Fundamentele limbajelor de programare
CURS 3

Traian Florin Șerbănuță și Andrei Sipoș

Facultatea de Matematică și Informatică, DL Info, Anul II
Semestrul II, 2023/2024

Din nou despre echivalența între programe

Reamintim că relația de echivalență pe instrucțiuni $\sim$ a fost definită în felul următor: pentru orice c_0, c_1, avem $c_0 \sim c_1$ exact atunci când, pentru orice σ, $\sigma' \in \Sigma$, $(c_0, \sigma) \Downarrow \sigma'$ dacă și numai dacă $(c_1, \sigma) \Downarrow \sigma'$.

Observăm că ultima condiție poate fi exprimată și ca:

$$\{(\sigma, \sigma') \in \Sigma^2 \mid (c_0, \sigma) \Downarrow \sigma'\} = \{(\sigma, \sigma') \in \Sigma^2 \mid (c_1, \sigma) \Downarrow \sigma'\},$$

sau chiar, notând cele două mulțimi corespunzător,

$$[\![c_0]\!] = [\![c_1]\!].$$

Aceasta ne conduce la ideea de a defini aceste mulțimi fără a face apel la vreo semantică operațională, dând practic un înțeles fiecărei instrucțiuni, înțeles care se va numi **semantică denotațională**, tradițional notată cu semnul $[\![\cdot]\!]$.

Definirea semanticii denotaționale: expresii

Pentru evaluarea expresiilor aritmetice, definim pur și simplu $[\![\cdot]\!] : \mathit{ExprArit} \to \mathbb{Z}^{\Sigma}$, pentru orice expresie aritmetică a și orice $\sigma \in \Sigma$, prin $[\![a]\!] := \sigma \mapsto e_\sigma(a)$.

Similar, pentru evaluarea expresiilor booleene, definim $[\![\cdot]\!] : \mathit{ExprBool} \to \{0,1\}^{\Sigma}$, pentru orice expresie booleană b și orice $\sigma \in \Sigma$, prin $[\![b]\!] := \sigma \mapsto e_\sigma(b)$.

Definirea semanticii denotaționale: instrucțiuni

Pentru instrucțiuni, vom defini o funcție

$$[\![\cdot]\!] : \mathit{Instrucțiuni} \to \mathcal{P}(\Sigma^2),$$

în mod recursiv, în felul următor:

$$[\![\textbf{skip}]\!] := \Delta_\Sigma \qquad [\![X := a]\!] := \left\{ \left(\sigma, \sigma_{X \mapsto [\![a]\!](\sigma)}\right) \mid \sigma \in \Sigma \right\}$$

$$\begin{aligned} [\![c_0; c_1]\!] &:= [\![c_1]\!] \circ [\![c_0]\!] \\ &= \{(\sigma, \sigma') \mid \text{există } \sigma'' \text{ cu } (\sigma, \sigma'') \in [\![c_0]\!] \text{ și } (\sigma'', \sigma') \in [\![c_1]\!]\} \end{aligned}$$

$$\begin{aligned} [\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!] := \{(\sigma, \sigma') \in [\![c_0]\!] \mid [\![b]\!](\sigma) = 1\} \\ \cup \{(\sigma, \sigma') \in [\![c_1]\!] \mid [\![b]\!](\sigma) = 0\} \end{aligned}$$

Definirea semanticii lui **while** va fi mai problematică, dat fiind că ambele semantici operaționale o definesc în funcție de ea însăși, nerespectând principiul recursiei structurale. Acest lucru a fost posibil doar cu prețul introducerii structurii inductive a deducțiilor. Avantajul semanticii denotaționale va fi că se va baza exclusiv pe principiile de inducție și recursie pe instrucțiuni în sine.

Despre **while**

Reamintim că am demonstrat (și apoi am și inclus în definiția semanticii small-step) că, pentru orice expresii potrivite b, c,

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

așadar am dori:

$$[\![\textbf{while } b \textbf{ do } c]\!] = [\![\textbf{if } b \textbf{ then } (c; (\textbf{while } b \textbf{ do } c)) \textbf{ else skip}]\!].$$

Notând, ca mai înainte, $w := \textbf{while } b \textbf{ do } c$, am avea:

$$\begin{aligned}
[\![w]\!] &= \{(\sigma, \sigma') \in [\![c; w]\!] \mid [\![b]\!](\sigma) = 1\} \cup \{(\sigma, \sigma) \mid [\![b]\!](\sigma) = 0\} \\
&= \{(\sigma, \sigma') \in [\![w]\!] \circ [\![c]\!] \mid [\![b]\!](\sigma) = 1\} \cup \{(\sigma, \sigma) \mid [\![b]\!](\sigma) = 0\} \\
&= \{(\sigma, \sigma') \mid [\![b]\!](\sigma) = 1, \text{există } \sigma'' \text{ cu } (\sigma, \sigma'') \in [\![c]\!] \text{ și } (\sigma'', \sigma') \in [\![w]\!]\} \\
&\qquad \cup \{(\sigma, \sigma) \mid [\![b]\!](\sigma) = 0\}.
\end{aligned}$$

Punct fix

Pentru orice $h : \Sigma \to \{0, 1\}$ și orice $C \subseteq \Sigma^2$, definim un operator $\Gamma_C^h : \mathcal{P}(\Sigma^2) \to \mathcal{P}(\Sigma^2)$, pentru orice $A \subseteq \Sigma^2$, prin:

$$\Gamma_C^h(A) := \{(\sigma, \sigma') \mid h(\sigma) = 1, \text{ există } \sigma'' \text{ cu } (\sigma, \sigma'') \in C \text{ și } (\sigma'', \sigma') \in A\} \cup \{(\sigma, \sigma) \mid h(\sigma) = 0\}.$$

Atunci va trebui să avem $\Gamma_{[\![c]\!]}^{[\![b]\!]}([\![w]\!]) = [\![w]\!]$.

În general, pentru orice mulțime X și orice $F : \mathcal{P}(X) \to \mathcal{P}(X)$, spunem că un $Y \subseteq X$ este **punct fix** al lui F dacă $F(Y) = Y$ și **cel mai mic punct fix** (**least fixed point**, **lfp**; dacă există, se notează cu μF) dacă e punct fix și, pentru orice punct fix Z, avem $Y \subseteq Z$. Vom vrea, deci, ca $[\![\textbf{while } b \textbf{ do } c]\!]$ să fie punct fix al lui $\Gamma_{[\![c]\!]}^{[\![b]\!]}$. Vom arăta că acest operator are chiar lfp și vom defini

$$[\![\textbf{while } b \textbf{ do } c]\!] := \mu\Gamma_{[\![c]\!]}^{[\![b]\!]}.$$

Alegerea va fi justificată prin faptul că semantica astfel definită va coincide cu semantica operațională structurală.

Operatori monotoni

Pentru orice mulțime X și orice $F : \mathcal{P}(X) \to \mathcal{P}(X)$, spunem că F este **monoton** dacă, pentru orice $A, B \subseteq X$ cu $A \subseteq B$, avem $F(A) \subseteq F(B)$. Se observă imediat că, pentru orice $h : \Sigma \to \{0, 1\}$ și orice $C \subseteq \Sigma^2$, Γ^h_C este monoton.

Teorema Knaster-Tarski

Fie X o mulțime și $F : \mathcal{P}(X) \to \mathcal{P}(X)$ monoton. Atunci F are lfp. Mai mult, pentru orice $Y \subseteq X$ cu $F(Y) \subseteq Y$, $\mu F \subseteq Y$.

Demonstrație

Notăm $Z := \bigcap\{Y \subseteq X \mid F(Y) \subseteq Y\}$ (bine definită, de ce?). Este suficient să arătăm că Z este punct fix (ultima parte, și implicit că este lfp, este imediată). Arătăm că $F(Z) \subseteq Z$. Fie $x \in F(Z)$ și $Y \subseteq X$ cu $F(Y) \subseteq Y$. Vrem $x \in Y$. Cum $Z \subseteq Y$, avem $F(Z) \subseteq F(Y) \subseteq Y$, deci $x \in Y$. Arătăm că $Z \subseteq F(Z)$. Fie $x \in Z$, deci, pentru orice $Y \subseteq X$ cu $F(Y) \subseteq Y$, $x \in Y$. E suficient, deci, să arătăm $F(F(Z)) \subseteq F(Z)$, care rezultă din monotonie și din faptul că $F(Z) \subseteq Z$.

Din nou despre echivalența între semantici

Teoremă

Fie c o instrucțiune și σ, $\sigma' \in \Sigma$. Atunci

$$(c, \sigma) \Downarrow \sigma' \text{ dacă și numai dacă } (\sigma, \sigma') \in [\![c]\!].$$

Demonstrație

Demonstrăm implicația „$\Rightarrow$”. Facem inducție după regulile pentru relația $\Downarrow$. Tratăm doar cazul ultimei reguli pentru **while**.

Fie b, c, σ, σ', σ'' ca acolo, cu $e_\sigma(b) = 1$. Notăm $w :=$ **while** b **do** c. Din ipoteza de inducție, avem $(\sigma, \sigma'') \in [\![c]\!]$ și $(\sigma'', \sigma') \in [\![w]\!]$. Vrem $(\sigma, \sigma') \in [\![w]\!]$. Dar aceasta rezultă exact din ecuația de punct fix pe care o satisface $[\![w]\!]$ (a se observa că aici nu avem nevoie de faptul că este chiar cel mai mic punct fix).

Din nou despre echivalența între semantici

Demonstrație (cont.)

Demonstrăm implicația „$\Leftarrow$”. Vom arăta, deci, prin inducție structurală după c, că, pentru orice σ, $\sigma' \in \Sigma$ cu $(\sigma, \sigma') \in [\![c]\!]$, avem $(c, \sigma) \Downarrow \sigma'$.

Din nou, cazul instrucțiunii **while** este cel mai complex. Fie b, c și notăm $w := \textbf{while } b \textbf{ do } c$. Notând $W := \{(\sigma, \sigma') \mid (w, \sigma) \Downarrow \sigma'\}$, vrem să arătăm $[\![w]\!] \subseteq W$. Cum $[\![w]\!] = \mu\Gamma^{[\![b]\!]}_{[\![c]\!]}$, e suficient să arătăm $\Gamma^{[\![b]\!]}_{[\![c]\!]}(W) \subseteq W$.

Fie $(\sigma, \sigma') \in \Gamma^{[\![b]\!]}_{[\![c]\!]}(W)$ și tratăm cazul $e_\sigma(b) = 1$. Atunci există σ'' cu $(\sigma, \sigma'') \in [\![c]\!]$ și $(\sigma'', \sigma') \in W$. Aplicând ipoteza de inducție, avem $(c, \sigma) \Downarrow \sigma''$ și $(w, \sigma'') \Downarrow \sigma'$, de unde scoatem $(w, \sigma) \Downarrow \sigma'$, adică $(\sigma, \sigma') \in W$.

Domenii

Se poate arăta ușor, prin inducție după derivări, că, pentru orice instrucțiune c și orice σ, σ', $\sigma'' \in \Sigma$ cu $(c, \sigma) \Downarrow \sigma'$ și $(c, \sigma) \Downarrow \sigma''$, avem $\sigma' = \sigma''$. Așadar, din echivalența între semantici, rezultă că $[\![c]\!]$ este graficul unei funcții parțiale, adică un element al mulțimii

$$\mathcal{R} := \{R \subseteq \Sigma^2 \mid \text{pentru orice } \sigma, \sigma', \sigma'' \in \Sigma \text{ cu } (\sigma, \sigma'), (\sigma, \sigma'') \in R, \text{ avem } \sigma' = \sigma''\}.$$

Apare întrebarea: putem defini din start semantica denotațională a unei instrucțiuni ca element al lui $\mathcal{R}$? Ideea este că, în acest caz, nu vom mai putea folosi teorema Knaster-Tarski, care se aplică (în forma dată aici) doar pentru întreaga mulțime a părților. Ca urmare, va trebui să studiem mai mult structura lui $\mathcal{R}$. Această observație stă la temelia **teoriei domeniilor**.

Mulțimi parțial ordonate

Definiție

O **mulțime parțial ordonată** (**mpo**) este o pereche $(D, \leq)$, unde $\leq \subseteq D^2$ astfel încât:

- pentru orice $x \in D$, $x \leq x$;
- pentru orice $x, y \in D$ cu $x \leq y$ și $y \leq x$, $x = y$;
- pentru orice $x, y, z \in D$ cu $x \leq y$ și $y \leq z$, $x \leq z$.

Definiție

Fie $(D, \leq)$ o mpo, $X \subseteq D$ și $x \in D$. Spunem că x este **supremum** pentru X (și notăm $x = \sup X$, el fiind unic) dacă este majorant pentru X (pentru orice $y \in X$, $y \leq x$) și, pentru orice majorant z pentru X, $x \leq z$.

Definiție

Fie $(D, \leq)$ o mpo. Un ω-lanț este un șir $(a_n)_{n \in \mathbb{N}}$ în D astfel încât, pentru orice $n \in \mathbb{N}$, $a_n \leq a_{n+1}$.

Cpo

Definiție

O mpo $(D, \leq)$ se numește **cpo** dacă are minim (notat de obicei cu $\bot$) și, pentru orice ω-lanț (a_n) în D, există $\sup\{a_n \mid n \in \mathbb{N}\}$, notat cu $\sup_{a \in \mathbb{N}} a_n$.

Exemplu: $(\mathcal{R}, \subseteq)$ este cpo (exercițiu!).

Definiție

Fie $(D, \leq)$, $(E, \leq)$ mpo și $f : D \to E$. Spunem că f este **monotonă** dacă, pentru orice x, $y \in D$ cu $x \leq y$, avem $f(x) \leq f(y)$.

Definiție

Fie $(D, \leq)$, $(E, \leq)$ cpo și $f : D \to E$. Spunem că f este ω-**continuă** dacă este monotonă și, pentru orice ω-lanț (a_n) în D, $f(\sup_{a \in \mathbb{N}} a_n) = \sup_{a \in \mathbb{N}} f(a_n)$.

Punct fix în cpo

Pentru orice mpo $(D, \leq)$ și orice $f : D \to D$, spunem că un $d \in D$ este **punct fix** al lui f dacă $f(d) = d$ și **cel mai mic punct fix** (**least fixed point**, **lfp**; dacă există, se notează cu μf) dacă e punct fix și, pentru orice punct fix e, avem $d \leq e$.

Teorema Kleene

Fie $(D, \leq)$ o cpo și $f : D \to D$ ω-continuă. Atunci f are lfp. Mai mult, pentru orice $e \in D$ cu $f(e) \leq e$, $\mu f \leq e$.

Demonstrație

Observăm că $\left(f^{(n)}(\bot)\right)_{n \in \mathbb{N}}$ este lanț (se arată prin inducție pornind de la $\bot \leq f(\bot)$). Notăm $d := \sup_{n \in \mathbb{N}} f^{(n)}(\bot)$ și avem:

$$f(d) = f(\sup_{n \in \mathbb{N}} f^{(n)}(\bot)) = \sup_{n \in \mathbb{N}} f^{(n+1)}(\bot) = \sup_{n \in \mathbb{N}} f^{(n)}(\bot) = d,$$

iar, luând un e cu $f(e) \leq e$, se arată prin inducție că, pentru orice $n \in \mathbb{N}$, $f^{(n)}(\bot) \leq e$ (pornind de la $\bot \leq e$), rezultând $d \leq e$.

Definirea semanticii

În acest moment, putem defini, analog cu tipul precedent de operator, pentru orice $h : \Sigma \to \{0, 1\}$ și orice $C \in \mathcal{R}$, o funcție $\tilde{\Gamma}^h_C : \mathcal{R} \to \mathcal{R}$ (trebuie arătat că este bine-definită – exercițiu!), despre care se arată ușor că este ω-continuă (exercițiu!). Așadar, putem defini o nouă variantă de semantică denotațională unde știm de la bun început că valorile instrucțiunilor sunt graficele unor funcții parțiale.

Se pot relua apoi ideile din demonstrația de echivalență cu semantica big-step (folosind în mod crucial acum că acel W din demonstrație este un element al lui $\mathcal{R}$, lucru care rezultă din faptul că semantica big-step produce funcții parțiale), așadar și această semantică coincide cu toate cele definite anterior.

Semantica este un morfism

Reamintim modul cum era definită semantica în logica propozițională: pentru orice $e : V \to \{0,1\}$ există și este unică $e^+ : \mathit{Form} \to \{0,1\}$ astfel încât:

- pentru orice $v \in V$, $e^+(v) = e(v)$;
- pentru orice $\varphi \in \mathit{Form}$, $e^+(\neg\varphi) = \neg e^+(\varphi)$;
- pentru orice $\varphi, \psi \in \mathit{Form}$, $e^+(\varphi \to \psi) = e^+(\varphi) \to e^+(\psi)$.

Dacă notăm, pentru orice φ și e, $[\![\varphi]\!](e) := e^+(\varphi)$, definind astfel o funcție $[\![\cdot]\!] : \mathit{Form} \to \{0,1\}^{\{0,1\}^V}$, și definind corespunzător operații $\tilde{\neg}$, $\tilde{\to}$, avem că:

- pentru orice $v \in V$, $[\![v]\!] = e \mapsto e(v)$;
- pentru orice $\varphi \in \mathit{Form}$, $[\![\neg\varphi]\!] = \tilde{\neg}[\![\varphi]\!]$;
- pentru orice $\varphi, \psi \in \mathit{Form}$, $[\![\varphi \to \psi]\!] = [\![\varphi]\!] \tilde{\to} [\![\psi]\!]$.

Vedem că semantica „scoate operații în afară”, deci are trăsături de morfism. Ne putem întreba: se poate formaliza această idee, definind anumite structuri pentru care funcțiile chiar devin morfisme? putem face aceasta și pentru semantica denotațională a lui IMP pe care tocmai am definit-o?

Signaturi algebrice

O **signatură algebrică (multisortată)** este un triplet (S, E, r), unde elementele lui S se vor numi **sorturi**, elementele lui E se vor numi **simboluri de funcție**, iar $r : E \to S^* \times S$ este **funcția de aritate** (îi notăm componentele cu r_1, r_2).

Observăm că acestea seamănă cu limbajele logicii de ordinul I, cu următoarele precizări:

- putem avea mai multe tipuri de date (sorturi) – lucru care este posibil și în logica de ordinul I, doar că, în mod tradițional, aceasta nu se face;
- nu avem simboluri de relație, ceea ce simplifică mult studiul structurilor corespunzătoare.

Algebre, morfisme

Fie (S, E, r) o signatură. Numim o **algebră** peste ea un obiect de forma $\mathcal{A} = (\{A_s\}_{s \in S}, \{A_e\}_{e \in E})$, unde, pentru orice $e \in E$, $A_e : A_{r_1(e)} \to A_{r_2(e)}$.

Dacă avem două algebre $\mathcal{A} = (\{A_s\}_{s \in S}, \{A_e\}_{e \in E})$, $\mathcal{B} = (\{B_s\}_{s \in S}, \{B_e\}_{e \in E})$, numim **morfism** de la $\mathcal{A}$ la $\mathcal{B}$ o familie de funcții $\{f_s : A_s \to B_s\}_{s \in S}$, unde, pentru orice $e \in E$, $f_{r_2(e)} \circ A_e = B_e \circ f_{r_1(e)}$. Un **izomorfism** va fi un morfism cu toate componentele bijective (echivalent: un morfism care are un morfism invers).

Cum se instanțiază acest formalism la structuri algebrice cunoscute (de exemplu, la grupuri)?

Logica privită algebric

Cum formalizăm acum logica propoziţională?

Definim signatura corespunzătoare (S, E, r) în felul următor: punem $S := \{f\}$ (avem un singur sort, al formulelor), iar $E := V \cup \{\neg, \to\}$. Avem: pentru orice $v \in V$, $r(v) = (\lambda, f)$; $r(\neg) = (f, f)$; $r(\to) = (ff, f)$. Formulele devin astfel, în mod natural, o structură algebrică notată cu $\mathcal{F}$.

Putem defini şi o algebră $\mathcal{E}$ cu $E_f := \{0,1\}^{\{0,1\}^V}$, $E_\neg := \tilde{\neg}$, $E_\to := \tilde{\to}$, iar, pentru orice $v \in V$, $E_v := e \mapsto e(v)$. Funcţia $[\![\cdot]\!]$ devine astfel, cum preconizam, un morfism (unicul!) de la $\mathcal{F}$ la $\mathcal{E}$.

Ducând aceste idei mult mai departe, ajungem la **logica algebrică**.

Algebre inițiale

Se observă și că Principiul recursiei pe formule poate fi reformulat astfel (de ce?): pentru orice algebră $\mathcal{B}$ există și este unic un morfism $F : \mathcal{F} \to \mathcal{B}$.

Aceasta ne conduce la următoarea definiție: fixând o signatură oarecare, o algebră $\mathcal{A}$ se numește **inițială** dacă, pentru orice algebră $\mathcal{B}$, există și este unic un morfism $F : \mathcal{A} \to \mathcal{B}$.

Așadar, $\mathcal{F}$ este algebră inițială. În general algebrele „de termeni" vor fi algebre inițiale, gândindu-ne profund la modul cum am demonstrat Principiul recursiei.

Proprietățile algebrelor inițiale

Propoziție

Fie $\mathcal{A}$, $\mathcal{A}'$ algebre inițiale. Atunci ele sunt izomorfe.

Demonstrație

Avem morfisme $f : \mathcal{A} \to \mathcal{A}'$ și $g : \mathcal{A}' \to \mathcal{A}$. Cum $g \circ f : \mathcal{A} \to \mathcal{A}$ și $\mathrm{id}_{\mathcal{A}} : \mathcal{A} \to \mathcal{A}$, rezultă că $g \circ f = \mathrm{id}_{\mathcal{A}}$. Analog, $f \circ g = \mathrm{id}_{\mathcal{A}'}$.

Propoziție

Fie $\mathcal{A}$, $\mathcal{A}'$ algebre izomorfe. Dacă $\mathcal{A}$ e inițială, atunci $\mathcal{A}'$ e inițială.

Demonstrație

Avem un izomorfism $f : \mathcal{A} \to \mathcal{A}'$. Fie $\mathcal{A}''$ o algebră. Vrem să arătăm că există un unic morfism $h : \mathcal{A}' \to \mathcal{A}''$. Pentru existență, știm că există un morfism $g : \mathcal{A} \to \mathcal{A}''$. Luăm $h := g \circ f^{-1}$. Pentru unicitate, fie h, $h' : \mathcal{A}' \to \mathcal{A}''$ morfisme. Atunci $h \circ f$, $h' \circ f : \mathcal{A} \to \mathcal{A}''$ sunt morfisme, deci $h \circ f = h' \circ f$. Compunând la dreapta cu f^{-1}, obținem $h = h'$.

Algebra IMP

Cum formalizăm acum limbajul IMP?

Vom pune $S := \{a, b, i\}$ (sorturi pentru expresii aritmetice, booleene și instrucțiuni), iar în E toate operațiile pe care le-am definit, pe care nu le vom enumera. Vom spune doar care este aritatea fiecărei operații care produce elemente de sort i:

- $r(\textbf{skip}) = (\lambda, i)$;
- pentru orice $X \in L$, $r(X :=) = (a, i)$;
- $r(;) = (ii, i)$;
- $r(\textbf{if}) = (bii, i)$;
- $r(\textbf{while}) = (bi, i)$.

Expresiile aritmetice, booleene și instrucțiunile, așa cum le-am definit, vor forma algebra inițială peste această signatură, algebră pe care o vom nota cu IMP.

Algebra de valori

Formalizăm acum algebra în care semantica denotațională ia valori și o notăm cu $\mathcal{A}$. Punem $A_a := \mathbb{Z}^\Sigma$, $A_b := \{0,1\}^\Sigma$ și $A_i := \mathcal{P}(\Sigma^2)$. Descriem în continuare operațiile care produc elemente de sort i:

- $A_{\textbf{skip}} := \Delta_\Sigma$;
- pt. orice $X \in L$, $j \in \mathbb{Z}^\Sigma$, $A_{X:=}(j) := \left\{\left(\sigma, \sigma_{X \mapsto j(\sigma)}\right) \mid \sigma \in \Sigma\right\}$;
- pentru orice $C_1,\ C_2 \subseteq \Sigma^2$, $A_;(C_1, C_2) := C_2 \circ C_1$;
- pentru orice $h \in \{0,1\}^\Sigma$, $C_1,\ C_2 \subseteq \Sigma^2$, $A_{\textbf{if}}(h, C_1, C_2) := \{(\sigma, \sigma') \in C_1 \mid h(\sigma) = 1\} \cup \{(\sigma, \sigma') \in C_2 \mid h(\sigma) = 0\}$;
- pentru orice $h \in \{0,1\}^\Sigma$, $C \subseteq \Sigma^2$, $A_{\textbf{while}}(h, C) := \mu\Gamma^h_C$.

Atunci semantica $[\![\cdot]\!] : \text{IMP} \to \mathcal{A}$ va fi unicul morfism, cel dat de inițialitate.

Același lucru îl putem face și cu grafice de funcții parțiale, obținând o algebră $\mathcal{B}$. Avem o funcție naturală de incluziune de la $\mathcal{B}$ la $\mathcal{A}$. Este aceasta morfism? Confirmați sau infirmați! (Nu este ușor.)