SUPORT EXAMEN PROGRAMARE LOGICĂ

1. Programare Logică - cazul logicii clauzelor definite propoziţionale

Logica propoziţională.

- *Formulele* sunt definite inductiv din *atomi propoziţionali* şi conectorii logic $\neg, \wedge, \vee, \rightarrow$.
- O *interpretare* este o funcţie care dă valori de adevăr (adevărat şi fals) atomilor.
- O formulă F este *adevărată* într-o interpretare $\mathcal{I}$, notând $\mathcal{I} \models F$, dacă valoarea de adevăr a formulei obţinută folosind *tabelele de adevăr* şi interpretarea $\mathcal{I}$ este adevărat.
- O formulă G este o *consecinţă logică* a unor formule $F_1, F_2, \ldots, F_n$, notat $F_1, \ldots, F_n \models G$, dacă pentru orice interpretare $\mathcal{I}$, dacă $\mathcal{I} \models F_i$, $i = 1, \ldots, n$, atunci $\mathcal{I} \models G$.
- O *clauză definită* este o formulă care este fie de forma (1) q (clauză unitate), sau (2) $p_1 \wedge \ldots \wedge p_k \rightarrow q$, unde $q, p_1, \ldots, p_n$ sunt atomi.
- Un *"program logic"* este o listă $F_1, \ldots, F_n$ de clauze definite.
- *Problema programării logice propoziţionale:* $F_1, \ldots, F_n \models q$, unde $F_1, \ldots, F_n, q$ clauze definite.

Sistem de deducţie pentru clauze definite.

- Pentru o mulţime S de clauze definite propoziţionale, considerăm un *sistem de deducţie* care are ca *axiome* orice clauză din S şi următoarele *reguli de deducţie:*

$$\boxed{\frac{P \quad P \rightarrow Q}{Q}\ (MP)} \qquad \boxed{\frac{P \quad Q}{P \wedge Q}\ (andI)}$$

- O clauză Q *se deduce* din S, notat $S \vdash Q$, dacă există o secvenţă de clauze $Q_1, \ldots, Q_n$ astfel încât $Q_n = Q$ şi fiecare Q_i: fie aparţine lui S, fie se poate deduce din $Q_1, \ldots, Q_{i-1}$ folosind regulile de deducţie.
- Dacă X, Y mulţimi de mulţimi, $f : X \rightarrow Y$ este *monotonă* dacă pentru orice mulţimi $X_1, X_2 \in X$, dacă $X_1 \subseteq X_2$, atunci $f(X_1) \subseteq f(X_2)$.
- Un *punct fix* al unei funcţii $f : \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ este o mulţime $Y \subseteq X$ astfel încât $f(Y) = Y$, unde $\mathcal{P}(X)$ este mulţimea părţilor lui X.
- Un *cel mai mic punct fix* (lfp) Y al lui f este un punct fix şi dacă Z este tot un punct fix, atunci $Y \subseteq Z$.

Teoremă 1 (*)**.** *Dacă $f : \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ este monotonă, atunci f are un cel mai mic punct fix.*

- Fie A mulţimea atomilor care apar în S şi $Baza = \{p_i \mid p_i \in S\}$ mulţimea atomilor care apar în clauzele unitate din S. Definim funcţia $f_S : \mathcal{P}(A) \rightarrow \mathcal{P}(A)$ prin

$$f_S(Y) = Y \cup Baza \cup \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \rightarrow a) \text{ este în } S, s_1 \in Y, \ldots, s_n \in Y\}$$

Propoziţie 1. *Funcţia f_S este monotonă.*

Teoremă 2. *Fie X este cel mai mic punct fix al funcţiei f_S. Atunci $q \in X$ dacă şi numai dacă $S \models q$.*

Corolar 1. *Sistemul de deducţie pentru clauze definite propoziţionale este complet pentru a arăta clauze unitate, i.e. dacă $S \models q$, atunci $S \vdash q$.*

- *Metodă de decizie* pentru a verifica $S \vdash Q$:
  - calculăm cel mai mic punct fix X al funcţiei f_S
  - dacă $q \in X$ atunci returnăm adevărat, altfel returnăm fals.

2. Programare Logică - cazul logicii Horn

Logica de ordinul I (calculul cu predicate).

- Un *limbaj $\mathcal{L}$ de ordinul I* este format dintr-o mulţime numărabilă de variabile $V = \{x_n \mid n \in \mathbb{N}\}$, conectorii $\neg, \rightarrow, \wedge, \vee$, paranteze, cuantificatorul universal $\forall$ şi cuantificatorul existenţial $\exists$, o mulţime $\mathbf{P}$ de simboluri de relaţii, o mulţime $\mathbf{F}$ de simboluri de funcţii, o mulţime $\mathbf{C}$ de simboluri de constante, o funcţie aritate $ar : \mathbf{F} \cup \mathbf{P} \rightarrow \mathbb{N}^*$.
- *Termenii* lui $\mathcal{L}$ sunt definiţi inductiv asftel: orice variabilă este un termen; orice simbol de constantă este un termen; dacă $f \in \mathbf{F}$, $ar(f) = n$ şi $t_1, \ldots, t_n$ sunt termeni, atunci $f(t_1, \ldots, t_n)$ este termen. Mulţimea termenilor lui $\mathcal{L}$ este notată cu $Trm_{\mathcal{L}}$.
- Dacă $R \in \mathbf{R}$, $ar(R) = n$ şi $t_1, \ldots, t_n$ sunt termeni, atunci $R(t_1, \ldots, t_n)$ este *formulă atomică*.
- *Formulele* lui $\mathcal{L}$ sunt definite asftel: orice formulă atomică este o formulă; dacă A este o formulă, atunci $\neg A$ este o formulă; dacă A şi B sunt formule, atunci $A \vee B$, $A \wedge B$, $A \rightarrow B$ sunt formule; dacă A este o formulă şi x_i este o variabilă, atunci $(\forall x_i)A$, $(\exists x_i)A$ sunt formule.
- O *structură* este de forma $\mathcal{S} = (S, \mathbf{F}^{\mathcal{S}}, \mathbf{P}^{\mathcal{S}}, \mathbf{C}^{\mathcal{S}})$, unde S este o mulţime nevidă, $\mathbf{F}^{\mathcal{S}} = \{f^{\mathcal{S}} \mid f \in \mathbf{F}\}$ este o mulţime de operaţii pe A (dacă f are aritatea n, atunci $f^{\mathcal{S}} : S^m \rightarrow S$), $\mathbf{R}^{\mathcal{S}} = \{R^{\mathcal{S}} \mid R \in \mathbf{R}\}$ este o mulţime de relaţii pe A (dacă R are aritatea n, atunci $R^{\mathcal{S}} \subseteq S^m$), şi $\mathbf{C}^{\mathcal{S}} = \{c^{\mathcal{S}} \in S \mid c \in \mathbf{C}\}$.
- O *interpretare a variabilelor* lui $\mathcal{L}$ în $\mathcal{S}$ este o funcţie $I : V \rightarrow S$. *Interpretarea termenului* t în $\mathcal{S}$ sub I ($t_I^{\mathcal{S}}$) este definită inductiv prin:
  - dacă $t = x_i \in V$, atunci $t_I^{\mathcal{S}} := I(x_i)$
  - dacă $t = c \in \mathbf{C}$, atunci $t_I^{\mathcal{S}} := c^{\mathcal{S}}$
  - dacă $t = f(t_1, \ldots, t_n)$, atunci $t_I^{\mathcal{S}} := f^{\mathcal{S}}((t_1)_I^{\mathcal{S}}, \ldots, (t_n)_I^{\mathcal{S}})$
- O formulă este adevărată în $\mathcal{S}$ sub interpretarea I dacă:
  - $\mathcal{S}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{S}}(t_1^{\mathcal{S}}, \ldots, t_n^{\mathcal{S}})$
  - $\mathcal{S}, I \models \neg B$ dacă $\mathcal{S}, I \not\models B$
  - $\mathcal{S}, I \models B \vee C$ dacă $\mathcal{S}, I \models B$ sau $\mathcal{S}, I \models C$
  - $\mathcal{S}, I \models B \wedge C$ dacă $\mathcal{S}, I \models B$ si $\mathcal{S}, I \models C$
  - $\mathcal{S}, I \models B \rightarrow C$ dacă $\mathcal{S}, I \not\models B$ sau $\mathcal{S}, I \models C$
  - $\mathcal{S}, I \models (\forall x)B$ dacă pentru orice interpretare $I_{x \leftarrow a}$ avem $\mathcal{S}, I_{x_i \leftarrow a} \models B$
  - $\mathcal{S}, I \models (\exists x)B$ dacă există o interpretare $I_{x \leftarrow a}$ astfel încât $\mathcal{S}, I_{x_i \leftarrow a} \models B$

  unde pentru orice $a \in S$, $I_{x \leftarrow a}(y) = \begin{cases} I(y) & \text{dacă } y \neq x \\ a & \text{dacă } y = x \end{cases}$

- O formulă A este *adevărată într-o structură* $\mathcal{S}$, notat $S \models A$, dacă este adevărată în $\mathcal{S}$ sub orice interpretare. Spunem că $\mathcal{S}$ este *model* al lui A. O formulă A este *adevărată în logica de ordinul I*, notat $\models A$, dacă este adevărată în orice structură.
- O formulă G este o *consecinţă logică* a formulelor $F_1, \ldots, F_n$, notat $F_1, \ldots, F_n \models G$, dacă pentru orice structură $\mathcal{S}$, dacă $S \models F_i$, $i = 1, \ldots, n$, atunci $S \models G$.
- *Logica clauzelor definite/Logica Horn* – un fragment al logicii de ordinul I în care singurele formule admise sunt *clauze definite*:
  - formule atomice: $P(t_1, \ldots, t_n)$
  - $A_1 \wedge \ldots \wedge A_n \to B$, unde toate A_i, B sunt formule atomice.
- *Problema programării logice:* $T \models A_1 \wedge \ldots \wedge A_n$, unde T mulţime de clauze definite şi toate A_i sunt formule atomice.

Algoritmul de unificare.

- O *subtituţie* σ este o funcţie (parţială) de la variabile la termeni, adică $\sigma : V \to Trm_{\mathcal{L}}$.
- Doi termeni t_1 şi t_2 *se unifică* dacă există o substituţie θ astfel încât $\theta(t_1) = \theta(t_2)$.
- Un unificator ν pentru U este un *cel mai general unificator* (cgu) dacă pentru orice alt unificator ν' pentru U, există o substituţie μ astfel încât $\nu' = \nu; \mu$, unde $\nu; \mu$ este compunerea substitutiilor ν şi μ.

	Lista soluţie S	Lista de rezolvat R
Iniţial	$\emptyset$	$t_1 \doteq t'_1, \ldots, t_n \doteq t'_n$
SCOATE	S	$R',\ t \doteq t$
	S	R'
DESCOMPUNE	S	$R',\ f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n)$
	S	$R',\ t_1 \doteq t'_1, \ldots t_n \doteq t'_n$
REZOLVĂ	S	$R',\ x \doteq t$ sau $t \doteq x$, x nu apare în t
	$x \doteq t,\ S[x \leftarrow t]$	$R'[x \leftarrow t]$
Final	S	$\emptyset$

Sistem de deducţie *backchain* pentru logica Horn (clauze definite).

- Pentru o mulţime T de clauze Horn, considerăm un *sistem de deducţie* (sistemul backchain) care are ca *axiome* orice clauză din T şi *regulă de deducţie backchain:*

$$\boxed{\frac{\theta(p_1) \quad \theta(p_2) \quad \ldots \quad \theta(p_n) \quad (p_1 \wedge p_2 \wedge \ldots \wedge p_n \to q)}{\theta(q')}}$$

  unde $p_1 \wedge p_2 \wedge \ldots \wedge p_n \to q \in T$, iar θ este cgu pentru q' şi q.
- Notăm $T \vdash_b Q$ dacă există o derivare a lui Q din T folosind sistemul de deducţie *backchain* (definit ca în cazul propoziţional).

Propoziţie 2 (Corectitudine *)**.** *Dacă* $T \vdash_b Q$*, atunci* $T \models Q$.

- Fie $\mathcal{L}$ un limbaj de ordinul I cu cel puţin un simbol de constantă.
- *Universul Herbrand* este mulţimea $T_{\mathcal{L}}$ a tututor termenilor lui $\mathcal{L}$ fără variabile.
- Un *model Herbrand* este o structură $\mathcal{H} = (T_{\mathcal{L}}, \mathbf{F}^{\mathcal{H}}, \mathbf{P}^{\mathcal{H}}, \mathbf{C}^{\mathcal{H}})$, unde
  - pentru orice simbol de constantă c, $c^{\mathcal{H}} = c$
  - pentru orice simbol de funcţie f de aritate n, $f^{\mathcal{H}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$
- O *instanţă de bază* a unei clauze $Q_1(X_1) \wedge \ldots \wedge Q_n(X_n) \to P(Y)$ este rezultatul obţinut prin înlocuirea variabilelor cu termeni fără variabile.
- Pentru o mulţime de clauze T, o formulă atomică P şi o mulţime de formule atomice X, predicatul $oneStep_T(P, X)$ este adevărat dacă există o instanţă de bază a unei clauze $Q_1(X_1) \wedge \ldots \wedge Q_n(X_n) \to P(Y)$ din T astfel încât P este instanţa lui $P(Y)$ şi instanţa lui $Q_i(X_i)$ este în X, pentru orice $i = 1, \ldots, n$.
- *Baza Herbrand* $B_{\mathcal{L}}$ este mulţimea formulelor atomice fără variabile.
- Pentru o mulţime de clauze T, definim $f_T : \mathcal{P}(B_{\mathcal{L}}) \to \mathcal{P}(B_{\mathcal{L}})$, $f_T(X) = \{P \in B_{\mathcal{L}} \mid oneStep_T(P, X)\}$.
- Fie FP_T cel mai mic punct fix al funcţiei monotone f_T.
- Pentru o mulţime de clauze T, definim *cel mai mic model Herbrand* $\mathcal{LH}$ ca fiind un model Herbrand în care un predicat $R(t_1, \ldots, t_n)$ este adevărat ddacă $R(t_1, \ldots, t_n) \in FP_T$.

Propoziţie 3. $\mathcal{LH} \models T$.

Propoziţie 4 (*)**.** *Pentru orice formulă atomică* Q*,* $T \vdash_B Q$ *ddacă* $\mathcal{LH} \models Q$.

Teoremă 3 (Completitudine *)**.** *Dacă* $T \models Q$*, atunci* $T \vdash_b Q$.

Rezoluţie SLD.

- Regula *backchain* este implementată în programarea logică prin *rezoluţia SLD* (Selected, Linear, Definite).
- O clauză definită $P_1 \wedge \ldots \wedge P_n \to Q$ poate fi gândită ca formula $Q \vee \neg P_1 \vee \ldots \vee \neg P_n$.
- Pentru o mulţime de clauze definite T, regula rezolutiei SLD este

$$SLD \ \boxed{\frac{\neg P_1 \vee \cdots \vee \neg P_i \vee \cdots \vee \neg P_n}{(\neg P_1 \vee \cdots \vee \neg Q_1 \vee \cdots \vee \neg Q_m \vee \cdots \vee \neg P_n)\theta}}$$

  unde $Q \vee \neg Q_1 \vee \cdots \vee \neg Q_m$ este o clauză definită din T (în care toate variabilele au fost redenumite) şi θ este c.g.u pentru P_i şi Q.
- Fie T o mulţime de clauze definite şi $P_1 \wedge \ldots \wedge P_m$ o ţintă, unde P_i sunt formule atomice. *O derivare* din T prin rezoluţie SLD este o secvenţă $G_0 := \neg P_1 \vee \ldots \vee \neg P_m$, G_1, $\ldots$, G_k, $\ldots$ în care G_{i+1} se obţine din G_i prin regula SLD. Dacă există un k cu $G_k = \square$ (clauza vidă), atunci derivarea se numeşte *SLD-respingere*.

Teoremă 4 (Completitudinea SLD-rezoluţiei)**.** *Sunt echivalente:*

(1) *există o* SLD-respingere *a lui* $P_1 \wedge \ldots \wedge P_m$ *din* T*,*
(2) $T \vdash_b P_1 \wedge \ldots \wedge P_m$*,*
(3) $T \models P_1 \wedge \cdots \wedge P_m$.

- Fie T o mulţime de clauze definite şi o ţintă $G_0 = \neg P_1 \vee \ldots \vee \neg P_m$. Un *arbore SLD* este definit astfel:
  - Fiecare nod al arborelui este o ţintă (posibil vidă)
  - Rădăcina este G_0
  - Dacă arborele are un nod G_i, iar G_{i+1} se obţine din G_i folosind regula SLD folosind o clauză $C_i \in T$, atunci nodul G_i are copilul G_{i+1}. Muchia dintre G_i şi G_{i+1} este etichetată cu C_i.
- Dacă un arbore SLD cu rădăcina G_0 are o frunză $\Box$ (clauza vidă), atunci există o SLD-respingere a lui G_0 din T.

3. Algebre multisortate

Signaturi multisortate. Mulţimi şi funcţii multisortate.

- O *signatură multisortată* este o pereche (S, Σ), unde $S \neq \emptyset$ este o mulţime de sorturi şi Σ este o mulţime de simboluri de operaţii $\sigma : s_1 s_2 \ldots s_n \to s$. Dacă $n = 0$, atunci $\sigma : \to s$ este simbolul unei constante.

Fixăm o mulţime de sorturi S.

- O *mulţime S-sortată* este o familie de mulţimi $A = \{A_s\}_{s\in S}$.
- O *funcţie S-sortată* $f : A \to B$ este o familie de funcţii $f = \{f_s\}_{s\in S}$, unde $f_s : A_s \to B_s$, pt. or. $s \in S$. Dacă $f : A \to B$ şi $g : B \to C$, definim *compunerea* $f; g : A \to C$, $(f; g)_s(a) = g_s(f_s(a))$, or. $a \in A_s$.
- O funcţie S-sortată $f : A \to B$ este *injectivă, (surjectivă, bijectivă)* dacă f_s este injectivă, (surjectivă, bijectivă), or. $s \in S$. O funcţie S-sortată $f = \{f_s\}_{s\in S} : A \to B$ este *inversabilă* dacă există $g : B \to A$ astfel încât $f; g = 1_A$ şi $g; f = 1_B$.

Propoziţie 5. *O funcţie S-sortată $f : A \to B$ este inversabilă $\Leftrightarrow$ este bijectivă.*

Algebre multisortate.

- O *algebră multisortată de tip* (S, Σ) este $\mathcal{A} = (A_S, A_\Sigma)$ unde $A_S = \{A_s\}_{s\in S}$ este o mulţime S-sortată şi $A_\Sigma = \{A_\sigma\}_{\sigma\in\Sigma}$ este o familie de operaţii astfel încât
  - dacă $\sigma : s_1 \ldots s_n \to s$ în Σ, atunci $A_\sigma : A_{s_1} \times \ldots \times A_{s_n} \to A_s$.
  - dacă $\sigma : \to s$ în Σ, atunci $A_\sigma \in A_s$.

Morfisme de algebre multisortate.

- Un *morfism de (S, Σ)-algebre* $h : \mathcal{A} \to \mathcal{B}$ este o funcţie S-sortată $h = \{h_s\}_{s\in S} : \{A_s\}_{s\in S} \to \{B_s\}_{s\in S}$ care verifică condiţia de compatibilitate:
  - $h_s(A_\sigma) = B_\sigma$, or. $\sigma : \to s \in \Sigma$,
  - $h_s(A_\sigma(a_1, \ldots, a_n)) = B_\sigma(h_{s_1}(a_1), \ldots, h_{s_n}(a_n))$, or. $\sigma : s_1 \ldots s_n \to s \in \Sigma$ şi or. $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$.

Propoziţie 6. *Compunerea a două Σ-morfisme este un Σ-morfism.*

Izomorfisme de algebre multisortate.

- Un Σ-morfism $h : \mathcal{A} \to \mathcal{B}$ se numeşte *izomorfism* dacă există un Σ-morfism $g : \mathcal{B} \to \mathcal{A}$ astfel încât $h; g = 1_A$ şi $g; h = 1_B$. Deoarece g este unic, se notează cu h^{-1}.
- Două Σ-algebre $\mathcal{A}$ şi $\mathcal{B}$ sunt *izomorfe* ($\mathcal{A} \simeq \mathcal{B}$) dacă există un izomorfism $f : \mathcal{A} \to \mathcal{B}$.

Propoziţie 7. *Fie $h : \mathcal{A} \to \mathcal{B}$ un Σ-morfism. Atunci h este izomorfism $\Leftrightarrow$ este funcţie S-sortată bijectivă.*

Propoziţie 8. *Compunerea a două izomorfisme $f : \mathcal{A} \to \mathcal{B}$ şi $g : \mathcal{B} \to \mathcal{C}$ este un izomorfism. Mai mult, $(f; g)^{-1} = g^{-1}; f^{-1}$.*

Tipuri abstracte de date.

- Un tip abstract de date este o clasă $\mathfrak{C}$ de (S, Σ)-algebre cu proprietatea că oricare două (S, Σ)-algebre din $\mathfrak{C}$ sunt izomorfe.
- $\mathfrak{I}_{(S,\Sigma)} = \{\mathcal{I} \mid \mathcal{I}\ (S, \Sigma)\text{-algebră iniţială}\}$ este un tip abstract de date.

Termeni. Algebre de termeni.

- O *mulţime de variabile* este o mulţime S-sortată $X = \{X_s\}_{s\in S}$ astfel încât $X_s \cap X_{s'} = \emptyset$, or. $s, s' \in S$, $s \neq s'$, $X_s \cap \{\sigma\}_{\sigma:s_1\ldots s_n\to s\in\Sigma} = \emptyset$ şi $X_s \cap \{\sigma\}_{\sigma:\to s\in\Sigma} = \emptyset$.
- Mulţimea S-sortată a *termenilor cu variabile din X*, $T_\Sigma(X)$, este cea mai mică mulţime de şiruri finite peste alfabetul $L = \bigcup_{s\in S} X_s \cup \bigcup_{w,s} \Sigma_{w,s} \cup \{(,)\} \cup \{,\}$ care verifică:
  - $X \subseteq T_\Sigma(X)$,
  - dacă $\sigma : \to s$ în Σ, atunci $\sigma \in T_\Sigma(X)_s$,
  - dacă $\sigma : s_1 \ldots s_n \to s$ în Σ şi $t_i \in T_\Sigma(X)_{s_i}$, or. $1 \le i \le n$, atunci $\sigma(t_1, \ldots, t_n) \in T_\Sigma(X)_s$.
- *Inducţia pe termeni*: Fie $\mathbf{P}$ o proprietate astfel încât:
  - pasul iniţial: $\mathbf{P}(x) = true$, or. $x \in X$, şi $\mathbf{P}(\sigma) = true$, or. $\sigma : \to s$.
  - pasul de inducţie: pt. or. $\sigma : s_1 \ldots s_n \to s$ şi or. $t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$, dacă $\mathbf{P}(t_1) = \ldots = \mathbf{P}(t_n) = true$, atunci $\mathbf{P}(\sigma(t_1, \ldots, t_n)) = true$.

  Atunci $\mathbf{P}(t) = true$, oricare $t \in T_\Sigma(X)$.
- Mulţimea S-sortată a termenilor $T_\Sigma(X)$ este o (S, Σ)-algebră, numită *algebra termenilor cu variabile din X*, cu operaţiile definite astfel: pt. or. $\sigma : \to s$ din Σ, operaţia corespunzătoare este $T_\sigma := \sigma \in T_\Sigma(X)_s$ şi pt. or. $\sigma : s_1 \ldots s_n \to s$ din Σ, operaţia corespunzătoare este $T_\sigma : T_\Sigma(X)_{s_1\ldots s_n} \to T_\Sigma(X)_s$, $T_\sigma(t_1, \ldots, t_n) := \sigma(t_1, \ldots, t_n)$, or. $t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$. T_Σ *algebra termenilor fără variabile* ($X = \emptyset$).
- O *substituţie* a variabilelor din X cu termeni din $T_\Sigma(Y)$ este o funcţie S-sortată $\tau : X \to T_\Sigma(Y)$.

Algebră iniţială.

- O (S, Σ)-algebră $\mathcal{I}$ este iniţială într-o clasă de (S, Σ)-algebre $\mathfrak{K}$ dacă pentru orice $\mathcal{B} \in \mathfrak{K}$, există un unic (S, Σ)-morfism $f : \mathcal{I} \to \mathcal{B}$.

Propoziţie 9.

(1) Dacă $\mathcal{I}$ este iniţială în $\mathfrak{K}$ şi $\mathcal{A} \in \mathfrak{K}$ astfel încât $\mathcal{A} \simeq \mathcal{I}$, atunci $\mathcal{A}$ este iniţială în $\mathfrak{K}$.

(2) Dacă $\mathcal{A}_1$ şi $\mathcal{A}_2$ sunt iniţiale în $\mathfrak{K}$, atunci $\mathcal{A}_1 \simeq \mathcal{A}_2$.

Teoremă 5. *Pentru orice (S, Σ)-algebră $\mathcal{B}$, există un unic morfism $f : T_\Sigma \to \mathcal{B}$.*

Corolar 2. *T_Σ este (S, Σ)-algebra iniţială.*

Algebre libere.

- O (S,Σ)-algebră $\mathcal{A}=(A_S,A_\Sigma)$ este *liber generată* de X dacă $X\subseteq A_S$, i.e. există funcţia S-sortată incluziune a lui X în A_S $i_A: X\hookrightarrow A_S$, şi pt. orice (S,Σ)-algebră $\mathcal{B}=(B_S,B_\Sigma)$ şi orice funcţie S-sortată $f:X\to B_S$, există un unic (S,Σ)-morfism $\tilde{f}:\mathcal{A}\to\mathcal{B}$ astfel încât $i_A;\tilde{f}=f$.

Teoremă 6. *Dacă $\mathcal{A}$ şi $\mathcal{B}$ sunt liber generate de X, atunci $\mathcal{A}\simeq\mathcal{B}$.*

Teoremă 7. *Fie $\mathcal{B}=(B_S,B_\Sigma)$ o (S,Σ)-algebră. Orice funcţie S-sortată $e:X\to B_S$ se extinde unic la un (S,Σ)-morfism $\tilde{e}:T_\Sigma(X)\to\mathcal{B}$.*

Corolar 3. *$T_\Sigma(X)$ este (S,Σ)-algebra liber generată de X.*

Propoziţie 10. *Fie $h:\mathcal{A}\to\mathcal{B}$ un (S,Σ)-morfism surjectiv şi X o mulţime de variabile. Pentru orice (S,Σ)-morfism $f:T_\Sigma(X)\to\mathcal{B}$, există un (S,Σ)-morfism $g:T_\Sigma(X)\to\mathcal{A}$ astfel încât $g;h=f$.*

Dacă $f:\mathcal{A}\to\mathcal{B}$ este un (S,Σ)-morfism şi $X\subseteq A_S$, $f\restriction_X$ este *restricţia* lui f la X, i.e. $(f\restriction_X)_s(x)=f_s(x)$, or. $x\in X_s$.

Propoziţie 11. *Fie $\mathcal{B}$ o (S,Σ)-algebră , X o mulţime de variabile şi $f:T_\Sigma(X)\to\mathcal{B}$, $g:T_\Sigma(X)\to\mathcal{B}$ morfisme. Atunci $g=f\Leftrightarrow g\restriction_X=f\restriction_X$.*

Propoziţie 12. *Dacă $X\simeq Y$, atunci $T_\Sigma(X)\simeq T_\Sigma(Y)$.*

Congruenţe.

- O relaţie S-sortată $\equiv=\{\equiv_s\}_{s\in S}\subseteq A_S\times A_S$ este o congruenţă dacă $\equiv_s\subseteq A_s\times A_s$ este echivalenţă, or. $s\in S$, şi $\equiv$ este compatibilă cu operaţiile: pt. or. $\sigma: s_1\ldots s_n\to s$ şi or. $a_i,b_i\in A_{s_i}$, $i=1,\ldots,n$, $a_i\equiv_{s_i} b_i$, or. $i=1,\ldots,n\Rightarrow A_\sigma(a_1,\ldots,a_n)\equiv_s A_\sigma(b_1,\ldots,b_n)$.
- Fie $\mathcal{A}$ o (S,Σ)-algebră şi $\equiv$ o congruenţă pe $\mathcal{A}$. Definim:
  - $[a]_{\equiv_s}:=\{a'\in A_s\mid a\equiv_s a'\}$ (clasa de echivalenţă a lui a) şi $A_s/_{\equiv_s}:=\{[a]_{\equiv_s}\mid a\in A_s\}$, or. $s\in S$.
  - *algebră cât* a lui $\mathcal{A}$ prin congruenţa $\equiv$ notată $\mathcal{A}/_\equiv$: $A/_\equiv:=\{A_s/_{\equiv_s}\}$ cu operaţiile: $(A/_\equiv)_\sigma:=[A_\sigma]_{\equiv_s}$, or. $\sigma:\to s$, şi $(A/_\equiv)_\sigma([a_1]_{\equiv_{s_1}},\ldots,[a_n]_{\equiv_{s_n}}):=[A_\sigma(a_1,\ldots,a_n)]_{\equiv_s}$, or. $\sigma:s_1\ldots s_n\to s$ şi $a_1\in A_{s_1},\ldots,a_n\in A_{s_n}$.
  - $[\cdot]_\equiv:\mathcal{A}\to\mathcal{A}/_\equiv$, $a\mapsto[a]_{\equiv_s}$, or. $a\in A_s$, morfism surjectiv. Avem $[a]_{\equiv_s}=[b]_{\equiv_s}\Leftrightarrow a\equiv_s b\Leftrightarrow(a,b)\in\equiv_s$.
- Dacă $f:\mathcal{A}\to\mathcal{B}$ un morfism de (S,Σ)-algebre, *nucleul* lui f este $Ker(f)=\{Ker(f_s)\}_{s\in S}$, unde $Ker(f_s):=\{(a,a')\in A_s\times A_s\mid f_s(a)=f_s(a')\}$, or. $s\in S$.

Propoziţie 13.

(1) $Ker(f)$ este o congruenţă pe $\mathcal{A}$.
(2) Dacă $\equiv$ este o congruenţă pe $\mathcal{A}$, atunci $Ker([\cdot]_\equiv)=\equiv$.

Teoremă 8 (Proprietatea de universalitate a algebrei cât). *Pentru orice (S,Σ)-algebră $\mathcal{B}$ şi pentru orice morfism $h:\mathcal{A}\to\mathcal{B}$ a.î. $\equiv\subseteq Ker(h)$, există un unic morfism $\overline{h}:\mathcal{A}/_\equiv\to\mathcal{B}$ a.i. $[\cdot]_\equiv;\overline{h}=h$.*

Propoziţie 14 (*). *Fie $\mathfrak{K}$ o clasă de (S,Σ)-algebre. Dacă $\equiv_{\mathfrak{K}}:=\bigcap\{Ker(h)\mid h:T_\Sigma\to\mathcal{B}\in\mathfrak{K}$ morfism$\}$, atunci:*

(1) $\equiv_{\mathfrak{K}}$ este congruenţa pe T_Σ,
(2) pt. or. $\mathcal{B}\in\mathfrak{K}$, există un unic morfism $\overline{h}:T_\Sigma/_{\equiv_{\mathfrak{K}}}\to\mathcal{B}$.

Ecuaţii. Relaţia de satisfacere.

- O (S,Σ)-*ecuaţie* $(\forall X)t\doteq_s t'$ este formată dintr-o mulţime de variabile X şi doi termeni $t,t'\in T_\Sigma(X)_s$.
- O (S,Σ)-algebră $\mathcal{A}=(A_S,A_\Sigma)$ *satisface o ecuaţie* $(\forall X)t\doteq_s t'$
  - dacă pentru orice funcţie S-sortată $e:X\to A_S$, $\tilde{e}_s(t)=\tilde{e}_s(t')$.
  - dacă pentru orice morfism $f:T_\Sigma(X)\to A$, $f_s(t)=f_s(t')$.
- O (S,Σ)-*ecuaţie condiţionată* $(\forall X)t\doteq_s t'$ *if* H este formată dintr-o mulţime de variabile X, doi termeni de acelaşi sort $t,t'\in T_\Sigma(X)_s$ şi o mulţime H de ecuaţii $u\doteq_{s'}v$, cu $u,v\in T_\Sigma(X)_{s'}$.
- O (S,Σ)-algebră $\mathcal{A}=(A_S,A_\Sigma)$ *satisface o ecuaţie condiţionată* $(\forall X)t\doteq_s t'$ *if* H
  - dacă pentru orice funcţie S-sortată $e:X\to A_S$, $\tilde{e}_{s'}(u)=\tilde{e}_{s'}(v)$, or. $u\doteq_{s'}v\in H\Rightarrow\tilde{e}_s(t)=\tilde{e}_s(t')$.
  - dacă pentru orice morfism $f:T_\Sigma(X)\to A$, $f_{s'}(u)=f_{s'}(v)$, or. $u\doteq_{s'}v\in H\Rightarrow f_s(t)=f_s(t')$.

Γ-algebre.

- Dacă Γ este o mulţime de ecuaţii condiţionate, o (S,Σ)-algebră $\mathcal{A}$ este o Γ-*algebră* dacă $\mathcal{A}\models\gamma$, or. $\gamma\in\Gamma$.
- Notăm cu $Alg(S,\Sigma,\Gamma)$ clasa tuturor Γ-algebrelor.

Teoremă 9. *Fie $\mathcal{A}$ şi $\mathcal{B}$ două (S,Σ)-algebre a.î. $\mathcal{A}\simeq\mathcal{B}$ şi $\gamma:=(\forall X)t\doteq_s t'$ if H. Atunci $\mathcal{A}\models\gamma\Leftrightarrow\mathcal{B}\models\gamma$.*

- O ecuaţie condiţionată θ este *consecinţă semantică* a lui Γ dacă $\mathcal{A}\models\Gamma$ implică $\mathcal{A}\models\theta$, pentru orice (S,Σ)-algebră $\mathcal{A}$.
- O congruenţă $\equiv$ pe $\mathcal{A}$ este *închisă la substituţie* dacă

$$\mathrm{CS}(\Gamma,\mathcal{A})\quad\boxed{\begin{array}{l}\text{or. } (\forall X)t\doteq_s t' \text{ if } H\in\Gamma,\text{ or. } e:X\to A_S\\ \tilde{e}_{s'}(u)\equiv_{s'}\tilde{e}_{s'}(v),\text{ or. } u\doteq_{s'}v\in H\Rightarrow\tilde{e}_s(t)\equiv_s\tilde{e}_s(t').\end{array}}$$

Propoziţie 15 (*). *Dacă $\equiv$ este o congruenţă pe $\mathcal{A}$ închisă la substituţie, atunci $\mathcal{A}/_\equiv\models\Gamma$.*

- Pentru Γ şi $\mathcal{A}=(A_S,A_\Sigma)$, definim $\equiv_{\Gamma,\mathcal{A}}:=\bigcap\{Ker(h)\mid h:\mathcal{A}\to\mathcal{B},\ \mathcal{B}\models\Gamma\}$.
- Dacă $\mathcal{A}=T_\Sigma(X)$, notăm $\equiv_{\Gamma,T_\Sigma(X)}$ cu $\equiv_\Gamma$. Avem $t\equiv_{\Gamma_s}t'\Leftrightarrow\Gamma\models(\forall X)t\doteq_s t'$.

Propoziţie 16 (*). *$\equiv_{\Gamma,\mathcal{A}}$ este o congruenţă pe $\mathcal{A}$ închisă la substituţie.*

Propoziţie 17 (*). *$\equiv_{\Gamma,\mathcal{A}}$ este cea mai mică congruenţă pe $\mathcal{A}$ închisă la substituţie.*

Teoremă 10 (*). *$T_\Sigma/_{\equiv_{\Gamma,T_\Sigma}}$ este Γ-algebra iniţială.*

Specificaţii algebrice.

- O *specificaţie algebrică* este un triplet (S,Σ,Γ), unde (S,Σ) este o signatură multisortată şi Γ este o mulţime de ecuaţii condiţionate. Specificaţia (S,Σ,Γ) defineşte clasa modelelor $Alg(S,\Sigma,\Gamma)$.
- Două specificaţii (S,Σ,Γ_1) şi (S,Σ,Γ_2) sunt *echivalente* dacă definesc aceeaşi clasă de modele.
- O specificaţie (S,Σ,Γ) este *adecvată* pentru $\mathcal{A}$ dacă $\mathcal{A}$ este Γ-algebră iniţială, i.e. $\mathcal{A}\in\mathfrak{I}_{(S,\Sigma,\Gamma)}$.

Deducţie ecuaţională - cazul necondiţionat.

- E mulţime de ecuaţii necondiţionate

R $\dfrac{}{(\forall X)t \doteq_s t}$ S $\dfrac{(\forall X)t_1 \doteq_s t_2}{(\forall X)t_2 \doteq_s t_1}$ T $\dfrac{(\forall X)t_1 \doteq_s t_2,\ (\forall X)t_2 \doteq_s t_3}{(\forall X)t_1 \doteq_s t_3}$

CΣ $\dfrac{(\forall X)t_1 \doteq_{s_1} t'_1, \ldots, (\forall X)t_n \doteq_{s_n} t'_n}{(\forall X)\sigma(t_1, \ldots, t_n) \doteq_s \sigma(t'_1, \ldots, t'_n)}$, unde $\sigma : s_1 \ldots s_n \to s \in \Sigma$

Sub$_E$ $\dfrac{}{(\forall X)\theta(t) \doteq_s \theta(t')}$, $(\forall Y)t \doteq_s t' \in E$ şi $\theta : Y \to T_\Sigma(X)$

- Ecuaţia $\epsilon := (\forall X)t \doteq_s t'$ se deduce din E dacă ex. o secvenţă $\epsilon_1, \ldots, \epsilon_n$ a.î. $\epsilon_n = \epsilon$ şi pt. or. $1 \le i \le n$:
  - $\epsilon_i \in E$ sau
  - ϵ_i se obţine din $\epsilon_1, \ldots, \epsilon_{i-1}$ aplicând una din reg. R, S, T, CΣ, Sub$_E$.

Deducţie ecuaţională - cazul condiţionat.

- Γ mulţime de ecuaţii condiţionate

R $\dfrac{}{(\forall X)t \doteq_s t}$ S $\dfrac{(\forall X)t_1 \doteq_s t_2}{(\forall X)t_2 \doteq_s t_1}$ T $\dfrac{(\forall X)t_1 \doteq_s t_2,\ (\forall X)t_2 \doteq_s t_3}{(\forall X)t_1 \doteq_s t_3}$

CΣ $\dfrac{(\forall X)t_1 \doteq_{s_1} t'_1, \ldots, (\forall X)t_n \doteq_{s_n} t'_n}{(\forall X)\sigma(t_1, \ldots, t_n) \doteq_s \sigma(t'_1, \ldots, t'_n)}$, unde $\sigma : s_1 \ldots s_n \to s \in \Sigma$

Sub$_\Gamma$ $\dfrac{(\forall X)\theta(u_1) \doteq_{s_1} \theta(v_1), \ldots, (\forall X)\theta(u_n) \doteq_{s_n} \theta(v_n)}{(\forall X)\theta(t) \doteq_s \theta(t')}$, unde $(\forall Y)t \doteq_s t'$ if $\{u_1 \doteq_{s_1} v_1, \ldots, u_n \doteq_{s_n} v_n\} \in \Gamma$, $\theta : Y \to T_\Sigma(X)$.

- Ecuaţia $\epsilon := (\forall X)t \doteq_s t'$ *se deduce* din Γ dacă ex. o secvenţă $\epsilon_1, \ldots, \epsilon_n$ a.î. $\epsilon_n = \epsilon$ si pt. or. $1 \le i \le n$:
  - $\epsilon_i \in \Gamma$ sau
  - ϵ_i se obţine din $\epsilon_1, \ldots, \epsilon_{i-1}$ aplicând una din reg. R, S, T, CΣ, Sub$_\Gamma$.

Corectitudinea logicii ecuaţionale.

- O regulă de deducţie $\dfrac{\epsilon_1, \ldots, \epsilon_n}{\epsilon}$ este corectă dacă $\Gamma \models \epsilon_1, \ldots, \Gamma \models \epsilon_n \Rightarrow \Gamma \models \epsilon$.

Propoziţie 18. *Regulile de deducţie R, S, T, CΣ, Sub$_\Gamma$ sunt corecte.*

Teoremă 11 (Corectitudinea deducţiei). $\Gamma \vdash (\forall X)t \doteq_s t' \Rightarrow \Gamma \models (\forall X)t \doteq_s t'$.

Completitudinea logicii ecuaţionale.

- O relaţie binară $\sim \subseteq T_\Sigma(X) \times T_\Sigma(X)$ este închisă la regula Reg $\dfrac{(\forall X)t_1 \doteq_{s_1} t'_1, \ldots, (\forall X)t_n \doteq_{s_n} t'_n}{(\forall X)t \doteq_s t'}$ dacă $t_1 \sim_{s_1} t'_1, \ldots, t_n \sim_{s_n} t'_n \Rightarrow t \sim_s t'$.

Propoziţie 19. *Sunt echivalente:*

(1) $\sim$ este congruenţă pe $T_\Sigma(X)$,
(2) $\sim$ este închisă la R, S, T, CΣ.

Propoziţie 20. *Sunt echivalente:*

(1) $\sim$ verifică CS($\Gamma, T_\Sigma(X)$) (i.e. închisă la substituţie),
(2) $\sim$ este închisă la Sub$_\Gamma$.

- Definim $t \sim_{\Gamma_s} t' \Leftrightarrow \Gamma \vdash (\forall X)t \doteq_s t'$, or. $s \in S$.

Propoziţie 21. *$\sim_\Gamma$ este o congruenţă pe $T_\Sigma(X)$ închisă la substituţie.*

Teoremă 12 (Completitudinea deducţiei). $\Gamma \models (\forall X)t \doteq_s t' \Rightarrow \Gamma \vdash (\forall X)t \doteq_s t'$.

Teorema de completitudine.

- Echivalenţa sintactică: $t \sim_{\Gamma_s} t' \Leftrightarrow \Gamma \vdash (\forall X)t \doteq_s t'$.
- Echivalenţa semantică: $t \equiv_{\Gamma_s} t' \Leftrightarrow \Gamma \models (\forall X)t \doteq_s t'$.
- Corectitudinea deducţiei: $\sim_\Gamma \subseteq \equiv_\Gamma$.
- Completitudinea deducţiei: $\equiv_\Gamma \subseteq \sim_\Gamma$.

Teoremă 13 (Teorema de completitudine). $\Gamma \models (\forall X)t \doteq_s t' \Leftrightarrow \Gamma \vdash (\forall X)t \doteq_s t'$ $(\equiv_\Gamma = \sim_\Gamma)$

Contexte.

- $nr_y(t)$ = numărul de apariţii ale lui y în t
- Fie z a.î. $z \notin X$. Un termen $c \in T_\Sigma(X \cup \{z\})$ se numeşte *context* dacă $nr_z(c) = 1$.
- Dacă $t_0 \in T_\Sigma(X)$ şi t_0 are acelaşi sort cu z, definim substituţia $\{z \leftarrow t_0\} : X \cup \{z\} \to T_\Sigma(X)$, prin $\{z \leftarrow t_0\}(x) = \begin{cases} t_0, & \text{dacă } x = z \\ x, & \text{altfel} \end{cases}$.
  Pentru un context $c \in T_\Sigma(X \cup \{z\})$, notăm $c[z \leftarrow t_0] := \{z \leftarrow t_0\}(c)$.

Sistem de rescriere.

- O regulă de rescriere $l \to_s r$ (peste Y) este formată din $l, r \in T_\Sigma(Y)_s$ astfel încât l nu este variabilă şi $Var(r) \subseteq Var(l)$.
- Un sistem de rescriere (TRS) este o mulţime finită de reguli de rescriere.
- Dacă R este un sistem de rescriere, pentru $t, t' \in T_\Sigma(X)_s$ definim relaţia $t \to_R t'$ astfel:

$$\boxed{\begin{array}{lll} t \to_R t' & \Leftrightarrow & t \text{ este } c[z \leftarrow \theta_s(l)] \text{ şi} \\ & & t' \text{ este } c[z \leftarrow \theta_s(r)], \text{ unde} \\ & & c \in T_\Sigma(X \cup \{z\}) \text{ context}, \\ & & l \to_s r \in R \text{ cu } Var(l) = Y, \\ & & \theta : Y \to T_\Sigma(X) \text{ substituţie} \end{array}}$$

- Dacă E este o mulţime de ecuaţii astfel încât, pt. or. $(\forall Y) l \doteq_s r \in E$, $l \notin Y$ (nu este variabilă) şi $Var(r) \subseteq Var(l)$, definim *sistemul de rescriere determinat de E* $R_E := \{l \to_s r \mid (\forall Y) l \doteq_s r \in E\}$. Notăm *relaţia de rescriere generată de* R_E prin $\to_E := \to_{R_E}$.

Logica ecuaţională şi rescrierea termenilor.

- E mulţime de ecuaţii necondiţionate: $\boxed{\mathrm{SR}_E \quad \dfrac{}{(\forall X) c[z \leftarrow \theta(t)] \doteq_{s'} c[z \leftarrow \theta(t')]}}$, unde
  $(\forall Y) t \doteq_s t' \in E$, $\theta : Y \to T_\Sigma(X)$, $c \in T_\Sigma(X \cup \{z\})_{s'}$, $z \notin X$, $nr_z(c) = 1$.
- Γ mulţime de ecuaţii condiţionate: $\boxed{\mathrm{SR}_\Gamma \quad \dfrac{(\forall X)\theta(u_1) \doteq_{s_1} \theta(v_1), \ldots, (\forall X)\theta(u_n) \doteq_{s_n} \theta(v_n)}{(\forall X) c[z \leftarrow \theta(t)] \doteq_{s'} c[z \leftarrow \theta(t')]}}$, unde
  $(\forall Y) t \doteq_s t'$ if $\{u_1 \doteq_{s_1} v_1, \ldots, u_n \doteq_{s_n} v_n\} \in \Gamma$, $\theta : Y \to T_\Sigma(X)$, $c \in T_\Sigma(X \cup \{z\})_{s'}$, $z \notin X$, $nr_z(c) = 1$.

Propoziţie 22. *SR_Γ este regulă de deducţie corectă.*

Teoremă 14. *Sunt echivalente:*

(1) $\Gamma \vdash_{R,S,T,C\Sigma,Sub_\Gamma} (\forall X) t \doteq_s t'$,

(2) $\Gamma \vdash_{R,S,T,SR_\Gamma} (\forall X) t \doteq_s t'$.

Teoremă 15. $E \vdash (\forall X) t \doteq_s t' \quad \Leftrightarrow \quad t \overset{*}{\leftrightarrow}_E t'$.

Sisteme de rescriere abstracte.

- Un *sistem de rescriere abstract* este o pereche $(T, \to)$ unde T este o mulţime şi $\to \subseteq T \times T$.
- $t \in T$ este *reductibil* dacă există $t' \in T$ a.î. $t \to t'$.
- $t \in T$ este *în formă normală* (ireductibil) dacă nu este reductibil.
- t_0 este *o formă normală a lui t* dacă $t \overset{*}{\to} t_0$ şi t_0 este în formă normală.
- t_1 şi t_2 *se intâlnesc* ($t_1 \downarrow t_2$) dacă există $t \in T$ a.î. $t_1 \overset{*}{\to} t \overset{*}{\leftarrow} t_2$.
- $(T, \to)$ se numeşte
  - *noetherian:* dacă nu există reduceri infinite $t_0 \to t_1 \to t_2 \to \ldots$.
  - *confluent:* $t_1 \overset{*}{\leftarrow} t \overset{*}{\to} t_2 \Rightarrow t_1 \downarrow t_2$.
  - *local confluent:* $t_1 \leftarrow t \to t_2 \Rightarrow t_1 \downarrow t_2$.
  - *Church-Rosser*: $t_1 \overset{*}{\leftrightarrow} t_2 \Rightarrow t_1 \downarrow t_2$.
  - *normalizat*: orice element are o formă normală.
  - *complet* (convergent, canonic): confluent şi noetherian.

Propoziţie 23. *Fie $(T, \to)$ sistem de rescriere. Dacă $t \downarrow t'$, atunci $t \overset{*}{\leftrightarrow} t'$.*

Propoziţie 24. *Dacă $(T, \to)$ este un sistem de rescriere noetherian, atunci orice element are o formă normală.*

Propoziţie 25. *Dacă $(T, \to)$ este un sistem de rescriere complet, atunci orice element are o unică formă normală.*

Propoziţie 26. *Un sistem de rescriere este confluent ddacă este Church-Rosser.*

Propoziţie 27. *Dacă $(T, \to)$ este un sistem de rescriere confluent, atunci este local confluent.*

Propoziţie 28. *Dacă $(T, \to)$ este un sistem de rescriere noetherian şi local confluent, atunci este confluent.*

Propoziţie 29. *Fie $(T, \to)$ sistem de rescriere complet. Atunci $t \overset{*}{\leftrightarrow} t' \Leftrightarrow fn(t) = fn(t')$.*

Corolar 4. *Dacă sistemul de rescriere $(T_\Sigma(X), R_E)$ este complet, atunci $E \vdash (\forall X) t \doteq_s t' \Leftrightarrow t \overset{*}{\leftrightarrow}_E t' \Leftrightarrow fn(t) = fn(t')$.*

Terminarea sistemelor de rescriere. Fie (S, Σ) o signatură, Y mulţime de variabile şi R un TRS.

- *Arborele de reducere* al termenului t este definit astfel:
  - rădăcina arborelui are eticheta t,
  - descendenţii nodului cu eticheta u sunt etichetaţi cu termenii u' care verifică $u \to_R u'$.

Propoziţie 30. *Sunt echivalente:*

(1) R este noetherian,
(2) oricărui termen t îi poate fi asociat un număr natural $\mu(t) \in \mathbb{N}$ astfel încât $t \to_R t'$ implică $\mu(t) > \mu(t')$.

Propoziţie 31 (*). *Fie $\mathcal{A}$ o (S, Σ)-algebră astfel încât:*

- *$A_s = \mathbb{N}$ or. $s \in S$,*
- *or. $\sigma : s_1 \dots s_n \to s$, dacă $k_i > k_i'$ atunci $A_\sigma(k_1, \dots, k_i, \dots k_n) > A_\sigma(k_1, \dots, k_i', \dots k_n)$,*
- *$\tilde{\mathbf{e}}(l) > \tilde{\mathbf{e}}(r)$, or. $l \to r \in R$ şi or. $\mathbf{e} : Var(l) \to A$.*

Atunci R este noetherian.

- O ordine strictă $>$ pe $T_\Sigma(X)$ se numeşte *o ordine de reducere* dacă:
  - este well-founded: orice mulţime de termeni are un cel mai mic element în raport cu relaţia $>$;
  - este *compatibilă cu operaţiile*: dacă $s_1 > s_2$, atunci $\sigma(t_1, \dots, t_{i-1}, s_1, t_{i+1}, \dots, t_n) > \sigma(t_1, \dots, t_{i-1}, s_2, t_{i+1}, \dots, t_n)$, pentru orice $\sigma : s_1 \dots s_n \to s$;
  - este *închisă la substituţii*: dacă $s_1 > s_2$, atunci $\theta(s_1) > \theta(s_2)$, pentru orice substituţie θ.
- Relaţia de ordine strictă $>$ pe $T_\Sigma(X)$ definită prin $s > t$ ddacă $|s| > |t|$ şi $nr_x(s) \geq nr_x(t)$, pentru orice $x \in X$ este o ordine de reducere.
- Ordinea lexicografică $>_{lpo}$ indusă pe mulţimea de termeni $T_\Sigma(X)$ de o relaţie de ordine strictă $>$ pe signatură este o ordine de reducere. Avem $s >_{lpo} t$ ddacă
  - (LPO1) $t \in X$ şi $s \neq t$, sau
  - (LPO2) $s = f(s_1, \dots, s_m)$, $t = g(t_1, \dots, t_n)$ şi
    - (LPO2a) există i, $1 \leq i \leq m$ astfel încât $s_i \geq_{lpo} t$, sau
    - (LPO2b) $f > g$ şi $s >_{lpo} t_j$, pentru orice j, $1 \leq j \leq n$
    - (LPO2c) $f = g$, $s >_{lpo} t_j$, pentru orice j, $1 \leq j \leq n$, şi există i, $1 \leq i \leq m$ astfel încât $s_1 = t_1, \dots, s_{i-1} = t_{i-1}$ şi $s_i >_{lpo} t_i$.

Teoremă 16 (*). *Următoarele sunt echivalente:*

(1) Un sistem de rescrire R este noetherian.
(2) Există o ordine de reducere $>$ care satisface $l > r$ pentru orice $l \to r \in R$.

Confluenţă şi perechi critice. Fie (S, Σ) o signatură, Y mulţime de variabile şi R un TRS.

- Fie $l_1 \to r_1$, $l_2 \to r_2 \in R$ astfel încât:
  (1) $Var(l_1) \cap Var(l_2) = \emptyset$,
  (2) există t un subtermen al lui l_1 care nu este variabilă ($l_1 = c[z \leftarrow t]$, unde $nr_z(c) = 1$, t nu este variabilă)
  (3) există θ c.g.u pentru t şi l_2 (i.e. $\theta(t) = \theta(l_2)$).

Perechea $(\theta(r_1), \theta(c)[z \leftarrow \theta(r_2)])$ se numeşte *pereche critică*.

$$\theta(r_1) \xleftarrow{R} \theta(l_1) \xrightarrow{R} \theta(c)[z \leftarrow \theta(r_2)]$$

Teoremă 17 (Teorema Perechilor Critice $*$). *Dacă R este noetherian, atunci sunt echivalente:*

(1) R este confluent,
(2) $t_1 \downarrow_R t_2$ pentru orice pereche critică (t_1, t_2).

Algoritmul Knuth-Bendix.

- INTRARE: R un sistem de rescriere (TRS) noetherian.
- INIŢIALIZARE: T := R şi $>$ ordine de reducere pentru T
- Se execută următorii paşi, cât timp este posibil:
  (1) CP := CP(T) = $\{(t_1, t_2) \mid (t_1, t_2)$ pereche critică în $T\}$
  (2) Dacă $t_1 \downarrow t_2$, oricare $(t_1, t_2) \in CP$, atunci STOP *(T completarea lui R).*
  (3) Dacă $(t_1, t_2) \in CP$, $t_1 \not\downarrow t_2$ atunci:
    - dacă $fn(t_1) > fn(t_2)$ atunci T := T $\cup \{fn(t_1) \to fn(t_2)\}$,
    - dacă $fn(t_2) > fn(t_1)$ atunci T := T $\cup \{fn(t_2) \to fn(t_1)\}$,
    - altfel, STOP *(completare eşuată).*
- IEŞIRE: T completarea lui R sau eşec.