# Curs 14

# Cuprins

# Logica propozițională (recap.)

## Semantica logicii propoziționale

- Mulțimea valorilor de adevăr este $\{0, 1\}$ pe care considerăm următoarele operații:

| $x$ | $\neg x$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

| $x$ | $y$ | $x \rightarrow y$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$x \vee y := max\{x, y\} \qquad x \wedge y := min\{x, y\}$$

## Semantica logicii propoziționale

- o funcție $e : Var \to \{0,1\}$ se numește evaluare (interpretare)
- pentru orice evaluare $e : Var \to \{0,1\}$ există o unică funcție $e^+ : Form \to \{0,1\}$ care verifică următoarele proprietăți:
  - $e^+(v) = e(v)$
  - $e^+(\neg\varphi) = \neg e^+(\varphi)$
  - $e^+(\varphi \to \psi) = e^+(\varphi) \to e^+(\psi)$
  - $e^+(\varphi \wedge \psi) = e^+(\varphi) \wedge e^+(\psi)$
  - $e^+(\varphi \vee \psi) = e^+(\varphi) \vee e^+(\psi)$

  oricare ar fi $v \in Var$ și $\varphi$, $\psi \in Form$.

# Semantica logicii propoziționale

Cum verificăm că o formulă este tautologie: $\models \varphi$?

- Fie $v_1, \ldots, v_n$ variabilele care apar în $\varphi$.
- Cele $2^n$ evaluări posibile $e_1, \ldots, e_{2^n}$ pot fi scrise într-un tabel:

| $v_1$ | $v_2$ | $\ldots$ | $v_n$ | $\varphi$ |
|---|---|---|---|---|
| $e_1(v_1)$ | $e_1(v_2)$ | $\ldots$ | $e_1(v_n)$ | $e_1^+(\varphi)$ |
| $e_2(v_1)$ | $e_2(v_2)$ | $\ldots$ | $e_2(v_n)$ | $e_2^+(\varphi)$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $e_{2^n}(v_1)$ | $e_{2^n}(v_2)$ | $\ldots$ | $e_{2^n}(v_n)$ | $e_{2^n}^+(\varphi)$ |

  Fiecare evaluare corespunde unei linii din tabel!

- $\models \varphi$ dacă și numai dacă $e_1^+(\varphi) = \cdots = e_{2^n}^+(\varphi) = 1$

# Sistemul Hilbert

- Oricare ar fi $\varphi$, $\psi$, $\chi \in$ *Form* următoarele formule sunt axiome:

  (A1) $\varphi \to (\psi \to \varphi)$
  (A2) $(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$
  (A3) $(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$.

- Regula de deducție este modus ponens: $\dfrac{\varphi, \ \varphi \to \psi}{\psi}$ MP

- O demonstrație din ipotezele $\Gamma$ (sau $\Gamma$-demonstrație) pentru $\varphi$ este o secvență de formule $\gamma_1, \ldots, \gamma_n$ astfel încât $\gamma_n = \varphi$ și, pentru fiecare $i \in \{1, \ldots, n\}$, una din următoarele condiții este satisfăcută:
  - $\gamma_i$ este axiomă,
  - $\gamma_i \in \Gamma$
  - $\gamma_i$ se obține din formulele anterioare prin MP:
    există $j$, $k < i$ astfel încât $\gamma_j = \gamma_k \to \gamma_i$

- O formulă $\varphi$ este $\Gamma$-teoremă dacă are o $\Gamma$-demonstrație.
  Notăm prin $\Gamma \vdash \varphi$ faptul că $\varphi$ este o $\Gamma$-teoremă

### Teorema deducției TD (Herbrand, 1930)

Fie $\Gamma \cup \{\varphi\} \subseteq Form$. Atunci

$$\Gamma \vdash \varphi \to \psi \text{ dacă și numai dacă } \Gamma \cup \{\varphi\} \vdash \psi$$

# Sistemul Hilbert

## Exercițiu

Fie $\varphi$ și $\psi$ formule în logica propozițională. Să se arate sintactic că

$$\vdash \varphi \rightarrow (\neg\varphi \rightarrow \psi).$$

## Soluție

Avem următoarea demonstrație:

| | | | |
|---|---|---|---|
| (1) | $\{\varphi, \neg\varphi\}$ | $\vdash \neg\varphi \rightarrow (\neg\psi \rightarrow \neg\varphi)$ | (A1) |
| (2) | $\{\varphi, \neg\varphi\}$ | $\vdash \neg\varphi$ | (ipoteză) |
| (3) | $\{\varphi, \neg\varphi\}$ | $\vdash \neg\psi \rightarrow \neg\varphi$ | (1), (2), MP |
| (4) | $\{\varphi, \neg\varphi\}$ | $\vdash (\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$ | (A3) |
| (5) | $\{\varphi, \neg\varphi\}$ | $\vdash \varphi \rightarrow \psi$ | (3), (4), MP |
| (6) | $\{\varphi, \neg\varphi\}$ | $\vdash \varphi$ | (ipoteză) |
| (7) | $\{\varphi, \neg\varphi\}$ | $\vdash \psi$ | (5), (6), MP |
| (8) | $\{\varphi\}$ | $\vdash \neg\varphi \rightarrow \psi$ | (7) Teorema Deducției |
| (9) | | $\vdash \varphi \rightarrow (\neg\varphi \rightarrow \psi)$ | (8) Teorema Deducției |

# Deducția naturală

# Deducția naturală

- Numim secvent o expresie de forma

$$\varphi_1, \ldots, \varphi_n \vdash \psi$$

- Formulele $\varphi_1, \ldots, \varphi_n$ se numesc premise, iar $\psi$ se numește concluzie.
- Un secvent este valid dacă există o demonstrație folosind regulile de deducție.
- O teoremă este o formulă $\psi$ astfel încât $\vdash \psi$
  (adică $\psi$ poate fi demonstrată din mulțimea vidă de ipoteze).
- Pentru fiecare conector logic vom avea reguli de introducere și reguli de eliminare.

# Regulile deducției naturale

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi}\ (\wedge i) \qquad \frac{\varphi \wedge \psi}{\varphi}\ (\wedge e_1) \qquad \frac{\varphi \wedge \psi}{\psi}\ (\wedge e_2)$$

$$\frac{\boxed{\begin{array}{c}\varphi \\ \vdots \\ \psi\end{array}}}{\varphi \to \psi}\ (\to i) \qquad \frac{\varphi \quad \varphi \to \psi}{\psi}\ (\to e)$$

$$\frac{\varphi}{\varphi \vee \psi}\ (\vee i_1) \qquad \frac{\psi}{\varphi \vee \psi}\ (\vee i_2) \qquad \frac{\varphi \vee \psi \quad \boxed{\begin{array}{c}\varphi \\ \vdots \\ \chi\end{array}} \quad \boxed{\begin{array}{c}\psi \\ \vdots \\ \chi\end{array}}}{\chi}\ (\vee e)$$

$$\frac{\varphi}{\neg\neg\varphi}\ (\neg\neg i) \qquad \frac{\boxed{\begin{array}{c}\varphi \\ \vdots \\ \bot\end{array}}}{\neg\varphi}\ (\neg i) \qquad \frac{\neg\neg\varphi}{\varphi}\ (\neg\neg e) \qquad \frac{\varphi \quad \neg\varphi}{\bot}\ (\neg e) \qquad \frac{\bot}{\varphi}\ (\bot e)$$

# Deducție naturală

## Exercițiu

Demonstrați că următorul secvent este valid:

$$p \wedge q \to r \vdash p \to (q \to r)$$

## Soluție

| | | |
|---|---|---|
| 1 | $p \wedge q \to r$ | *premiza* |
| 2 | $p$ | *ipoteza* |
| 3 | $q$ | *ipoteza* |
| 4 | $p \wedge q$ | $(\wedge i)$,2,3 |
| 5 | $r$ | $(\to e)$,1,4 |
| 6 | $q \to r$ | $(\to i)$,3–5 |
| 7 | $p \to (q \to r)$ | $(\to i)$,2–6 |

# Deducție naturală

## Exercițiu

Demonstrați că următorul secvent este valid:

$$p \rightarrow q, p \rightarrow \neg q \vdash \neg p$$

## Soluție

| | | |
|---|---|---|
| 1 | $p \rightarrow q$ | *premiza* |
| 2 | $p \rightarrow \neg q$ | *premiza* |
| 3 | $p$ | *ipoteza* |
| 4 | $q$ | $(\rightarrow e)$,1,3 |
| 5 | $\neg q$ | $(\rightarrow e)$,2,3 |
| 6 | $\perp$ | $(\neg e)$,4,5 |
| 7 | $\neg p$ | $(\neg i)$,3–6 |

# Deducție naturală

## Exercițiu

*Echivalența logică* este definită prin $\varphi \leftrightarrow \psi = (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)$.
Găsiți reguli de introducere și eliminare pentru $\leftrightarrow$.

## Soluție

Observăm că $\leftrightarrow$ este o combinație între $\rightarrow$ și $\land$. Regulile pentru $\leftrightarrow$ se obțin combinând regulile pentru $\rightarrow$ și $\land$.

Introducerea $(\leftrightarrow i)$: pentru a introduce $\varphi \leftrightarrow \psi$ trebuie să introducem $\varphi \rightarrow \psi$ și $\psi \rightarrow \varphi$, apoi să introducem $\land$.

$$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \psi\end{array}} \quad \boxed{\begin{array}{c}\psi\\ \vdots\\ \varphi\end{array}}}{\varphi \leftrightarrow \psi}\ (\leftrightarrow i)$$

## Deducție naturală

### Soluție (cont.)

Eliminarea $(\leftrightarrow i)$: pentru a elimina $\varphi \leftrightarrow \psi$ trebuie să eliminăm $\wedge$ apoi să eliminăm o $\rightarrow$; vom avea două variante:

$$\frac{\varphi \leftrightarrow \psi \quad \psi}{\varphi}\ (\leftrightarrow e_1) \qquad \frac{\varphi \leftrightarrow \psi \quad \varphi}{\psi}\ (\leftrightarrow e_2)$$

# Clauze propoziționale definite

# Clauze propoziționale definite

O clauză definită este o formulă care poate avea una din formele:

1. $q$ (clauză unitate)
2. $p_1 \wedge \ldots \wedge p_k \rightarrow q$

unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale.

## Sistem de deducție pentru clauze definite propoziționale

Pentru o mulțime $S$ de clauze definite propoziționale, avem

- Axiome (premise): orice clauză din $S$
- Reguli de deducție:

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

# Mulțimi parțial ordonate

- O mulțime parțial ordonată (mpo) este o pereche $(M, \leq)$ unde $\leq \subseteq M \times M$ este o relație de ordine.
  - relație de ordine: reflexivă, antisimetrică, tranzitivă

- O mpo $(L, \leq)$ se numește lanț dacă este total ordonată, adică $x \leq y$ sau $y \leq x$ pentru orice $x, y \in L$. Vom considera lanțuri numărabile, i.e.
$$x_1 \leq x_2 \leq x_3 \leq \ldots$$

- O mpo $(C, \leq)$ este completă (CPO) dacă:
  - $C$ are prim element $\perp$ ($\perp \leq x$ oricare $x \in C$),
  - $\bigvee_n x_n$ există pentru orice lanț $x_1 \leq x_2 \leq x_3 \leq \ldots$

# Funcții monotone și continue

- Fie $(A, \leq_A)$ și $(B, \leq_B)$ mulțimi parțial ordonate.
  O funcție $f : A \to B$ este monotonă (crescătoare)
  dacă $a_1 \leq_A a_2$ implică $f(a_1) \leq_B f(a_2)$ oricare $a_1, a_2 \in A$.

- Fie $(A, \leq_A)$ și $(B, \leq_B)$ mulțimi parțial ordonate complete.
  O funcție $f : A \to B$ este continuă dacă
  $f(\bigvee_n a_n) = \bigvee_n f(a_n)$ pentru orice lanț $\{a_n\}_n$ din $A$.

- Observăm că orice funcție continuă este crescătoare.

# Teorema de punct fix

Un element $a \in C$ este punct fix al unei funcții $f : C \to C$ dacă $f(a) = a$.

### Teorema Knaster-Tarski pentru CPO

Fie $(C, \leq)$ o mulțime parțial ordonată completă și $\mathbf{F} : C \to C$ o funcție continuă. Atunci

$$a = \bigvee_n \mathbf{F}^n(\perp)$$

este cel mai mic punct fix al funcției $\mathbf{F}$.

# Puncte fixe

## Exercițiu

Care sunt punctele fixe ale următoarei funcții? Dar cel mai punct fix?

$$f_1 : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\}), \quad f_1(Y) = Y \cup \{1\}$$

## Soluție

Se observă că punctele fixe ale lui $f_1$ sunt submulțimile $Y$ ale lui $\{1,2,3\}$ care îl conțin pe 1 (dacă $1 \notin Y$, atunci $f_1(Y) = Y \cup \{1\}$ și evident $Y \neq Y \cup \{1\}$).

Deci punctele fixe ale lui $f_1$ sunt $\{1\}, \{1,2\}, \{1,3\}, \{1,2,3\}$.

Evident, cel mai mic punct fix este $\{1\}$.

# Puncte fixe

## Exercițiu

Care sunt punctele fixe ale următoarei funcții? Dar cel mai punct fix?

$$f_2 : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\}), \quad f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \emptyset & \text{altfel} \end{cases}$$

## Soluție

Se observă că singurele puncte fixe ale lui $f_2$ sunt $\emptyset$ și $\{1\}$. Evident $\emptyset$ este cel mai mic punct fix.

# Clauze definite și funcții monotone

Fie $A$ mulțimea atomilor $p_1, p_2, \ldots$ care apar în $S$.

Fie $Baza = \{p_i \mid p_i \in S\}$ mulțimea atomilor care apar în clauzele unitate din $S$.

Definim funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ prin

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

# Clauze definite și funcții monotone

## Exercițiu

Arătați că funcția $f_S$ este monotonă.

## Soluție

Fie $Y_1, Y_2 \subseteq A$ astfel încât $Y_1 \subseteq Y_2$. Trebuie să arătăm că $f_S(Y_1) \subseteq f_S(Y_2)$. Fie următoarele mulțimi:

$$\begin{aligned} Z_1 &= \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } S, s_1 \in Y_1, \ldots, s_n \in Y_1\}, \\ Z_2 &= \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } S, s_1 \in Y_2, \ldots, s_n \in Y_2\}. \end{aligned}$$

Deci $f_S(Y_1) = Y_1 \cup Baza \cup Z_1$ și $f_S(Y_2) = Y_2 \cup Baza \cup Z_2$.
Cum $Y_1 \subseteq Y_2$, rămâne să arătăm doar că $Z_1 \subseteq Z_2$.
Fie $a \in Z_1$. Atunci există $s_1 \wedge \ldots \wedge s_n \to a \in S$ și $s_1, \ldots, s_n \in Y_1$.
Deci $s_1, \ldots, s_n \in Y_2$, de unde rezultă că $a \in Z_2$.

# Clauze definite și funcții monotone

Pentru funcția continuă $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$

$$\boxed{\begin{aligned} f_S(Y) = Y \cup Baza & \\ & \cup \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } S, \\ & \qquad\qquad s_1 \in Y, \ldots, s_n \in Y\} \end{aligned}}$$

aplicând Teorema Knaster-Tarski pentru CPO, obținem că

$$\bigcup_n f_S^n(\emptyset)$$

este cel mai mic punct fix al lui $f_S$.

# Cel mai mic punct fix

## Exercițiu

Calculați cel mai mic punct fix pentru functia $f_{S_1}$ unde

$$S_1 = \{x_1 \wedge x_2 \to x_3,\ x_4 \wedge x_2 \to x_5,\ x_2,\ x_6,\ x_6 \to x_1\}$$

## Soluție

Observăm că $A = \{x_1, x_2, \ldots, x_6\}$ și $Baza = \{x_2, x_6\}$.
Cum $f_S$ este continuă, aplicăm Teorema Knaster-Tarski pentru a calcula cel mai mic punct fix:

$$\begin{aligned}
&f_{S_1}(\emptyset) = Baza = \{x_2, x_6\}\\
&f_{S_1}(\{x_2, x_6\}) = \{x_2, x_6, x_1\}\\
&f_{S_1}(\{x_2, x_6, x_1\}) = \{x_2, x_6, x_1, x_3\}\\
&f_{S_1}(\{x_2, x_6, x_1, x_3\}) = \{x_2, x_6, x_1, x_3\}
\end{aligned}$$

În concluzie, cel mai mic punct fix căutam este $\{x_2, x_6, x_1, x_3\}$.

# Programe logice și cel mai mic punct fix

**Teoremă**

*Fie $X$ este cel mai mic punct fix al funcției $f_S$. Atunci $q \in X$ ddacă $S \models q$.*

Intuiție: Cel mai mic punct fix al funcției $f_S$ este mulțimea tuturor atomilor care sunt consecințe logice ale programului.

Avem o metodă de decizie *(decision procedure)* pentru a verifica $S \vdash q$. Metoda constă în:

- calcularea celui mai mic punct fix $X$ al funcției $f_S$
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Această metodă se termină.

# Logica de ordinul I (recap.)

# Logica de ordinul I - sintaxa

Limbaj de ordinul I $\mathcal{L}$

- unic determinat de $\tau = (\mathbf{R}, \mathbf{F}, \mathbf{C}, ari)$

Termenii lui $\mathcal{L}$, notați $Trm_{\mathcal{L}}$, sunt definiți inductiv astfel:

- orice variabilă este un termen;
- orice simbol de constantă este un termen;
- dacă $f \in \mathbf{F}$, $ar(f) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $f(t_1, \ldots, t_n)$ este termen.

Formulele atomice ale lui $\mathcal{L}$ sunt definite astfel:

- dacă $R \in \mathbf{R}$, $ar(R) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $R(t_1, \ldots, t_n)$ este formulă atomică.

Formulele lui $\mathcal{L}$ sunt definite astfel:

- orice formulă atomică este o formulă
- dacă $\varphi$ este o formulă, atunci $\neg\varphi$ este o formulă
- dacă $\varphi$ și $\psi$ sunt formule, atunci $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ sunt formule
- dacă $\varphi$ este o formulă și $x$ este o variabilă, atunci $\forall x\, \varphi$, $\exists x\, \varphi$ sunt formule

# Logica de ordinul I - semantică

O structură este de forma $\mathcal{A} = (A, \mathbf{F}^{\mathcal{A}}, \mathbf{R}^{\mathcal{A}}, \mathbf{C}^{\mathcal{A}})$, unde

- $A$ este o mulțime nevidă
- $\mathbf{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathbf{F}\}$ este o mulțime de operații pe $A$; dacă $f$ are aritatea $n$, atunci $f^{\mathcal{A}} : A^n \to A$.
- $\mathbf{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathbf{R}\}$ este o mulțime de relații pe $A$; dacă $R$ are aritatea $n$, atunci $R^{\mathcal{A}} \subseteq A^n$.
- $\mathbf{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathbf{C}\}$.

O interpretare a variabilelor lui $\mathcal{L}$ în $\mathcal{A}$ ($\mathcal{A}$-interpretare) este o funcție $I : V \to A$.

Inductiv, definim interpretarea termenului $t$ în $\mathcal{A}$ sub $I$ notat $t_I^{\mathcal{A}}$.

Inductiv, definim când o formulă este adevărată în $\mathcal{A}$ în interpretarea $I$ notat $\mathcal{A}, I \vDash \varphi$. În acest caz spunem că $(\mathcal{A}, I)$ este model pentru $\varphi$.

O formulă $\varphi$ este adevărată într-o structură $\mathcal{A}$, notat $\mathcal{A} \vDash \varphi$, dacă este adevărată în $\mathcal{A}$ sub orice interpretare. Spunem că $\mathcal{A}$ este model al lui $\varphi$.

O formulă $\varphi$ este adevărată în logica de ordinul I, notat $\vDash \varphi$, dacă este adevărată în orice structură. O formulă $\varphi$ este validă dacă $\vDash \varphi$.

O formulă $\varphi$ este satisfiabilă dacă există o structură $\mathcal{A}$ și o $\mathcal{A}$-interpretare $I$ astfel încât $\mathcal{A}, I \vDash \varphi$.

# Validitate și satisfiabilitate

**Propoziție**

Dacă $\varphi$ este o formulă atunci

$\varphi$ este validă  dacă și numai dacă  $\neg\varphi$ nu este satisfiabilă.

# Algoritmul de unificare

# Unificare

- O subtituție $\sigma$ este o funcție (parțială) de la variabile la termeni,

$$\sigma : V \to Trm_{\mathcal{L}}$$

- Doi termeni $t_1$ și $t_2$ se unifică dacă există o substituție $\theta$ astfel încât

$$\theta(t_1) = \theta(t_2).$$

- În acest caz, $\theta$ se numește unificatorul termenilor $t_1$ și $t_2$.
- Un unificator $\nu$ pentru $t_1$ și $t_2$ este un cel mai general unificator (cgu,mgu) dacă pentru orice alt unificator $\nu'$ pentru $t_1$ și $t_2$, există o substituție $\mu$ astfel încât

$$\nu' = \nu ; \mu.$$

# Algoritmul de unificare

- Pentru o mulțime finită de termeni $\{t_1, \ldots, t_n\}$, $n \geq 2$, algoritmul de unificare stabilește dacă există un cgu.

- Algoritmul lucrează cu două liste:
  - Lista soluție: $S$
  - Lista de rezolvat: $R$

- Inițial:
  - Lista soluție: $S = \emptyset$
  - Lista de rezolvat: $R = \{t_1 \doteq t_2, \ldots, t_{n-1} \doteq t_n\}$

- $\doteq$ este un simbol nou care ne ajută sa formăm perechi de termeni (ecuații).

# Algoritmul de unificare

Algoritmul constă în aplicarea regulilor de mai jos:

- SCOATE
  - orice ecuație de forma $t \doteq t$ din $R$ este eliminată.
- DESCOMPUNE
  - orice ecuație de forma $f(t_1, \ldots, t_n) \doteq f(t_1', \ldots, t_n')$ din $R$ este înlocuită cu ecuațiile $t_1 \doteq t_1', \ldots, t_n \doteq t_n'$.
- REZOLVĂ
  - orice ecuație de forma $x \doteq t$ sau $t \doteq x$ din $R$, unde variabila $x$ nu apare în termenul $t$, este mutată sub forma $x \doteq t$ în $S$.
    În toate celelalte ecuații (din $R$ și $S$), $x$ este înlocuit cu $t$.

# Algoritmul de unificare

Algoritmul se termină normal dacă $R = \emptyset$. În acest caz, $S$ dă cgu.

Algoritmul este oprit cu concluzia inexistenței unui cgu dacă:

1. În $R$ există o ecuație de forma
$$f(t_1, \ldots, t_n) \doteq g(t'_1, \ldots, t'_k) \text{ cu } f \neq g.$$
2. În $R$ există o ecuație de forma $x \doteq t$ sau $t \doteq x$ și variabila $x$ apare în termenul $t$.

# Algoritmul de unificare - schemă

| | Lista soluție<br>S | Lista de rezolvat<br>R |
|---|---|---|
| Inițial | $\emptyset$ | $t_1 \doteq t'_1, \ldots, t_n \doteq t'_n$ |
| 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 $t$ |
| | $x \doteq t, \; S[x/t]$ | $R'[x/t]$ |
| Final | $S$ | $\emptyset$ |

$S[x/t]$: în toate ecuațiile din $S$, $x$ este înlocuit cu $t$

# Exemplu

## Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)\}$ au gcu?

| $S$ | $R$ | |
|---|---|---|
| $\emptyset$ | $g(y) \doteq x,\ f(x, h(x), y) \doteq f(g(z), w, z)$ | REZOLVĂ |
| $x \doteq g(y)$ | $f(g(y), h(g(y)), y) \doteq f(g(z), w, z)$ | DESCOMPUNE |
| $x \doteq g(y)$ | $g(y) \doteq g(z),\ h(g(y)) \doteq w,\ y \doteq z$ | REZOLVĂ |
| $w \doteq h(g(y))$, $x \doteq g(y)$ | $g(y) \doteq g(z),\ y \doteq z$ | REZOLVĂ |
| $y \doteq z, x \doteq g(z)$, $w \doteq h(g(z))$ | $g(z) \doteq g(z)$ | SCOATE |
| $y \doteq z, x \doteq g(z)$, $w \doteq h(g(z))$ | $\emptyset$ | |

- $\nu = \{y/z,\ x/g(z),\ w/h(g(z))\}$ este cgu.

# Exemplu

## Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)\}$ au gcu?

| $S$ | $R$ | |
|---|---|---|
| $\emptyset$ | $g(y) \doteq x,\ f(x, h(y), y) \doteq f(g(z), b, z)$ | REZOLVĂ |
| $x \doteq g(y)$ | $f(g(y), h(y), y) \doteq f(g(z), b, z)$ | DESCOMPUNE |
| $x \doteq g(y)$ | $g(y) \doteq g(z),\ h(y) \doteq b,\ y \doteq z$ | - EȘEC - |

- *h* și *b* sunt simboluri de operații diferite!
- Nu există unificator pentru ecuațiile din $U$.

# Exemplu

## Exemplu

- Ecuațiile $\{g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)\}$ au gcu?

| $S$ | $R$ | |
|---|---|---|
| $\emptyset$ | $g(y) \doteq x,\ f(x, h(x), y) \doteq f(y, w, z)$ | REZOLVĂ |
| $x \doteq g(y)$ | $f(g(y), h(g(y)), y) \doteq f(y, w, z)$ | DESCOMPUNE |
| $x \doteq g(y)$ | $g(y) \doteq y,\ h(g(y)) \doteq w,\ y \doteq z$ | - EȘEC - |

- În ecuația $g(y) \doteq y$, variabila $y$ apare în termenul $g(y)$.
- Nu există unificator pentru ecuațiile din $U$.

# Forme prenex și Skolem. Modele Herbrand

# Variabile libere. Variabile legate. Enunțuri

Fie $\varphi$ o formulă și $Var(\varphi)$ mulțimea variabilelor care apar în $\varphi$.

- Variabilele libere ale unei formule $\varphi$ sunt variabilele care nu sunt cuantificate.
- Mulțimea $FV(\varphi)$ a variabilelor libere ale unei formule $\varphi$ poate fi definită prin inducție după formule:

$$
\begin{array}{lcll}
FV(\varphi) & = & Var(\varphi), & \text{dacă } \varphi \text{ este formulă atomică} \\
FV(\neg\varphi) & = & FV(\varphi) & \\
FV(\varphi \circ \psi) & = & FV(\varphi) \cup FV(\psi), & \text{dacă } \circ \in \{\rightarrow, \vee, \wedge\} \\
FV(\forall x\, \varphi) & = & FV(\varphi) - \{x\} & \\
FV(\exists x\, \varphi) & = & FV(\varphi) - \{x\} &
\end{array}
$$

- O variabilă $v \in Var(\varphi)$ care nu este liberă se numește legată în $\varphi$.
- Un enunț este o formulă fără variabile libere.
- Pentru orice structură $\mathcal{A}$ și orice enunț $\varphi$, o $\mathcal{A}$-interpretare $I$ nu joacă niciun rol în a determina dacă $\mathcal{A}, I \vDash \varphi$.

# Enunțuri

Fie $\varphi$ o formulă și $FV(\varphi) = \{x_1, \ldots, x_n\}$.

**Propozitie**

Pentru orice structură $\mathcal{A}$ avem

$$\mathcal{A} \vDash \varphi \text{ dacă și numai dacă } \mathcal{A} \vDash \forall x_1 \cdots \forall x_n\, \varphi.$$

A verifica validitatea unei formule revine la
a verifica validitatea enunțului asociat.

# Substituții și formule echivalente

- Substituțiile înlocuiesc variabilele libere cu termeni.
- O substituție aplicată unui termen întoarce un alt termen.

- Fie $\varphi$ o formulă și $t_1, \ldots, t_n$ termeni care nu conțin variabile din $\varphi$. Notăm $\varphi[x_1/t_1, \ldots, x_n/t_n]$ formula obținută din $\varphi$ substituind toate aparițiile libere ale lui $x_1, \ldots, x_n$ cu $t_1, \ldots, t_n$.

$$\varphi[x_1/t_1, \ldots, x_n/t_n] = \{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}\varphi$$

- Notăm prin $\varphi \mathrel{=\!\!\!\mid\!\!\!\models} \psi$ faptul că $\models \varphi \leftrightarrow \psi$, adică $\varphi$ și $\psi$ au aceleași modele.

# Forma rectificată

- O formulă $\varphi$ este în formă rectificată dacă:
  1. nici o variabilă nu apare și liberă și legată
  2. cuantificatori distincți leagă variabile distincte

- Pentru orice formulă $\varphi$ există o formulă $\varphi^r$ în formă rectificată astfel încât $\varphi \mathrel{\text{⫤⊨}} \varphi^r$.

- Intuitiv, forma rectificată a unei formule se obține prin redenumirea variabilelor astfel încât să nu apară conflicte.

În continuare vom presupune că
toate formulele sunt în formă rectificată.

# Forma prenex

O formulă prenex este o formulă de forma

$$Q_1 x_1 \, Q_2 x_2 \ldots Q_n x_n \, \varphi$$

unde $Q_i \in \{\forall, \exists\}$ pentru orice $i \in \{1, \ldots, n\}$, $x_1, \ldots, x_n$ sunt variabile distincte și $\varphi$ nu conține cuantificatori.

# Cum calculăm forma prenex?

- Se înlocuiesc $\rightarrow$ și $\leftrightarrow$ :

$$\begin{aligned} \varphi \rightarrow \psi &\quad \mathrel{\text{⊨}\!\!\!\text{⫤}} \quad \neg\varphi \vee \psi \\ \varphi \leftrightarrow \psi &\quad \mathrel{\text{⊨}\!\!\!\text{⫤}} \quad (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi) \end{aligned}$$

- Se aplică următoarele echivalențe:

$$\begin{aligned} \neg\exists x\, \neg\varphi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \forall x\, \varphi & \qquad \forall x\, \varphi \wedge \forall x\, \psi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \forall x\, (\varphi \wedge \psi) \\ \neg\forall x\, \neg\varphi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \exists x\, \varphi & \qquad \exists x\, \varphi \vee \exists x\, \psi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \exists x\, (\varphi \vee \psi) \end{aligned}$$

$$\begin{aligned} \neg\exists x\, \varphi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \forall x\, \neg\varphi & \qquad \forall x\, \forall y\, \varphi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \forall y\, \forall x\, \varphi \\ \neg\forall x\, \varphi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \exists x\, \neg\varphi & \qquad \exists x\, \exists y\, \varphi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \exists y\, \exists x\, \varphi \end{aligned}$$

$$\begin{aligned} \forall x\, \varphi \vee \psi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \forall x\, (\varphi \vee \psi) \text{ dacă } x \notin FV(\psi) \\ \forall x\, \varphi \wedge \psi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \forall x\, (\varphi \wedge \psi) \text{ dacă } x \notin FV(\psi) \\ \exists x\, \varphi \vee \psi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \exists x\, (\varphi \vee \psi) \text{ dacă } x \notin FV(\psi) \\ \exists x\, \varphi \wedge \psi &\;\mathrel{\text{⊨}\!\!\!\text{⫤}}\; \exists x\, (\varphi \wedge \psi) \text{ dacă } x \notin FV(\psi) \end{aligned}$$

# Forma prenex

## Exerciţiu

Considerăm un limbaj de ordinul I cu $\mathbf{R} = \{P, R, Q\}$ cu $ari(P) = 1$ şi $ari(R) = ari(Q) = 2$.
Găsiţi forma echivalentă prenex pentru următoarea formulă:

$$\forall x \exists y (R(x, y) \to R(y, x)) \to \exists x R(x, x)$$

## Soluţie

$$\begin{array}{lll}
 & \forall x \exists y (R(x, y) \to R(y, x)) \to \exists x R(x, x) & \\
\boxminus & \forall x \exists y (R(x, y) \to R(y, x)) \to \exists z R(z, z) & \text{(redenumim variabile)} \\
\boxminus & \neg \forall x \exists y (\neg R(x, y) \vee R(y, x)) \vee \exists z R(z, z) & \\
\boxminus & \exists x \forall y (R(x, y) \wedge \neg R(y, x)) \vee \exists z R(z, z) & \\
\boxminus & \exists z (\exists x \forall y (R(x, y) \wedge \neg R(y, x)) \vee R(z, z)) & \\
\boxminus & \exists z \exists x (\forall y (R(x, y) \wedge \neg R(y, x)) \vee R(z, z)) & \\
\boxminus & \exists z \exists x \forall y ((R(x, y) \wedge \neg R(y, x)) \vee R(z, z)) &
\end{array}$$

# Forma Skolem

Fie $\varphi$ enunț în formă prenex. Definim $\varphi^{sk}$ și $\mathcal{L}^{sk}(\varphi)$ astfel:

- dacă $\varphi$ este liberă de cuantificatori, atunci $\varphi^{sk} = \varphi$ și $\mathcal{L}^{sk}(\varphi) = \mathcal{L}$,
- dacă $\varphi$ este universală, atunci $\varphi^{sk} = \varphi$ și $\mathcal{L}^{sk}(\varphi) = \mathcal{L}$,
- dacă $\varphi = \exists x\, \psi$ atunci introducem un nou simbol de constantă $c$ și considerăm $\varphi^1 = \psi[x/c]$, $\mathcal{L}^1 = \mathcal{L} \cup \{c\}$.
- dacă $\varphi = \forall x_1 \ldots \forall x_k\, \exists x\, \psi$ atunci introducem un nou simbol de funcție $f$ de aritate $k$ și considerăm $\mathcal{L}^1 = \mathcal{L} \cup \{f\}$,

$$\varphi^1 = \forall x_1 \ldots \forall x_k\, \psi[x/f(x_1 \ldots x_k)]$$

În ambele cazuri, $\varphi^1$ are cu un cuantificator existențial mai puțin decât $\varphi$. Dacă $\varphi^1$ este liberă de cuantificatori sau universală, atunci $\varphi^{sk} = \varphi^1$. Dacă $\varphi^1$ nu este universală, atunci formăm $\varphi^2, \varphi^3, \ldots$, până ajungem la o formulă universală și aceasta este $\varphi^{sk}$.

**Definiție**

$\varphi^{sk}$ este o formă Skolem a lui $\varphi$.

# Forma Skolem

## Exercițiu

Considerăm un limbaj de ordinul I cu $\mathbf{C} = \{b\}$ și $\mathbf{R} = \{P, R, Q\}$ cu $ari(P) = 1$ și $ari(R) = ari(Q) = 2$.

Găsiți forma Skolem pentru următoarea formulă în formă prenex

$$\varphi = \forall x \exists y \forall z \exists w (R(x, y) \wedge (R(y, z) \rightarrow (R(z, w) \wedge R(w, w))))$$

## Soluție

$\varphi_1 = \forall x \forall z \exists w (R(x, f(x)) \wedge (R(f(x), z) \rightarrow (R(z, w) \wedge R(w, w))))$
$(y \mapsto f(x))$

$\varphi_2 = \forall x \forall z (R(x, f(x)) \wedge (R(f(x), z) \rightarrow$
$(R(z, g(x, z)) \wedge R(g(x, z), g(x, z)))))$
$(w \mapsto g(x, z))$

$\varphi^{sk} = \varphi_2$

# Model Herbrand

Fie $\mathcal{L}$ un limbaj de ordinul I.

- Presupunem că are cel puţin un simbol de constantă!
- Dacă nu are, adăugăm un simbol de constantă.

Universul Herbrand este mulţimea $T_{\mathcal{L}}$ a tututor termenilor fără variabile.

O structură Herbrand este o structură $\mathcal{H} = (T_{\mathcal{L}}, \mathbf{F}^{\mathcal{H}}, \mathbf{R}^{\mathcal{H}}, \mathbf{C}^{\mathcal{H}})$, unde

- pentru orice simbol de constantă $c$, $c^{\mathcal{H}} = c$
- pentru orice simbol de funcţie $f$ de aritate $n$, $f^{\mathcal{H}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$

Atenţie! Într-o structură Herbrand nu fixăm o definiţie pentru relaţii: pentru orice simbol de relaţie $R$ de aritate $n$, $R^{\mathcal{H}}(t_1, \ldots, t_n) \subseteq (T_{\mathcal{L}})^n$

O interpretare Herbrand este o interpretare $H : V \to T_{\mathcal{L}}$

O structură Herbrand $\mathcal{H}$ este model al unei formule $\varphi$ dacă $\mathcal{H} \vDash \varphi$. În acest caz spunem că $\mathcal{H}$ este model Herbrand al lui $\varphi$.

# Teorema lui Herbrand

**Teorema lui Herbrand**

Fie $n \geq 0$ și $\varphi = \forall x_k \ldots \forall x_1 \, \psi$ un enunț în forma Skolem.
Atunci $\varphi$ are un model dacă și numai dacă are un model Herbrand.

Teorema lui Herbrand reduce problema satisfiabilității la găsirea unui model Herbrand.

# Universul Herbrand al unei formule

Fie $\varphi$ un enunț în forma Skolem, adică $\varphi = \forall x_1 \ldots \forall x_n \, \psi$.

Definim $T(\varphi)$, universul Herbrand al formulei $\varphi$, astfel:

- dacă $c$ este o constantă care apare în $\varphi$ atunci $c \in T(\varphi)$,
- dacă $\varphi$ nu conține nicio constantă atunci alegem o constantă arbitrară $c$ și considerăm că $c \in T(\varphi)$,
- dacă $f$ este un simbol de funcție care apare în $\varphi$ cu $ari(f) = n$ și $t_1, \ldots, t_n \in T(\varphi)$ atunci $f(t_1, \ldots, t_n) \in T(\varphi)$.

Intuitiv, $T(\varphi)$ este mulțimea termenilor care se pot construi folosind simbolurile de funcții care apar în $\varphi$.

Definim extensia Herbrand a lui $\varphi$ astfel

$$\mathcal{H}(\varphi) = \{\psi[x_1/t_1, \ldots, x_n/t_n] \mid t_1, \ldots, t_n \in T(\varphi)\}$$

# Extensia Herbrand a unei formule

Fie $\varphi$ un enunț în forma Skolem, adică $\varphi = \forall x_1 \ldots \forall x_n \, \psi$.

**Teoremă**

Sunt echivalente:

- $\varphi$ este satisfiabilă,
- $\varphi$ are un model Herbrand $\mathcal{H}$ cu proprietatea că $\mathbf{R}^{\mathcal{H}} \subseteq T(\varphi)^n$ pentru orice relație $R \in \mathbf{R}$ cu $ari(R) = n$ care apare în $\varphi$,
- mulțimea de formule $\mathcal{H}(\varphi)$ este satisfiabilă.

# Extensia Herbrand a unei formule

## Exercițiu

Considerăm un limbaj de ordinul I cu $\mathbf{F} = \{f, g\}$ cu $ari(f) = 2$ și $ari(g) = 1$, $\mathbf{C} = \{b, c\}$ și $\mathbf{R} = \{P, Q\}$ cu $ari(P) = 3, ari(Q) = 2$ .
Descrieți termenii din universul Herbrand și formulele din expansiunea Herbrand a următoarei formule:

$$\varphi := \forall x \forall y\, P(c, f(x, b), g(y))$$

## Soluție

Universul Herbrand
$T(\varphi) = \{b, c, g(b), g(c), g(g(b)), g(g(c)), \ldots,$
$f(b, c), f(b, g(b)), f(b, g(c)), f(g(c), b), f(g(c), g(c)), \ldots\}$

Expansiunea Herbrand
$\mathcal{H}(\varphi) = \{P(c, f(b, b), g(b)), P(c, f(b, b), g(c)),$
$P(c, f(c, b), g(b)), P(c, f(g(b), b), g(g(g(b)))), \ldots\}$

# Logica de ordinul I

- Cercetarea validității poate fi redusă la cercetarea satisfiabilității.
- Cercetarea satisfiabilității unei formule poate fi redusă la cercetarea satisfiabilității unui enunț în forma Skolem.
- Teorema lui Herbrand reduce verificarea satisfiabilitătii unui enunț în forma Skolem la verificarea satisfiabilității în universul Herbrand.
- În situații particulare Teorema lui Herbrand ne dă o procedură de decizie a satisfiabilității, dar acest fapt nu este adevărat în general:

  dacă limbajul $\mathcal{L}$ conține cel putin o constantă și cel puțin un simbol de funcție $f$ cu $ari(f) \geq 1$ atunci universul Herbrand $T_{\mathcal{L}}$ este infinit.

# Logica de ordinul I

Problema validității

- nu este decidabilă.
- este semi-decidabilă.

Problema satisfiabilității

- nu este decidabilă.
- nu este semi-decidabilă.

# Formă clauzală. Rezoluție

# Literali. FNC

- În logica propozițională un literal este o variabilă sau negația unei variabile.

$$literal := \; p \mid \neg p \qquad \text{unde } p \text{ este variabilă propozițională}$$

- În logica de ordinul I un literal este o formulă atomică sau negația unei formule atomice.

$$literal := \; P(t_1, \ldots, t_n) \mid \neg P(t_1, \ldots, t_n)$$

unde $P \in \mathbf{R}, ari(P) = n$, și $t_1, \ldots, t_n$ sunt termeni.

- Pentru un literal $L$ vom nota cu $L^c$ literalul complement.

O formulă este în formă normală conjunctivă (FNC) dacă este o conjuncție de disjuncții de literali.

# Forma clauzală în logica propozițională

- Pentru orice formulă $\alpha$ există o FNC $\alpha^{fc}$ astfel încât $\alpha \mathrel{\text{⫤}\!\models} \alpha^{fc}$.
- Pentru o formulă din logica propozițională determinăm FNC corespunzătoare prin următoarele transformări:

1. înlocuirea implicațiilor și echivalențelor

$$\begin{aligned} \varphi \to \psi &\quad \mathrel{\text{⫤}\!\models} \quad \neg\varphi \vee \psi \\ \varphi \leftrightarrow \psi &\quad \mathrel{\text{⫤}\!\models} \quad (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi) \end{aligned}$$

2. regulile De Morgan

$$\begin{aligned} \neg(\varphi \vee \psi) &\quad \mathrel{\text{⫤}\!\models} \quad \neg\varphi \wedge \neg\psi \\ \neg(\varphi \wedge \psi) &\quad \mathrel{\text{⫤}\!\models} \quad \neg\varphi \vee \neg\psi \end{aligned}$$

3. principiului dublei negații

$$\neg\neg\psi \quad \mathrel{\text{⫤}\!\models} \quad \psi$$

4. distributivitatea

$$\begin{aligned} \varphi \vee (\psi \wedge \chi) &\quad \mathrel{\text{⫤}\!\models} \quad (\varphi \vee \psi) \wedge (\varphi \vee \chi) \\ (\psi \wedge \chi) \vee \varphi &\quad \mathrel{\text{⫤}\!\models} \quad (\psi \vee \varphi) \wedge (\chi \vee \varphi) \end{aligned}$$

# Forma clauzală în logica de ordinul I

- O formulă este formă normală conjunctivă prenex (FNCP) dacă
  - este în formă prenex $Q_1 x_1 \ldots Q_n x_n \psi$ ($Q_i \in \{\forall, \exists\}$ oricare $i$)
  - $\psi$ este FNC
- O formulă este formă clauzală dacă este enunț universal și FNCP:

$$\forall x_1 \ldots \forall x_n \psi \text{ unde } \psi \text{ este FNC}$$

- Pentru orice formulă $\varphi$ din logica de ordinul I există o formă clauzală $\varphi^{fc}$ astfel încât

$$\varphi \text{ este satisfiabilă dacă și numai dacă } \varphi^{fc} \text{ este satisfiabilă}$$

- Pentru o formulă $\varphi$, forma clauzală $\varphi^{fc}$ se poate calcula astfel:
  1. se determină forma rectificată
  2. se cuantifică universal variabilele libere
  3. se determină forma prenex
  4. se determină forma Skolem

     în acest moment am obținut o formă Skolem $\forall x_1 \ldots \forall x_n \psi$
  5. se determină o FNC $\psi'$ astfel încât $\psi \mathrel{\text{⫤}}\models \psi'$
  6. $\varphi^{fc}$ este $\forall x_1 \ldots \forall x_n \psi'$

# Clauze

- O clauză este o disjuncție de literali.
- Dacă $L_1, \ldots, L_n$ sunt literali atunci clauza $L_1 \vee \ldots \vee L_n$ o vom scrie ca mulțimea $\{L_1, \ldots, L_n\}$

  clauză = mulțime de literali

- Clauza $C = \{L_1, \ldots, L_n\}$ este satisfiabilă dacă $L_1 \vee \ldots \vee L_n$ este satisfiabilă.
- O clauză $C$ este trivială dacă conține un literal și complementul lui.
- Când $n = 0$ obținem clauza vidă, care se notează $\square$
- Prin definiție, clauza $\square$ nu este satisfiabilă.

# Forma clauzală

- Observăm că o FNC este o conjuncție de clauze.
- Dacă $C_1, \ldots, C_k$ sunt clauze atunci $C_1 \wedge \ldots \wedge C_k$ o vom scrie ca mulțimea $\{C_1, \ldots, C_k\}$

$$\text{FNC} = \text{mulțime de clauze}$$

- O mulțime de clauze $\mathcal{C} = \{C_1, \ldots, C_k\}$ este satisfiabilă dacă $C_1 \wedge \ldots \wedge C_k$ este satisfiabilă
- Când $k = 0$ obținem mulțimea de clauze vidă, pe care o notăm $\{\}$
- Prin definiție, mulțimea de clauze vidă $\{\}$ este satisfiabilă.

$$\{\} \text{ este satisfiabilă, dar } \{\square\} \text{ nu este satisfiabilă}$$

# Forma clauzală

- Dacă $\varphi$ este o formulă în calculul propozițional, atunci

$$\varphi^{fc} = \bigwedge_{i=1}^{k} \bigvee_{j=1}^{n_i} L_{ij} \text{ unde } L_{ij} \text{ sunt literali}$$

- Dacă $\varphi$ o formulă în logica de ordinul I, atunci

$$\varphi^{fc} = \forall x_1 \ldots \forall x_n \left( \bigwedge_{i=1}^{k} \bigvee_{j=1}^{n_i} L_{ij} \right) \text{ unde } L_{ij} \text{ sunt literali}$$

$\varphi$ este satisfiabilă dacă și numai dacă

$\varphi^{fc}$ este satisfiabilă dacă și numai dacă

$\{\{L_{11}, \ldots, L_{1n_1}\}, \ldots, \{L_{k1}, \ldots, L_{kn_k}\}\}$ este satisfiabilă

# Rezoluția în logica propozițională (recap.)

# Regula rezoluției

$$Rez \quad \frac{C_1 \cup \{p\}, C_2 \cup \{\neg p\}}{C_1 \cup C_2}$$

unde $C_1, C_2$ clauze, iar $p$ este variabila propozițională astfel încât $\{p, \neg p\} \cap C_1 = \emptyset$ și $\{p, \neg p\} \cap C_2 = \emptyset$.

Fie $\mathcal{C}$ o mulțime de clauze. O derivare prin rezoluție din $\mathcal{C}$ este o secvență finită de clauze astfel încât fiecare clauză este din $\mathcal{C}$ sau rezultă din clauzele anterioare prin rezoluție (este rezolvent).

# Procedura Davis-Putnam DPP (informal)

**Intrare:** o mulțime $\mathcal{C}$ de clauze

Se repetă următorii pași:

- se elimină clauzele triviale
- se alege o variabilă $p$
- se adaugă la mulțimea de clauze toți rezolvenții obținuti prin aplicarea *Rez* pe variabila $p$
- se șterg toate clauzele care conțin $p$ sau $\neg p$

**Ieșire:** dacă la un pas s-a obținut $\square$, mulțimea $\mathcal{C}$ nu este satisfiabilă; altfel $\mathcal{C}$ este satisfiabilă.

# Procedura Davis-Putnam DPP

## Exercițiu

Folosind algoritmul Davis-Putnam, cercetați dacă următoarea mulțime de clauze din calculul propozițional este satisfiabilă:

$$\mathcal{C} = \{\{v_0\}, \{\neg v_0, v_1\}, \{\neg v_1, v_2, v_3\}, \{\neg v_3, v_4\}, \{\neg v_4\}, \{\neg v_2\}\}$$

## Soluție

**Pasul 1.**
Alegem variabila $v_0$ și selectăm $\mathcal{C}_0^{v_0} := \{\{v_0\}\}$, $\mathcal{C}_0^{\neg v_0} = \{\{\neg v_0, v_1\}\}$.
Mulțimea rezolvenților posibili este $\mathcal{R}_0 := \{\{v_1\}\}$.
Se elimină clauzele în care apare $v_0$, adăugăm rezolvenții și obținem:

$$\mathcal{C}_1 := \{\{\neg v_1, v_2, v_3\}, \{\neg v_3, v_4\}, \{\neg v_4\}, \{\neg v_2\}, \{v_1\}\}$$

## Soluție (cont.)

**Pasul 2.**
Alegem variabila $v_1$ și selectăm $\mathcal{C}_1^{v_1} := \{\{v_1\}\}$ și $\mathcal{C}_1^{\neg v_1} := \{\{\neg v_1, v_2, v_3\}\}$.
Mulțimea rezolvenților posibili este $\mathcal{R}_1 := \{\{v_2, v_3\}\}$.
Se elimină clauzele în care apare $v_1$, adăugăm rezolvenții și obținem:
$\mathcal{C}_2 := \{\{\neg v_3, v_4\}, \{\neg v_4\}, \{\neg v_2\}, \{v_2, v_3\}\}$.

**Pasul 3.**
Alegem variabila $v_2$ și selectăm $\mathcal{C}_2^{v_2} := \{\{v_2, v_3\}\}$, $\mathcal{C}_2^{\neg v_2} := \{\{\neg v_2\}\}$.
Mulțimea rezolvenților posibili este $\mathcal{R}_2 := \{\{v_3\}\}$.
Se elimină clauzele în care apare $v_2$, adăugăm rezolvenții și obținem:
$\mathcal{C}_3 := \{\{\neg v_3, v_4\}, \{\neg v_4\}, \{v_3\}\}$.

# Procedura Davis-Putnam DPP

## Soluție (cont.)

**Pasul 4.**
Alegem variabila $v_3$ și selectăm $\mathcal{C}_3^{v_3} := \{\{v_3\}\}$, $\mathcal{C}_3^{\neg v_3} := \{\{\neg v_3, v_4\}\}$.
Mulțimea rezolvenților posibili este $\mathcal{R}_3 := \{\{v_4\}\}$.
Se elimină clauzele în care apare $v_3$, adăugăm rezolvenții și obținem:
$\mathcal{C}_4 := \{\{\neg v_4\}, \{v_4\}\}$.

**Pasul 5.**
Alegem variabila $v_4$ și selectăm $\mathcal{C}_4^{v_4} := \{\{v_4\}\}$, $\mathcal{C}_4^{\neg v_4} := \{\{\neg v_4\}\}$.
Mulțimea rezolvenților posibili este $\mathcal{R}_4 := \{\square\}$.
Se elimină clauzele în care apare $v_4$, adăugăm rezolvenții și obținem:
$\mathcal{C}_5 := \{\square\}$.
Deoarece $\mathcal{C}_5 = \{\square\}$, obținem că mulțimea de clauze $\mathcal{C}$ nu este satisfiabilă.

# Rezoluția în logica de ordinul I

# Clauze închise

- Fie $C$ o clauză. Spunem că $C'$ este o instață a lui $C$ dacă există o substituție $\theta : V \to Trm_{\mathcal{L}}$ astfel încât $C' = \theta(C)$.

  Spunem că $C'$ este o instanță închisă a lui $C$ dacă există o substituție $\theta : V \to T_{\mathcal{L}}$ such that $C' = \theta(C)$ ( $C'$ se obține din $C$ înlocuind variabilele cu termeni din universul Herbrand)

- Fie $\mathcal{C}$ o mulțime de clauze. Definim

  $$\mathcal{H}(\mathcal{C}) := \{\theta(C) \mid C \in \mathcal{C}, \theta : V \to T_{\mathcal{L}}\}$$

  $\mathcal{H}(\mathcal{C})$ este mulțimea instanțelor închise ale clauzelor din $\mathcal{C}$.

# Rezoluția pe clauze închise

$$Rez \quad \frac{C_1 \cup \{L\}, C_2 \cup \{\neg L\}}{C_1 \cup C_2}$$

unde $C_1, C_2$ clauze închise, iar $L$ este o formulă atomică închisă astfel încât $\{L, \neg L\} \cap C_1 = \emptyset$ și $\{L, \neg L\} \cap C_2 = \emptyset$.

**Teoremă**

Fie $\varphi$ o formulă arbitrară în logica de ordinul I. Atunci $\vDash \varphi$ dacă și numai dacă există o derivare pentru $\square$ din $\mathcal{H}(\mathcal{C})$ folosind *Rez*, unde $\mathcal{C}$ este mulțimea de clauze asociată lui $(\neg\varphi)^{fc}$.

### Exercițiu

Considerăm următoarea mulțime de clauze în logica de ordinul I:

$$\mathcal{C} = \{\, \{\neg P(f(a)), Q(y)\}, \{P(y)\}, \{\neg Q(b)\}\, \}$$

Arătați că $\mathcal{C}$ nu este satisfiabilă prin următoarele metode:

1) Găsiți o submulțime finită nesatisfiabilă lui $\mathcal{H}(\mathcal{C})$.
2) Găsiți o derivare pentru $\square$ folosind rezoluția pe clauze închise.

## Soluție

1) $\mathcal{H}(\mathcal{C}) = \{ \{\neg Q(b)\}, \{\neg P(f(a)), Q(a)\}, \{\neg P(f(a)), Q(b)\},$
$\{P(a)\}, \{P(b)\}, \{P(f(a))\}, \cdots \}$

   O submulțime nesatisfiabilă este
   $\{\{\neg P(f(a)), Q(b)\}, \{P(f(a))\}, \{\neg Q(b)\}\}$

2) Derivare pentru $\square$:
   1. $\{\neg P(f(a)), Q(b)\}$
   2. $\{P(f(a))\}$
   3. $\{Q(b)\}$
   4. $\{\neg Q(b)\}$
   5. $\square$

# Rezoluția pe clauze arbitrare

Regula rezoluției pentru clauze arbitrare

$$Rez \ \frac{C_1, C_2}{(\sigma C_1 \setminus \sigma Lit_1) \cup (\sigma C_2 \setminus \sigma Lit_2)}$$

dacă următoarele condiții sunt satisfăcute:

1. $C_1, C_2$ clauze care nu au variabile comune,
2. $Lit_1 \subseteq C_1$ și $Lit_2 \subseteq C_2$ sunt mulțimi de literali,
3. $\sigma$ este un cgu pentru $Lit_1$ și $Lit_2^c$, adică $\sigma$ unifică toți literalii din $Lit_1$ și $Lit_2^c$.

   O clauză $C$ se numește rezolvent pentru $C_1$ și $C_2$ dacă există o redenumire de variabile $\theta : V \to V$ astfel încât $C_1$ și $\theta C_2$ nu au variabile comune și $C$ se obține din $C_1$ și $\theta C_2$ prin $Rez$.

# Rezoluția în logica de ordinul I

## Exercițiu

Găsiți o derivare prin rezoluție a $\square$ pentru următoarea mulțime de clauze:

$C_1 = \{\, \neg P(x),\ R(x, f(x)) \,\}$
$C_2 = \{\, \neg R(a, x), Q(x) \,\}$
$C_3 = \{P(a)\}$
$C_4 = \{\neg Q(f(x))\}$

unde $P, Q, R$ sunt simboluri de relații, $f$ e simbol de funție, $a$ este o constantă, $x, y$ sunt variabile.

## Soluție

$C_5 = \{R(a, f(a))\}$ din $Rez, C_1, C_3, \theta = \{x \leftarrow a\}$
$C_4' = \{\neg Q(f(z))\}$ redenumire
$C_6 = \{\neg R(a, f(z))\}$ din $Rez, C_4', C_2, \theta = \{y \leftarrow f(z)\}$
$\square$ din $Rez, C_6, C_5, \theta = \{z \leftarrow a\}$

# Logica Horn

# Clauze definite. Programe logice. Clauze Horn

- clauză:

  $$\{\neg Q_1, \ldots, \neg Q_n, P_1, \ldots, P_k\} \quad \text{sau} \quad Q_1 \wedge \ldots \wedge Q_n \to P_1 \vee \ldots \vee P_k$$

  unde $n, k \geq 0$ și $Q_1, \ldots, Q_n, P_1, \ldots, P_k$ sunt formule atomice.

- clauză program definită: $k = 1$
  - cazul $n > 0$ : $Q_1 \wedge \ldots \wedge Q_n \to P$
  - cazul $n = 0$ : $\top \to P$ (clauză unitate, fapt)

  Program logic definit = mulțime finită de clauze definite

- scop definit (țintă, întrebare): k=0
  - $Q_1 \wedge \ldots \wedge Q_n \to \bot$

- clauza vidă $\square$: $n = k = 0$

  Clauza Horn = clauză program definită sau clauză scop ($k \leq 1$)

# Programare logica

- Logica clauzelor definite/Logica Horn: un fragment al logicii de ordinul I în care singurele formule admise sunt clauze Horn
  - formule atomice: $P(t_1, \ldots, t_n)$
  - $Q_1 \wedge \ldots \wedge Q_n \to P$
    unde toate $Q_i, P$ sunt formule atomice, $\top$ sau $\bot$

- Problema programării logice: reprezentăm cunoștințele ca o mulțime de clauze definite $T$ și suntem interesați să aflăm răspunsul la o întrebare de forma $Q_1 \wedge \ldots \wedge Q_n$, unde toate $Q_i$ sunt formule atomice

$$T \vDash Q_1 \wedge \ldots \wedge Q_n$$

  - Variabilele din $T$ sunt cuantificate universal.
  - Variabilele din $Q_1, \ldots, Q_n$ sunt cuantificate existențial.

Limbajul PROLOG are la bază logica clauzelor Horn.

# Modele Herbrand

Definim o ordine între modelele Herbrand:

$\mathcal{H}_1 \leq \mathcal{H}_2$ este definită astfel:

*pentru orice $R \in \mathbf{R}$ cu $ari(R) = n$ și pentru orice termeni $t_1, \ldots, t_n$*
dacă $\mathcal{H}_1 \vDash R(t_1, \ldots, t_n)$, atunci $\mathcal{H}_2 \vDash R(t_1, \ldots, t_n)$

Semantica unui program logic definit $T$ este dată de
cel mai mic model Herbrand al lui $T$!

- Definim $\mathcal{LH}_T := \bigcap\{\mathcal{H} \mid \mathcal{H}$ model Herbrand pentru $T\}$
- $\mathcal{LH}_T \vDash T$.
- Vom caracteriza cel mai mic model Herbrand $\mathcal{LH}_T$ printr-o construcție de punct fix.

# Cel mai mic model Herbrand

- O instanță de bază a unei clauze $Q_1(x_1) \land \ldots \land Q_n(x_n) \to P(y)$ este rezultatul obținut prin înlocuirea variabilelor cu termeni fără variabile.

- Pentru o mulțime de clauze definite $T$, o formulă atomică $P$ și o mulțime de formule atomice $X$,

  $$oneStep_T(P, X) \text{ este adevărat}$$

  dacă există o instanță de bază a unei clauze $Q_1(x_1) \land \ldots \land Q_n(x_n) \to P(y)$ din $T$ astfel încât $P$ este instanța lui $P(y)$ și instanța lui $Q_i(x_i)$ este în $X$, pentru orice $i = 1, \ldots, n$.

- Baza Herbrand $B_{\mathcal{L}}$ este mulțimea formulelor atomice fără variabile.

- Pentru o mulțime de clauze definite $T$, definim

  $$f_T : \mathcal{P}(B_{\mathcal{L}}) \to \mathcal{P}(B_{\mathcal{L}})$$

  $$f_T(X) = \{P \in B_{\mathcal{L}} \mid oneStep_T(P, X)\}$$

# Cel mai mic model Herbrand

Fie $T$ un program logic definit.

- $f_T$ este continuă
- Din teorema Knaster-Tarski, $f_T$ are un cel mai mic punct fix $FP_T$.
- $FP_T$ este reuniunea tuturor mulțimilor

$$f_T(\{\}),\ f_T(f_T(\{\})),\ f_T(f_T(f_T(\{\}))), \ldots$$

**Propoziție (caracterizarea $\mathcal{LH}_T$)**

Pentru orice $R \in \mathbf{R}$ cu $ari(R) = n$ și pentru orice $t_1, \ldots, t_n$ termeni, avem

$$(t_1, \ldots, t_n) \in R^{\mathcal{LH}_T} \text{ ddacă } R(t_1, \ldots, t_n) \in FP_T$$

# Sistem de deducție *backchain*

## Sistem de deducție pentru clauze Horn

Pentru un program logic definit $T$ avem

- Axiome: orice clauză din $T$
- Regula de deducție: regula *backchain*

$$\frac{\theta(Q_1) \quad \theta(Q_2) \quad \ldots \quad \theta(Q_n) \quad (Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \to P)}{\theta(Q)}$$

unde $Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \to P \in T$, iar $\theta$ este cgu pentru $Q$ și $P$.

# Rezoluția SLD

Fie $T$ o mulțime de clauze definite.

$$\text{SLD}\ \boxed{\dfrac{\neg Q_1 \vee \cdots \vee \neg Q_i \vee \cdots \vee \neg Q_n}{(\neg Q_1 \vee \cdots \vee \neg P_1 \vee \cdots \vee \neg P_m \vee \cdots \vee \neg Q_n)\theta}}$$

unde

- $Q \vee \neg P_1 \vee \cdots \vee \neg P_m$ este o clauză definită din $T$ (în care toate variabilele au fost redenumite) și
- variabilele din $Q \vee \neg P_1 \vee \cdots \vee \neg P_m$ și $Q_i$ se redenumesc
- $\theta$ este c.g.u pentru $Q_i$ și $Q$

# Rezoluția SLD

Fie $T$ o mulțime de clauze definite și $Q_1 \land \ldots \land Q_m$ o întrebare, unde $Q_i$ sunt formule atomice.

- O derivare din $T$ prin rezoluție SLD este o secvență
$$G_0 := \neg Q_1 \lor \ldots \lor \neg Q_m, \quad G_1, \quad \ldots, \quad G_k, \ldots$$
în care $G_{i+1}$ se obține din $G_i$ prin regula SLD.
- Dacă există un $k$ cu $G_k = \square$ (clauza vidă), atunci derivarea se numește SLD-respingere.

# Rezoluția SLD

## Exercițiu

Găsiți o SLD-respingere pentru următorul program Prolog și ținta:

```
1.  p(X) :- q(X,f(Y)), r(a).        ?- p(X), q(Y,Z).
2.  p(X) :- r(X).
3.  q(X,Y) :- p(Y).
4.  r(X) :- q(X,Y).
5.  r(f(b)).
```

## Soluție

| | |
|---|---|
| $G_0 = \neg p(X) \vee \neg q(Y, Z)$ | |
| $G_1 = \neg r(X_1) \vee \neg q(Y, Z)$ | (2 cu $\theta(X) = X_1$) |
| $G_2 = \neg q(Y, Z)$ | (5 cu $\theta(X_1) = f(b)$) |
| $G_3 = \neg p(Z_1)$ | (3 cu $\theta(X) = Y_1$ și $\theta(Y) = Z_1$) |
| $G_4 = \neg r(X)$ | (2 cu $\theta(Z_1) = X$) |
| $G_5 = \square$ | (5 cu $\theta(X) = f(b)$) |

# Rezoluția SLD - arbori de căutare

## Arbori SLD

- Presupunem că avem o mulțime de clauze definite $T$ și o țintă $G_0 = \neg Q_1 \vee \ldots \vee \neg Q_m$

- Construim un arbore de căutare (arbore SLD) astfel:
  - Fiecare nod al arborelui este o țintă (posibil vidă)
  - Rădăcina este $G_0$
  - Dacă arborele are un nod $G_i$, iar $G_{i+1}$ se obține din $G_i$ folosind regula SLD folosind o clauză $C_i \in T$, atunci nodul $G_i$ are copilul $G_{i+1}$. Muchia dintre $G_i$ și $G_{i+1}$ este etichetată cu $C_i$.

- Dacă un arbore SLD cu rădăcina $G_0$ are o frunză $\square$ (clauza vidă), atunci există o SLD-respingere a lui $G_0$ din $T$.

# Rezoluția SLD - arbori de căutare

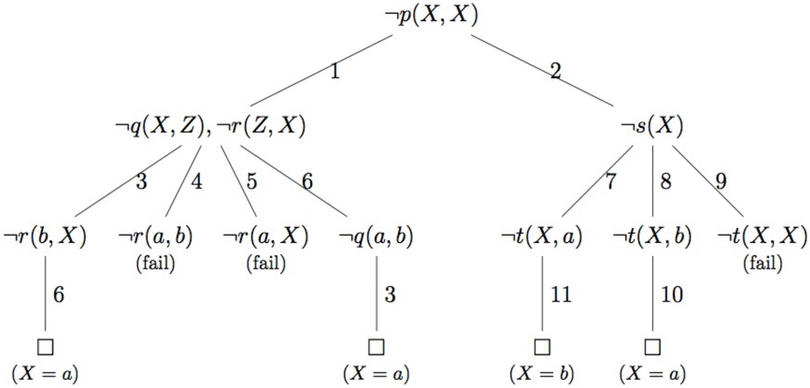
## Exercițiu

Desenați arborele SLD pentru programul Prolog de mai jos și ținta
`?- p(X,X).`

```
1.  p(X,Y) :- q(X,Z), r(Z,Y).
2.  p(X,X) :- s(X).
3.  q(X,b).
4.  q(b,a).
5.  q(X,a) :- r(a,X).
6.  r(b,a).
7.  s(X) :- t(X,a).
8.  s(X) :- t(X,b).
9.  s(X) :- t(X,X).
10.  t(a,b).
11.  t(b,a).
```

# Rezoluția SLD - arbori de căutare

## Soluție



# Sisteme de rescriere

# Sisteme de rescriere abstracte

# Sisteme de rescriere abstracte

## Definiție

Un sistem de rescriere abstract este o pereche $(T, \to)$ unde:

- $T$ este o mulțime,
- $\to \subseteq T \times T$ ($\to$ este o relație binară pe $T$).

Definiții:

- $\leftarrow \; := \; \to^{-1}$ (relația inversă)
- $\leftrightarrow \; := \; \to \cup \leftarrow$ (închiderea simetrică)
- $\stackrel{*}{\to} \; := (\to)^*$ (închiderea reflexivă și tranzitivă)
- $\stackrel{*}{\leftrightarrow} \; := (\leftrightarrow)^*$ (echivalența generată)

# Sisteme de rescriere abstracte

## Definiție

Fie $(T, \to)$ sistem de rescriere.

- $t \in T$ este reductibil dacă există $t' \in T$ a.î. $t \to t'$.
- O reducere este un șir $t_0 \to t_1 \to t_2 \to \ldots$
- $t \in T$ este în formă normală (ireductibil) dacă nu este reductibil.
- $t_0$ este o formă normală a lui $t$ dacă
  - $t \xrightarrow{*} t_0$ și
  - $t_0$ este în formă normală.
- $t_1$ și $t_2$ se intâlnesc dacă există $t \in T$ a.î. $t_1 \xrightarrow{*} t \xleftarrow{*} t_2$.
  - notație: $t_1 \downarrow t_2$.

# Sisteme de rescriere abstracte

## Definiție

Un sistem de rescriere $(T, \to)$ se numește

- noetherian (se termină): dacă nu există reduceri infinite $t_0 \to t_1 \to t_2 \to \ldots$.
  - orice rescriere se termină.
- confluent: $t_1 \stackrel{*}{\leftarrow} t \stackrel{*}{\to} t_2 \Rightarrow t_1 \downarrow t_2$.
- local confluent: $t_1 \leftarrow t \to t_2 \Rightarrow t_1 \downarrow t_2$.
- Church-Rosser: $t_1 \stackrel{*}{\leftrightarrow} t_2 \Rightarrow t_1 \downarrow t_2$.
- Normalizat: orice element are o formă normală.
- Complet (convergent, canonic): confluent și noetherian.

Am arătat diverse legături între proprietățile de mai sus.

# Sisteme de rescriere pentru termeni

# Rescrierea termenilor

Fie $\mathcal{L}$ un limbaj de ordinul I.

## Definiție

O regulă de rescriere (pentru termeni) este formată din doi termeni $l, r \in Trm_{\mathcal{L}}$ astfel încât:

1. $l$ nu este variabilă,
2. $Var(r) \subseteq Var(l)$.

Vom nota o regulă de rescriere prin:

$$l \to r.$$

Un sistem de rescriere pentru termeni (TRS) este o mulțime finită de reguli de rescriere pentru termeni.

# Contexte

- Fie $\mathcal{L}$ un limbaj de ordinul I
- Dacă $t \in Trm_{\mathcal{L}}$ și $x \in Var$ notăm
$$nr_x(t) = \text{numărul de apariții ale lui } x \text{ în } t$$

**Definiție**

Fie $z$ a.î. $z \notin Var$ (o variabilă nouă). Un termen $c$ se numește context dacă $nr_z(c) = 1$.

- Dacă $t_0 \in Trm_{\mathcal{L}}$, definim substituția $\{z \leftarrow t_0\} : Var \cup \{z\} \to Trm_{\mathcal{L}}$
$$\{z \leftarrow t_0\}(x) = \begin{cases} t_0, & \text{dacă } x = z \\ x, & \text{altfel} \end{cases}$$
- Pentru un context $c$, notăm:
$$c[z \leftarrow t_0] := \{z \leftarrow t_0\}(c)$$

# Relația de rescriere generată de $R$

- Fie $\mathcal{L}$ un limbaj de ordinul I și $R$ sistem de rescriere pentru termeni
- Pentru $t, t' \in Trm_{\mathcal{L}}$ definim relația $t \to_R t'$ astfel:

$$\boxed{\begin{array}{lcl} t \to_R t' & \Leftrightarrow & t \text{ este } c[z \leftarrow \theta(l)] \text{ și} \\ & & t' \text{ este } c[z \leftarrow \theta(r)], \text{ unde} \\ & & c \text{ context}, \\ & & l \to r \in R, \\ & & \theta \text{ substituție} \end{array}}$$

- Observați că $t \to_R t'$ ddacă $t'$ se poate obține din $t$ înlocuind o instanță a lui $l$ cu o instanță a lui $r$, unde $l \to r \in R$.
- $\to_R$ este relația de rescriere generată de sistemul de rescriere $R$.

# Confluență. Perechi critice.

# Perechi critice

Fie $\mathcal{L}$ un limbaj de ordinul I și $R$ un sistem de rescriere pentru termeni.

## Definiție

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 un subtermen $t$ 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 $\theta$ c.g.u pentru $t$ și $l_2$ (i.e. $\theta(t) = \theta(l_2)$).

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

$$\begin{tikzcd}
 & \theta(l_1) \arrow[dl, "R"'] \arrow[dr, "R"] & \\
\theta(r_1) & & \theta(c)[z \leftarrow \theta(r_2)]
\end{tikzcd}$$

# Confluență și perechi critice

Fie $\mathcal{L}$ un limbaj de ordinul I și $R$ un sistem de rescriere pentru termeni.

$$\begin{tikzcd}[column sep=small] & \theta(l_1) \arrow[dl] \arrow[dr] & \\ \theta(r_1) & & \theta(c)[z \leftarrow \theta(r_2)] \end{tikzcd}$$

$l_1 \to r_1,\ l_2 \to r_2 \in R$,
$Var(l_1) \cap Var(l_2) = \emptyset$,
$l_1 = c[z \to t]$,
$\theta(t) = \theta(l_2)$ c.g.u

## Teoremă (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 $(t_1, t_2)$.*

# Consecință

## Corolar

*Confluența unui TRS noetherian este decidabilă.*

Algoritm:

- pt. or. pereche de reguli de rescriere $l_1 \to r_1$ și $l_2 \to r_2$
- se încearcă generarea perechilor critice $(t_1, t_2)$
- pt. or. pereche critică $(t_1, t_2)$, se arată că $t_1 \downarrow_R t_2$

# Confluență și perechi critice

## Exercițiu

Fie $\mathcal{L}$ un limbaj de ordinul I cu două simboluri de funcție $f$ și $g$ de aritate 1. Cercetați dacă sistemul de rescriere de mai jos este confluent:

$$R = \{f(f(x)) \to g(x)\}.$$

## Soluție

Se observă că $R$ este noetherian, deci putem aplica Teorema Perechilor Critice. Determinăm perechile critice ale sistemului $R$. Redenumind variabilele, considerăm $l_1 \to r_1, l_2 \to r_2 \in R$ ca fiind $f(f(x)) \to g(x)$ și $f(f(y)) \to g(y)$, respectiv. Subtermenii lui $l_1$ care nu sunt variabile sunt $f(x)$ și $f(f(x))$. Investigăm fiecare caz:

- $t := f(x)$. Observăm că $l_1 = c[z \leftarrow t]$ pentru contextul $c = f(z)$. Mai mult, $\theta(x) = f(y)$ este c.g.u. pentru $t$ și $l_2$. Obținem perechea critică $P_1 = (g(f(y)), f(g(y)))$.
- $t := f(f(x))$. Observăm că $l_1 = c[z \leftarrow t]$ pentru contextul $c = z$. Mai mult, $\theta(x) = y$ este c.g.u. pentru $t$ și $l_2$. Obținem perechea critică $P_1 = (g(y), g(y))$.

Evident $g(y) \downarrow g(y)$, dar $g(f(y)) \not\downarrow f(g(y))$ deoarece $g(f(y))$ și $f(g(y))$ sunt deja în formă normală. Din Teorema Perechilor Critice obținem că $R$ nu este confluent.

# Confluență și perechi critice

## Exercițiu

Fie $\mathcal{L}$ un limbaj de ordinul I cu trei simboluri de constantă $a$, $b$ și $c$, un simbol de funcție $g$ de aritate 1 și un simbol de funcție $f$ de aritate 2. Cercetați dacă sistemul de rescriere de mai jos este confluent:

$$R = \{f(x,x) \to a, \quad f(x,g(x)) \to b, \quad c \to g(c)\}.$$

## Soluție

Se observă că $R$ nu se termină:

$$c \to_R g(c) \to_R g(g(c)) \to_R \dots .$$

În concluzie nu putem aplica Teorema perechilor critice pentru a stabili confluența. Se observă că:

$$\begin{tikzcd} & f(c,c) \arrow[dl] \arrow[dr] & \\ a & & f(c,g(c)) \arrow[dr] \\ & & & b \end{tikzcd}$$

Cum $a \not\downarrow b$, sistemul $R$ nu este confluent.

# Alte subiecte prezentate

Aceste subiecte nu intră în materia pentru examen.

- Terminarea sistemelor de rescriere
- Algoritmul Knuth-Bendix

# Prolog - Backtracking, Cut, Negații

# Semantica programelor

# Semantica programelor

- Semantica operațională
  - Semantica small-step
  - Semantica big-step

- Semantica denotațională

# Limbajul IMP

IMP este un limbaj IMPerativ foarte simplu.

Conține:

- Expresii
  - Aritmetice `x + 3`
  - Booleene `(x > 7)`
- Blocuri de instrucțiuni
  - De atribuire `x = 5;`
  - Condiționale `if (x > 7) {x =5; } else {x = 0;}`
  - De ciclare `while (x > 7) {x = x - 1;}`

Implementare în Prolog.

# Despre examen

# Notare

- **Laborator: 30 puncte**
- **Examen: 60 puncte**
- Se acordă 10 puncte din oficiu!

- Condiție minimă pentru promovare:
  laborator: minim 15 puncte și
  examen: minim 25 puncte.

- Chiar dacă nu ați promovat testul de laborator, puteți să vă prezentați la examen.
- În sesiunea de restanțe trebuie să refaceți problele pe care nu le-ați promovat.

# Examen: 60 puncte

- Subiecte de tip exerciţiu:
  - în stilul exemplelor de la curs;
  - în stilul exerciţiilor rezolvate la seminarii şi laboratoare.
- Timp de lucru: 2 ore
- Aveţi voie doar cu materialele de la curs printate.
- Pentru a trece această probă, trebuie să obţineţi minim 25 puncte.

Baftă la examen!