FMI, Master Sisteme Distribuite, Anul II
Semestrul II, 2024/2025

.

Procese concurente
(note de curs)

Andrei Sipoş

1 Preliminarii algebrice (parţial recapitulativ)

O privire detaliată asupra algebrei universale (multisortate), logicii ecuaţionale şi rescrierii, concentrată pe aşa-numitul *punct de vedere local*, se poate găsi în cartea [3].

1.1 Despre mulţimea vidă

Reamintim că, pentru orice A şi B, o funcţie de la A la B este un triplet $f = (A, B, R)$ unde $R \subseteq A \times B$ şi pentru orice $a \in A$ există şi este unic un $b \in B$ cu $(a, b) \in R$ şi care este notat cu $f(a)$. Sintagma „f este o funcţie de la A la B" se prescurtează prin $f : A \to B$.

Fie A o mulţime. Fie $f = (\emptyset, A, R)$ o funcţie de la $\emptyset$ la A. Atunci $R \subseteq \emptyset \times A = \emptyset$, deci $R = \emptyset$. Invers, avem că $(\emptyset, A, \emptyset)$ verifică în mod trivial condiţia din definiţia funcţiei. Aşadar, pentru orice mulţime A există şi este unică o funcţie de la $\emptyset$ la A, anume $(\emptyset, A, \emptyset)$, şi spunem că $\emptyset$ este **mulţimea iniţială**. De aceea, avem că A^0 este o mulţime cu un singur element, şi o definim ca fiind $\{\emptyset\}$.

Fie $f = (A, \emptyset, R)$ o funcţie de la A la $\emptyset$. Dacă $A = \emptyset$, atunci, din cele de mai sus, $f = (\emptyset, \emptyset, \emptyset)$. Dacă $A \neq \emptyset$, atunci, luând un $a \in A$ arbitrar, am avea că există un $b \in \emptyset$ cu $(a, b) \in R$, ceea ce este o contradicţie, dat fiind că nu există $b \in \emptyset$. Aşadar, nu există nicio funcţie de la o mulţime nevidă la mulţimea vidă.

Fie X acum o mulţime cu un element – de exemplu, putem lua $X := \{\emptyset\}$, dar alegerea precisă a lui X nu va conta.

Atunci există o unică funcţie de la A la X, anume cea care duce orice element din A în unicul element al lui X. De aceea, spunem că orice mulţime cu un singur element este **mulţime terminală**.

Invers, dacă vrem să găsim o funcţie de la X la A, trebuie doar să selectăm elementul din A în care va fi dus acel unic element al lui X. Aşadar, funcţiile de la X la A sunt în corespondenţă biunivocă cu elementele lui A.

1.2 Signaturi, termeni, ecuaţii

Definiţia 1.1. *O* **signatură (ecuaţională, monosortată)** *este o pereche* $\Sigma = (F, r)$ *unde* $r : F \to \mathbb{N}$. *Elementele lui* F *se numesc* **simbolurile** *lui* Σ. *Pentru orice* $f \in F$, $r(f)$ *se numeşte* **aritatea** *lui* f, *iar dacă aceasta este* 0, f *se numeşte* **constantă**.

Exemplul 1.2. *Considerăm signatura* $\Sigma_a = (F_a, r_a)$ *(cu care vom tot lucra), unde* $F_a = \{0, s, a, m\}$, $r_a(0) = 0$, $r_a(s) = 1$, $r_a(a) = r_a(m) = 2$.

Fie acum $\Sigma = (F, r)$ o signatură şi X o mulţime oarecare cu $X \cap F = \emptyset$. Notăm

$$S_{\Sigma,X} := X \cup F.$$

Vom defini $T_\Sigma(X)$ ca fiind o mulţime de şiruri finite cu elemente din $S_{\Sigma,X}$ (în fapt, un limbaj formal peste alfabetul $S_{\Sigma,X}$), anume ca respectând condiţiile:

- $X \subseteq T_\Sigma(X)$;
- pentru orice $f \in F$, notând $n := r(f)$, şi orice $t_1, \ldots, t_n \in T_\Sigma(X)$, avem $ft_1 \ldots t_n \in T_\Sigma(X)$,

şi ca fiind cea mai mică mulţime ce respectă aceste condiţii. Elementele lui $T_\Sigma(X)$ se vor numi **Σ-termeni** peste X. Pentru orice $t \in T_\Sigma(X)$, vom nota cu $Var(t)$ (citim „variabilele lui t") mulţimea elementelor lui X ce apar în t.

Proprietatea de a fi „cea mai mică" se poate ilustra şi sub forma unui principiu de inducţie (numită **inducţie structurală**): anume că pentru orice $B \subseteq T_\Sigma(X)$, dacă $X \subseteq B$, iar pentru orice $f \in F$, notând $n := r(f)$, şi orice $t_1, \ldots, t_n \in B$, avem $ft_1 \ldots t_n \in B$, atunci $B = T_\Sigma(X)$.

Dacă t este astfel încât $Var(t) = \emptyset$, spunem că t este **închis**. Pentru orice Σ şi X, avem că

$$\{t \in T_\Sigma(X) \mid Var(t) = \emptyset\} = T_\Sigma(\emptyset) =: T_\Sigma.$$

Pentru Σ_a de mai sus şi $X = \{x, y\}$, exemple de Σ_a-termeni peste X vor fi x, mxy (gândit informal ca $m(x, y)$), $amxyx$ (gândit informal ca $a(m(x, y), x)$). Faptul că parantezele nu sunt necesare în definirea termenilor este datorat formei prefixate şi arităţii fixe a simbolurilor.

Definiţia 1.3. *Fie Σ o signatură. O* **Σ-ecuaţie** *este un triplet (X, s, t) unde s, $t \in T_\Sigma(X)$. Ea va fi deseori notată ca $s = t$, X fiind subînţeles din context.*

Definiţia 1.4. *O* **teorie (ecuaţională)** *este o pereche (Σ, E) unde Σ este o signatură, iar E este o mulţime de Σ-ecuaţii.*

Exemplul 1.5. *O teorie ecuaţională poate fi $T_a = (\Sigma_a, E_a)$ unde*

$$E_a = \{(\{x\}, ax0, x), (\{x, y\}, axsy, saxy), (\{x\}, mx0, 0), (\{x, y\}, mxsy, amxyx)\}.$$

Ce interpretare intuitivă are T_a?

Dacă $\Sigma_1 = (F_1, r_1)$ şi $\Sigma_2 = (F_2, r_2)$ sunt două signaturi, vom nota cu $\Sigma_1 \leq \Sigma_2$ dacă $F_1 \subseteq F_2$ şi $(r_2)_{|F_1} = r_1$.

1.3 Semantica: algebre, morfisme

Definiţia 1.6. *Fie $\Sigma = (F, r)$ o signatură. O* **Σ-algebră** *este o pereche $\mathcal{A} = (A, \{A_f\}_{f \in F})$, unde pentru orice $f \in F$, avem $A_f : A^{r(f)} \to A$. A se numeşte* **universul** *sau* **mulţimea-suport** *a lui $\mathcal{A}$.*

Dacă $f \in F$ este constantă, atunci $A_f : \{\emptyset\} \to A$, deci, din cele spuse în paragrafele anterioare, putem gândi pe A_f ca fiind un element al lui A.

Exemplul 1.7. *Avem că Σ_a-algebra "canonică" este $\mathcal{N} = (\mathbb{N}, \{N_f\}_{f \in F_a})$, unde $N_0 = 0$, pentru orice $n \in \mathbb{N}$, $N_s(n) = n + 1$, iar pentru orice n, $k \in \mathbb{N}$, $N_a(n, k) = n + k$ şi $N_m(n, k) = n \cdot k$.*

În general, pentru orice signatură $\Sigma = (F, r)$ şi orice X, mulţimea $T_\Sigma(X)$ poate fi înzestrată natural cu o structură de Σ-algebră ce va fi notată tot cu $T_\Sigma(X)$ şi va fi numită **Σ-algebra de termeni peste** X – aşadar,

$$T_\Sigma(X) = (T_\Sigma(X), \{(T_\Sigma(X))_f\}_{f \in F}),$$

unde, pentru orice $f \in F$, notând $n := r(f)$, şi orice $t_1, \ldots, t_n \in T_\Sigma(X)$, avem

$$(T_\Sigma(X))_f(t_1, \ldots, t_n) = ft_1 \ldots t_n.$$

Definiţia 1.8. *Fie Σ o signatură şi $\mathcal{A} = (A, \{A_f\}_{f \in F})$, $\mathcal{B} = (B, \{B_f\}_{f \in F})$ două Σ-algebre. Numim* **morfism de la $\mathcal{A}$ la $\mathcal{B}$** *o funcţie $g : A \to B$ astfel încât pentru orice $f \in F$, notând $n := r(f)$, şi orice $a_1, \ldots, a_n \in A$,*

$$g(A_f(a_1, \ldots, a_n)) = B_f(g(a_1), \ldots, g(a_n)).$$

Se observă (exerciţiu uşor!) că dacă g este un morfism bijectiv, atunci şi g^{-1} este un morfism. În acest caz, g se numeşte **izomorfism**.

Teorema 1.9 (Proprietatea de universalitate a algebrei de termeni). *Fie Σ o signatură, X o mulţime şi $\mathcal{A} = (A, \{A_f\}_{f \in F})$ o Σ-algebră.*

Avem că pentru orice $h : X \to A$ există şi este unic un morfism $\widetilde{h}$ de la $T_\Sigma(X)$ la $\mathcal{A}$ astfel încât $\widetilde{h}_{|X} = h$, i.e. pentru orice $x \in X$, $\widetilde{h}(x) = h(x)$.

Această proprietate justifică denumirea alternativă a lui $T_\Sigma(X)$ de **Σ-algebră liberă peste** *X.*

Observaţia 1.10. *Dacă luăm $X = \emptyset$, obţinem că există şi este unic un morfism de la T_Σ la orice altă Σ-algebră. Spunem că T_Σ este* **Σ-algebra iniţială**.

Exemplul 1.11. *Cel mai frecvent caz de aplicare a teoremei, şi care ne va şi convinge intuitiv de adevărul ei, este substituţia de variabile prin termeni.*

Vom lua $\Sigma := \Sigma_a$, $X = \{x, y\}$ şi $\mathcal{A} := T_\Sigma(X)$.

Fie $h_1 : X \to T_\Sigma(X)$ cu $h_1(x) = ss0$ şi $h_1(y) = 0$. Atunci $\widetilde{h_1}(amxyx) = amss00ss0$ (gândit informal ca $a(m(ss0, 0), ss0)$).

Fie $h_2 : X \to T_\Sigma(X)$ cu $h_2(x) = ssy$ şi $h_2(y) = x$. Atunci $\widetilde{h_2}(amxyx) = amssyxssy$ (gândit informal ca $a(m(ssy, x), ssy)$).

Definiţia 1.12. *Fie Σ o signatură, (X, s, t) o Σ-ecuaţie şi $\mathcal{A} = (A, \{A_f\}_{f \in F})$ o Σ-algebră. Spunem că $\mathcal{A}$* **satisface** *(X, s, t) sau este* **model** *pentru (X, s, t), şi notăm prin*

$$\mathcal{A} \models (X, s, t),$$

dacă pentru orice $h : X \to A$ avem $\widetilde{h}(s) = \widetilde{h}(t)$.

Definiţia 1.13. *Fie $T = (\Sigma, E)$ o teorie şi $\mathcal{A} = (A, \{A_f\}_{f \in F})$ o Σ-algebră. Spunem că $\mathcal{A}$* **satisface** *T sau este* **model** *pentru T, şi notăm prin*

$$\mathcal{A} \models T,$$

dacă pentru orice $e \in E$, avem $\mathcal{A} \models e$.

Exemplul 1.14. *Avem $\mathcal{N} \models T_a$.*

Definiţia 1.15. *Fie $T = (\Sigma, E)$ o teorie şi e o Σ-ecuaţie. Spunem că din T* **se deduce semantic** *e, şi notăm prin*

$$T \models e,$$

dacă pentru orice Σ-algebră $\mathcal{A}$ cu $\mathcal{A} \models T$, avem $\mathcal{A} \models e$.

Definiţia 1.16. *Fie $\Sigma_1 = (F_1, r_1)$ şi $\Sigma_2 = (F_2, r_2)$ două signaturi cu $\Sigma_1 \leq \Sigma_2$ şi $\mathcal{A} = (A, \{A_f\}_{f \in F_2})$ o Σ_2-algebră. Notăm*

$$\mathcal{A}_{|\Sigma_1} := (A, \{A_f\}_{f \in F_1})$$

este o Σ_1-algebră şi o numim **redusa** *lui $\mathcal{A}$ la Σ_1.*

Definiţia 1.17. *Fie $T_1 = (\Sigma_1, E_1)$ şi $T_2 = (\Sigma_2, E_2)$ două teorii. Spunem că T_2 este* **extindere** *a lui T_1 şi notăm $T_1 \leq T_2$ dacă $\Sigma_1 \leq \Sigma_2$ şi $E_1 \subseteq E_2$.*

Exemplul 1.18. *Fie $\Sigma_e = (F_e, r_e)$, unde $F_e = F_a \cup \{e\}$, $(r_e)_{|F_1} = r_a$ şi $r_e(e) = 2$. Dacă luăm*

$$E_e = E_a \cup \{(\{x\}, ex0, s0), (\{x, y\}, exsy, mexyx)\},$$

atunci $T_e = (\Sigma_e, E_e)$ este o extindere a lui T_a.

Ce însemnătate are T_e? Cum arată un model al lui T_e?

Propoziţia 1.19. *Fie $T_1 = (\Sigma_1, E_1)$ şi $T_2 = (\Sigma_2, E_2)$ două teorii cu $T_1 \leq T_2$ şi $\mathcal{A}$ o Σ_2-algebră. Avem că dacă $\mathcal{A} \models T_2$, atunci $\mathcal{A}_{|\Sigma_1} \models T_1$.*

Definiţia 1.20. *Fie $\Sigma = (F, r)$ o signatură şi $\mathcal{A} = (A, \{A_f\}_{f \in F})$ o Σ-algebră. O* **relaţie de congruenţă pe** *$\mathcal{A}$ este o relaţie de echivalenţă $\sim$ pe A astfel încât pentru orice $f \in F$, notând $n := r(f)$, şi orice $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$, astfel încât pentru orice $i \in \{1, \ldots, n\}$ avem $a_i \sim b_i$, avem că*

$$A_f(a_1, \ldots, a_n) \sim A_f(b_1, \ldots, b_n).$$

Dacă $\Sigma = (F, r)$ este o signatură, $\mathcal{A} = (A, \{A_f\}_{f \in F})$ o Σ-algebră şi $\sim$ o relaţie de congruenţă pe $\mathcal{A}$, atunci mulţimea cât $A/\sim$ poate fi înzestrată cu o structură de Σ-algebră, numită Σ-algebră-cât şi notată cu $\mathcal{A}/\sim$ şi definită prin

$$\mathcal{A}/\sim := (A/\sim, \{(\mathcal{A}/\sim)_f\}_{f \in F}),$$

unde, pentru orice $f \in F$, notând $n := r(f)$, şi orice $a_1, \ldots, a_n \in A$, avem

$$(\mathcal{A}/\sim)_f(\widehat{a_1}, \ldots, \widehat{a_n}) = A_f(\widehat{a_1, \ldots, a_n}).$$

Aşadar, surjecţia canonică $\pi_\sim : A \to A/\sim$ este un morfism de la $\mathcal{A}$ la $\mathcal{A}/\sim$.

1.4 Deducţia sintactică ecuaţională

Definiţia 1.21. *Fie $T = (\Sigma, E)$ o teorie, unde $\Sigma = (F, r)$, şi e o Σ-ecuaţie. Spunem că din T* **se deduce sintactic** *e, şi notăm $T \vdash e$, dacă există $n \in \mathbb{N}$ şi $e_1, \ldots, e_n$ Σ-ecuaţii astfel încât $e_n = e$, iar pentru orice $i \in \{1, \ldots, n\}$, se întâmplă una din următoarele:*

- *$e_i \in E$;*
- *există Y, $j < i$ cu $e_j = (X, s, t)$ şi $h : X \to T_\Sigma(Y)$ astfel încât $e_i = (Y, \widetilde{h}(s), \widetilde{h}(t))$;*
- *există X şi $t \in T_\Sigma(X)$ cu $e_i = (X, t, t)$;*
- *există $j < i$ cu $e_j = (X, s, t)$ cu $e_i = (X, t, s)$;*
- *există j, $k < i$ cu $e_j = (X, s, t)$, $e_k = (X, t, u)$ şi $e_i = (X, s, u)$;*
- *există $j < i$ şi $f \in F$ astfel încât, notând $n := r(f)$, există X şi $t_1, \ldots, t_n, s \in T_\Sigma(X)$ cu $e_j = (X, t_i, s)$, iar $e_i = (X, ft_1 \ldots t_{i-1}t_it_{i+1} \ldots t_n, ft_1 \ldots t_{i-1}st_{i+1} \ldots t_n)$.*

Exemplul 1.22. *Avem că $T_a \vdash (\emptyset, as0s0, ss0)$, deoarece putem considera şirul*

$$(\{x, y\}, axsy, saxy), \quad (\emptyset, as0s0, sas00), \quad (\{x\}, ax0, x),$$

$$(\emptyset, as00, s0), \quad (\emptyset, sas00, ss0), \quad (\emptyset, as0s0, ss0).$$

Ce „reguli” sunt aplicate mai sus?

Vom putea scrie demonstraţii de genul de mai sus mai pe scurt, sub forme precum:

$$T_a \vdash as0s0 = sas00 = ss0.$$

Definiţia 1.23. *Fie $T_1 = (\Sigma_1, E_1)$ şi $T_2 = (\Sigma_2, E_2)$ două teorii. Spunem că T_2 este o* **extindere-închisă conservativă** *a lui T_1 dacă pentru orice p, $q \in T_{\Sigma_1}$, avem*

$$T_1 \vdash (\emptyset, p, q) \Leftrightarrow T_2 \vdash (\emptyset, p, q).$$

Observaţia 1.24. *Atenţie, noţiunea de mai sus* **nu** *este un caz particular al noţiunii de extindere, de aceea este folosită cratima.*

1.5 Structura termenilor aritmetici

Definiţia 1.25. *Numim un termen $u \in T_{\Sigma_a}$* **bazal** *dacă există $n \in \mathbb{N}$ cu $u = s^n0$.*

Propoziţia 1.26. *Pentru orice n, $k \in \mathbb{N}$, $T_a \vdash a(s^n0, s^k0) = s^{n+k}0$.*

Demonstraţie. Demonstrăm prin inducţie după k. Pentru $k = 0$, avem $T_a \vdash a(s^n0, 0) = s^n0$. Presupunem adevărat pentru k şi demonstrăm pentru $k + 1$: avem $T_a \vdash a(s^n0, s^{k+1}0) = s(a(s^n0, s^k0)) = ss^{n+k}0 = s^{n+k+1}0$. □

Propoziţia 1.27. *Pentru orice n, $m \in \mathbb{N}$, $T_a \vdash m(s^n0, s^k0) = s^{nk}0$.*

Teorema 1.28. *Pentru orice $t \in T_{\Sigma_a}$ există $u \in T_{\Sigma_a}$ bazal cu $T_a \vdash t = u$, i.e. există $n \in \mathbb{N}$ cu $T_a \vdash t = s^n0$.*

Demonstraţie. Demonstrăm prin inducţie structurală după t:

- dacă $t = 0$, suntem OK;
- dacă t este de forma sq, atunci avem că există n cu $T_a \vdash q = s^n 0$ şi deci $T_a \vdash t = sq = ss^n 0 = s^{n+1} 0$;
- dacă t este de forma aqr, atunci avem că există n, k cu $T_a \vdash q = s^n 0$ şi $T_a \vdash r = s^k 0$ şi deci $T_a \vdash t = aqr = a(s^n 0, s^k 0) = s^{n+k} 0$;
- dacă t este de forma mqr, atunci avem că există n, k cu $T_a \vdash q = s^n 0$ şi $T_a \vdash r = s^m 0$ şi deci $T_a \vdash t = mqr = m(s^n 0, s^k 0) = s^{nk} 0$.

□

Propoziţia 1.29. *Pentru orice p, $q \in T_{\Sigma_a}$, avem $T_a \vdash apq = aqp$.*

Demonstraţie. Avem că există n, $k \in \mathbb{N}$ cu $T_a \vdash p = s^n 0$ şi $T_a \vdash q = s^k 0$. Atunci avem

$$T_a \vdash apq = as^n 0 s^k 0 = s^{n+k} 0 = s^{k+n} 0 = as^k 0 s^n 0 = aqp.$$

□

Propoziţia 1.30. *Pentru orice p, $q \in T_{\Sigma_a}$, avem $T_a \vdash mpq = mqp$.*

1.6 Legătura dintre semantică şi deducţie sintactică

Teorema 1.31 (Teorema de corectitudine). *Fie $T = (\Sigma, E)$ o teorie. Atunci, pentru orice Σ-ecuaţie e cu $T \vdash e$, avem $T \models e$.*

Exerciţiul 1.32. *Este adevărat că $T_a \vdash (\emptyset, as0s0, s0)$?*

Demonstraţie. Nu, nu este adevărat! Presupunem că ar fi.

Cum $\mathcal{N} \models T_a$, am avea, din Teorema de corectitudine, $\mathcal{N} \models (\emptyset, as0s0, s0)$. Fie $h : \emptyset \to \mathbb{N}$ funcţia unică. Atunci avem $\widetilde{h}(as0s0) = \widetilde{h}(s0)$.

Dar aceasta înseamnă $1 + 1 = 1$, fals! □

Dacă $T = (\Sigma, E)$ este o teorie şi X este o mulţime, atunci pe $T_\Sigma(X)$ pot defini relaţia $\sim_T$ astfel: pentru orice t, $s \in T_\Sigma(X)$,

$$t \sim_T s :\Leftrightarrow T \vdash (X, t, s).$$

Atunci $\sim_T$ este o relaţie de congruenţă (exerciţiu!).

Vom nota $I_T(X) := T_\Sigma(X)/\sim_T$.

Propoziţia 1.33. $I_T(X) \models T$.

Demonstraţie. Fie $(Y, s, t) \in E$. Vrem $I_T(X) \models (Y, s, t)$, i.e. că pentru orice $h : Y \to I_T(X)$ avem $\widetilde{h}(s) = \widetilde{h}(t)$. Fie un asemenea h şi fie $j : Y \to T_\Sigma(X)$ astfel încât pentru orice $y \in Y$, $h(y) = \widehat{j(y)}$. Atunci, din unicitatea morfismului din Proprietatea de universalitate a algebrei de termeni, avem $\widetilde{h} = \pi_{\sim_t} \circ \widetilde{j}$.

Cum $(Y, s, t) \in E$, avem $T \vdash (Y, s, t)$, deci $T \vdash (X, \widetilde{j}(s), \widetilde{j}(t))$, i.e. $\widetilde{j}(s) \sim_T \widetilde{j}(t)$. Dar asta înseamnă $\pi_{\sim_T}(\widetilde{j}(s)) = \pi_{\sim_T}(\widetilde{j}(t))$, i.e. $\widetilde{h}(s) = \widetilde{h}(t)$. □

Teorema 1.34 (Teorema de completitudine). *Fie $T = (\Sigma, E)$ o teorie. Atunci, pentru orice Σ-ecuaţie e cu $T \models e$, avem $T \vdash e$.*

Demonstraţie. Considerăm $e = (X, s, t)$. Fie $h : X \to I_T(X)$ astfel încât, pentru orice $x \in X$, $h(x) = \widehat{x}$. Atunci, din unicitatea morfismului din Proprietatea de universalitate a algebrei de termeni, avem $\widetilde{h} = \pi_{\sim_T}$.

Cum, din propoziţia anterioară, avem $I_T(X) \models T$, rezultă că $I_T(X) \models (X, s, t)$, în particular că $\widetilde{h}(s) = \widetilde{h}(t)$, deci (din paragraful anterior) că $\widehat{s} = \widehat{t}$, i.e. $s \sim_T t$. Dar aceasta înseamnă că $T \vdash (X, s, t)$, ceea ce trebuia demonstrat. □

Definiţia 1.35. *Fie $T = (\Sigma, E)$ o teorie şi $\mathcal{A}$ o Σ-algebră. Spunem că $\mathcal{A}$ este* **închis-completă** *pentru T dacă pentru orice p, $q \in T_\Sigma$ cu $\mathcal{A} \models (\emptyset, p, q)$ avem că $T \vdash (\emptyset, p, q)$ (sau, echivalent, $T \models (\emptyset, p, q)$).*

1.7 Rescrierea termenilor

Definiţia 1.36. *Fie Σ o signatură. O* **Σ-regulă de rescriere** *este o Σ-ecuaţie (X, t_1, t_2) astfel încât $t_1 \notin X$ şi $Var(t_2) \subseteq Var(t_1)$. Ea va fi deseori notată ca $t_1 \to t_2$, X fiind subînţeles din context.*

Definiţia 1.37. *Numim* **sistem de rescriere de termeni** *(***term rewriting system**, **TRS***) o pereche (Σ, R) unde Σ este o signatură şi R este o mulţime de Σ-reguli de rescriere.*

Fie (Σ, R) un TRS şi X o mulţime. Definim $\to_R$ ca fiind cea mai mică submulţime a lui $T_\Sigma(X) \times T_\Sigma(X)$ (i.e. o relaţie binară pe $T_\Sigma(X)$) ce verifică următoarele condiţii:

- pentru orice $(Y, s, t) \in R$ şi orice $h : Y \to T_\Sigma(X)$, avem $\widetilde{h}(s) \to_R \widetilde{h}(t)$;
- pentru orice $h : X \to T_\Sigma(X)$ şi orice s, t cu $s \to_R t$, avem $\widetilde{h}(s) \to_R \widetilde{h}(t)$;
- pentru orice $f \in F$, notând $n := r(f)$, pentru orice $t_1, \ldots, t_n, s \in T_\Sigma(X)$ şi i cu $t_i \to_R s$, avem

$$ft_1 \ldots t_{i-1}t_it_{i+1} \ldots t_n \to_R ft_1 \ldots t_{i-1}st_{i+1} \ldots t_n.$$

Notăm cu $\to_R^*$ închiderea reflexiv-tranzitivă a lui $\to_R$, i.e. cea mai mică relaţie binară reflexivă şi tranzitivă pe $T_\Sigma(X)$ ce include pe $\to_R$. Se observă că pentru orice s, $t \in T_\Sigma(X)$, avem $s \to_R^* t$ dacă şi numai dacă există $n \in \mathbb{N}^*$ şi $t_1, \ldots, t_n \in T_\Sigma(X)$ cu $t_1 = s$, $t_n = t$, iar pentru orice $i < n$, $t_i \to_R t_{i+1}$.

Definiţia 1.38. *Fie (Σ, R) un TRS şi X o mulţime. Spunem că $t \in T_\Sigma(X)$ este o* **formă normală** *a lui (Σ, R) dacă nu există s cu $t \to_R s$.*

Definiţia 1.39. *Fie (Σ, R) un TRS, X o mulţime şi $t \in T_\Sigma(X)$. Spunem că $u \in T_\Sigma(X)$ este o* **formă normală** *pentru t dacă u este o formă normală a lui (Σ, R) şi $t \to_R^* u$.*

Definiţia 1.40. *Un TRS (Σ, R) se numeşte* **noetherian** *dacă nu admite rescrieri infinite, i.e. nu există X şi $(t_n)_{n\in\mathbb{N}} \subseteq T_\Sigma(X)$ astfel încât pentru orice $n \in \mathbb{N}$, $t_n \to_R t_{n+1}$.*

Teorema 1.41. *Fie (Σ, R) un TRS noetherian şi X o mulţime. Atunci pentru orice $t \in T_\Sigma(X)$ există $u \in T_\Sigma(X)$ formă normală pentru t.*

Demonstraţie. Presupunem că ar exista un t fără formă normală. Vom construi atunci $(t_n)_{n\in\mathbb{N}} \subseteq T_\Sigma(X)$ astfel încât pentru orice $n \in \mathbb{N}$, $t_n \to_R t_{n+1}$, contrazicând ipoteza de noetherianitate. Punem $t_0 := t$. Presupunem că am construit $t_0, \ldots, t_n$ astfel încât pentru orice $i < n$, $t_i \to_R t_{i+1}$, deci $t_0 \to_R^* t_n$. Dacă t_n este o formă normală, atunci este o formă normală pentru t, ceea ce contrazice presupunerea. Aşadar există s cu $t_n \to_R s$ şi vom pune $t_{n+1} := s$. □

Exerciţiul 1.42. *Să se arate că sistemul de rescriere dat de unica regulă $ffx \to fgfx$ (unde f şi g sunt simboluri de aritate 1) este noetherian.*

Demonstraţie. Observăm că la orice aplicare a regulii numărul de f-uri din termen urmate de alt f scade cu 1. □

Exerciţiul 1.43. *Să se arate că sistemul de rescriere dat de unica regulă $fgx \to ggffx$ (unde f şi g sunt simboluri de aritate 1) nu este noetherian.*

Demonstraţie. Arătam că orice termen de forma $f^m g^n t$ cu:

$$m \geq 1, \quad n \geq 1, \quad [m \geq 2 \text{ sau } n \geq 2]$$

se rescrie într-unul de lungime mai lungă (aceasta este evident) ce admite un subtermen de aceeaşi formă. Aşadar, obţinem un şir infinit de rescrieri.

Avem că $f^m g^n t \to f^{m-1} g^2 f^2 g^{n-1} t$. Notăm $s := f^2 g^{n-1} t$.

Dacă $m \geq 2$, luăm subtermenul $f^{m-1} g^2 s$ (termenul însuşi).

Dacă $n \geq 2$, luăm subtermenul s. □

Definiţia 1.44. *Un TRS (Σ, R) se numeşte* **confluent** *dacă pentru orice X şi orice t_1, t_2, $t_3 \in T_\Sigma(X)$ cu $t_1 \to_R^* t_2$ şi $t_1 \to_R^* t_3$ există $t_4 \in T_\Sigma(X)$ cu $t_2 \to_R^* t_4$ şi $t_3 \to_R^* t_4$.*

Teorema 1.45. *Fie (Σ, R) un TRS confluent şi X o mulţime. Atunci pentru orice $t \in T_\Sigma(X)$ şi orice u, $v \in T_\Sigma(X)$ forme normale pentru t, avem $u = v$.*

Demonstraţie. Din ipoteza de confluenţă, rezultă că există w cu $u \to_R^* w$ şi $v \to_R^* w$. Dar, cum u şi v sunt forme normale, rezultă $u = w = v$. □

Aşadar, dacă un TRS este noetherian şi confluent, orice termen t admite exact o formă normală, ce va fi notată cu $\mathrm{nf}(t)$.

2 Sisteme de tranziţie etichetate

Reamintim de la cursul de Limbaje formale şi automate că, pentru orice mulţime L, notăm cu L^* mulţimea cuvintelor peste L, iar cu $\lambda \in L^*$ notăm cuvântul vid.

2.1 Introducere. Bisimulare

Definiţia 2.1. *Un* **sistem de tranziţie etichetat** *(***labelled transition system, LTS***) este un tuplu $(S, L, \to, \downarrow)$ unde $\to \subseteq S \times L \times S$ şi $\downarrow \subseteq S$. Pentru orice $(p, a, q) \in S \times L \times S$, notăm $(p, a, q) \in \to$ cu $p \to^a q$. Pentru orice $(p, a) \in S \times L$, notăm faptul că nu există $q \in S$ cu $p \to^a q$ prin $p \not\to^a$. Pentru orice $p \in S$, notăm faptul că $p \in \downarrow$ cu $p \downarrow$ şi faptul că $p \notin \downarrow$ cu $p \not\downarrow$.*

Definiţia 2.2. *Fie $(S, L, \to, \downarrow)$ un LTS. Definim $\twoheadrightarrow \subseteq S \times L^* \times S$ ca fiind cea mai mică mulţime astfel încât avem:*

- *pentru orice $s \in S$, avem $(s, \lambda, s) \in \twoheadrightarrow$,*
- *pentru orice s, t, $u \in S$, $w \in L^*$, $a \in L$ cu $(s, w, t) \in \twoheadrightarrow$ şi $t \to^a u$, avem $(s, wa, u) \in \twoheadrightarrow$,*

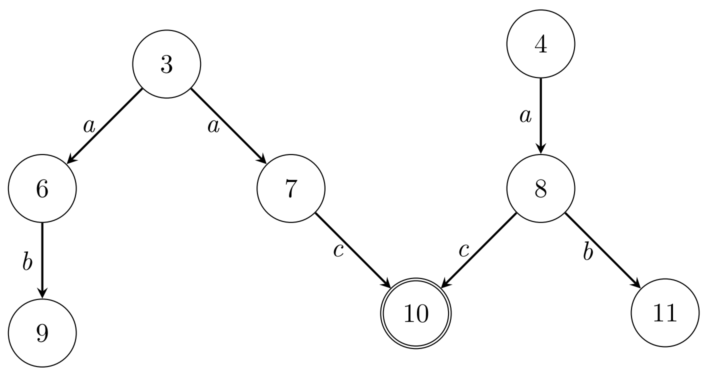
i.e. $\twoheadrightarrow$ este închiderea reflexiv-tranzitivă a lui $\to$.

Pentru orice $(p, w, q) \in S \times L^* \times S$, notăm $(p, w, q) \in \twoheadrightarrow$ cu $p \twoheadrightarrow^w q$. Pentru orice p, $q \in S$, spunem că q este **accesibil** *din p dacă există $w \in L^*$ cu $p \twoheadrightarrow^w q$.*

Dacă $\mathcal{S} = (S, L, \to, \downarrow)$ este un LTS şi $s \in S$, notăm

$$\mathcal{L}_s(\mathcal{S}) = \{w \in L^* \mid \text{există } t \downarrow \text{ cu } s \twoheadrightarrow^w t\}.$$

Exemplul 2.3. *Fie următorul LTS, notat cu $\mathcal{S}$:*



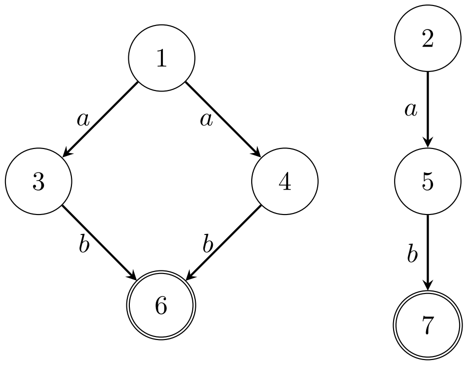
Avem că $\mathcal{L}_3(\mathcal{S}) = \mathcal{L}_4(\mathcal{S}) = \{ac\}$.

Definiţia 2.4. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. O* **relaţie de bisimulare** *pe $\mathcal{S}$ este o relaţie binară R pe S, i.e. $R \subseteq S \times S$, astfel încât:*

- *pentru orice s, t, $s' \in S$, $a \in L$ cu $(s, t) \in R$ şi $s \to^a s'$, există $t' \in S$ cu $t \to^a t'$ şi $(s', t') \in R$;*
- *pentru orice s, t, $t' \in S$, $a \in L$ cu $(s, t) \in R$ şi $t \to^a t'$, există $s' \in S$ cu $s \to^a s'$ şi $(s', t') \in R$;*
- *pentru orice s, $t \in S$ cu $(s, t) \in R$ şi $s \downarrow$, avem $t \downarrow$;*

- *pentru orice s, $t \in S$ cu $(s,t) \in R$ şi $t\downarrow$, avem $s\downarrow$.*

Fie următorul LTS, notat cu $\mathcal{S}$:



Atunci $R = \{(1,2),(3,5),(4,5),(6,7)\}$ este o relaţie de bisimulare pe $\mathcal{S}$.

Exerciţiul 2.5. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. Atunci:*

- *$\Delta_S = \{(s,s) \mid s \in S\} \subseteq S \times S$ este o relaţie de bisimulare pe $\mathcal{S}$;*
- *dacă R este o relaţie de bisimulare pe $\mathcal{S}$, atunci $R^{\mathrm{op}} = \{(s,r) \mid (r,s) \in R\}$ este o relaţie de bisimulare pe $\mathcal{S}$;*
- *dacă R_1 şi R_2 sunt relaţii de bisimulare pe $\mathcal{S}$, atunci*

$$R_2 \circ R_1 = \{(r,t) \mid \text{ există } s \text{ cu } (r,s) \in R_1 \text{ şi } (s,t) \in R_2\}$$

  este o relaţie de bisimulare pe $\mathcal{S}$.

Definiţia 2.6. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. Definim*

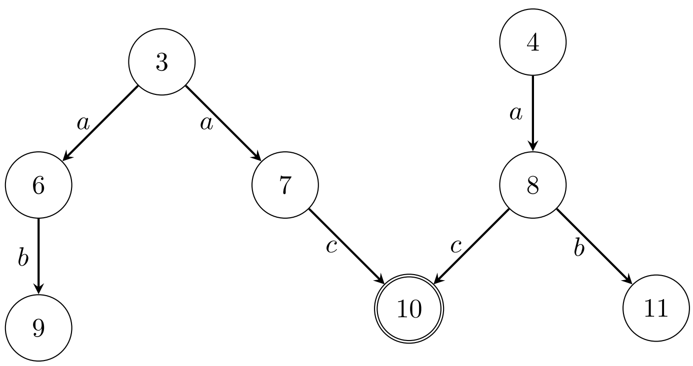
$$B_{\mathcal{S}} := \bigcup_{R \text{ este relaţie de bisimulare pe } \mathcal{S}} R$$

şi o numim **relaţia de bisimilaritate** *pe $\mathcal{S}$.*

Propoziţia 2.7. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. Atunci $B_{\mathcal{S}}$ este o relaţie de bisimulare pe $\mathcal{S}$ (şi este, deci, cea mai mare asemenea relaţie).*

Definiţia 2.8. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. Pentru orice s, $t \in S$, notăm prin $s \underline{\leftrightarrow} t$ faptul că $(s,t) \in B_{\mathcal{S}}$, i.e. că există o relaţie de bisimulare R pe $\mathcal{S}$ cu $(s,t) \in R$. Spunem că s şi t sunt* **bisimilare**.

Reluăm acum primul exemplu dat de LTS:

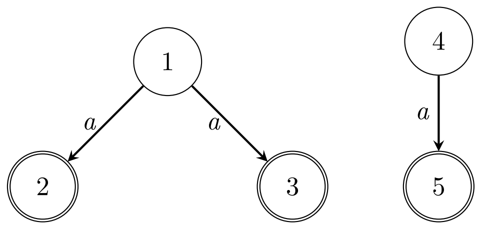


În acest LTS, avem că $3 \not\underline{\leftrightarrow} 4$. De ce? Presupunem că am avea $3 \underline{\leftrightarrow} 4$. Atunci există o relaţie R de bisimulare pe $\mathcal{S}$ cu $(3,4) \in R$. Cum $4 \to^a 8$, există s' cu $3 \to^a s'$ şi $(s',8) \in R$. Deci $(6,8) \in R$ sau

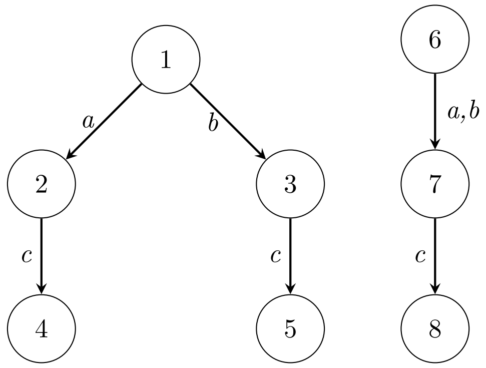
$(7,8) \in R$. Presupunem că $(6,8) \in R$. Cum $8 \rightarrow^c 10$, există s' cu $6 \rightarrow^c s'$ şi $(s', 10) \in R$. Contradicţie cu $6 \not\rightarrow^c$! Presupunem acum că $(7,8) \in R$. Cum $8 \rightarrow^b 11$, există s' cu $7 \rightarrow^b s'$ şi $(s', 11) \in R$. Contradicţie cu $7 \not\rightarrow^b$!

Exerciţiul 2.9. *Să se decidă dacă:*

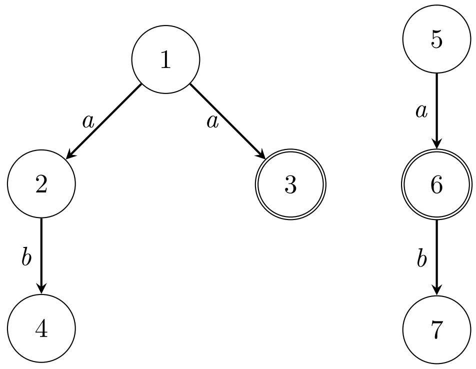
- $1 \underline{\leftrightarrow} 4$ *în LTS-ul*



- $1 \underline{\leftrightarrow} 6$ *în LTS-ul*



- $1 \underline{\leftrightarrow} 5$ *în LTS-ul*



Teorema 2.10. *Fie $\mathcal{S} = (S, L, \rightarrow, \downarrow)$ un LTS. Atunci $B_{\mathcal{S}}$ este o relaţie de echivalenţă.*

Demonstraţie. Pentru a arăta reflexivitatea, ne folosim de faptul că Δ_S este o relaţie de bisimulare pe $\mathcal{S}$ şi deci $\Delta_S \subseteq B_{\mathcal{S}}$.

Pentru simetrie, fie $s, t \in S$ cu $s \underline{\leftrightarrow} t$. Atunci există R relaţie de bisimulare pe $\mathcal{S}$ cu $(s,t) \in R$ şi deci $(t,s) \in R^{\text{op}}$, care este şi ea o relaţie de bisimulare. Ca urmare, $t \underline{\leftrightarrow} s$.

Similar, pentru tranzitivitate, folosim faptul că o compunere a două relaţii de bisimulare este tot o relaţie de bisimulare. $\square$

Definiţia 2.11. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. Spunem că $s \in S$ se numeşte* **deadlock** *dacă $s \not\downarrow$ şi, pentru orice $a \in L$, $s \not\to^a$. Spunem că $s \in S$* **are deadlock** *dacă există $u \in S$ deadlock accesibil din s; în caz contrar, s se numeşte* **liber de deadlock**.

Exerciţiul 2.12. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. Fie $s, t \in S$ cu $s \underline{\leftrightarrow} t$. Să se arate că s are deadlock dacă şi numai dacă t are.*

Demonstraţie. Din simetria problemei, este suficient să arătăm implicaţia „$\Rightarrow$".

Presupunem că s are deadlock, i.e. există $u \in S$ deadlock şi $w \in L^*$ cu $s \twoheadrightarrow^w u$. Fie R relaţie de bisimulare pe $\mathcal{S}$ cu $(s, t) \in R$. Deducem (prin inducţie după lungimea lui w) că există un v cu $(u, v) \in R$ şi $t \twoheadrightarrow^w v$. Rezultă că v este deadlock, deoarece, cum u este deadlock, avem $u \not\downarrow$, deci $v \not\downarrow$, şi apoi că pentru orice $a \in L$, $u \not\to^a$, deci $v \not\to^a$. □

Definiţia 2.13. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS. El se numeşte* **regulat** *dacă S şi $\to$ sunt finite (v. limbajele regulate de la cursul de Limbaje formale şi automate).*

Definiţia 2.14. *Fie $\mathcal{S} = (S, L, \to, \downarrow)$ un LTS şi $s \in S$. Definim*

$$S_s := \{t \in S \mid t \text{ este accesibil din } s\}$$

şi

$$\mathcal{S}_s := (S_s, L, \to \cap (S_s \times L \times S_s), \downarrow \cap S_s).$$

2.2 Semantica operaţională

Definiţia 2.15. *Fie $\Sigma = (F, r)$ o signatură şi L o mulţime.*

Dacă X e o mulţime, numim o **formulă** *peste Σ, L, X o expresie de forma $s \to^a t$ cu $s, t \in T_\Sigma(X)$ şi $a \in L$ sau o expresie de forma $t \downarrow$ cu $t \in T_\Sigma(X)$.*

Numim o **regulă de deducţie** *peste Σ, L un triplet (X, Φ, ψ) (notat deseori doar prin perechea (Φ, ψ)) astfel încât Φ este mulţime de formule peste Σ, L, X, iar ψ este formulă peste Σ, L, X.*

O regulă de deducţie peste Σ, L sub forma (X, Φ, ψ) se numeşte **în format cale** *(path) dacă:*

- *pentru orice $s \to^a t \in \Phi$, $t \in X$;*
- *dacă ψ este de forma $s \to^a t$, atunci fie $s \in X$, fie există $f \in F$ şi $x_1, \ldots, x_{r(f)} \in X$ toate distincte astfel încât $s = f x_1 \ldots x_{r(f)}$;*
- *t-urile de la primul punct sunt toate diferite şi niciunul dintre ele nu apare în s de la cel de-al doilea punct.*

Numim un **sistem de deducţie** *peste Σ, L o mulţime R de reguli peste Σ, L.*

Un sistem de deducţie peste Σ, L se numeşte **în format cale** *dacă fiecare dintre elementele sale este în format cale.*

Dacă R este un sistem de deducţie peste Σ, L, definim

$$\mathcal{S}_R := (T_\Sigma, L, \to, \downarrow),$$

LTS-ul indus de *R, unde $\to$ şi $\downarrow$ sunt cele mai mici mulţimi cu proprietatea că pentru orice $(X, \Phi, \psi) \in R$ şi orice $h : X \to T_\Sigma$, dacă avem:*

- *pentru orice $s \to^a t \in \Phi$, $\widetilde{h}(s) \to^a \widetilde{h}(t)$;*
- *pentru orice $s \downarrow \in \Phi$, $\widetilde{h}(s) \downarrow$*

atunci:

- *dacă ψ este de forma $s \to^a t$, avem $\widetilde{h}(s) \to^a \widetilde{h}(t)$;*
- *dacă ψ este de forma $s \downarrow$, $\widetilde{h}(s) \downarrow$.*

Teorema 2.16 (Baeten/Verhoef, 1993)**.** *Fie Σ o signatură, L o mulţime, iar R un sistem de deducţie peste Σ, L în format cale.*

Atunci $B_{\mathcal{S}_R}$ este o relaţie de congruenţă pe Σ-algebra T_Σ.

În restul acestei secţiuni, vom ilustra aceste noţiuni pe un exemplu concret. Vom construi un sistem de deducţie peste Σ_a şi $\{1\}$ format din următoarele reguli (notând $\to^1$ mai simplu cu $\to$):

- Reguli referitoare la zero şi succesor:

$$(\emptyset, 0\downarrow), \quad (\emptyset, sx \to x).$$

  În general, interpretarea intuitivă a unui $t \in T_{\Sigma_a}$ va fi că t „reprezintă un $n \in \mathbb{N}$" dacă se poate ajunge de la el la o stare finală în exact n paşi.

- Reguli referitoare la adunare:

$$(\{y \to y'\}, axy \to axy'), \quad (\{x \to x', y\downarrow\}, axy \to x'), \quad (\{x\downarrow, y\downarrow\}, axy\downarrow).$$

  Exemple de aplicare a acestor reguli: $0\downarrow$, $s0 \to 0$, $as00 \to 0$, $as0s0 \to as00$.

- Reguli referitoare la înmulţire:

$$(\{x \to x', y \to y'\}, mxy \to amxy'x'), \quad (\{x\downarrow\}, mxy\downarrow), \quad (\{y\downarrow\}, mxy\downarrow).$$

Se observă că sistemul de deducţie este în format cale, deci $\underline{\leftrightarrow}$ este o relaţie de congruenţă pe T_{Σ_a}. Notăm Σ_a-algebra-cât cu $M_a := T_{\Sigma_a}/\underline{\leftrightarrow}$.

Ce vom arăta până la sfârşitul secţiunii va fi că pentru orice t_1, $t_2 \in T_{\Sigma_a}$, avem că $t_1 \underline{\leftrightarrow} t_2$ dacă şi numai dacă $T_a \vdash t_1 = t_2$.

Teorema 2.17. $M_a \models T_a$.

Demonstraţie. Vom exemplifica demonstraţia pe una dintre axiomele lui T_a (restul rămânând ca exerciţiu), anume vom arăta că $M_a \models axsy = saxy$, i.e. că pentru orice p, $q \in T_{\Sigma_a}$, avem $apsq \underline{\leftrightarrow} sapq$.

Este suficient să furnizăm o relaţie de bisimulare R astfel încât pentru orice $p, q \in T_{\Sigma_a}$, $(apsq, sapq) \in R$. Fie $R := \{(apsq, sapq) \mid p,\ q \in T_{\Sigma_a}\} \cup \{(p,p) \mid p \in T_{\Sigma_a}\}$.

Rămâne de arătat că R este o relaţie de bisimulare. Este suficient să arătăm acest lucru pentru primul tip de elemente ale lui R. Fie, deci, p, $q \in T_{\Sigma_a}$. Ne uităm la cele patru condiţii din definiţia bisimulării:

- Dacă avem $r \in T_{\Sigma_a}$ cu $apsq \to r$, atunci r trebuie să fie apq (de ce?). Avem şi că $sapq \to apq$, iar $(apq, apq) \in R$, deci suntem OK.
- Dacă avem $r \in T_{\Sigma_a}$ cu $sapq \to r$, atunci r trebuie să fie apq (de ce?). Avem şi că $apsq \to apq$, iar $(apq, apq) \in R$, deci, din nou, suntem OK.
- Dacă avem $apsq\downarrow$, atunci ar trebui să avem $sq\downarrow$, ceea ce este imposibil.
- Dacă avem $sapq\downarrow$, atunci din nou obţinem o contradicţie.

$\square$

Corolarul 2.18. *Pentru orice t_1, $t_2 \in T_{\Sigma_a}$ cu $T_a \vdash t_1 = t_2$, avem că $t_1 \underline{\leftrightarrow} t_2$.*

Lema 2.19. *Pentru orice $p \in T_{\Sigma_a}$ cu $p\downarrow$, avem $T_a \vdash p = 0$.*

Demonstraţie. Demonstrăm prin inducţie structurală după p:

- dacă $p = 0$, atunci clar $T_a \vdash p = 0$;
- dacă $p = sq$, atunci nu putem avea $p\downarrow$;
- dacă $p = aqr$, atunci avem $q\downarrow$ şi $r\downarrow$ - prin inducţie avem $T_a \vdash q = 0$ şi $T_a \vdash r = 0$ şi deci $T_a \vdash aqr = a00 = 0$;
- dacă $p = mqr$, atunci fie $q\downarrow$, fie $r\downarrow$:
  - dacă $q\downarrow$, atunci $T_a \vdash q = 0$ şi deci $T_a \vdash mqr = mrq = mr0 = 0$;
  - dacă $r\downarrow$, atunci $T_a \vdash r = 0$ şi deci $T_a \vdash mqr = mq0 = 0$.

$\square$

Lema 2.20. *Pentru orice p, $p' \in T_{\Sigma_a}$ cu $p \to p'$, avem $T_a \vdash p = sp'$.*

Demonstraţie. Demonstrăm prin inducţie structurală după p:

- dacă $p = 0$, atunci $p \not\to$;
- dacă $p = sq$, atunci $p' = q$, aşadar $T_a \vdash p = sq = sp'$;
- dacă $p = aqr$, atunci:
  - fie $p' = aqr'$ şi $r \to r'$, caz în care prin inducţie avem $T_a \vdash r = sr'$ şi deci $T_a \vdash p = aqr = aqsr' = saqr' = sp'$;
  - fie $q \to p'$ şi $r \downarrow$, caz în care prin inducţie avem $T_a \vdash q = sp'$ şi din lema anterioară $T_a \vdash r = 0$ şi deci $T_a \vdash p = aqr = a(sp')0 = sp'$;
- dacă $p = mqr$ – exerciţiu!

$\square$

Teorema 2.21. *Pentru orice p, $q \in T_{\Sigma_a}$ cu $p \underline{\leftrightarrow} q$, avem că $T_a \vdash p = q$.*

Demonstraţie. Din cele arătate mai demult, există termeni bazali p_1, q_1 cu $T_a \vdash p = p_1$ şi $T_a \vdash q = q_1$. Din cele de mai devreme, avem $p \underline{\leftrightarrow} p_1$ şi $q \underline{\leftrightarrow} q_1$, deci $p_1 \underline{\leftrightarrow} q_1$ şi este suficient să arătăm că $T_a \vdash p_1 = q_1$.

Demonstrăm prin inducţie structurală după p_1 şi distingem doar două cazuri, cum p_1 este bazal:

- dacă $p_1 = 0$, atunci $p_1 \downarrow$, iar cum $p_1 \underline{\leftrightarrow} q_1$, avem că $q_1 \downarrow$, deci $T_a \vdash q_1 = 0$ şi deci $T_a \vdash p_1 = q_1$;
- dacă $p_1 = sr$, atunci $p_1 \to r$ şi deci, cum $p_1 \underline{\leftrightarrow} q_1$, există $t \in T_{\Sigma_a}$ cu $r \underline{\leftrightarrow} t$ şi $q_1 \to t$, deci şi t este tot bazal: din ipoteza de inducţie, avem că $T_a \vdash r = t$, iar cum $q_1 \to t$, avem că $T_a \vdash q_1 = st$, deci $T_a \vdash p_1 = sr = st = q_1$ şi suntem OK.

$\square$

3 Teoria elementară a proceselor

Fixăm, pentru aceasta şi următoarele secţiuni, o mulţime A – elementele sale se vor numi **acţiuni**.

3.1 $\mathsf{MPT}(A)$ – Minimal Process Theory

Descriem în continuare una dintre cele mai simple teorii posibile pentru descrierea proceselor.

Signatura $\Sigma_{\mathsf{MPT},A}$ conţine o constantă 0 (ce semnifică, informal, un proces în stare de deadlock), o operaţie binară $+$ (ce semnifică o alegere între procesele reprezentate de operanzii săi) şi, pentru fiecare $a \in A$, câte o operaţie unară notată prin $a.$ (astfel încât, pentru orice proces x, $a.x$ semnifică executarea acţiunii a urmată de executarea procesului x).

De exemplu, dacă $a \in A$, atunci avem $a.0 \in T_{\Sigma_{\mathsf{MPT},A}}$.

Vom nota operaţia $+$ în mod infixat, pentru claritate, de exemplu vom scrie $p + 0$ în loc de $+p0$, iar de aceea vom folosi şi paranteze (chiar dacă ele de fapt nu există, tehnic vorbind). Vom mai nota, pentru orice x (inclusiv un x definibil doar mai târziu) şi orice $a \in A$, $a^0x := x$, iar pentru orice $n \in \mathbb{N}$, $a^{n+1}x := a.(a^nx)$.

Definim mulţimea $E_{\mathsf{MPT},A}$ ca fiind formată din următoarele ecuaţii (scrise aici informal):

- $x + y = y + x$;
- $(x + y) + z = x + (y + z)$;
- $x + x = x$;
- $x + 0 = x$.

Definim acum $\mathsf{MPT}(A) := (\Sigma_{\mathsf{MPT},A}, E_{\mathsf{MPT},A})$.

Dăm acum un exemplu de raţionament informal în $\mathsf{MPT}(A)$:

$$a.x + (b.y + a.x) = a.x + (a.x + b.y) = (a.x + a.x) + b.y = a.x + b.y.$$

Aşadar, pentru orice a, $b \in A$, avem

$$\mathsf{MPT}(A) \vdash a.x + (b.y + a.x) = a.x + b.y.$$

Exerciţiul 3.1. *În prezenţa celei de-a treia axiome, primele două axiome sunt echivalente cu*

$$(x + y) + z = (y + z) + x.$$

Demonstraţie. Clar, date fiind primele două axiome, avem

$$(x + y) + z = x + (y + z) = (y + z) + x.$$

Acum, dată fiind axioma $(x + y) + z = (y + z) + x$, avem

$$x+y = (x+x)+y = (x+y)+x = (y+x)+x = ((y+y)+x)+x = ((y+x)+y)+x = (y+x)+(y+x) = y+x$$

şi

$$(x + y) + z = (y + z) + x = x + (y + z).$$

□

Exerciţiul 3.2. *Fie X o mulţime oarecare. Pe $T_{\Sigma_{\mathsf{MPT},A}}(X)$ definim relaţia $\leq$, pentru orice t_1, $t_2 \in T_{\Sigma_{\mathsf{MPT},A}}(X)$, prin*

$$t_1 \leq t_2 :\Leftrightarrow \mathsf{MPT}(A) \vdash t_1 + t_2 = t_2.$$

Să se arate următoarele:

1. *Pentru orice t_1, $t_2 \in T_{\Sigma_{\mathsf{MPT},A}}(X)$, $t_1 \leq t_2$ dacă şi numai dacă există $t_3 \in T_{\Sigma_{\mathsf{MPT},A}}(X)$ cu*

$$\mathsf{MPT}(A) \vdash t_1 + t_3 = t_2.$$

2. *Relaţia $\leq$ este o relaţie de ordine parţială.*

Demonstraţie. Raţionăm peste tot în $\mathsf{MPT}(A)$. Avem:

1. Într-un sens, luăm $t_3 := t_2$. În celălalt, dacă avem t_3 cu $t_1 + t_3 = t_2$, atunci

$$t_1 + t_2 = t_1 + (t_1 + t_3) = (t_1 + t_1) + t_3 = t_1 + t_3 = t_2.$$

2. Pentru orice t, avem $t + t = t$, deci $t \leq t$ şi $\leq$ este reflexivă. Fie acum t_1 şi t_2 cu $t_1 \leq t_2$ şi $t_2 \leq t_1$. Atunci

$$t_1 = t_2 + t_1 = t_1 + t_2 = t_2,$$

deci $\leq$ este antisimetrică. Fie t_1, t_2, t_3 cu $t_1 \leq t_2$ şi $t_2 \leq t_3$. Atunci

$$t_1 + t_3 = t_1 + (t_2 + t_3) = (t_1 + t_2) + t_3 = t_2 + t_3 = t_3.$$

□

Exemplul 3.3. *Pe universul $\mathbb{N}$, dacă interpretăm 0 prin 0, iar $+$ prin operaţia de maxim, toate axiomele $\mathsf{MPT}(A)$ sunt satisfăcute, indiferent de interpretările operaţiilor corespunzătoare acţiunilor. Obţinem aşadar o clasă largă de modele ale lui $\mathsf{MPT}(A)$.*

Exerciţiul 3.4. *Fie $a \in A$. Atunci $\mathsf{MPT}(A) \not\vdash a.(x + y) = a.x + a.y$.*

Demonstraţie. În tipul de structură exemplificat mai sus, interpretăm $a.$ prin operaţia care duce 0 în 1, 1 în 0, iar restul numerelor sunt lăsate pe loc. Atunci

$$a.(0+1) = a.1 = 0 \neq 1 = 1 + 0 = a.0 + a.1.$$

□

onsiderăm sistemul de deducţie peste $\Sigma_{\mathsf{MPT},A}$ şi A, format din următoarele reguli (pentru orice $a \in A$):

- $(\emptyset, a.x \to^a x)$;
- $(\{x \to^a x'\}, x + y \to^a x')$;
- $(\{y \to^a y'\}, x + y \to^a y')$.

Cum sistemul de deducţie este în format cale, avem că relaţia de bisimilaritate $\underline{\leftrightarrow}$ pe LTS-ul indus de el, ce are ca mulţime de stări pe $T_{\Sigma_{\mathsf{MPT},A}}$, este o congruenţă pe $T_{\Sigma_{\mathsf{MPT},A}}$. Notăm $\Sigma_{\mathsf{MPT},A}$-algebra-cât cu $M_{\mathsf{MPT},A} := T_{\Sigma_{\mathsf{MPT},A}}/\underline{\leftrightarrow}$. Elementele sale (precum şi cele ale algebrelor asemănătoare ce se vor construi în subsecţiunile şi secţiunile următoare) se vor numi **procese**.

Teorema 3.5. *Această algebră este într-adevăr un model al lui* $\mathsf{MPT}(A)$*, i.e.* $M_{\mathsf{MPT},A} \models \mathsf{MPT}(A)$.

Demonstraţie. Pentru simplitate, demonstrăm doar satisfacerea axiomei de element neutru, $x + 0 = x$. Aşadar, ce trebuie arătat este că pentru orice $p \in T_{\Sigma_{\mathsf{MPT},A}}$, avem $p + 0\underline{\leftrightarrow}p$.

Este suficient să arătăm că

$$R := \{(p + 0, p) \mid p \in T_{\Sigma_{\mathsf{MPT},A}}\} \cup \{(p, p) \mid p \in T_{\Sigma_{\mathsf{MPT},A}}\}$$

este o relaţie de bisimulare pe $T_{\Sigma_{\mathsf{MPT},A}}$. De asemenea, este suficient să arătăm condiţiile de bisimulare pentru primul tip de perechi.

Fie, aşadar, $p \in T_{\Sigma_{\mathsf{MPT},A}}$. Presupunem că avem $a \in A$ şi $p' \in T_{\Sigma_{\mathsf{MPT},A}}$ cu $p+0 \to^a p'$. Atunci, conform regulilor de deducţie, această tranziţie ar fi putut fi dedusă doar din $p \to^a p'$, iar $(p', p') \in R$. Invers, dacă avem că există $a \in A$ şi $p' \in T_{\Sigma_{\mathsf{MPT},A}}$ cu $p \to^a p'$, conform regulilor de deducţie avem $p + 0 \to^a p'$ şi din nou $(p', p') \in R$. □

Corolarul 3.6. *Fie p, $q \in T_{\Sigma_{\mathsf{MPT},A}}$ cu* $\mathsf{MPT}(A) \vdash p = q$*. Atunci $p\underline{\leftrightarrow}q$.*

În continuare, vom demonstra (analog cu secţiunea precedentă) reciproca acestui corolar.

Lema 3.7. *Fie p, $p' \in T_{\Sigma_{\mathsf{MPT},A}}$ şi $a \in A$ cu $p \to^a p'$. Atunci* $\mathsf{MPT}(A) \vdash p = a.p' + p$.

Demonstraţie. Demonstrăm prin inducţie după p.

Dacă $p = 0$, atunci, cum $0 \not\to^a$, am terminat.

Dacă p este de forma $b.p''$, atunci, conform regulilor de deducţie, trebuie să avem $b = a$ şi $p'' = p'$, deci $p = a.p'$. În $\mathsf{MPT}(A)$ deducem:

$$p = p + p = a.p' + p.$$

Dacă p este de forma $p_1 + p_2$, atunci tranziţia din ipoteză trebuie să fi fost dedusă fie din $p_1 \to^a p'$, fie din $p_2 \to^a p'$. Fără a restrânge generalitatea, presupunem $p_1 \to^a p'$. Din ipoteza de inducţie, avem $\mathsf{MPT}(A) \vdash p_1 = a.p' + p_1$. Aşadar, în $\mathsf{MPT}(A)$ deducem:

$$p = p_1 + p_2 = (a.p' + p_1) + p_2 = a.p' + (p_1 + p_2) = a.p' + p.$$

□

Lema 3.8. *Fie p, q, $r \in T_{\Sigma_{\mathsf{MPT},A}}$ cu $(p + q) + r\underline{\leftrightarrow}r$. Atunci $p + r\underline{\leftrightarrow}r$ şi $q + r\underline{\leftrightarrow}r$.*

Demonstraţie. Exerciţiu! □

Teorema 3.9. *Fie p, $q \in T_{\Sigma_{\mathsf{MPT},A}}$ cu $p\underline{\leftrightarrow}q$. Atunci* $\mathsf{MPT}(A) \vdash p = q$.

Demonstraţie. Vom demonstra că pentru orice p, $q \in T_{\Sigma_{\mathsf{MPT},A}}$, avem că dacă $p + q \underline{\leftrightarrow} q$, atunci $\mathsf{MPT}(A) \vdash p + q = q$ şi, similar, că dacă $p \underline{\leftrightarrow} p + q$, atunci $\mathsf{MPT}(A) \vdash p = p + q$.

De ce ne rezolvă aceste afirmaţii teorema? Dacă avem, ca în ipoteză, $p \underline{\leftrightarrow} q$, atunci, din faptul că $\underline{\leftrightarrow}$ este congruenţă, avem $p = p + p \underline{\leftrightarrow} p + q$ şi $p + q \underline{\leftrightarrow} q + q = q$, deci, din cele două afirmaţii, deducem $\mathsf{MPT}(A) \vdash p + q = q$ şi $\mathsf{MPT}(A) \vdash p = p + q$, de unde scoatem $\mathsf{MPT}(A) \vdash p = q$.

Demonstrăm, deci, cele două afirmaţii (a doua, cum am zis, este similară primeia şi de aceea nu indicăm şi argumentele demonstrării ei) prin inducţie completă după **numărul total de simboluri** din p şi q.

Pasul de bază este $p = q = 0$. Dar este imediat atunci că $\mathsf{MPT}(A) \vdash 0+0 = 0$, deci $\mathsf{MPT}(A) \vdash p+q = q$.

Pentru pasul de inducţie, distingem trei cazuri, după forma lui p.

Dacă $p = 0$, atunci în $\mathsf{MPT}(A)$ deducem $p + q = 0 + q = q + 0 = q$.

Dacă p este de forma $a.p'$, atunci avem $p \to^a p'$ şi deci $p + q \to^a p'$. Cum $p + q \underline{\leftrightarrow} q$, există q' cu $q \to^a q'$ şi $p' \underline{\leftrightarrow} q'$. Din prima lemă, avem că $\mathsf{MPT}(A) \vdash q = a.q' + q$. Cum $p' \underline{\leftrightarrow} q'$, avem ca mai înainte că $p' + q' \underline{\leftrightarrow} q'$ şi $q' + p' \underline{\leftrightarrow} p'$, deci, din ipoteza de inducţie, $\mathsf{MPT}(A) \vdash p' + q' = q'$ şi $\mathsf{MPT}(A) \vdash q' + p' = p'$. Atunci, în $\mathsf{MPT}(A)$, deducem:

$$p' = q' + p' = p' + q' = q'$$

şi

$$p + q = a.p' + q = a.q' + q = q.$$

Dacă p este de forma $p_1 + p_2$, deci avem $(p_1 + p_2) + q \underline{\leftrightarrow} q$, deci, din lema anterioară, avem că $p_1 + q \underline{\leftrightarrow} q$ şi $p_2 + q \underline{\leftrightarrow} q$. Din ipoteza de inducţie, avem $\mathsf{MPT}(A) \vdash p_1 + q = q$ şi $\mathsf{MPT}(A) \vdash p_2 + q = q$. Atunci, în $\mathsf{MPT}(A)$, deducem:

$$p + q = (p_1 + p_2) + q = p_1 + (p_2 + q) = p_1 + q = q.$$

$\square$

3.2 $\mathsf{BSP}(A)$ – Basic Sequential Processes

Vom spune că $\Sigma_{\mathsf{BSP},A}$ este $\Sigma_{\mathsf{MPT},A}$ la care adăugăm constanta 1 (ce va reprezenta o terminare cu succes, spre deosebire de 0, ce reprezenta un deadlock), iar $E_{\mathsf{BSP},A}$ va fi acelaşi cu $E_{\mathsf{MPT},A}$.

Definim acum $\mathsf{BSP}(A) := (\Sigma_{\mathsf{BSP},A}, E_{\mathsf{BSP},A})$.

Teorema 3.10. $\mathsf{BSP}(A)$ *este o extindere-închisă conservativă a lui* $\mathsf{MPT}(A)$*, i.e. pentru orice* p*,* $q \in T_{\Sigma_{\mathsf{MPT},A}}$*, avem*

$$\mathsf{MPT}(A) \vdash p = q \Leftrightarrow \mathsf{BSP}(A) \vdash p = q.$$

Demonstraţie. Direcţia „$\Rightarrow$" este evidentă. Pentru „$\Leftarrow$", fie p, $q \in T_{\Sigma_{\mathsf{MPT},A}}$ cu $\mathsf{BSP}(A) \vdash p = q$, deci $\mathsf{BSP}(A) \models p = q$. Vrem $\mathsf{MPT}(A) \vdash p = q$, i.e. $\mathsf{MPT}(A) \models p = q$. Fie $\mathcal{A}$ o $\Sigma_{\mathsf{MPT},A}$-algebră cu $\mathcal{A} \models \mathsf{MPT}(A)$. Iau $\mathcal{B}$ o $\Sigma_{\mathsf{BSP},A}$-algebră care este expansiunea lui $\mathcal{A}$ la $\Sigma_{\mathsf{BSP},A}$ interpretând pe 1 la fel ca pe 0. Atunci $\mathcal{B} \models \mathsf{BSP}(A)$ şi deci $\mathcal{B} \models p = q$. Avem, deci, $\mathcal{A} \models p = q$ şi am terminat. $\square$

Considerăm sistemul de deducţie peste $\Sigma_{\mathsf{BSP},A}$ şi A, ce extinde pe cel definit în secţiunea anterioară cu următoarele reguli:

- $(\emptyset, 1\downarrow)$;
- $(\{x\downarrow\}, (x + y)\downarrow)$;
- $(\{y\downarrow\}, (x + y)\downarrow)$.

Pentru orice $x \in T_{\Sigma_{\mathsf{BSP},A}}$, gândim $x\downarrow$ ca însemnând faptul că procesul reprezentat de x se termină cu succes.

Cum sistemul de deducţie este în format cale, avem că relaţia de bisimilaritate $\underline{\leftrightarrow}$ pe LTS-ul indus de el, ce are ca mulţime de stări pe $T_{\Sigma_{\mathsf{BSP},A}}$, este o congruenţă pe $T_{\Sigma_{\mathsf{BSP},A}}$. Notăm $\Sigma_{\mathsf{BSP},A}$-algebra-cât cu $M_{\mathsf{BSP},A} := T_{\Sigma_{\mathsf{BSP},A}}/\underline{\leftrightarrow}$.

Teorema 3.11. *Această algebră este un model al lui* $\mathsf{BSP}(A)$*, i.e.* $M_{\mathsf{BSP},A} \models \mathsf{BSP}(A)$*.*

Teorema 3.12. *Fie* p*,* $q \in T_{\Sigma_{\mathsf{BSP},A}}$*. Atunci* $p \underline{\leftrightarrow} q$ *dacă şi numai dacă* $\mathsf{BSP}(A) \vdash p = q$*.*

Teorema 3.13. *Fie $p \in T_{\Sigma_{\mathsf{BSP},A}}$. Atunci p este liber de deadlock în LTS-ul indus dacă şi numai dacă există $q \in T_{\Sigma_{\mathsf{BSP},A}}$ în care nu apare 0 astfel încât* $\mathsf{BSP}(A) \vdash p = q$.

Demonstraţie. De la dreapta la stânga, cum $\mathsf{BSP}(A) \vdash p = q$, atunci $p \underline{\leftrightarrow} q$, iar cum q este liber de deadlock, avem că şi p este.

De la stânga la dreapta, demonstrăm prin inducţie după p.

Dacă $p = 0$, atunci p este deadlock şi nu avem, deci, nimic de demonstrat.

Dacă $p = 1$, atunci p este liber de deadlock, şi putem lua $q := 1$.

Dacă p este de forma $a.p'$, atunci, cum singura tranziţie din p este $p \to^a p'$, avem că p' este liber de deadlock, deci (din ipoteza de inducţie) există $q' \in T_{\Sigma_{\mathsf{BSP},A}}$ în care nu apare 0 astfel încât $\mathsf{BSP}(A) \vdash p' = q'$. Trebuie demonstrat că există $q \in T_{\Sigma_{\mathsf{BSP},A}}$ în care nu apare 0 astfel încât $\mathsf{BSP}(A) \vdash p = q$. Avem, raţionând în $\mathsf{BSP}(A)$, că

$$p = a.p' = a.q'$$

şi deci putem lua $q := a.q'$.

Luăm acum p de forma $p_1 + p_2$. Cum $p = p_1 + p_2$ este liber de deadlock, avem (din regulile de deducţie) şi p_1, şi p_2 sunt libere de deadlock sau unul dintre ele este, iar celălalt este o sumă de zerouri. Avem, în primul caz, că există q_1, q_2 ce nu conţin 0 cu $\mathsf{BSP}(A) \vdash p_1 = q_1$ şi $\mathsf{BSP}(A) \vdash p_2 = q_2$ şi deci putem lua $q := q_1 + q_2$, iar în al doilea caz, că există q_1 ce nu conţine 0 cu $\mathsf{BSP}(A) \vdash p_1 = q_1$ şi deci putem lua $q := q_1$. □

3.3 $(\mathsf{BSP{+}PR})(A)$

Definim $\Sigma_{\mathsf{BSP+PR},A}$ adăugând la $\Sigma_{\mathsf{BSP},A}$, pentru fiecare $n \in \mathbb{N}$, un operator unar π_n. Dacă x reprezintă un proces şi $n \in \mathbb{N}$, atunci $\pi_n x$ va însemna intuitiv comportamentul lui x în primii n paşi.

Definim $E_{\mathsf{BSP+PR},A}$ adăugând la $E_{\mathsf{BSP},A}$ următoarele **scheme** de axiome, pentru orice $n \in \mathbb{N}$ şi $a \in A$ (după caz):

- $\pi_n(1) = 1$;
- $\pi_n(0) = 0$;
- $\pi_0(a.x) = 0$;
- $\pi_{n+1}(a.x) = a.\pi_n(x)$;
- $\pi_n(x + y) = \pi_n(x) + \pi_n(y)$.

Definim acum $(\mathsf{BSP{+}PR})(A) := (\Sigma_{\mathsf{BSP+PR},A}, E_{\mathsf{BSP+PR},A})$.

Exemplu de raţionament în $(\mathsf{BSP{+}PR})(A)$:

$$\pi_2(a.(a.0 + b.c.1)) = a.\pi_1(a.0 + b.c.1) = a.(\pi_1(a.0) + \pi_1(b.c.1)) = a.(a.\pi_0(0) + b.\pi_0(c.1)) = a.(a.0 + b.0).$$

Acest exemplu ne conduce la următorul rezultat.

Propoziţia 3.14. *Pentru orice $p \in T_{\Sigma_{\mathsf{BSP+PR},A}}$ există $q \in T_{\Sigma_{\mathsf{BSP},A}}$ cu* $(\mathsf{BSP{+}PR})(A) \vdash p = q$.

Demonstraţie. Demonstrăm prin inducţie completă după **lungimea** lui p, observând şi că, la fiecare pas, obţinem un termen de lungime mai scurtă. Distingem mai multe cazuri decât de obicei.

Dacă p este 0 sau 1, suntem OK.

Dacă p este de forma $p_1 + p_2$ sau $a.p_1$, scriem p_1 şi p_2 ca q_1, respectiv q_2 şi scriem respectiv pe p ca $q_1 + q_2$, $a.q_1$.

Dacă p este de forma $\pi_n(0)$, $\pi_n(1)$ sau $\pi_0(a.p_1)$, îl scriem respectiv ca 0, 1, 0.

Dacă p este de forma $\pi_n(p_1 + p_2)$, $\pi_{n+1}(a.p_1)$ sau $\pi_n(\pi_m(p_1))$, scriem $\pi_n(p_1)$, $\pi_n(p_2)$ şi $\pi_m(p_1)$ ca q_1, q_2, respectiv q_3 şi scriem respectiv pe p ca $q_1 + q_2$, $a.q_1$, $\pi_n(q_3)$, iar ultimul dintre ele are lungimea mai scurtă şi deci putem aplica ipoteza de inducţie.

Se observă că am acoperit toate cazurile posibile. □

Teorema 3.15. $(\mathsf{BSP{+}PR})(A)$ *este o extindere-închisă conservativă a lui* $\mathsf{BSP}(A)$*, i.e. pentru orice p, $q \in T_{\Sigma_{\mathsf{BSP},A}}$, avem*

$$\mathsf{BSP}(A) \vdash p = q \Leftrightarrow (\mathsf{BSP{+}PR})(A) \vdash p = q.$$

Exerciţiul 3.16. *Fie a, b, c, $d \in A$ şi $p = a.(b.1 + c.d.0) + a.b.1 \in T_{\Sigma_{\mathsf{BSP+PR},A}}$. Pentru orice $k \in \mathbb{N}$, să se găsească $q \in T_{\Sigma_{\mathsf{BSP},A}}$ cu $(\mathsf{BSP+PR})(A) \vdash \pi_k p = q$.*

Demonstraţie. Raţionând în $(\mathsf{BSP+PR})(A)$, avem:

$$\begin{aligned}
\pi_0(p) &= 0 + 0 = 0\\
\pi_1(p) &= a.0 + a.0 = a.0\\
\pi_2(p) &= a.(b.1 + c.0) + a.b.1\\
\pi_3(p) &= a.(b.1 + c.d.0) + a.b.1 = p\\
\text{pentru orice } k \geq 3,\ \pi_k(p) &= p.
\end{aligned}$$

□

Acest exerciţiu ne conduce la următorul rezultat.

Teorema 3.17. *Pentru orice $p \in T_{\Sigma_{\mathsf{BSP+PR},A}}$, există $n \in \mathbb{N}$ astfel încât pentru orice $k \geq n$,*

$$(\mathsf{BSP+PR})(A) \vdash \pi_k p = p.$$

Demonstraţie. Din propoziţia demonstrată anterior, este suficient să arătăm enunţul pentru orice $p \in T_{\Sigma_{\mathsf{BSP},A}}$.

Dacă $p = 0$ sau $p = 1$, luăm $n = 0$.

Dacă p este de forma $a.q$, atunci, luând m numărul corespunzător pentru q, putem lua $n := m + 1$, deoarece pentru orice $k \geq n$, avem $k - 1 \geq m$ şi deci, raţionând în $\mathsf{BSP}(A)$,

$$\pi_k p = \pi_k(a.q) = a.\pi_{k-1} q = a.q = p.$$

Dacă p este de forma $p_1 + p_2$, atunci, luând m_1 numărul corespunzător pentru p_1 şi m_2 numărul corespunzător pentru p_2, putem lua $n := \max(m_1, m_2)$, deoarece pentru orice $k \geq n$, avem $k \geq m_1$ şi $k \geq m_2$ şi deci, raţionând în $\mathsf{BSP}(A)$,

$$\pi_k p = \pi_k(p_1 + p_2) = \pi_k(p_1) + \pi_k(p_2) = p_1 + p_2 = p.$$

□

Considerăm sistemul de deducţie peste $\Sigma_{\mathsf{BSP+PR},A}$ şi A, ce extinde pe cel definit în secţiunea anterioară cu următoarele reguli (pentru orice $a \in A$ şi orice $n \in \mathbb{N}$):

- $(\{x \downarrow\}, \pi_n x \downarrow)$;
- $(\{x \to^a x'\}, \pi_{n+1} x \to^a \pi_n x')$.

Cum sistemul de deducţie este în format cale, avem că relaţia de bisimilaritate $\underline{\leftrightarrow}$ pe LTS-ul indus de el, ce are ca mulţime de stări pe $T_{\Sigma_{\mathsf{BSP+PR},A}}$, este o congruenţă pe $T_{\Sigma_{\mathsf{BSP+PR},A}}$. Notăm $\Sigma_{\mathsf{BSP+PR},A}$-algebra-cât cu $M_{\mathsf{BSP+PR},A} := T_{\Sigma_{\mathsf{BSP+PR},A}}/\underline{\leftrightarrow}$.

Teorema 3.18. *Această algebră este un model al lui $(\mathsf{BSP+PR})(A)$, i.e. $M_{\mathsf{BSP+PR},A} \models (\mathsf{BSP+PR})(A)$.*

Teorema 3.19. *Fie p, $q \in T_{\Sigma_{\mathsf{BSP+PR},A}}$. Atunci $p \underline{\leftrightarrow} q$ dacă şi numai dacă $(\mathsf{BSP+PR})(A) \vdash p = q$.*

3.4 $\mathsf{BSP^*}(A)$

Definim $\Sigma_{\mathsf{BSP^*},A}$ adăugând la $\Sigma_{\mathsf{BSP},A}$, pentru fiecare $a \in A$, un operator unar a^*. Dacă x reprezintă un proces şi $a \in A$, atunci $a^* x$ va însemna intuitiv executarea lui a de un număr finit, arbitrar, de ori urmată de executarea lui x.

Definim $E_{\mathsf{BSP^*},A}$ adăugând la $E_{\mathsf{BSP},A}$ următoarele scheme de axiome, pentru orice $a \in A$:

- $a.(a^* x) + x = a^* x$;
- $a^*(a^* x) = a^* x$.

Definim acum $\mathsf{BSP*}(A) := (\Sigma_{\mathsf{BSP*},A}, E_{\mathsf{BSP*},A})$.

Considerăm sistemul de deducţie peste $\Sigma_{\mathsf{BSP*},A}$ şi A, ce extinde pe cel definit în secţiunea despre $\mathsf{BSP}(A)$ cu următoarele reguli (pentru orice a, $b \in A$):

- $(\emptyset, a^*x \to^a a^*x)$;
- $(\{x \to^b x'\}, a^*x \to^b x')$;
- $(\{x \downarrow\}, a^*x \downarrow)$.

Cum sistemul de deducţie este în format cale, avem că relaţia de bisimilaritate $\underline{\leftrightarrow}$ pe LTS-ul indus de el, ce are ca mulţime de stări pe $T_{\Sigma_{\mathsf{BSP*},A}}$, este o congruenţă pe $T_{\Sigma_{\mathsf{BSP*},A}}$. Notăm $\Sigma_{\mathsf{BSP*},A}$-algebra-cât cu $M_{\mathsf{BSP*},A} := T_{\Sigma_{\mathsf{BSP*},A}}/\underline{\leftrightarrow}$.

Teorema 3.20. *Această algebră este un model al lui* $\mathsf{BSP*}(A)$*, i.e.* $M_{\mathsf{BSP*},A} \models \mathsf{BSP*}(A)$.

Teorema 3.21. *Fie* p, $q \in T_{\Sigma_{\mathsf{BSP*},A}}$. *Atunci* $p\underline{\leftrightarrow}q$ *dacă şi numai dacă* $\mathsf{BSP*}(A) \vdash p = q$.

Teorema 3.22. $\mathsf{BSP*}(A)$ *este o extindere-închisă conservativă a lui* $\mathsf{BSP}(A)$*, i.e. pentru orice* p, $q \in T_{\Sigma_{\mathsf{BSP},A}}$*, avem*

$$\mathsf{BSP}(A) \vdash p = q \Leftrightarrow \mathsf{BSP*}(A) \vdash p = q.$$

3.5 $(\mathsf{BSP*+PR})(A)$

Definim $\Sigma_{\mathsf{BSP*+PR},A}$ şi $E_{\mathsf{BSP*+PR},A}$ în modul cel mai natural cu putinţă, iar apoi punem

$$(\mathsf{BSP*+PR})(A) := (\Sigma_{\mathsf{BSP*+PR},A}, E_{\mathsf{BSP*+PR},A}).$$

Definiţia 3.23. *Un* $p \in T_{\Sigma_{\mathsf{BSP*+PR},A}}$ *se numeşte* **de adâncime mărginită** *(bounded depth) dacă există* $n \in \mathbb{N}$ *astfel încât pentru orice* $k \geq n$,

$$(\mathsf{BSP*+PR})(A) \vdash \pi_k p = p.$$

Exerciţiul 3.24. *Fie* $a \in A$. *Pentru orice* $n \in \mathbb{N}$, *avem* $(\mathsf{BSP*+PR})(A) \vdash \pi_n(a^*0) = a^n0$.

Demonstraţie. Raţionăm peste tot în $(\mathsf{BSP*+PR})(A)$. Demonstrăm prin inducţie după n.

Pentru $n = 0$, avem

$$\pi_n(a^*0) = \pi_0(a^*0) = \pi_0(a.a^*0 + 0) = \pi_0(a.a^*0) + \pi_0(0) = 0 + 0 = 0 = a^00.$$

Presupunem adevărat pentru n şi demonstrăm pentru $n + 1$. Atunci avem

$$\pi_{n+1}(a^*0) = \pi_{n+1}(a.a^*0 + 0) = \pi_{n+1}(a.a^*0) + \pi_{n+1}(0) = a.\pi_n(a^*0) + 0 = a.\pi_n(a^*0) = a.(a^n0) = a^{n+1}0.$$

□

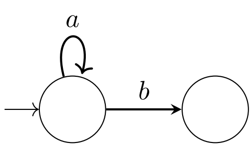
4 Procese recursive

4.1 Motivaţie

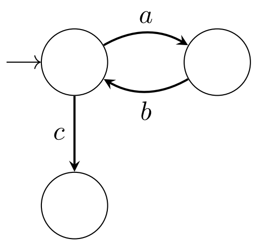
În secţiunea anterioară, am introdus teoria $\mathsf{BSP*}(A)$, în care puteam vorbi, pentru fiecare $a \in A$, despre un operator unar a^*, şi reamintim că acest operator era guvernat de o ecuaţie de forma

$$a^*x = x + a.(a^*x).$$

De exemplu, dacă a, $b \in A$, termenul $a^*(b.0)$ făcea referire la un sistem de tranziţie de forma următoare:



Dacă avem a, b, $c \in A$, şi vrem să modelăm un sistem de tranziţie precum următorul:



atunci aceşti operatori nu mai sunt suficienţi. Fireşte, am putea să introducem pentru fiecare cuvânt $w \in A^*$ un nou „operator" w^*, guvernat de o ecuaţie de forma

$$w^*x = x + w.(w^*x),$$

unde notăm, dacă $w = a_1 \dots a_n$, prin $w.x$ termenul $a_1.(\dots(a_n.x)\dots)$. Atunci sistemul de tranziţie de mai sus ar fi descris de termenul

$$X := (ab)^*(c.0).$$

Nu vom face acest lucru, deoarece vom introduce o soluţie chiar mai generală de atât. Totuşi, exemplul de mai sus ne va conduce spre acea soluţie. Dacă aplicăm ecuaţia definitorie a acestui „operator" termenului X, obţinem

$$X = c.0 + (ab).((ab)^*(c.0)) = c.0 + (ab).X = c.0 + a.b.X,$$

i.e., mai pe scurt,

$$X = c.0 + a.b.X.$$

Observăm că X apare în ambii membri ai ecuaţiei, şi de aceea putem numi o asemenea ecuaţie „recursivă". Despre acest tip de ecuaţii vom vorbi în continuare.

4.2 Ecuaţii şi specificaţii recursive

Fie $\Sigma = (F, r)$ o signatură şi W o mulţime cu $W \cap F = \emptyset$, ale cărei elemente se vor numi **variabile recursive** şi se vor nota de obicei cu litere mari. O **ecuaţie recursivă** peste Σ şi W este o ecuaţie de forma $X = t$ unde $X \in W$ şi $t \in T_\Sigma(W)$. O **specificaţie recursivă** peste Σ şi W este o mulţime R de ecuaţii recursive peste Σ şi W astfel încât pentru orice $X \in W$ există şi este unică o ecuaţie din R al cărei membru stâng este X.

Vom da acum exemple, pornind de la signatura $\Sigma_{\mathsf{BSP},A}$ şi considerând a, b, c, $d \in A$:

- $W_1 = \{X\}$, $R_1 = \{X = c.0 + a.b.X\}$ (exemplul dat în motivaţie);
- $W_2 = \{X\}$, $R_2 = \{X = a.(b.1 + c.d.0)\}$ (vedem că membrul drept poate să nu conţină variabile recursive);
- $W_3 = \{X, Y\}$, $R_3 = \{X = a.Y, Y = b.X + a.Y\}$;
- $W_4 = \{X_n \mid n \in \mathbb{N}\}$, $R_4 = \{X_0 = a.X_1\} \cup \{X_{n+1} = a.X_{n+2} + b.X_n \mid n \in \mathbb{N}\}$ (atât W_4, cât şi R_4 sunt mulţimi infinite, numărabile).

Fie R o specificaţie recursivă peste $\Sigma_{\mathsf{BSP},A}$ şi un W corespunzător.

Notăm cu $\Sigma^W_{\mathsf{BSP},A}$ extinderea lui $\Sigma_{\mathsf{BSP},A}$ adăugând elementele lui W pe post de constante, iar cu $E^R_{\mathsf{BSP},A}$ mulţimea de $\Sigma^W_{\mathsf{BSP},A}$-ecuaţii care se obţine adăugând la $E_{\mathsf{BSP},A}$ elementele lui R.

Notăm apoi $(\mathsf{BSP}{+}R)(A) := (\Sigma^W_{\mathsf{BSP},A}, E^R_{\mathsf{BSP},A})$.

De exemplu, să zicem că avem $W = \{X, Y\}$ şi $R = \{X = a.Y, Y = b.X\}$. Atunci, raţionând în $(\mathsf{BSP}{+}R)(A)$, avem că

$$X = a.Y = a.(b.X)$$

şi apoi

$$X = a.(b.X) = a.(b.(a.(b.X))).$$

Dacă vom vrea să lucrăm cu mai multe specificaţii recursive, unde, să zicem, am notat variabile din specificaţii diferite cu acelaşi simbol, vom putea dezambiguiza acele variabile folosind notaţii precum

$$\mu X.R,$$

semnificând „acel X din specificaţia R".

Vom nota cu $\mathsf{BSP}_{\mathsf{rec}}(A) = (\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}, E_{\mathsf{BSP}_{\mathsf{rec}},A})$ teoria care se obţine din $\mathsf{BSP}(A)$ adăugând **toate** specificaţiile recursive posibile (cu oarecare grijă, ca să nu intrăm în situaţii de genul paradoxului lui Russell: de exemplu, putem fixa de la început o mulţime numărabilă de variabile şi doar peste submulţimi ale ei să construim specificaţii), folosind notaţia de mai sus pentru dezambiguizare între variabile (şi similar vom avea variante recursive pentru toate teoriile de procese din secţiunea precedentă). În mod similar, vom defini extensii ale lui $(\mathsf{BSP}+\mathsf{PR})(A)$ – adică $(\mathsf{BSP}+\mathsf{PR})_{\mathsf{rec}}(A)$ – şi ale altor teorii de procese.

Dat fiind că în teoriile de procese operaţia $+$ este asociativă şi comutativă, putem folosi notaţia $\sum$ pentru a aduna elementele dintr-o mulţime finită dată.

4.3 Exemplu: descrierea unei stive

Fie D o mulţime finită ce va reprezenta tipul de date ce pot fi elemente ale unei stive.

Pentru orice $w \in D^*$ introducem o variabilă S_w ce va reprezenta stiva ce va conţine cuvântul (şirul finit) w.

Vom considera şi că pentru orice $d \in D$ avem acţiunile $\mathrm{push}(d), \mathrm{pop}(d) \in A$.

Definim specificaţia recursivă formată din ecuaţia

$$S_\lambda = 1 + \sum_{d \in D} \mathrm{push}(d).S_d$$

şi, pentru orice $d \in D$ şi $w \in D^*$,

$$S_{dw} = \mathrm{pop}(d).S_w + \sum_{e \in D} \mathrm{push}(e).S_{edw}.$$

În particular, pentru orice $d \in D$, avem

$$S_d = \mathrm{pop}(d).S_\lambda + \sum_{e \in D} \mathrm{push}(e).S_{ed}.$$

Se observă că au loc următoarele egalităţi:

$$\pi_0(S_\lambda) = 1 + 0 = 1.$$

$$\pi_1(S_\lambda) = 1 + \sum_{d \in D} \mathrm{push}(d).\pi_0(S_d) = 1 + \sum_{d \in D} \mathrm{push}(d).0.$$

$$\pi_2(S_\lambda) = 1 + \sum_{d \in D} \mathrm{push}(d).\pi_1(S_d) = 1 + \sum_{d \in D} \mathrm{push}(d).\left(\mathrm{pop}(d).1 + \sum_{e \in D} \mathrm{push}(e).0\right).$$

4.4 Soluţii

Fie $T = (\Sigma, E)$ o teorie, W o mulţime corespunzătoare (de variabile recursive), iar R o specificaţie recursivă peste Σ şi W. Definim, prin generalizare a notaţiilor prezentate anterior, Σ^W ca fiind extinderea lui Σ adăugând elementele lui W pe post de constante, iar E^R ca fiind mulţimea de Σ^W-ecuaţii care se obţine adăugând la E elementele lui R. Notăm apoi $T^R := (\Sigma^W, E^R)$. Fie $\mathcal{M}$ o Σ-algebră care satisface T.

În acest cadru, spunem că o **soluţie** pentru R în $\mathcal{M}$ este o Σ^W-algebră $\mathcal{M}'$ ce extinde pe $\mathcal{M}$ astfel încât $\mathcal{M}' \models T^R$.

În continuare, vom da câteva exemple.

Dacă lucrăm cu teoria T_a, punând $W = \{X\}$ şi $R = \{X = X\}$. Atunci Σ_a^W va fi Σ_a la care adăugăm pe X pe post de constantă, iar E_a^R va fi E_a la care adăugăm ecuaţia $X = X$.

Ştim că $\mathcal{N} \models T_a$. A găsi o soluţie pentru R în $\mathcal{N}$ este echivalent cu a găsi un $n \in \mathbb{N}$ (o interpretare pentru X) astfel încât $n = n$. Vedem că orice $n \in \mathbb{N}$ satisface aceasta, aşadar soluţia nu este unică.

Dacă aveam $R = \{X = X + X\}$, aveam o singură soluţie, dată de interpretarea lui X prin 0.

Dacă aveam $R = \{X = X + s0\}$, atunci nu existau soluţii.

Lucrăm acum cu teoria $(\mathsf{BSP}+\mathsf{PR})(A)$. Luăm $W = \{X\}$ şi $R = \{X = a.X\}$. Căutăm o soluţie pentru R în $M_{\mathsf{BSP},A}$, iar aceasta este echivalent cu a găsi un $t \in T_{\Sigma_{\mathsf{BSP},A}}$ cu $t \underline{\leftrightarrow} a.t$.

Din cele arătate data trecută, ştim că există $n \in \mathbb{N}$ astfel încât pentru orice $k \geq n$ avem

$$(\mathsf{BSP}+\mathsf{PR})(A) \vdash \pi_k t = t,$$

deci $t \underline{\leftrightarrow} \pi_k t$. Arătăm acum că pentru orice $m \in \mathbb{N}$, avem că $(\mathsf{BSP}+\mathsf{PR}+R)(A) \vdash \pi_m X = a^m 0$.

Demonstrăm prin inducţie după m. Pentru $m = 0$, avem, raţionând în $(\mathsf{BSP}+\mathsf{PR}+R)(A)$, că

$$\pi_0 X = \pi_0(a.X) = 0 = a^0 0.$$

Presupunând adevărat pentru un m, avem că

$$\pi_{m+1} X = \pi_{m+1}(a.X) = a.\pi_m(X) = a.(a^m 0) = a^{m+1} 0.$$

Aşadar, pentru orice $m \in \mathbb{N}$, avem $\pi_m t \underline{\leftrightarrow} a^m 0$, deci

$$a^n 0 \underline{\leftrightarrow} \pi_n t \underline{\leftrightarrow} t \underline{\leftrightarrow} \pi_{n+1} t \underline{\leftrightarrow} a^{n+1} 0,$$

dar avem că $a^n 0 \not\underline{\leftrightarrow} a^{n+1} 0$ (exerciţiu!), ceea ce este o contradicţie. Aşadar, nu avem soluţii, iar aceasta motivează căutarea de **modele noi**.

În primul rând, dacă W e o mulţime corespunzătoare, şi R este o specificaţie recursivă peste $\Sigma_{\mathsf{BSP},A}$ şi W, vom extinde notaţia $\mu t.R$ de la $t \in W$ la $t \in T_{\Sigma^W_{\mathsf{BSP},A}}$, recursiv, pentru orice s, $t \in T_{\Sigma^W_{\mathsf{BSP},A}}$, prin:

- $\mu 1.R = 1$, $\mu 0.R = 0$;
- $\mu(a.t).R = a.(\mu t.R)$;
- $\mu(s + t).R = (\mu s.R) + (\mu t.R)$.

Considerăm sistemul de deducţie peste $\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}$ şi A, ce extinde pe cel definit în secţiunea despre $\mathsf{BSP}(A)$ cu următoarele reguli (pentru orice W, orice specificaţie R peste $\Sigma_{\mathsf{BSP}(A),A}$ şi W, orice $X \in W$ şi orice $t \in T_{\Sigma^W_{\mathsf{BSP},A}}$ cu $X = t \in R$):

- $(\{\mu t.R \downarrow\}, \mu X.R \downarrow)$;
- $(\{\mu t.R \to^a y\}, \mu X.R \to^a y)$.

Cum sistemul de deducţie este în format cale, avem că relaţia de bisimilaritate $\underline{\leftrightarrow}$ pe LTS-ul indus de el, ce are ca mulţime de stări pe $T_{\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}}$, este o congruenţă pe $T_{\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}}$. Notăm $\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}$-algebra-cât cu $M_{\mathsf{BSP}_{\mathsf{rec}},A} := T_{\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}} / \underline{\leftrightarrow}$.

Teorema 4.1. *Această algebră este un model al lui* $\mathsf{BSP}_{\mathsf{rec}}(A)$*, i.e.* $M_{\mathsf{BSP}_{\mathsf{rec}},A} \models \mathsf{BSP}_{\mathsf{rec}}(A)$.

Demonstraţie. Fie W şi o specificaţie R peste $\Sigma_{\mathsf{BSP},A}$ şi W. Luăm $X = t \in R$. Ceea ce trebuie arătat aici este că $\mu X.R \underline{\leftrightarrow} \mu t.R$. Pentru aceasta, se ia relaţia

$$\{(\mu X.R, \mu t.R)\} \cup \{(p, p) \mid p \in T_{\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}}\}$$

şi se arată că este o bisimulare. □

Teorema 4.2. $\mathsf{BSP}_{\mathsf{rec}}(A)$ *este o extindere-închisă conservativă a lui* $\mathsf{BSP}(A)$*, i.e. pentru orice* p, $q \in T_{\Sigma_{\mathsf{BSP}_{\mathsf{rec}},A}}$*, avem*

$$\mathsf{BSP}(A) \vdash p = q \Leftrightarrow \mathsf{BSP}_{\mathsf{rec}}(A) \vdash p = q.$$

În mod analog, se construieşte şi $M_{(\mathsf{BSP}+\mathsf{PR})_{\mathsf{rec}},A}$.

4.5 Principii de recursie. Specificaţii gardate

Dacă $T = (\Sigma, E)$ este o teorie, iar M o Σ-algebră ce satisface T, spunem că M satisface RDP (*Recursive Definition Principle*) şi notăm $M \models \mathsf{RDP}$ dacă pentru orice W şi orice specificaţie recursivă R peste Σ şi W, avem că există o soluţie a lui R în M.

De exemplu, avem că $M_{\mathsf{BSP},A} \not\models \mathsf{RDP}$, dar $M_{\mathsf{BSP_{rec}},A} \models \mathsf{RDP}$.

În continuare, vom oferi nişte metode pentru a garanta **unicitatea** soluţiilor. Nu vom intra mereu în toate detaliile, ci doar vom schiţa ideile principale şi vom raţiona mai informal.

Dacă X este o mulţime, $s \in T_{\Sigma_{\mathsf{BSP_{rec}},A}}(X)$ şi $x \in X \cup W$, spunem că o apariţie a lui x în s este **gardată** (*guarded*) dacă este în cadrul unui subtermen de forma $a.t$. Termenul s se numeşte **complet gardat** dacă pentru orice $x \in X \cup W$ avem că fiecare apariţie a lui x în s este gardată. O specificaţie recursivă se numeşte **complet gardată** dacă orice membru drept al unei ecuaţii din ea este complet gardat.

Oferim câteva exemple:

- În termenul $a.X + Y + c.(b.0 + X)$, avem că apariţiile lui X sunt gardate, iar apariţia lui Y nu este.
- În termenul $a.X + Y + a.(b.0 + Y)$, avem că apariţia lui X este gardată, iar prima apariţie a lui Y nu este gardată, în vreme ce a doua este.
- În termenul $a.(X + Y)$, şi X, şi Y apar doar gardat, deci termenul este complet gardat.
- Specificaţia $R_1 = \{X_1 = a.X_1, Y_1 = a.X_1\}$ este complet gardată.
- Specificaţia $R_2 = \{X_2 = a.X_2, Y_2 = X_2\}$ nu este complet gardată.
- Specificaţia dată mai devreme pentru o stivă este complet gardată.

Observăm că, raţionând în $(\mathsf{BSP}+R_1)(A)$, avem că $X_1 = a.X_1$ şi $Y_1 = a.X_1 = X_1$, iar, raţionând în $(\mathsf{BSP}+R_2)(A)$, avem că $X_2 = a.X_2$ şi $Y_2 = X_2 = a.X_2$, deci într-un anume sens cele două specificaţii sunt „echivalente", deci definiţia de mai sus este prea restrictivă.

Spunem că $s \in T_{\Sigma_{\mathsf{BSP_{rec}},A}}(X)$ este **gardat** dacă există $t \in T_{\Sigma_{\mathsf{BSP_{rec}},A}}(X)$ complet gardat astfel încât $\mathsf{BSP_{rec}}(A) \vdash s = t$. Spunem că R este o specificaţie **gardată** dacă există o specificaţie complet gardată R', în care apar în membrii stângi exact aceleaşi variabile ca în R, astfel încât pentru orice $e \in R'$, avem

$$(\mathsf{BSP}+R)(A) \vdash e.$$

De exemplu, R_2 de mai sus este gardată, deoarece putem lua $R' = \{X_2 = a.X_2, Y_2 = a.X_2\}$.

Dacă $T = (\Sigma, E)$ este o teorie, iar M o Σ-algebră ce satisface T, spunem că M satisface RSP (*Recursive Specification Principle*) şi notăm $M \models \mathsf{RSP}$ dacă pentru orice W şi orice specificaţie recursivă gardată R peste Σ şi W, avem că există cel mult o soluţie a lui R în M.

Putem folosi RSP ca pe o regulă de deducţie.

De exemplu, dacă avem specificaţia $R_1 = \{X = a.X\}$ şi $R_2 = \{Y = a.a.Y\}$, luând X o soluţie pentru R_1, obţinem că

$$X = a.X = a.a.X$$

şi deci X este o soluţie pentru R_2. Aplicând RSP, obţinem $X = Y$.

De asemenea, vom mai nota cu RDP^- varianta lui RDP care cere doar specificaţiilor gardate să aibă soluţii.

Introducem acum un nou principiu, numit AIP (*Approximation Induction Principle*), care, ca regulă de deducţie spune că dacă avem doi termeni s şi t astfel încât pentru orice $n \in \mathbb{N}$, avem $\pi_n s = \pi_n t$, atunci putem deduce $s = t$. Este, aşadar, o regulă de deducţie **infinitară**, ce necesită un număr numărabil de premise.

De pildă, folosind tot specificaţiile de mai devreme, putem deduce (ca într-un exemplu anterior) că pentru orice $n \in \mathbb{N}$, avem $\pi_n X = a^n 0$ şi $\pi_n Y = a^n 0$, deci, folosind AIP, deducem că $X = Y$. Se observă aşadar o legătură între AIP şi RSP, pe care o vom schiţa în continuare.

Dacă punem în descrierea lui AIP condiţia ca s să fie gardat, obţinem ceea ce se notează cu AIP^-. Remarcăm că deducţia precedentă încă funcţionează, deoarece avem $X = a.X$, iar $a.X$ este complet gardat, deci X este gardat.

Teorema 4.3. $M_{(\mathsf{BSP+PR})_{\mathsf{rec}},A} \models \mathsf{AIP}^-$.

Demonstraţie. Trebuie arătat că pentru orice p complet gardat şi orice q astfel încât pentru orice $n \in \mathbb{N}$ avem $\pi_n p \underline{\leftrightarrow} \pi_n q$, avem $p \underline{\leftrightarrow} q$. Pentru aceasta, se ia relaţia

$$\{(u, v) \mid u \text{ complet gardat şi pentru orice } n \in \mathbb{N},\ \pi_n u \underline{\leftrightarrow} \pi_n v\}$$

şi se arată că este o bisimulare – o demonstraţie destul de complexă. □

Pentru o teorie T ce extinde $\mathsf{BSP}(A)$, definim în mod recursiv că un termen este în formă normală *head* (HNF), în felul următor:

- 0 şi 1 sunt în HNF;
- pentru orice $a \in A$ şi **orice** t, $a.t$ este în HNF;
- pentru orice s, t în HNF, $s + t$ este în HNF.

Spunem că T are proprietatea HNF dacă pentru orice termen gardat s există un termen t în HNF cu $T \vdash s = t$.

Prin inducţie, se poate arăta că dacă T extinde $\mathsf{BSP}(A)$, atunci pentru orice t în HNF există $a_1, \ldots, a_n \in A$ şi termeni $t_1, \ldots, t_n$ cu

$$T \vdash t = \sum_{i=1}^{n} a_i.t_i (+1).$$

Tot prin inducţie, se poate arăta că $(\mathsf{BSP+PR})_{\mathsf{rec}}(A)$ are proprietatea HNF.

Propoziţia 4.4. *Fie R o specificaţie recursivă gardată pentru $(\mathsf{BSP+PR})(A)$, $X \in W$ şi $n \in \mathbb{N}$.*

Atunci există un termen $p \in T_{\Sigma_{\mathsf{BSP+PR},A}}$ astfel încât pentru orice $t \in T_{\Sigma_{(\mathsf{BSP+PR})_{\mathsf{rec}},A}}$ cu

$$(\mathsf{BSP+PR})_{\mathsf{rec}}(A) \vdash R[X := t],$$

avem

$$(\mathsf{BSP+PR})_{\mathsf{rec}}(A) \vdash \pi_n t = p.$$

Teorema 4.5. *Fie R o specificaţie recursivă gardată pentru $(\mathsf{BSP+PR})(A)$, $X \in W$ şi $n \in \mathbb{N}$. Fie s, $t \in T_{\Sigma_{(\mathsf{BSP+PR})_{\mathsf{rec}},A}}$ cu*

$$(\mathsf{BSP+PR})_{\mathsf{rec}}(A) \vdash R[X := s],$$

şi

$$(\mathsf{BSP+PR})_{\mathsf{rec}}(A) \vdash R[X := t].$$

Atunci

$$(\mathsf{BSP+PR})_{\mathsf{rec}}(A) \vdash \pi_n s = \pi_n t.$$

Demonstraţie. Iau p-ul din propoziţia anterioară şi atunci avem

$$(\mathsf{BSP+PR})_{\mathsf{rec}}(A) \vdash \pi_n s = p = \pi_n t.$$

□

Se poate arăta acum că, într-o teorie ce extinde $(\mathsf{BSP+PR})(A)$, în prezenţa AIP^-, proprietatea HNF implică RSP. Intuitiv vorbind, luăm s şi t care satisfac ecuaţiile unei specificaţii recursive gardate, iar, din rezultatele anterioare, obţinem că pentru orice n, $\pi_n s = \pi_n t$. Aplicând AIP^-, obţinem $s = t$.

În felul acesta se arată următorul rezultat.

Teorema 4.6. $M_{(\mathsf{BSP+PR})_{\mathsf{rec}},A} \models \mathsf{RSP}$.

Bibliografie

[1] J. C. M. Baeten, T. Basten, M. A. Reniers, *Process algebra: equational theories of communicating processes*. With forewords by Tony Hoare, Robin Milner and Jan Bergstra. Cambridge Tracts in Theoretical Computer Science, 50. Cambridge University Press, Cambridge, 2010.

[2] J. A. Bergstra, J. W. Klop, Process algebra for synchronous communication. *Inform. and Control* 60, no. 1-3, 109–137, 1984.

[3] V. E. Căzănescu, *Programare prin rescriere*. Editura Universităţii din Bucureşti, 2020.

[4] G. Ştefănescu, *Network algebra*. Springer-Verlag London, Ltd., London, 2000.