# Curs 3

2017-2018

Programare Logică

# Verificarea problemei consecinței logice

- În principiu, sistemul poate verifica problema consecinței logice construind un tabel de adevăr, cu câte o linie pentru fiecare interpretare posibilă.

- În cazul în care programul și ținta conțin $n$ atomi diferiți, tabelul de adevăr rezultat o să aibă $2^n$ rânduri.

- Această metodă este atât de costisitoare computațional, încât este irealizabilă. (Timp exponențial)

- Problemă deschisă de un milion de dolari:

  *Este posibil să găsim o metodă mai bună pentru a decide problema consecinței logice în cazul propozițional care să funcționeze în timp polinomial față de intrarea programului și a țintei?*

  Echivalent, este adevărată $P = NP$?
  (Institutul de Matematica Clay – Millennium Prize Problems)

# Cum salvăm situația?

1. Folosirea metodelor sintactice pentru a stabili problema consecinței logice (*proof search*)
2. Restricționarea formulelor din ”programele logice”

Această metodologie este eficientă și flexibilă.

# Clauze propoziționale definite

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate)
  2. $p_1 \wedge \ldots \wedge p_k \to q$

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

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \wedge \ldots \wedge p_k \to q$ (o regulă în Prolog `q :- p1,...,pk`)

  unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale (constante în Prolog).

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \land \ldots \land p_k \to q$ (o regulă în Prolog `q :- p1,...,pk`)

  unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale (constante în Prolog).
- Numim variabilele propoziționale și atomi.

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \wedge \ldots \wedge p_k \rightarrow q$ (o regulă în Prolog `q :- p1,...,pk`)

  unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale (constante în Prolog).
- Numim variabilele propoziționale și atomi.

## Programare logică – cazul logicii propoziționale

- Un "program logic" este o listă $F_1, \ldots, F_n$ de clauze definite.

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \wedge \ldots \wedge p_k \to q$ (o regulă în Prolog `q :- p1,...,pk`)

  unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale (constante în Prolog).
- Numim variabilele propoziționale și atomi.

## Programare logică – cazul logicii propoziționale

- Un "program logic" este o listă $F_1, \ldots, F_n$ de clauze definite.
- O țintă (întrebare, *goal*) este o listă $g_1, \ldots, g_m$ de atomi.

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \wedge \ldots \wedge p_k \rightarrow q$ (o regulă în Prolog `q :- p1,...,pk`)

  unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale (constante în Prolog).
- Numim variabilele propoziționale și atomi.

## Programare logică – cazul logicii propoziționale

- Un "program logic" este o listă $F_1, \ldots, F_n$ de clauze definite.
- O țintă (întrebare, *goal*) este o listă $g_1, \ldots, g_m$ de atomi.
- Sarcina sistemului este să stabilească:

$$F_1, \ldots, F_n \models g_1 \wedge \ldots \wedge g_m.$$

# Clauze propoziționale definite

- O clauză definită este o formulă care poate avea una din formele:
  1. $q$ (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \wedge \ldots \wedge p_k \to q$ (o regulă în Prolog `q :- p1,...,pk`)

  unde $q, p_1, \ldots, p_n$ sunt variabile propoziționale (constante în Prolog).
- Numim variabilele propoziționale și atomi.

## Programare logică – cazul logicii propoziționale

- Un "program logic" este o listă $F_1, \ldots, F_n$ de clauze definite.
- O țintă (întrebare, *goal*) este o listă $g_1, \ldots, g_m$ de atomi.
- Sarcina sistemului este să stabilească:

$$F_1, \ldots, F_n \models g_1 \wedge \ldots \wedge g_m.$$

- Scop: Vrem să găsim metode sintactice pentru a rezolva problema de mai sus!

# Sistem de deducție CDP

## Sistem de deducție CDP pentru clauze definite propoziționale

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

# Sistem de deducție CDP

Sistem de deducție CDP pentru clauze definite propoziționale

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

- Axiome (premise): orice clauză din $S$

# Sistem de deducție CDP

## Sistem de deducție CDP pentru clauze definite propoziționale

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

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

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

  - Aceste reguli ne permit să deducem formula de sub linie din formulele de deasupra liniei.
  - Sunt regulile $(\to e)$ și $(\wedge i)$ din deducția naturală pentru logica propozițională.

# Sistemul de deducție CDP

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

## Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \wedge \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

# Sistemul de deducție CDP

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

## Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \wedge \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

$$\dfrac{\dfrac{\dfrac{oslo \qquad oslo \to norway}{norway} \qquad norway \to cold}{cold} \qquad \dfrac{oslo \qquad oslo \to windy}{windy}}{cold \wedge windy}$$

# Sistemul de deducție CDP

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

## Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \wedge \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

$$\dfrac{\dfrac{\dfrac{oslo \qquad oslo \to norway}{norway} \qquad norway \to cold}{cold} \qquad \dfrac{oslo \qquad oslo \to windy}{windy}}{cold \wedge windy}$$

$$\frac{cold \wedge windy \qquad cold \wedge windy \to winterIsComing}{winterIsComing}$$

# Sistemul de deducție CDP

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

## Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \wedge \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

1. $oslo \to windy$
2. $oslo \to norway$
3. $norway \to cold$
4. $cold \wedge windy \to winterIsComing$
5. $oslo$
6. $norway$ (MP 5,2)
7. $cold$ (MP 6,3)
8. $windy$ (MP 5,1)
9. $cold \wedge windy$ (andI 7,8)
10. $winterIsComing$ (MP 9,4)

# Sistemul de deducție CDP

O formulă $Q$ se poate deduce din $S$ în sistemul de deducție CDP, notat

$$S \vdash Q,$$

dacă există o secvență de formule $Q_1, \ldots, Q_n$ astfel încât $Q_n = Q$ și fiecare $Q_i$:

- fie aparține lui $S$
- fie se poate deduce din $Q_1, \ldots, Q_{i-1}$ folosind regulile de deducție ($MP$) și ($andI$)

# Sistemul de deducție CDP

Putem folosi sistemul de deducție CDP pentru a deduce alți atomi:

# Sistemul de deducție CDP

Putem folosi sistemul de deducție CDP pentru a deduce alți atomi:

- Clauzele unitate din $S$ (atomii $p_i \in S$) sunt considerate adevărate.
  - Sunt deduși ca axiome.

# Sistemul de deducție CDP

Putem folosi sistemul de deducție CDP pentru a deduce alți atomi:

- Clauzele unitate din $S$ (atomii $p_i \in S$) sunt considerate adevărate.
  - Sunt deduși ca axiome.

- Putem deduce că un nou atom $r$ este adevărat dacă
  - am dedus că $p_1, \ldots, p_n$ sunt adevărați, și
  - $p_1 \wedge \ldots \wedge p_n \rightarrow r$ este în $S$.

  O astfel de derivare folosește de $n - 1$ ori (*andI*) și o data (*MP*).

# Sistemul de deducție CDP

Putem folosi sistemul de deducție CDP pentru a deduce alți atomi:

- Clauzele unitate din $S$ (atomii $p_i \in S$) sunt considerate adevărate.
  - Sunt deduși ca axiome.

- Putem deduce că un nou atom $r$ este adevărat dacă
  - am dedus că $p_1, \ldots, p_n$ sunt adevărați, și
  - $p_1 \wedge \ldots \wedge p_n \rightarrow r$ este în $S$.

  O astfel de derivare folosește de $n-1$ ori ($andI$) și o data ($MP$).

Deci putem construi mulțimi din ce în ce mai mari de atomi care sunt consecințe logice din $S$, și pentru care există derivări din $S$.

# Completitudinea sistemului de deducție CDP

- Se poate demonstra că aceste reguli sunt corecte, folosind tabelele de adevăr.
    - Dacă formulele de deasupra liniei sunt adevărate, atunci și formula de sub linie este adevărată.

# Completitudinea sistemului de deducție CDP

- Se poate demonstra că aceste reguli sunt corecte, folosind tabelele de adevăr.
  - Dacă formulele de deasupra liniei sunt adevărate, atunci și formula de sub linie este adevărată.
- Mai mult, sistemul de deducție este și complet. Adică

$$\text{dacă } S \models Q, \text{ atunci } S \vdash Q.$$

  - Dacă $Q$ este o consecință logică a lui $S$, atunci există o derivare a sa din $S$ folosind sistemul de deducție CDP

# Completitudinea sistemului de deducție CDP

- Se poate demonstra că aceste reguli sunt corecte, folosind tabelele de adevăr.
  - Dacă formulele de deasupra liniei sunt adevărate, atunci și formula de sub linie este adevărată.

- Mai mult, sistemul de deducție este și complet. Adică

$$\text{dacă } S \models Q, \text{ atunci } S \vdash Q.$$

  - Dacă $Q$ este o consecință logică a lui $S$, atunci există o derivare a sa din $S$ folosind sistemul de deducție CDP

- Pentru a demonstra completitudinea vom folosi teorema Knaster-Tarski.

# Puncte fixe. Teorema Knaster-Tarski

# 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

# 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.
$$x_1 \leq x_2 \leq x_3 \leq \ldots$$

# Mulțimi parțial ordonate complete

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 în $C$ pentru orice lanț $x_1 \leq x_2 \leq x_3 \leq \ldots$

# Mulțimi parțial ordonate complete

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 în $C$ pentru orice lanț $x_1 \leq x_2 \leq x_3 \leq \ldots$

### Exemplu

Fie $X$ o mulțime și $\mathcal{P}(X)$ mulțimea submulțimilor lui $X$.

$(\mathcal{P}(X), \subseteq)$ este o cpo:

- $\subseteq$ este o relație de ordine
- $\emptyset$ este prim element ($\emptyset \subseteq Q$ pentru orice $Q \in \mathcal{P}(X)$)
- pentru orice șir (numărabil) de submulțimi ale lui X $Q_1 \subseteq Q_2 \subseteq \ldots$ evident $\bigcup_n Q_n \in \mathcal{P}(X)$

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

**Exemplu**

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$ cu $i \in \{1,2,3\}$

- $f_1(Y) = Y \cup \{1\}$

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

**Exemplu**

Fie următoarele funcții $f_i : \mathcal{P}(\{1, 2, 3\}) \to \mathcal{P}(\{1, 2, 3\})$ cu $i \in \{1, 2, 3\}$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

## Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1, 2, 3\}) \to \mathcal{P}(\{1, 2, 3\})$ cu $i \in \{1, 2, 3\}$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.
- $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \emptyset & \text{altfel} \end{cases}$

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

### Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$ cu $i \in \{1,2,3\}$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.
- $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \emptyset & \text{altfel} \end{cases}$ este monotonă.

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

## Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$ cu $i \in \{1,2,3\}$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.
- $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \emptyset & \text{altfel} \end{cases}$ este monotonă.
- $f_3(Y) = \begin{cases} \emptyset & \text{dacă } 1 \in Y \\ \{1\} & \text{altfel} \end{cases}$

# Funcție monotonă

- 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 $a_1,\ a_2 \in A$.

## Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$ cu $i \in \{1,2,3\}$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.
- $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \emptyset & \text{altfel} \end{cases}$ este monotonă.
- $f_3(Y) = \begin{cases} \emptyset & \text{dacă } 1 \in Y \\ \{1\} & \text{altfel} \end{cases}$ nu este monotonă.

# Funcție continuă

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

# Funcție continuă

- Fie $(A, \leq_A)$ și $(B, \leq_B)$ mulțimi parțial ordonate complete.
  O funcție $f\colon A \to B$ este continuă dacă

  $$f(\bigvee_n a_n) = \bigvee_n f(a_n) \text{ pentru orice lanț } \{a_n\}_n \text{ din } A.$$

- Observăm că orice funcție continuă este crescătoare.

# Funcție continuă

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

- Observăm că orice funcție continuă este crescătoare.

## Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ cu $i \in \{1, 2\}$.

- $f_1(Y) = Y \cup \{1\}$ este continuă.

- $f_2(Y) = \begin{cases} \emptyset & \text{dacă } Y \text{ este finita} \\ Y & \text{altfel} \end{cases}$
  este monotonă, nu este continuă.

# Teorema de punct fix

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

# 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 $\mathbf{F}$.

# 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 $\mathbf{F}$.

- Observăm că în ipotezele ultimei teoreme secvența

$$\mathbf{F}^0(\perp) = \perp \leq \mathbf{F}(\perp) \leq \mathbf{F}^2(\perp) \leq \cdots \leq \mathbf{F}^n(\perp) \leq \cdots$$

este un lanț, deci $\bigvee_n \mathbf{F}^n(\perp)$ există.

# Teorema Knaster-Tarski pentru CPO

## Demonstrație

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

- Arătăm că $a = \bigvee_n \mathbf{F}^n(\perp)$ este punct fix, i.e. $\mathbf{F}(a) = a$

# Teorema Knaster-Tarski pentru CPO

## Demonstrație

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

- Arătăm că $a = \bigvee_n \mathbf{F}^n(\perp)$ este punct fix, i.e. $\mathbf{F}(a) = a$

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

# Teorema Knaster-Tarski pentru CPO

## Demonstrație

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

- Arătăm că $a = \bigvee_n \mathbf{F}^n(\bot)$ este punct fix, i.e. $\mathbf{F}(a) = a$

$$
\begin{aligned}
\mathbf{F}(a) &= \mathbf{F}(\textstyle\bigvee_n \mathbf{F}^n(\bot)) \\
&= \textstyle\bigvee_n \mathbf{F}(\mathbf{F}^n(\bot)) \text{ din continuitate}
\end{aligned}
$$

# Teorema Knaster-Tarski pentru CPO

## Demonstrație

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

- Arătăm că $a = \bigvee_n \mathbf{F}^n(\bot)$ este punct fix, i.e. $\mathbf{F}(a) = a$

$$\begin{aligned}
\mathbf{F}(a) &= \mathbf{F}(\textstyle\bigvee_n \mathbf{F}^n(\bot)) \\
&= \textstyle\bigvee_n \mathbf{F}(\mathbf{F}^n(\bot)) \text{ din continuitate} \\
&= \textstyle\bigvee_n \mathbf{F}^{n+1}(\bot)
\end{aligned}$$

# Teorema Knaster-Tarski pentru CPO

## Demonstrație

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

- Arătăm că $a = \bigvee_n \mathbf{F}^n(\perp)$ este punct fix, i.e. $\mathbf{F}(a) = a$

$$
\begin{aligned}
\mathbf{F}(a) &= \mathbf{F}(\textstyle\bigvee_n \mathbf{F}^n(\perp)) \\
&= \textstyle\bigvee_n \mathbf{F}(\mathbf{F}^n(\perp)) \text{ din continuitate} \\
&= \textstyle\bigvee_n \mathbf{F}^{n+1}(\perp) \\
&= \textstyle\bigvee_n \mathbf{F}^n(\perp) = a
\end{aligned}
$$

# Teorema Knaster-Tarski pentru CPO

## Demonstrație (cont.)

- Arătăm că *a* este cel mai mic punct fix.

# Teorema Knaster-Tarski pentru CPO

## Demonstrație (cont.)

- Arătăm că $a$ este cel mai mic punct fix.

Fie $b$ un alt punct fix, i.e. $\mathbf{F}(b) = b$.

Demonstrăm prin inducție după $n \geq 1$ că $\mathbf{F}^n(\perp) \leq b$.

# Teorema Knaster-Tarski pentru CPO

## Demonstrație (cont.)

- Arătăm că $a$ este cel mai mic punct fix.

Fie $b$ un alt punct fix, i.e. $\mathbf{F}(b) = b$.

Demonstrăm prin inducție după $n \geq 1$ că $\mathbf{F}^n(\perp) \leq b$.

Pentru $n = 0$, $\mathbf{F}^0(\perp) = \perp \leq b$ deoarece $\perp$ este prim element.

# Teorema Knaster-Tarski pentru CPO

## Demonstrație (cont.)

- Arătăm că $a$ este cel mai mic punct fix.

Fie $b$ un alt punct fix, i.e. $\mathbf{F}(b) = b$.

Demonstrăm prin inducție după $n \geq 1$ că $\mathbf{F}^n(\perp) \leq b$.

Pentru $n = 0$, $\mathbf{F}^0(\perp) = \perp \leq b$ deoarece $\perp$ este prim element.

Dacă $\mathbf{F}^n(\perp) \leq b$, atunci $\mathbf{F}^{n+1}(\perp) \leq \mathbf{F}(b)$, deoarece $\mathbf{F}$ este crescătoare.
Deoarece $\mathbf{F}(b) = b$ rezultă $\mathbf{F}^{n+1}(\perp) \leq b$.

# Teorema Knaster-Tarski pentru CPO

## Demonstrație (cont.)

- Arătăm că $a$ este cel mai mic punct fix.

Fie $b$ un alt punct fix, i.e. $\mathbf{F}(b) = b$.

Demonstrăm prin inducție după $n \geq 1$ că $\mathbf{F}^n(\perp) \leq b$.

Pentru $n = 0$, $\mathbf{F}^0(\perp) = \perp \leq b$ deoarece $\perp$ este prim element.

Dacă $\mathbf{F}^n(\perp) \leq b$, atunci $\mathbf{F}^{n+1}(\perp) \leq \mathbf{F}(b)$, deoarece $\mathbf{F}$ este crescătoare.
Deoarece $\mathbf{F}(b) = b$ rezultă $\mathbf{F}^{n+1}(\perp) \leq b$.

Știm $\mathbf{F}^n(\perp) \leq b$ oricare $n \geq 1$, deci $a = \bigvee_n \mathbf{F}^n(\perp) \leq b$.

# Teorema Knaster-Tarski pentru CPO

## Demonstrație (cont.)

- Arătăm că $a$ este cel mai mic punct fix.

Fie $b$ un alt punct fix, i.e. $\mathbf{F}(b) = b$.

Demonstrăm prin inducție după $n \geq 1$ că $\mathbf{F}^n(\perp) \leq b$.

Pentru $n = 0$, $\mathbf{F}^0(\perp) = \perp \leq b$ deoarece $\perp$ este prim element.

Dacă $\mathbf{F}^n(\perp) \leq b$, atunci $\mathbf{F}^{n+1}(\perp) \leq \mathbf{F}(b)$, deoarece $\mathbf{F}$ este crescătoare.
Deoarece $\mathbf{F}(b) = b$ rezultă $\mathbf{F}^{n+1}(\perp) \leq b$.

Știm $\mathbf{F}^n(\perp) \leq b$ oricare $n \geq 1$, deci $a = \bigvee_n \mathbf{F}^n(\perp) \leq b$.

Am arătat că $a$ este cel mai mic punct fix al funției $\mathbf{F}$. $\square$

# Completitudinea sistemului de deducție CDP

# Clauze definite și funcții monotone

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

# 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 clauzelor unitate din $S$.

# 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 clauzelor unitate din $S$.

## Exemplu

$$\begin{array}{rcl} \texttt{oslo} & \to & \texttt{windy} \\ \texttt{oslo} & \to & \texttt{norway} \\ \texttt{norway} & \to & \texttt{cold} \\ \texttt{cold} \wedge \texttt{windy} & \to & \texttt{winterIsComing} \\ & & \texttt{oslo} \end{array}$$

$A = \{oslo, windy, norway, cold, winterIsComing\}$

$Baza = \{oslo\}$

# 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

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

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

Exercițiu. Arătați că funcția $f_S$ este monotonă.

# Clauze definite și funcții monotone

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

**Propoziție**

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

# Clauze definite și funcții monotone

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

## Propoziție

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

## Demonstrație

Arătăm că dacă $Y_1 \subseteq Y_2 \subseteq Y_3 \subseteq \ldots$ atunci $f_S(\bigcup_k Y_k) = \bigcup_k f_S(Y_k)$.

# Clauze definite și funcții monotone

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

## Propoziție

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

## Demonstrație

Arătăm că dacă $Y_1 \subseteq Y_2 \subseteq Y_3 \subseteq \ldots$ atunci $f_S(\bigcup_k Y_k) = \bigcup_k f_S(Y_k)$.

Din faptul că $f_S$ este crescătoare rezultă $f_S(\bigcup_k Y_k) \supseteq \bigcup_k f_S(Y_k)$

# Clauze definite și funcții monotone

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

## Propoziție

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

## Demonstrație

Arătăm că dacă $Y_1 \subseteq Y_2 \subseteq Y_3 \subseteq \ldots$ atunci $f_S(\bigcup_k Y_k) = \bigcup_k f_S(Y_k)$.

Din faptul că $f_S$ este crescătoare rezultă $f_S(\bigcup_k Y_k) \supseteq \bigcup_k f_S(Y_k)$

Demonstrăm în continuare că $f_S(\bigcup_k Y_k) \subseteq \bigcup_k f_S(Y_k)$. Fie $a \in f_S(\bigcup_n Y_k)$.
Sunt posibile trei cazuri

# Clauze definite și funcții monotone

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

## Propoziție

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

## Demonstrație

Arătăm că dacă $Y_1 \subseteq Y_2 \subseteq Y_3 \subseteq \ldots$ atunci $f_S(\bigcup_k Y_k) = \bigcup_k f_S(Y_k)$.

Din faptul că $f_S$ este crescătoare rezultă $f_S(\bigcup_k Y_k) \supseteq \bigcup_k f_S(Y_k)$

Demonstrăm în continuare că $f_S(\bigcup_k Y_k) \subseteq \bigcup_k f_S(Y_k)$. Fie $a \in f_S(\bigcup_n Y_k)$. Sunt posibile trei cazuri

- $a \in \bigcup_k Y_k$
  Există un $k \geq 1$ astfel încât $a \in Y_k$, deci $a \in f_S(Y_k) \subseteq \bigcup_k f_S(Y_k)$.

# Clauze definite și funcții monotone

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

## Propoziție

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

## Demonstrație

Arătăm că dacă $Y_1 \subseteq Y_2 \subseteq Y_3 \subseteq \ldots$ atunci $f_S(\bigcup_k Y_k) = \bigcup_k f_S(Y_k)$.

Din faptul că $f_S$ este crescătoare rezultă $f_S(\bigcup_k Y_k) \supseteq \bigcup_k f_S(Y_k)$

Demonstrăm în continuare că $f_S(\bigcup_k Y_k) \subseteq \bigcup_k f_S(Y_k)$. Fie $a \in f_S(\bigcup_n Y_k)$. Sunt posibile trei cazuri

- $a \in \bigcup_k Y_k$
  Există un $k \geq 1$ astfel încât $a \in Y_k$, deci $a \in f_S(Y_k) \subseteq \bigcup_k f_S(Y_k)$.
- $a \in Baza \subseteq \bigcup_k f_S(Y_k)$

# Clauze definite și funcții monotone

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

**Propoziție**

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este continuă.

**Demonstrație**

Arătăm că dacă $Y_1 \subseteq Y_2 \subseteq Y_3 \subseteq \ldots$ atunci $f_S(\bigcup_k Y_k) = \bigcup_k f_S(Y_k)$.

Din faptul că $f_S$ este crescătoare rezultă $f_S(\bigcup_k Y_k) \supseteq \bigcup_k f_S(Y_k)$

Demonstrăm în continuare că $f_S(\bigcup_k Y_k) \subseteq \bigcup_k f_S(Y_k)$. Fie $a \in f_S(\bigcup_n Y_k)$.
Sunt posibile trei cazuri

- $a \in \bigcup_k Y_k$
  Există un $k \geq 1$ astfel încât $a \in Y_k$, deci $a \in f_S(Y_k) \subseteq \bigcup_k f_S(Y_k)$.
- $a \in Baza \subseteq \bigcup_k f_S(Y_k)$
- Există $s_1, \ldots, s_n$ în $\bigcup_k Y_k$ astfel încât $(s_1 \wedge \ldots \wedge s_n \to a)$ este în $S$.

# Clauze definite și funcții monotone

## Demonstrație (cont.)

- Există $s_1, \ldots, s_n$ în $\bigcup_k Y_k$ astfel încât $(s_1 \wedge \ldots \wedge s_n \to a)$ este în $S$.

# Clauze definite și funcții monotone

## Demonstrație (cont.)

- Există $s_1, \ldots, s_n$ în $\bigcup_k Y_k$ astfel încât $(s_1 \wedge \ldots \wedge s_n \to a)$ este în $S$.

  Pentru fiecare $i \in \{1, \ldots, n\}$ există $k_i \in \mathbb{N}$ astfel încât $s_i \in Y_{k_i}$.

# Clauze definite și funcții monotone

## Demonstrație (cont.)

- Există $s_1, \ldots, s_n$ în $\bigcup_k Y_k$ astfel încât $(s_1 \land \ldots \land s_n \to a)$ este în $S$.

  Pentru fiecare $i \in \{1, \ldots, n\}$ există $k_i \in \mathbb{N}$ astfel încât $s_i \in Y_{k_i}$.

  Dacă $k_0 = \max\{k_1, \ldots, k_n\}$ atunci $Y_{k_i} \subseteq Y_{k_0}$ pentru orice $i \in \{1, \ldots, n\}$.

# Clauze definite și funcții monotone

## Demonstrație (cont.)

- Există $s_1, \ldots, s_n$ în $\bigcup_k Y_k$ astfel încât $(s_1 \wedge \ldots \wedge s_n \to a)$ este în $S$.

  Pentru fiecare $i \in \{1, \ldots, n\}$ există $k_i \in \mathbb{N}$ astfel încât $s_i \in Y_{k_i}$.

  Dacă $k_0 = \max\{k_1, \ldots, k_n\}$ atunci $Y_{k_i} \subseteq Y_{k_0}$ pentru orice $i \in \{1, \ldots, n\}$.

  Rezultă că $s_1, \ldots, s_n \in Y_{k_0}$, deci $a \in f_S(Y_{k_0}) \subseteq \bigcup_k f_S(Y_k)$.

# Clauze definite și funcții monotone

## Demonstrație (cont.)

- Există $s_1, \ldots, s_n$ în $\bigcup_k Y_k$ astfel încât $(s_1 \wedge \ldots \wedge s_n \to a)$ este în $S$.

  Pentru fiecare $i \in \{1, \ldots, n\}$ există $k_i \in \mathbb{N}$ astfel încât $s_i \in Y_{k_i}$.

  Dacă $k_0 = \max\{k_1, \ldots, k_n\}$ atunci $Y_{k_i} \subseteq Y_{k_0}$ pentru orice $i \in \{1, \ldots, n\}$.

  Rezultă că $s_1, \ldots, s_n \in Y_{k_0}$, deci $a \in f_S(Y_{k_0}) \subseteq \bigcup_k f_S(Y_k)$.

Am demonstrat că $f_S$ este continuă. $\square$

# Clauze definite și funcții monotone

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

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

aplicând Teorema Knaster-Tarski pentru CPO, obținem că

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

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

# Clauze definite și funcții monotone

- Analizați ce se întamplă când considerăm succesiv

$$\emptyset, \quad f_S(\emptyset), \quad f_S(f_S(\emptyset)), \quad f_S(f_S(f_S(\emptyset))), \ldots$$

La fiecare aplicare a lui $f_S$, rezultatul fie se mărește, fie rămâne neschimbat.

# Clauze definite și funcții monotone

- Analizați ce se întamplă când considerăm succesiv

$$\emptyset, \quad f_S(\emptyset), \quad f_S(f_S(\emptyset)), \quad f_S(f_S(f_S(\emptyset))), \ldots$$

  La fiecare aplicare a lui $f_S$, rezultatul fie se mărește, fie rămâne neschimbat.

- Să presupunem că în $S$ avem $k$ atomi. Atunci după $k+1$ aplicări ale lui $f_S$, trebuie să existe un punct în șirul de mulțimi obținute de unde o nouă aplicare a lui $f_S$ nu mai schimbă rezultatul (punct fix):

$$f_S(X) = X$$

# Clauze definite și funcții monotone

- Analizați ce se întamplă când considerăm succesiv

$$\emptyset, \quad f_S(\emptyset), \quad f_S(f_S(\emptyset)), \quad f_S(f_S(f_S(\emptyset))), \ldots$$

  La fiecare aplicare a lui $f_S$, rezultatul fie se mărește, fie rămâne neschimbat.

- Să presupunem că în $S$ avem $k$ atomi. Atunci după $k+1$ aplicări ale lui $f_S$, trebuie să existe un punct în șirul de mulțimi obținute de unde o nouă aplicare a lui $f_S$ nu mai schimbă rezultatul (punct fix):

$$f_S(X) = X$$

- Dacă aplicăm $f_S$ succesiv ca mai devreme până găsim un $X$ cu proprietatea $f_S(X) = X$, atunci găsim cel mai mic punct fix al lui $f_S$.

# Cel mai mic punct fix

## Exemplu

$$\begin{array}{rcl} cold & \to & wet \\ wet \wedge cold & \to & scotland \end{array}$$

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

Se observă că $f_S(\emptyset) =$

# Cel mai mic punct fix

## Exemplu

$$\begin{array}{rcl} cold & \to & wet \\ wet \wedge cold & \to & scotland \end{array}$$

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

Se observă că $f_S(\emptyset) = \emptyset$, deci $\emptyset$ este cel mai mic punct fix.

De aici deducem că niciun atom nu este consecință logică a formulelor de mai sus.

# Exemplu

## Exemplu

$$
\begin{array}{rcl}
cold & & \\
cold & \to & wet \\
windy & \to & dry \\
wet \wedge cold & \to & scotland
\end{array}
$$

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

# Exemplu

## Exemplu

$$\begin{array}{rcl} cold & & \\ cold & \rightarrow & wet \\ windy & \rightarrow & dry \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

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

$$f_S(\emptyset) \quad = \quad \{\ cold\ \}$$

# Exemplu

## Exemplu

$$\begin{array}{rcl} cold & & \\ cold & \rightarrow & wet \\ windy & \rightarrow & dry \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

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

$$\begin{array}{rcl} f_S(\emptyset) & = & \{\ cold\ \} \\ f_S(\{\ cold\ \}) & = & \{\ cold, wet\ \} \end{array}$$

# Exemplu

## Exemplu

$$\begin{array}{rcl} cold & & \\ cold & \rightarrow & wet \\ windy & \rightarrow & dry \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

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

$$\begin{array}{rcl} f_S(\emptyset) & = & \{\ cold\ \} \\ f_S(\{\ cold\ \}) & = & \{\ cold, wet\ \} \\ f_S(\{\ cold, wet\ \}) & = & \{\ cold, wet, scotland\ \} \end{array}$$

# Exemplu

## Exemplu

$$\begin{array}{rcl} cold & & \\ cold & \rightarrow & wet \\ windy & \rightarrow & dry \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

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

$$\begin{array}{rcl} f_S(\emptyset) & = & \{\ cold\ \} \\ f_S(\{\ cold\ \}) & = & \{\ cold, wet\ \} \\ f_S(\{\ cold, wet\ \}) & = & \{\ cold, wet, scotland\ \} \\ f_S(\{\ cold, wet, scotland\ \}) & = & \{\ cold, wet, scotland\ \} \end{array}$$

# Exemplu

## Exemplu

$$\begin{array}{rcl} cold & & \\ cold & \rightarrow & wet \\ windy & \rightarrow & dry \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

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

$$\begin{array}{rcl} f_S(\emptyset) & = & \{\ cold\ \} \\ f_S(\{\ cold\ \}) & = & \{\ cold, wet\ \} \\ f_S(\{\ cold, wet\ \}) & = & \{\ cold, wet, scotland\ \} \\ f_S(\{\ cold, wet, scotland\ \}) & = & \{\ cold, wet, scotland\ \} \end{array}$$

Deci cel mai mic punct fix este $\{\ cold, wet, scotland\ \}$.

# Programe logice și cel mai mic punct fix

**Teoremă**

*Fie $X$ este cel mai mic punct fix al funcției $f_S$. Atunci*

$$q \in X \quad \textit{ddacă} \quad S \models q.$$

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

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este definită prin

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

unde $A$ este mulțimea atomilor din $S$ și $Baza = \{p_i \mid p_i \in S\}$ este mulțimea atomilor care apar în clauzele unitate din $S$.

# Programe logice și cel mai mic punct fix

## Demonstrație

$(\Rightarrow)$ $q \in X \Rightarrow S \models q$.

- Funcția $f_S$ conservă atomii adevărați.
- Deci, dacă fiecare clauză unitate din $S$ este adevărată, după fiecare aplicare a funcției $f_S$ obținem o mulțime adevărată de atomi.

# Programe logice și cel mai mic punct fix

### Demonstrație

$(\Rightarrow)$ $q \in X \Rightarrow S \models q$.

- Funcția $f_S$ conservă atomii adevărați.
- Deci, dacă fiecare clauză unitate din $S$ este adevărată, după fiecare aplicare a funcției $f_S$ obținem o mulțime adevărată de atomi.

$(\Leftarrow)$ $S \models q \Rightarrow q \in X$.

- Fie $S \models q$. Presupunem prin absurd că $q \notin X$.
- Căutăm o evaluare $e$ care face fiecare clauză din $S$ adevărată, dar $q$ falsă.

# Programe logice și cel mai mic punct fix

## Demonstrație (cont.)

- Fie evaluarea

$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$

## Demonstrație (cont.)

- Fie evaluarea

$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$

- Evident, această interpretare face $q$ falsă.

# Programe logice și cel mai mic punct fix

## Demonstrație (cont.)

- Fie evaluarea

$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$

- Evident, această interpretare face $q$ falsă.
- Arătăm că $e^{+}(P) = 1$, pentru orice clauză $P \in S$.

# Programe logice și cel mai mic punct fix

## Demonstrație (cont.)

- Fie evaluarea
$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$
- Evident, această interpretare face $q$ falsă.
- Arătăm că $e^{+}(P) = 1$, pentru orice clauză $P \in S$.
- Fie $P \in S$. Avem două cazuri:

## Programe logice și cel mai mic punct fix

### Demonstrație (cont.)

- Fie evaluarea

$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$

- Evident, această interpretare face $q$ falsă.
- Arătăm că $e^+(P) = 1$, pentru orice clauză $P \in S$.
- Fie $P \in S$. Avem două cazuri:

  1. $P$ este o clauză unitate. Atunci $P \in X$, deci $e(P) = 1$.

# Programe logice și cel mai mic punct fix

## Demonstrație (cont.)

- Fie evaluarea
$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$
- Evident, această interpretare face $q$ falsă.
- Arătăm că $e^+(P) = 1$, pentru orice clauză $P \in S$.
- Fie $P \in S$. Avem două cazuri:

  1. $P$ este o clauză unitate. Atunci $P \in X$, deci $e(P) = 1$.
  2. $P$ este de forma $p_1 \wedge \ldots \wedge p_n \rightarrow r$. Atunci avem două cazuri:

# Programe logice și cel mai mic punct fix

## Demonstrație (cont.)

- Fie evaluarea

$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$

- Evident, această interpretare face $q$ falsă.
- Arătăm că $e^+(P) = 1$, pentru orice clauză $P \in S$.
- Fie $P \in S$. Avem două cazuri:

1. $P$ este o clauză unitate. Atunci $P \in X$, deci $e(P) = 1$.
2. $P$ este de forma $p_1 \wedge \ldots \wedge p_n \rightarrow r$. Atunci avem două cazuri:
   - există un $p_i$, $i = 1, \ldots, n$, care nu este în $X$. Deci $e^+(P) = 1$.

# Programe logice și cel mai mic punct fix

## Demonstrație (cont.)

- Fie evaluarea

$$e(p) = \begin{cases} 1, & \text{dacă } p \in X \\ 0, & \text{altfel} \end{cases}$$

- Evident, această interpretare face $q$ falsă.
- Arătăm că $e^+(P) = 1$, pentru orice clauză $P \in S$.
- Fie $P \in S$. Avem două cazuri:

  1. $P$ este o clauză unitate. Atunci $P \in X$, deci $e(P) = 1$.
  2. $P$ este de forma $p_1 \wedge \ldots \wedge p_n \rightarrow r$. Atunci avem două cazuri:
     - există un $p_i$, $i = 1, \ldots, n$, care nu este în $X$. Deci $e^+(P) = 1$.
     - toți $p_i$, $i = 1, \ldots, n$, sunt în $X$. Atunci $r \in f_S(X) = X$, deci $e(r) = 1$. În concluzie $e^+(P) = 1$.

$\square$

# Sistemul de deducție CDP

## Corolar

Sistemul de deducție pentru clauze definite propoziționale este complet pentru a arăta clauze unitate:

$$\text{dacă } S \models q, \text{ atunci } S \vdash q.$$

# Sistemul de deducție CDP

## Corolar

Sistemul de deducție pentru clauze definite propoziționale este complet pentru a arăta clauze unitate:

$$\text{dacă } S \models q, \text{ atunci } S \vdash q.$$

## Demonstrație

- Presupunem $S \models q$.
- Atunci $q \in X$, unde $X$ este cel mai mic punct fix al funcției $f_S$.
- Fiecare aplicare a funcției $f_S$ produce o mulțime demonstrabilă de atomi.
- Cum cel mai mic punct fix este atins după un număr finit de aplicări ale lui $f_S$, orice $a \in X$ are o derivare.

$\square$

# Metodă de decizie

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**

# Metodă de decizie

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.
Exercițiu. De ce?

# Metodă de decizie

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.
Exercițiu. De ce?

Nu acesta este algoritmul folosit de Prolog!

Pe săptămâna viitoare!