conditional operator; then we have immediately Let us define a function G*b,^c* : (S! S?) ! S ! S?: def $\mathbf{J} \cdot \mathbf{J} = \mathbf{J} \cdot \mathbf{J} \cdot \mathbf{J} \cdot \mathbf{J} \cdot \mathbf{J}$ Let us define a function G*b,^c* : (S ! S?) ! S ! S?: def \Box **.** *b*²**l**, *l*, **J**_c <u>I</u>**D** *D CO D***_—l**, 2**l**, *l*, *d*, ¹ Let us define a function G*b,^c* : (S ! S?) ! S ! S?: def Γ \mathcal{O} \mathbb{L} -**b**², *b*_B \mathbb{L} \math $\sum_{n=1}^{\infty}$ \overline{C} \overline{C} \overline{D} \overline{D} \overline{D} \overline{D} \overline{C} \overline{C} $\overline{}$ \mathcal{O} $\mathbb{F}_{\text{--}}$ **,** $\mathbb{F}_{\text{--}}$ $\mathbb{F}_{\text{--}}$ $\sum_{n=1}^{\infty}$ $\mathcal{L} \circ \mathbb{R}$ $\mathbb{L} \circ \mathbb{L} \circ \mathbb$ Since commands can diverge, the codomain of *C* is the set of partial functions from memories to memories. As we have discussed in Example 5.14, for each partial function we can define an equivalent total function. So we define T denote the assignment evaluation of the assignment evaluation of the arithmetic expression of the ari *a* via *A* and then modifies the memory by assigning the corresponding value to the location *x*. *a* via *A* and then modifies the memory by assigning the corresponding value to the Ω Let us now consider the sequential composition of two commands. In interpreting rad sem (ctd) of the defining equation. In the presence of t *^C* ^Jwhile *^b* do *^c*K^s ⁼ *^B* ^J*b*K^s ! *^C* ^Jwhile *^b* do *^c*^K Oppion is not definition in the structure of the second is not a structure that the same expression, and the s *^C* ^Jwhile *^b* do *^c*^K whose meaning we want to define appears in the right-hand side not def A *Mal sem. (ctd)* $\overline{\mathcal{O}}$ obviously this definition is not a structural recursion, because the same expression, because *^C* ^J*^x* :⁼ *^a*K^s THOMAI SEM. (CIA) *a* via *A* and then modifies the memory by assigning the corresponding value to the $i = 1, 2, \ldots, n$ \mathbf{b} do $\mathbf{c}^{\mathbb{T}} - \mathbf{c}^{\mathbb{C}}$ def \mathbf{B} \mathbf{b} do \mathbf{a}^{T} \mathbf{c} \mathbf{c} $(\nabla \cdot \nabla)$ *m. (c* def $\mathbf{\dot{r}}$ *^A* ^J*a*K^s */x*] (6.2)

$$
\mathscr{C}[\![\text{while } b \text{ do } c]\!] = \Gamma_{b,c} \mathscr{C}[\![\text{while } b \text{ do } c]\!] \qquad \qquad \mathscr{C}: Com \to (\Sigma \to \Sigma_{\bot})
$$

 $\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda \varphi$. $\lambda \sigma$. $\mathscr{B}[[b]] \sigma \rightarrow \varphi^*(\mathscr{C}[[c]] \sigma)$, σ \mathcal{L}_\perp and \mathcal{L}_\perp $\sum \rightarrow \sum$ the definition allows only for the presence of the p $\begin{array}{c} \begin{array}{c} \begin{array}{c} \mathcal{L} \longrightarrow \Sigma_{\bot} \end{array} & \longrightarrow & \mathcal{L} \longrightarrow \Sigma_{\bot} \end{array} \end{array}$ def $\vec{p} = \lambda \varphi$. $\lambda \sigma$. $\mathscr{B}\llbracket b \rrbracket \sigma \rightarrow \varphi^*(\mathscr{C}\llbracket c \rrbracket \sigma)$, σ Σ_{\perp} $\Sigma \rightarrow \Sigma_{\perp}$ $(\Sigma \rightarrow \Sigma_{\perp}) \rightarrow \Sigma \rightarrow \Sigma_{\perp}$ \det one. \det is a monotone and continuous function, so that we continuous function, so that $\mathcal{L}_{b,c} = \lambda \varphi$. $\lambda \sigma$. $\mathscr{B} \llbracket b \rrbracket \sigma \rightarrow \varphi^*(\mathscr{C} \llbracket c \rrbracket \sigma)$, σ *^C* ^Jwhile *^b* do *^c*^K def $\overline{z} \rightarrow \overline{z}_{\perp}$ G *n* Γ def γ_{α} γ_{α} γ_{α} interpretation simply as γ_{α} \sum_{\perp} ⇤ (*^C* ^J*c*Ks)*,*^s $\mathcal{L} \subset \mathbb{R}_+$ for a structural recover $\mathscr{C} \llbracket c \rrbracket$ ($\begin{array}{rcl} (2 \rightarrow & \!\!\!\!\! \rightarrow & \!\!\!\!\! \searrow & \!\!\!\!\! \se$ $\frac{1}{\sqrt{2}}$ is the function of $\frac{1}{\sqrt{2}}$ is the function $\frac{1}{\sqrt{2}}$ is the function of $\frac{1}{\sqrt{2}}$ $\Sigma \rightarrow \Sigma_{\perp}$ ω^* $\frac{\mathbf{r}}{\mathbf{r}}$, $\frac{\mathbf{r}}{\mathbf{r}}$ is the function $\frac{\mathbf{r}}{\mathbf{r}}$ is set of function $\frac{\mathbf{r}}{\mathbf{r}}$ $\Sigma \rightarrow \Sigma_{\perp}$ $*(C\mathcal{O} \mathbb{F}_{\alpha} \mathbb{F}_{\alpha})$ \rightarrow $\frac{\mathbf{L}^2 \mathbf{L}^2 \math$ $\Sigma \rightarrow \Sigma_{\perp}$ Γ def 2 Ω $\frac{\mathbb{L} \mathbb{L} \math$ l s*. ^B* ^J*b*K^s ! ^j⇤(*^C* ^J*c*Ks)*,*^s Γ def $-\nu, c$ the function \mathbb{L}^2 is seen function \mathbb{L}^2 in \mathbb{L}^2 is seen function \mathbb{L}^2 l s*. ^B* ^J*b*K^s ! ^j⇤(*^C* ^J*c*Ks)*,*^s $\int f_1$ $\int g_2$ $\int g_1$ $\int g_2$ $\frac{1}{\sqrt{2}}$, $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ $\frac{1}{\sqrt{2}}$ l s*. ^B* ^J*b*K^s ! ^j⇤(*^C* ^J*c*Ks)*,*^s $\phi \to \boldsymbol{\phi}^*(\mathscr{C}\|\boldsymbol{c}\|\,\boldsymbol{\sigma}), \boldsymbol{\sigma} \qquad \qquad \boldsymbol{\phi}: \boldsymbol{\Sigma} \to \boldsymbol{\Sigma}_+$ necessary explanations. $\varphi^* \colon \Sigma_\bot \to \Sigma_\bot$ $\mathscr{B}[[b]]\sigma \to \varphi^*(\mathscr{C}[[c]]\sigma), \sigma$ $\varphi: \Sigma \to \Sigma$. $\frac{\varphi \llbracket \nu \rrbracket \mathbf{v} - \varphi \left(\varphi \llbracket \mathbf{c} \rrbracket \mathbf{v} \right), \mathbf{v}}{\Gamma}$, where $\varphi : \mathcal{L} \to \mathcal{L}_{\perp}$ \mathscr{C} \mathscr{L} \mathscr{L} \rightarrow \mathscr{L} $\mathscr{L}[\mathbb{I}] = \mathbb{I}$ follows: $\lambda \sigma \propto \mathbb{Z} \left[h \right] \sigma \propto \omega^* (\mathscr{L} \left[\right] \sigma) \sigma$: (0, i) def \det θ \in θ $\mathcal{F}_{\mathbf{a}}_{\mathbf{c}} \stackrel{\text{def}}{=} \lambda \varphi \cdot \lambda \sigma$. $\mathscr{B}[[b]] \sigma \rightarrow \varphi^*(\mathscr{C}[[c]] \sigma)$, σ $\qquad \qquad \varphi \cdot \mathbf{y} \rightarrow \mathbf{y}$ Let us define a function G*b,^c* : (S ! S?) ! S ! S?: G*b,^c* $\frac{\Sigma \rightarrow \Sigma_{\perp}}{\Sigma \rightarrow \Sigma_{\perp}}$ $\mathscr{C}[[c]] \sigma : \Sigma_{\perp}$ $\mathscr{B}[[b]]\sigma \to \varphi^*(\mathscr{C}[[c]]\sigma), \sigma$ $\varphi: \Sigma \to \Sigma_\bot$ Σ_{\perp} and Σ_{\perp} and φ^* : Σ_{\perp} \rightarrow Σ_{\perp} $\begin{CD} \varphi_{\perp} & \qquad \psi : \mathcal{L}_{\perp} \to \mathcal{L}_{\perp} \ & \qquad \mathcal{L}_{\perp} \to \mathcal{L}_{\perp} \end{CD}$ \det $\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda \varphi. \; \lambda \sigma. \; \mathscr{B}\llbracket b \rrbracket \, \sigma \to \varphi^*(\mathscr{C}\llbracket c \rrbracket \, \sigma), \sigma \qquad \qquad \varphi: \Sigma \to \Sigma_\bot$ $\Sigma \rightarrow \Sigma_{\perp}$ $\Sigma \rightarrow \Sigma_{\perp} \rightarrow \Sigma \rightarrow \Sigma_{\perp}$ $\Sigma \rightarrow \Sigma_{\perp}$ in S, $\mathscr{C} \llbracket c \rrbracket \boldsymbol{\sigma} : \Sigma_{\perp}$ $\mathcal{I}\rightarrow\mathcal{I}\rightarrow\mathcal{I}\rightarrow\mathcal{I}$ is taken in S $\mathbb{E}\left[\mathcal{C}\right]$ **o** $\cdot\mathcal{L}\perp$ $\begin{aligned} \lambda \, \pmb{\varphi}.\,\, \lambda \, \pmb{\sigma}.\,\, \mathscr{B}\, \llbracket b \rrbracket \, \pmb{\sigma} \rightarrow \pmb{\varphi}^* (\mathscr{C}\, \llbracket c \rrbracket \, \pmb{\sigma}), \pmb{\sigma} \ & \Sigma_\perp \end{aligned}$

 $\mathscr{C}\left[\!\left[c \right]\!\right]$ σ : 2 $\begin{CD} \vdash \bot \end{CD} \qquad \qquad \begin{CD} \forall \cdot \angle \bot \end{CD} \rightarrow \angle \bot$ φ (*v* [*x*] φ) \cdot φ \perp \mathscr{C} $\llbracket c \rrbracket$ σ), σ $\qquad \qquad \varphi : \Sigma \to \Sigma_{\perp}$ $\begin{CD} \begin{picture}(180,10) \put(0,0){\line(1,0){10}} \put(1,0){\line(1,0){10}} \put(1,0){\line(1,0){10$ $f: \Sigma_{\perp}^+$ $\bm{\varphi}: \bm{\varSigma} \rightarrow \bm{\varSigma}_{\perp}$ The function G*b,^c* takes a function j : S ! S?, and returns the function σ . $\llbracket c \rrbracket$ σ^2 $\varphi^*(\mathscr{C}\llbracket c\rrbracket \sigma) : \Sigma_\perp$

 $\mathscr{C}: \mathit{Com} \to (\Sigma \to \Sigma_{\perp})$

$$
T_{b,c}: (\Sigma \to \Sigma_{\perp}) \to \Sigma \to \Sigma_{\perp}
$$
\n
$$
\varphi^*(\mathscr{C}[[c]] \sigma): \Sigma_{\perp}
$$
\npartial functions\n
$$
\Sigma \to \Sigma
$$
\nsets of pairs\n
$$
(\sigma, \sigma')
$$
\n
$$
T_{b,c}: (\Sigma \to \Sigma_{\perp}) \to \Sigma \to \Sigma_{\perp}
$$
\n
$$
\varphi^*(\mathscr{C}[[c]] \sigma): \Sigma_{\perp}
$$

Monotone and continuous The function G*b,^c* takes a function j : S ! S?, and returns the function of subterms in the right-hand side, like *^B* ^J*b*^K and *^C* ^J*c*K. To solve this issue we will reduce the problem of defining the semantics of iteration to a fixpoint calculation. Let us define the control of the control of \overline{C}

$$
\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda \, \pmb{\varphi}. \,\, \lambda \, \pmb{\sigma}. \,\, \mathscr{B}\llbracket b \rrbracket \, \pmb{\sigma} \to \pmb{\varphi}^{\ast}(\mathscr{C}\llbracket c \rrbracket \, \pmb{\sigma}), \pmb{\sigma}
$$

$$
\text{Take } \quad R_{b,c} = \left\{ \frac{(\sigma'', \sigma')}{(\sigma, \sigma')} \mathcal{B} \llbracket b \rrbracket \sigma \wedge \mathcal{C} \llbracket c \rrbracket \sigma = \sigma'' \ , \ \frac{1}{(\sigma, \sigma)} \mathcal{B} \llbracket \neg b \rrbracket \sigma \right\}
$$

clearly *R* $R_{b,c} = \Gamma_{b,c} \;$ when we see $\Gamma_{b,c}$ as operating over partial functions \hat{C} C dearly $\hat{R}_{bc} = \Gamma_{bc}$ when we see Γ_{bc} as oneration over least one. Next we show that G*b,^c* is a monotone and continuous function, so that we *partial functions*

R $\widehat R_{b,c}\,$ is (monotone and) continuous, and so is $\,\Gamma_{b,c}\,$ outcome returns either ^j⇤(*^C* ^J*c*Ks) or ^s. Note that the definition of ^G*b,^c* refers only

$$
\mathscr{C}[\![\text{while } b \text{ do } c]\!] \stackrel{\text{def}}{=} \text{fix } \varGamma_{b,c} = \bigsqcup_{n \in \mathbb{N}} \varGamma_{b,c}^n(\perp_{\varSigma \to \varSigma_{\perp}}) \lambda \sigma. \perp
$$

Bottom

- Σ_{\perp} has a bottom element: \perp
- $\Sigma \rightarrow \Sigma_{\perp}$ has a bottom element: $\lambda \sigma$. \perp

to avoid ambiguities we denote the bottom element of a domain D by \perp_D

 \perp \geq \rightarrow \geq \parallel

Example $w =$ while true do skip which the skip true do skip true do skip $w =$ while true do skip *w* = while true do skip *w* = while true do skip $w =$ while true do skip $Example$ $W =$ **WHILE CALCE 40** SKIP \mathbb{R} 1. $w =$ while true do skip

$\Gamma_{\text{true,skip}}\varphi\sigma = \mathscr{B}[\![\text{true}]\!] \sigma \to \varphi^*(\mathscr{C}[\![\text{skip}]\!]\sigma),$ $\varGamma_{\textbf{true},\textbf{skip}}\varphi\sigma = \mathscr{B}\llbracket \textbf{true}\rrbracket\,\sigma \rightarrow \varphi^*\left(\mathscr{C}\llbracket \textbf{skip}\rrbracket\,\sigma\right),\sigma$ $\boldsymbol{\varphi} = \boldsymbol{\varphi}^*(\mathscr{C} [\![\textbf{skip}]\!]\, \boldsymbol{\sigma})$ $=\varphi^*\sigma$ $= \varphi \sigma$ $\mathbf{r} = \mathbf{true} \rightarrow \varphi^* \left(\mathscr{C} \left[\mathbf{skip} \right] \sigma \right), \sigma$ $= \varphi^* (\mathscr{C} [\![\text{skip}]\!] \, \sigma)$ $\text{Lie}, \text{skip} \varphi\sigma = \mathscr{B} \llbracket \text{true} \rrbracket\, \sigma \to \phi^* \left(\mathscr{C} \llbracket \text{skip} \rrbracket\, \sigma \right), \sigma$ $\mathbf{true}\rightarrow \boldsymbol{\varphi}^{*}\left(\mathscr{C}\left[\mathbf{skip}\right]\mathbf{\sigma}\right), \boldsymbol{\sigma}$ $= \varphi^* (\mathscr{C} [\![\textbf{skip}]\!] \, \sigma)$ $= \boldsymbol{\varphi}^* \boldsymbol{\sigma}$ $= \varphi \sigma$ $-\varphi\sigma$ $\varGamma_{\textbf{true},\textbf{skip}}\varphi\sigma = \mathscr{B}\llbracket \textbf{true}\rrbracket\,\sigma \rightarrow \varphi^*\left(\mathscr{C}\llbracket \textbf{skip}\rrbracket\,\sigma\right),\sigma$ $= \mathsf{true} \to \varphi^* \left(\mathscr{C} \left[\textsf{skip} \right] \sigma \right), \sigma$ $= \varphi^* (\mathscr{C} [\![\text{skip}]\!] \, \sigma)$ $= \, \boldsymbol{\varphi}^* \boldsymbol{\sigma}$ $\begin{aligned} \n\textbf{Example} \n\textbf{Example$

 $I_{true,skip} \varphi = \varphi$ $\int \frac{dx}{\sin x} dx = \int \frac{dx}{\sin x} dx$ $f(x)$ if $f(x)$ is a fixed for the looking for ΔE $I_{\text{true},\text{skip}}\varphi = \varphi$ $I_{\text{true},\text{skip}}$ is the identity function. function j is a fixpoint of Gtrue*,*skip, but we are looking for the least fixpoint. This $\Gamma_{\rm true, skip} \varphi = \varphi$ $\Gamma_{\rm true, skip}$ is the identity function. every element is a $\begin{array}{rcl}\n\text{fixpoint} \\
\text{fix } & \Gamma\n\end{array}$ $\lim_{x \to \infty} \frac{f(x)}{1 + x}$ $\Gamma_{\text{true, skin}} \varphi = \varphi$ $\Gamma_{\text{true, skin}}$ is the identity function. fix $\varGamma_{\rm true,skip} = \lambda \, \sigma$. $\perp_{\varSigma_{\perp}}$ $I_{\text{true},\text{skip}}\varphi = \varphi$ $I_{\text{true},\text{skip}}$ is the identity function. function j is a fixpoint of Gtrue*,*skip, but we are looking for the least fixpoint. This $f(x) = \frac{f(x)}{x}$ function is the least function in the CPO² is the CPO² is the CPO² is the CPO² is solved that S^2 every element is a
fixpoint fixpoint

$$
w \triangleq \text{while } x > 1 \text{ do } x := x - 1
$$

 $\Gamma_{b,c} \varphi \sigma = \mathcal{B}\llbracket x > 1 \rrbracket \sigma \rightarrow \varphi^*(\mathcal{C}\llbracket x := x - 1 \rrbracket \sigma), \sigma$ $= (\sigma(x) > 1) \rightarrow \varphi^*(\sigma[\sigma(x)-1/x]), \sigma$

$$
\widehat{R}_{b,c} \stackrel{\Delta}{=} \left\{ \begin{array}{c} \overline{\sigma, \sigma} \\ \overline{\sigma, \sigma} \end{array} \sigma(x) \leq 1 \quad , \quad \frac{(\sigma'', \sigma')}{(\sigma, \sigma')}\sigma(x) > 1 \land \sigma'' = \sigma[\sigma(x) - 1_{x}] \end{array} \right.
$$

$$
\widehat{R}_{b,c} \stackrel{\Delta}{=} \left\{ \begin{array}{c} \frac{\sigma(\sigma, \sigma)}{\sigma, \sigma} \end{array} \right. \sigma(x) \leq 1 \quad , \quad \frac{\sigma(\sigma(\sigma(x)-1)}{\sigma, \sigma')}{\sigma(x)} \cdot \sigma(x) > 1
$$

$$
w \stackrel{\Delta}{=} \textbf{while } x > 1 \textbf{ do } x := x - 1
$$

$$
\widehat{R}_{b,c} \triangleq \begin{cases} \frac{\sigma}{(\sigma,\sigma)} \sigma(x) \leq 1, & \frac{(\sigma[\sigma(x)-1]_x], \sigma'}{(\sigma,\sigma')} \sigma(x) > 1 \end{cases}
$$

$$
\widehat{R}_{b,c}^{0}(\emptyset) = \emptyset
$$
\n
$$
\widehat{R}_{b,c}^{1}(\emptyset) = \{(\sigma, \sigma) \mid \sigma(x) \le 1\}
$$
\n
$$
\widehat{R}_{b,c}^{2}(\emptyset) = \widehat{R}_{b,c}^{1}(\emptyset) \cup \{(\sigma, \sigma[1/x]) \mid \sigma(x) = 2\}
$$
\n
$$
\widehat{R}_{b,c}^{3}(\emptyset) = \widehat{R}_{b,c}^{2}(\emptyset) \cup \{(\sigma, \sigma[1/x]) \mid \sigma(x) = 3\}
$$
\n
$$
\dots
$$
\n
$$
\widehat{R}_{b,c}^{n}(\emptyset) = \{(\sigma, \sigma) \mid \sigma(x) \le 1\} \cup \{(\sigma, \sigma[1/x]) \mid 1 < \sigma(x) \le n\}
$$
\n
$$
\dots
$$
\n
$$
\mathcal{C}[\![w]\!] = \widehat{hx}(\widehat{R}_{b,c}) = \{(\sigma, \sigma) \mid \sigma(x) \le 1\} \cup \{(\sigma, \sigma[1/x]) \mid 1 < \sigma(x)\}
$$