Topici speciale în logică și securitate I

Elemente teoretice introductive

Ioana Leuștean

Master anul II, Sem. I, 2019-2020

Cuprins

Mulțimi parțial ordonate

Relații binare

A mulțime, $R \subseteq A \times A$

Relația binară R se numește:

- reflexivă: $(x, x) \in R$ or. $x \in A$
- simetrică: $(x, y) \in R$ implică $(y, x) \in R$ or. $x,\ y \in A$
- antisimetrică: $(x, y) \in R$ și $(y, x) \in R$ implică $x = y$ or. $x,\ y \in A$
- tranzitivă: $(x, y) \in R$ și $(y, z) \in R$ implică $(x, z) \in R$ or. $x,\ y,\ z \in A$
- relație de preordine: reflexivă, tranzitivă
- relație de ordine: reflexivă, antisimetrică, tranzitivă
- relație de echivalență: reflexivă, simetrică, tranzitivă

Mulțimi parțial ordonate

O mulțime parțial ordonată (**mpo**) este o pereche (A, R), unde A este o mulțime și R este o relație de ordine pe A.

Exemple.

- $(\mathcal{P}(T), \subseteq)$ unde T este o mulțime,
- $(\mathbb{N}, \leq)$, $(\mathbb{N}, |)$ cu $|$ relația de divizibilitate,
- $(Pf(A, B), \prec)$ unde
A, B sunt mulțimi,
$Pf(A, B) = \{f : A \rightharpoonup B | f \text{ functie parțială}\}$,
$f \prec g \Longrightarrow dom(f) \subseteq dom(g)$ si $f(a) = g(a)$ or. $a \in dom(f)$

Relațiile de ordine sunt uzual notate cu $\leq$.
Dacă $(A, \leq)$ este **mpo** si $\geq := \leq^{-1}$ atunci
$(A, \geq)$ este **mpo**.

Mulțimi total ordonate

Fie $(A, \leq)$ *mpo*. Două elemente a_1, $a_2 \in A$ se numesc **comparabile** dacă $a_1 \leq a_2$ sau $a_2 \leq a_1$.

Exemplu. In $(\mathbb{N}, |)$ elementele 2 si 4 sunt comparabile, dar elementele 3 si 7 nu sunt comparabile.

O relație de ordine parțială se numește **totală** (**liniară**) dacă oricare două elemente sunt comparabile. O **mulțime total ordonată** este o pereche $(A, \leq)$ unde A este o mulțime și $\leq$ este o relație de ordine totală pe A. Pentru mulțimile total ordonate este folosită și denumirea de **lanț**.

Exemplu. $(\mathcal{LR}, \leq_{lex})$ este lanț, unde $\mathcal{LR}$ este mulțimea cuvintelor din DEX, iar $\leq_{lex}$ este relația de ordine lexicografică.

Produsul cartezian de lanțuri

Fie $(A_1, \leq_1)$, $(A_2, \leq_2)$ lanțuri.

- Pe $A_1 \times A_2$ definim **relația de ordine pe componente**
  $(x_1, x_2) \leq (y_1, y_2) \Leftrightarrow x_1 \leq_1 y_1$ si $x_2 \leq_2 y_2$.
  $(A_1 \times A_2, \leq)$ este **mpo**.
  Dacă $|A_1|, |A_2| \geq 2$ atunci $(A_1 \times A_2, \leq)$ **nu e lanț**.

Exemplu. In $\mathbb{R} \times \mathbb{R}$ elementele $(2, 3)$ si $(4, 1)$ nu sunt comparabile.

- Pe $A_1 \times A_2$ definim **relația de ordine lexicografică**
  $(x_1, x_2) \leq_{lex} (y_1, y_2) \Leftrightarrow (x_1 \leq_1 y_1$ si $x_1 \neq y_1)$ sau
  $(x_1 = y_1$ si $x_2 \leq_2 y_2)$.
  $(A_1 \times A_2, \leq)$ este **lanț**.

Exercitiu. Definiți relația de ordine lexicografică pe produsul cartezian a n lanțuri $(A_1, \leq_1), \cdots, (A_n, \leq_n)$.

Elemente minimale si maximale

Fie $(A, \leq)$ **mpo**. Un element $e \in A$ se numește

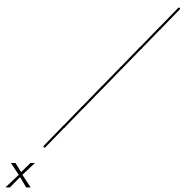
- **element minimal** dacă $(a \leq e \Rightarrow a = e)$;
- **prim element** (**minim**) dacă $e \leq a$ or $a \in A$;
- **element maximal** dacă $(e \leq a \Rightarrow a = e)$;
- **ultim element** (**maxim**) dacă $a \leq e$ or $a \in A$.

Exemplu. In **mpo** $(\{2, 4, 5, 10, 12, 20, 25\}, |)$, elementele minimale sunt 2 si 5, iar elementele maximale sunt 12, 20, 25.

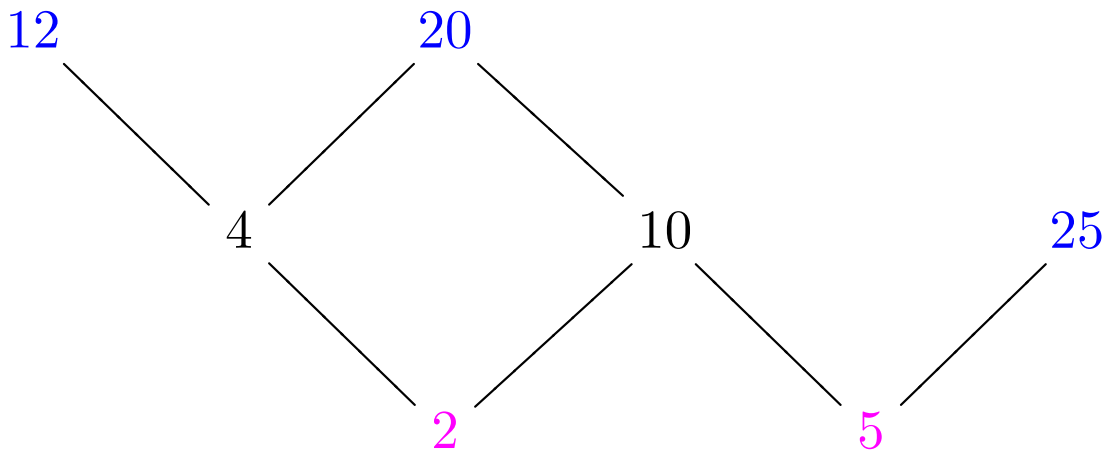
Orice minim (maxim) este element minimal (maximal).
Invers nu este adevărat.

Diagrame Hasse

$x \leq y$ este reprezentat prin



Exemplu. $(\{2, 4, 5, 10, 12, 20, 25\}, |)$ este reprezentata prin



Sortarea topologică

Este folosită în probleme de planificare. Presupunem că un project este format din mai multe procese care se conditioneaza intre ele: procesul p nu poate începe decât după terminarea procesului q. Mulțimea proceselor devine astfel mulțime parțial ordonată $(P, \leq)$. Se pune problema găsirii unei secvențe de execuție a proceselor.

O relație de ordine totală $\lhd \subseteq P \times P$ este compatibilă cu relația de ordine parțială $\leq$ dacă $x \leq y \Rightarrow x \lhd y$ oricare $x, y \in P$. Determinarea unei relații de ordine totale compatibile cu o relație de ordine parțială dată se numește **sortare topologică**.

Algoritm de sortare topologică

Intrare: $(P, \leq)$ **mpo**, $n = |P|$ $k := 1$

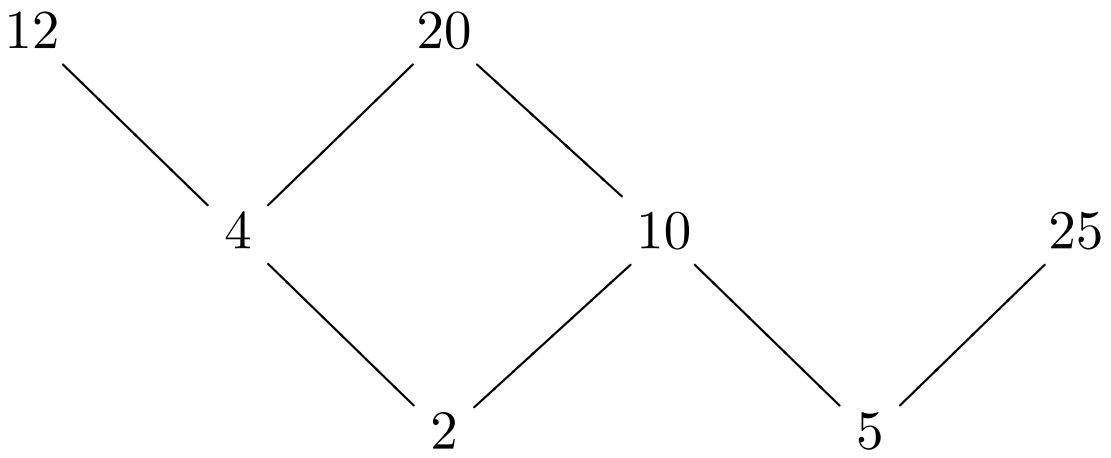
while $P \neq \emptyset$
{
$p_k :=$ un element minimal al lui P;

$P := P \setminus \{p_i\}$;

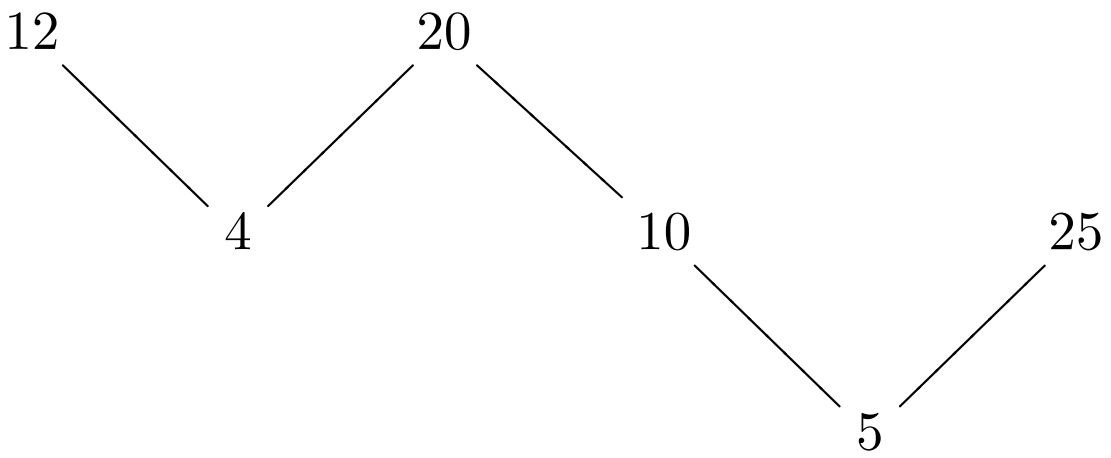
$k := k + 1$
}

Iesire: $p_1 \lhd \cdots \lhd p_n$ este o ordonare totala compatibilă

Sortare topologică

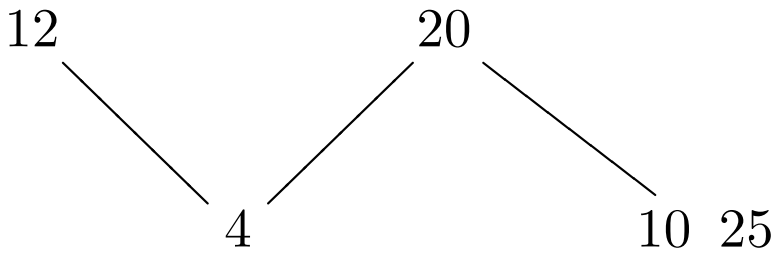


$x_1 := 2$

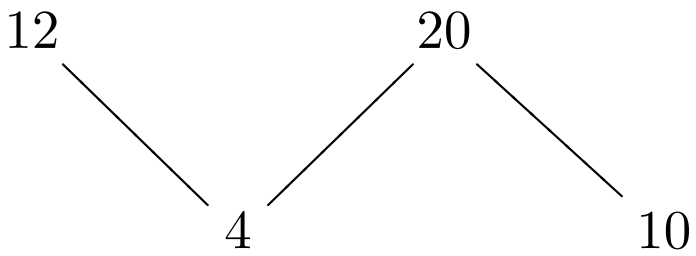


$x_2 := 5$

Sortare topologică



$x_3 := 25$



$x_4 := 4$

$\vdots$

$2 \lhd 5 \lhd 25 \lhd 4 \lhd 10 \lhd 12 \lhd 20$

Soluția nu este unică.

Infimum si supremum

Fie $(A, \leq)$ **mpo**. Un element $a \in A$ se numește

- **minorant** al lui X daca $a \leq x$ or. $x \in X$;
- **majorant** al lui X daca $x \leq a$ or. $x \in X$;
- **infimum** al lui X daca a este cel mai mare minorant (a este ultim element in multimea minoranților lui X);
- **supremum** al lui X dacă a este cel mai mic majorant (a este prim element in multimea majoranților lui X).

Exercițiu. Infimumul (supremumul) unei mulțimi, dacă există, este unic.

Infimumul (supremumul) lui X îl vom nota $\inf X$ ($\sup X$).
Dacă $X = \{x_1, x_2\}$ atunci notăm $\inf\{x_1, x_2\}$, $\sup\{x_1, x_2\}$.

Exemple

- $(\mathbb{R}, \leq)$, $X = (3, 4]$, $Y = \mathbb{N}$

mulțimea majoranților lui X este $[4, \infty)$,
4 este ultim element pt. X,
mulțimea minoranților lui X este $(-\infty, 3]$,
3 este infimum pt. X,
Y nu are majoranți (ultim element, supremum),
mulțimea minoranților lui Y este $(-\infty, 0)$,
0 este prim element pt. Y .

- $(\mathbb{N}, |)$, $X = \{n_1, n_2\}$

$\sup X = \sup\{n_1, n_2\} = cmmmc\{n_1, n_2\}$
$\inf X = \inf\{n_1, n_2\} = cmmdc\{n_1, n_2\}$

- $(\{2, 4, 5, 10, 12, 20, 25\}, |)$, $X = \{12, 20, 25\}$

X nu are minoranți si nici majoranți

- $(\mathcal{P}(T), \subseteq)$, $\mathcal{X} \subseteq \mathcal{P}(T)$

$\sup \mathcal{X} = \bigcup\{Y | Y \in \mathcal{X}\}$, $\inf \mathcal{X} = \bigcap\{Y | Y \in \mathcal{X}\}$

Latici

Latici

Definitia L1.
O **mpo** $(L, \leq)$ este latice dacă $\sup\{x_1, x_2\}$, $\inf\{x_1, x_2\}$ există oricare x_1, $x_2 \in L$.

Infimumul (supremumul) devin operații pe L:
$\vee : L \times L \to L$, $x_1 \vee x_2 := \sup\{x_1, x_2\}$,
$\wedge : L \times L \to L$, $x_1 \wedge x_2 := \inf\{x_1, x_2\}$.

Propoziție

Următoarele identități sunt satisfăcute:

- asociativitate:
  $(x \vee y) \vee z = x \vee (y \vee z)$, $(x \wedge y) \wedge z = x \wedge (y \wedge z)$,
- comutativitate: $x \vee y = y \vee x$, $x \wedge y = y \wedge x$,
- absorbție: $x \vee (x \wedge y) = x$, $x \wedge (x \vee y) = x$.

Demonstrație. exercitiu.

Laticea $(L, \leq)$ devine structură algebrică $(L, \vee, \wedge)$.

Definiția algebrică a Laticilor

Definitia L2.
O **latice** este o structură algebrică $(L, \vee, \wedge)$ unde $\vee$ si $\wedge$ sunt operații binare asociative, comutative și care verifică proprietățile de absorbție.

Lema. $x \vee y = y \Leftrightarrow x \wedge y = x$ or. $x, y \in L$.
Demonstrație. Daca $x \vee y = y$, atunci $x = x \wedge (x \vee y) = x \wedge y$.

Propozitie

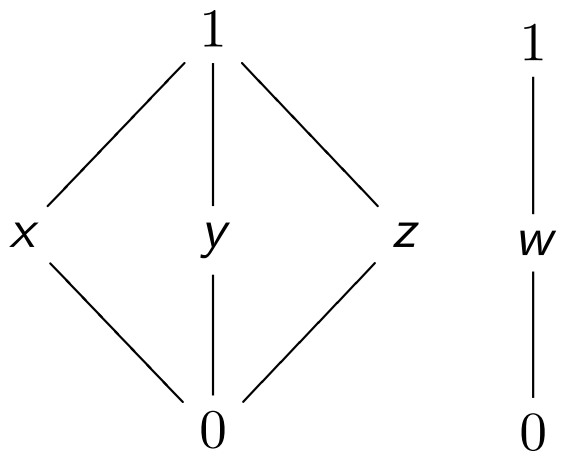
Fie $(L, \vee, \wedge)$ in sensul Definiției L2. Definim
$x \leq y \Leftrightarrow x \vee y = y \Leftrightarrow x \wedge y = x$.
Atunci $(L, \leq)$ este latice in sensul Definitiei L1. In plus
$\sup\{x, y\} = x \vee y$, $\inf\{x, y\} = x \wedge y$ or. $x, y \in L$.

Latici marginite

O latice este **marginită** dacă are prim și ultim element. Primul element se notează cu 0, iar ultimul element se notează cu 1.
O latice marginită va fi notata prin $(L, \leq, 0, 1)$, iar ca structură algebrică prin $(L, \vee, \wedge, 0, 1)$. Se observă că:
$x \vee 0 = x$, $x \wedge 0 = 0$, $x \vee 1 = 1$, $x \wedge 1 = x$ or. $x \in L$.

Elementul x este **complement** al lui y daca $x \vee y = 1$ si $x \wedge y = 0$.
O latice este **complementată** dacă orice element are cel puțin un complement.

Exemplu. Determinați elementele complementate în laticile:



Algebre Boole

O latice $(L, \vee, \wedge)$ este **distributivă** dacă, or x, $y \in L$:
$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ și $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

Lemă

Într-o latice distributivă și marginită, un element are cel mult un complement.
Demostrație. Fie y_1, y_2 complemente pentru x. Obținem
$y_1 = y_1 \wedge 1 = y_1 \wedge (y_2 \vee x) = y_1 \wedge y_2 = y_2 \wedge (y_1 \vee x) = y_2 \wedge 1 = y_2$

O **algebra Boole** este o latice distributivă și complementată cu prim si ultim element.Dacă B este o algebră Boole, atunci pentru orice element există un unic complement. Putem defini o operație

$^- : B \to B$ prin $\bar{x} :=$ complementul lui x.

O algebră Boole este o structură algebrică

$$(B, \vee, \wedge, ^-, 0, 1).$$

George Boole - *The Mathematical Analysis of Logic* (1847) Analiza raționamentelor prin metode asemănătoare calculului algebric.

Teoreme de punct fix

CPO. Latici

CPO

O **mpo completă** (CPO) este o mulțime parțial ordonată $(C, \leq)$ cu proprietățile:

- C are prim element $\bot$,
- $\sup X$ există pentru orice **lanț** $X \subseteq C$.

Exemplu. $(Pf(A, B), \prec)$ este CPO.

Latice

O **mpo** $(L, \leq)$ este **latice** dacă $\sup\{x_1, x_2\}$ si $\inf\{x_1, x_2\}$ există oricare ar fi x_1, $x_2 \in L$. Laticea $(L, \leq)$ este **completă** dacă $\inf X$ si $\sup X$ există oricare ar fi $X \subseteq L$.

Exemplu. $(\mathcal{P}(A), \subseteq)$ este latice completă.

Exercițiu.

(a) Orice latice completă este CPO.
(b) Orice latice completă are prim si ultim element.

Funcție crescătoare. Puncte fixe.

Funcție crescătoare

Dacă $(A, \leq_A)$ si $(B, \leq_B)$ sunt **mpo** atunci o funcție $f : A \to B$ este **crescătoare** dacă $a_1 \leq_A a_2 \Rightarrow f(a_1) \leq_B f(a_2)$ or. $a_1, a_2 \in A$.

Funcție continuă

Dacă $(A, \leq_A)$ si $(B, \leq_B)$ sunt CPO atunci o funcție $f : A \to B$ este **continuă** dacă pentru orice lanț $\{a_n | n \in \mathbb{N}\}$ din A

$$f(\sup\{a_n | n \in \mathbb{N}\}) = \sup\{f(a_n) | n \in N\}.$$

Exercitiu. Orice funcție continuă este crescătoare.

Punct fix

Un element $a \in A$ este **punct fix** al unei funcții $f : A \to A$ dacă

$$f(a) = a.$$

Teoreme de punct fix

Teorema Kleene pentru latici complete

Fie $(L, \leq)$ latice completă și $\mathbf{F} : L \to L$ o funcție crescătoare. Atunci $a = \inf\{x \in L | \mathbf{F}(x) \leq x\}$ este cel mai mic punct fix al funcției $\mathbf{F}$.

Teorema Knaster-Tarski pentru CPO

Fie $(C, \leq)$ o CPO si $\mathbf{F} : C \to C$ o funcție continuă. Atunci $a = \sup\{\mathbf{F}^n(\bot) | n \in \mathbb{N}\}$ cel mai mic punct fix al funcției $\mathbf{F}$.

Observăm că în ipotezele ultimei teoreme secvența $\mathbf{F}^0(\bot) = \bot \leq \mathbf{F}(\bot) \leq \mathbf{F}^2(\bot) \leq \cdots \leq \mathbf{F}^n(\bot) \leq \cdots$ este un lanț, deci a există.

Teoreme de punct fix

Exemplu.
$Pf(\mathbb{N},\mathbb{N})$ CPO, $\perp : \mathbb{N} \rightharpoonup \mathbb{N}$, $\perp(k)$ nedefinită or. $k \in \mathbb{N}$
$\mathbf{F} : Pf(\mathbb{N},\mathbb{N}) \to Pf(\mathbb{N},\mathbb{N})$

$$\mathbf{F}(g)(k) := \begin{cases} 1, & k = 0 \\ k * g(k-1), & k > 0 \text{ si } g(k-1) \text{ e definită} \\ \text{nedefinit}, & \text{altfel} \end{cases}$$

Deoarece $\mathbf{F}$ este continuă, conform Teoremei Knaster-Tarski, cel mai mic punct fix al funcției $\mathbf{F}$ este $f = \sup\{\mathbf{F}^n(\perp) | n \in \mathbb{N}\}$.

$$f(k) = \mathbf{F}(f)(k) := \begin{cases} 1, & k = 0 \\ k * f(k-1), & k > 0 \text{ si } f(k-1) \text{ e definită} \\ \text{nedefinit}, & \text{altfel} \end{cases}$$

f este *funcția factorial*.

Teoremele de punct fix sunt folosite in semantica denotațională pentru a defini semantica instrucțiunii `while`.

Knaster-Tarski pentru CPO

Demonstrație.
Fie $(C, \leq)$ CPO si $\mathbf{F} : C \to C$ funcție continuă.
(I) $a = \sup\{\mathbf{F}^n(\bot) | n \in \mathbb{N}\}$ este punct fix.

$$\begin{aligned}\mathbf{F}(a) &= \mathbf{F}(\sup\{\mathbf{F}^n(\bot) | n \in \mathbb{N}\}) \\ &= \sup\{\mathbf{F}(\mathbf{F}^n(\bot)) | n \in \mathbb{N}\} \text{ (continuitate)} \\ &= \sup\{\mathbf{F}^{n+1}(\bot) | n \in \mathbb{N}\} \\ &= \sup\{\mathbf{F}^n(\bot) | n \in \mathbb{N}\} = a\end{aligned}$$

(II) 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 $\mathbf{F}^n(\bot) \leq b$. Pentru $n = 0$, $\mathbf{F}^0(\bot) = \bot \leq b$ deoarece $\bot$ este prim element. Dacă $\mathbf{F}^n(\bot) \leq b$, atunci $\mathbf{F}^{n+1}(\bot) \leq \mathbf{F}(b)$, deoarece $\mathbf{F}$ este crescătoare. Dar $\mathbf{F}(b) = b$, deci $\mathbf{F}^{n+1}(\bot) \leq b$.

Kleene pentru latici complete

Demonstrație.
Fie $(L, \leq)$ latice completă și $\mathbf{F} : C \to C$ funcție crescătoare.
(I) $a = \inf\{x \in L | \mathbf{F}(x) \leq x\}$ este punct fix.
Fie $X := \{x \in L | \mathbf{F}(x) \leq x\}$ si fie $x \in X$. Atunci $a \leq x$, deci $\mathbf{F}(a) \leq \mathbf{F}(x) \leq x$. Am demonstrat ca $\mathbf{F}(a) \leq x$ oricare $x \in X$, deci $\mathbf{F}(a)$ este un minorant pentru X. În consecință, $\mathbf{F}(a) \leq a$. Observăm că $\mathbf{F}(a) \in X$, deci $a \leq \mathbf{F}(a)$. Din $\mathbf{F}(a) \leq a$ și $a \leq \mathbf{F}(a)$ rezulta $\mathbf{F}(a) = a$.

(II) a este cel mai mic punct fix.
Fie b un alt punct fix, i.e. $\mathbf{F}(b) = b$.Atunci $b \in X$, deci $a \leq b$.

Aplicații

Ce este analiza statică?

Analiza statică presupue analiza proprietăților programelor fără a le rula.

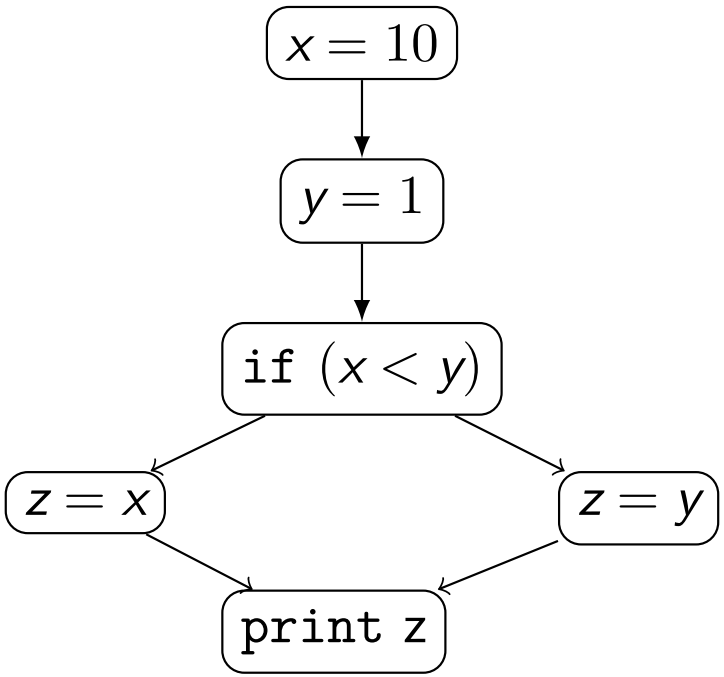
Exemple de proprietăți: analiza tipurilor, pointeri nuli, atribuiri nefolosite, vulnerabilitati de cod (de exemplu depășiri de indici), etc.

1. Mulțimi parțial ordonate
2. Latici
3. Teoreme de punct fix
4. Aplicații

Veți studia mai multe probleme de analiză statică în al doilea modul!

Control Flow Graphs (CFG)

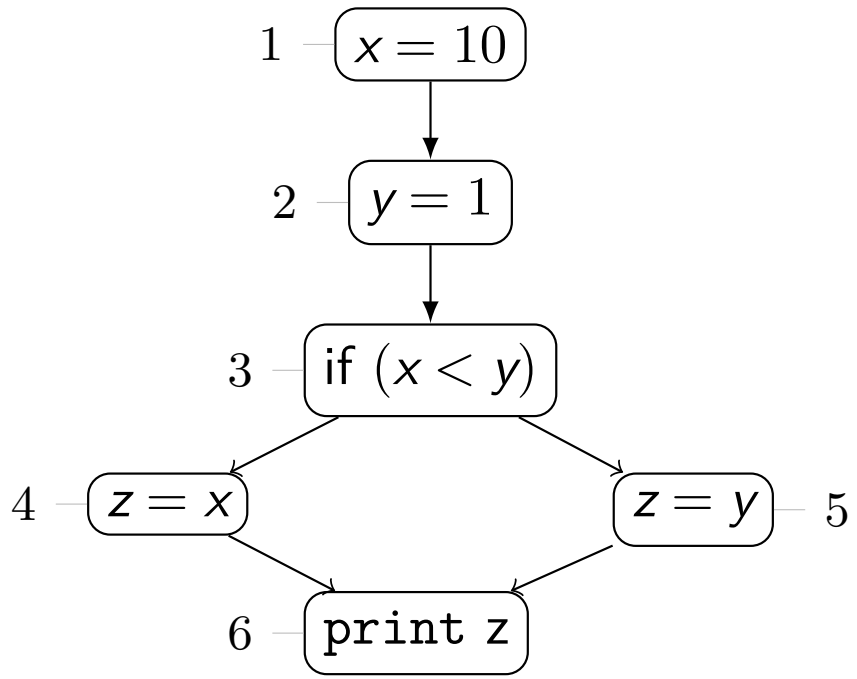
```
x = 10;
y = 1;
if  (x < y) z = x;
     else   z = y;
print z
```



Control Flow Graphs (CFG)

```
x = 10;
y = 1;
if  (x < y) z = x;
     else   z=y;
print z
```

Adăugăm etichete pentru nodurile grafului.



Exemplu: determinarea variabilelor *live*

https://cs420.epfl.ch/c/06_dataflow-analysis.html

O variabilă este activă (live) într-un punct al programului dacă este posibil ca valoarea curentă variabilei să fie citită înainte de a fi rescrisă.

Pentru un nod n din CFG vom nota cu v_n mulțimea variabilelor care sunt *live* înaintea lui.

Observăm că

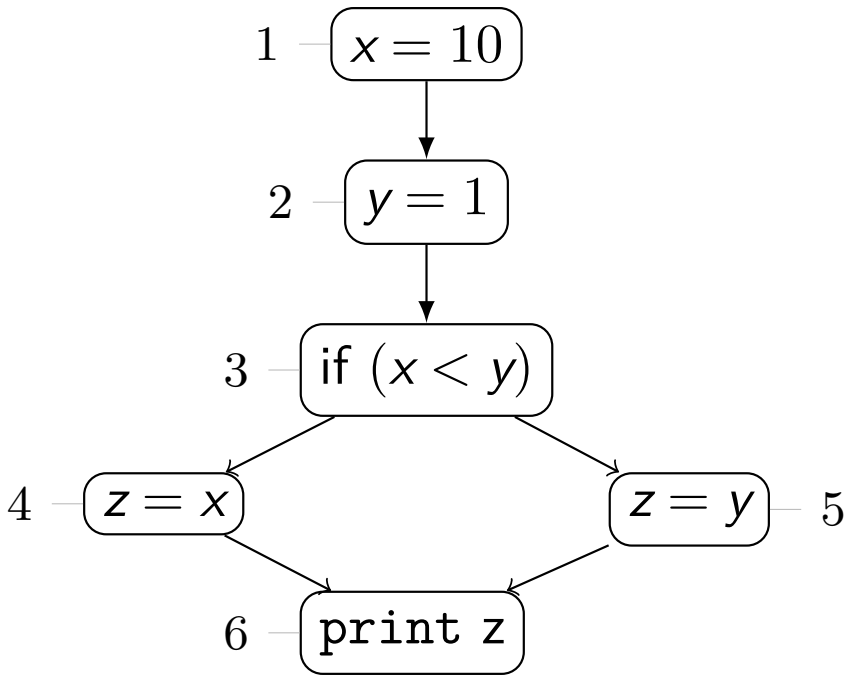
$$v_n = \left(\bigcup \{ v_i \mid i \text{ succesor al lui } n \} \setminus \mathit{Written}(n) \right) \cup \mathit{Read}(n)$$

$\mathit{Written}(n)$ este mulțimea variabilelor care primesc valori în nodul n,
$\mathit{Read}(n)$ este mulțimea variabilelor care sunt citite în nodul n.

Exemplu: determinarea variabilelor *live*

Ținând cont de condiția găsită obținem un sistem de constrângeri:

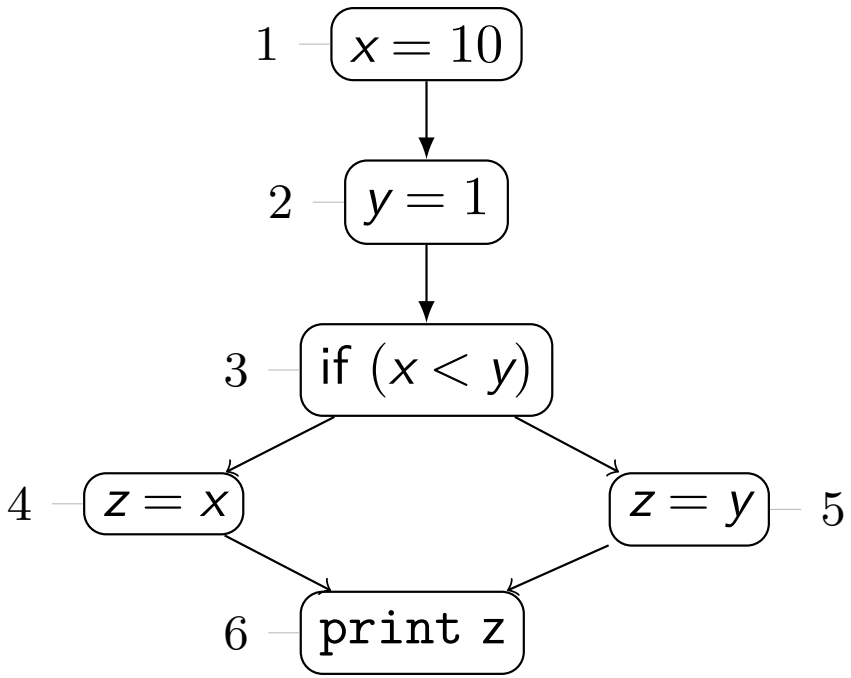
$$
\begin{aligned}
v_1 &= v_2 \setminus \{x\} \\
v_2 &= v_3 \setminus \{y\} \\
v_3 &= v_4 \cup v_5 \cup \{x, y\} \\
v_4 &= (v_6 \setminus \{z\}) \cup \{x\} \\
v_5 &= (v_6 \setminus \{z\}) \cup \{y\} \\
v_6 &= \{z\}
\end{aligned}
$$



Exemplu: determinarea variabilelor *live*

Rezolvând sistemul, găsim o soluție:

$$
\begin{array}{lcl}
v_1 & = & \emptyset \\
v_2 & = & \{x\} \\
v_3 & = & \{x, y\} \\
v_4 & = & \{x\} \\
v_5 & = & \{y\} \\
v_6 & = & \{z\}
\end{array}
$$



CFG - observații

- În analiza problemei anterioare, fiecărui nod din graful de control al fluxului i-am asociat o mulțime de variabile, ce reprezintă soluția unui sistem de ecuații in $\mathcal{P}(Var)$, unde Var este mulțimea variabilelor din program.

- În general, analiza unei probleme poate conduce la un sistem de ecuații într-o latice oarecare L. Există o metodă generală de rezolvare a unui astfel de sistem?

Laticea produs

Fie L o latice și $n \geq 1$.

Laticea produs L^n

Pe L^n definim relația de ordine pe componente:

$$(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n) \text{ ddacă } x_i \leq y_i \text{ pt. orice } i \in \{1, \ldots, n\}$$

Observăm că $(L^n, \leq)$ este latice. În plus, dacă L este latice completă atunci și L^n este latice completă.

Sisteme de ecuații în latici complete

Fie L o latice completă, $n \geq 1$ și
$F_1, \ldots, F_n : L^n \to L$ funcții monotone.
Vrem să rezolvăm sistemul

$$\begin{aligned} x_1 &= F_1(x_1, \ldots, x_n) \\ x_2 &= F_2(x_1, \ldots, x_n) \\ &\ldots \\ x_n &= F_n(x_1, \ldots, x_n) \end{aligned}$$

cu $x_1, \ldots, x_n \in L$.

Definim funcția $F : L^n \to L^n$ prin

$$F(x_1, \ldots, x_n) = (F_1(x_1, \ldots, x_n), \ldots, F_1(x_1, \ldots, x_n))$$

și observăm că sistemul se poate scrie

$$F(x_1, \ldots, x_n) = (x_1, \ldots, x_n)$$

Sistemul de ecuații poate fi rezolvat folosind Teorema de punct fix!

Exemplu: determinarea variabilelor *live*

În consecință, a rezolva sistemul:

$$\begin{aligned} v_1 &= v_2 \setminus \{x\} \\ v_2 &= v_3 \setminus \{y\} \\ v_3 &= v_4 \cup v_5 \cup \{x, y\} \\ v_4 &= (v_6 \setminus \{z\}) \cup \{x\} \\ v_5 &= (v_6 \setminus \{z\}) \cup \{y\} \\ v_6 &= \{z\} \end{aligned}$$

revine la a găsi o soluție a ecuației:

$$F(x_1, x_2, x_3, x_4, x_5, x_6) = \\ (x_2 \setminus \{x\}\,,\ x_3 \setminus \{y\}\,,\ x_4 \cup x_5 \cup \{x, y\}\,,\ (x_6 \setminus \{z\}) \cup \{x\}\,,\ (x_6 \setminus \{z\}) \cup \{y\}\,,\ \{z\})$$

în laticea completă $(\mathcal{P}(Var), \subseteq, \emptyset, Var)$.

Rezolvarea folosind teorema de punct fix

$$F(x_1, x_2, x_3, x_4, x_5, x_6) =$$
$$(x_2 \setminus \{x\}\,,\ x_3 \setminus \{y\}\,,\ x_4 \cup x_5 \cup \{x, y\}\,,\ (x_6 \setminus \{z\}) \cup \{x\}\,,\ (x_6 \setminus \{z\}) \cup \{y\}\,,\ \{z\})$$

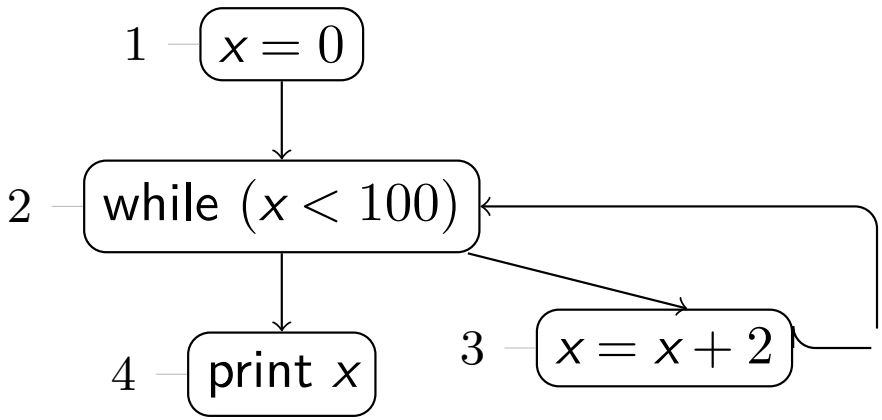
Determinarea celui mai mic punct fix:

	x_1	x_2	x_3	x_4	x_5	x_6
$\bot$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$F(\bot)$	$\emptyset$	$\emptyset$	$\{x, y\}$	$\{x\}$	$\{y\}$	$\{z\}$
$F^2(\bot)$	$\emptyset$	$\{x\}$	$\{x, y\}$	$\{x\}$	$\{y\}$	$\{z\}$
$F^3(\bot)$	$\emptyset$	$\{x\}$	$\{x, y\}$	$\{x\}$	$\{y\}$	$\{z\}$

unde $\bot = (\emptyset, \ldots, \emptyset)$

Exemplu: Semantica urmelor trace semantics

```
x = 0;
while (x<100)
      {x=x+2;}
```

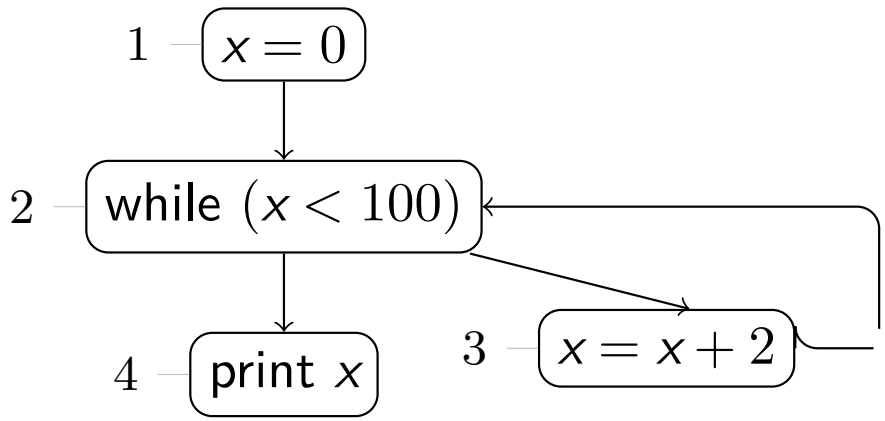


Semantica urmelor (trace semantics)

Semantica urmelor definește comportamentul programului printr-un sistem de tranziții între stări.

$Nodes = \{1, 2, 3, 4\}$, $Var = \{x\}$,
$\mathbb{Z}^{Var} \simeq \mathbb{Z}$
$St = \mathbb{Z} \times Nodes$

$\tau : \mathcal{P}(St) \to \mathcal{P}(St)$
$\tau(\{(x, 1)\}) = \{(x, 2)\}$
$\tau(\{(x, 2)\}) = \{(x, 3)\}$ dacă $x < 100$
$\tau(\{(x, 2)\}) = \{(x, 4)\}$ dacă $x \geq 100$
$\tau(\{(x, 3)\}) = \{(x + 2, 2)\}$
$\tau(S) = \bigcup_{s \in S} \tau(s)$



Fie $I \subseteq St$ mulțimea stărilor inițiale.
Definim $\Phi_I : \mathcal{P}(St) \to \mathcal{P}(St)$ $\Phi_I(S) = I \cup \tau(S)$

Aplicând Teorema de punct fix obținem $R = lfp(\Phi_I)$, cel mai mic punct fix al funcției Φ.

R este mulțimea stărilor accesibile din mulțimea de stări inițiale I.

Exemplu: Interpretări abstracte

Semantica urmelor este o semantica concretă, comportamentul unui program poate fi descris ca un șir de tranziții și asociază unei mulțimi de stări o altă mulțime de stări.
Dorim să analizăm comportamentul programului din punctul de vedere al unei proprietăți satisfăcute de o mulțime de stări. Acest proces se numește abtractizare și poate fi reprezentat matematic printr-o funcție

$$\alpha : \mathcal{P}(St) \to A$$

unde A este domeniul abstract prin care reprezentăm proprietatea pe care o analizăm. Procesul invers se numește concretizare și poate fi reprezentat printr-o funcție

$$\gamma : A \to \mathcal{P}(St)$$

În teoria interpretărilor abstracte, α și γ definesc o conexiune Galois:

$$\mathcal{P}(St) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} A$$

Conexiune Galois

Fie $(C, \leq_C)$ și $(A, \leq_A)$ mulțimi parțial ordonate. O conexiune Galois este definită de două funcții $C \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} A$ care satisfac următoarea proprietate:

$$\alpha(c) \leq_A a \text{ ddacă } c \leq_C \gamma(a)$$

pentru orice $a \in A$ și $c \in C$.

Proprietăți:

- α și γ sunt funcții crescătoare
- $c \leq \gamma(\alpha(c))$ pentru orice $c \in C$

Exemplu: interpretări abstracte

Domeniul abstract este $(A = \{pos, neg, zero, \bot, \top\}, \leq)$ cu relația de ordine:

$$\bot \leq a \leq \top \text{ pentru orice } a \in \{pos, neg, zero\}.$$

Fie $St = \mathbb{Z} \times Nodes$ din exemplul anterior. Definim

$\alpha : \mathcal{P}(St) \to A$
$\alpha(\emptyset) = \bot$
$\alpha(S) = pos$ dacă $x > 0$ oricare $(x, n) \in S$
$\alpha(S) = zero$ dacă $x = 0$ oricare $(x, n) \in S$
$\alpha(S) = neg$ dacă $x < 0$ oricare $(x, n) \in S$
$\alpha(S) = \top$ altfel.

$\gamma : A \to \mathcal{P}(St)$
$\gamma(\bot) = \emptyset$, $\gamma(\top) = St$,
$\gamma(pos) = \{(x, n) \in St \mid x > 0\}$,
$\gamma(neg) = \{(x, n) \in St \mid x < 0\}$,
$\gamma(zero) = \{(0, n) \mid n \in Nodes\}$.

$$\mathcal{P}(St) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} A$$

Transformări abstracte

Am calculat stările aplicând folosind teorema de punct fix pentru funția $\Phi_I : \mathcal{P}(St) \to \mathcal{P}(St)$.

Un exemplu de transformare abstractă este $\Phi_I^{\#} : A \to A$ definită prin $\Phi_I^{\#} = \alpha \circ \Phi_I \circ \gamma$

Se observă că și pentru $\Phi_I^{\#}$ se poate aplica teorema de punct fix. În plus, $\alpha(\mathit{lfp}(\Phi_I)) = \mathit{lfp}(\Phi_I^{\#})$.

În abordarea concretă $\mathit{lfp}(\Phi_I)$ calculează starile în care se poate ajunge dintr-o stare inițială dată.

În abordarea abstractă $\mathit{lfp}(\Phi_I^{\#})$ calculează semnul pe care îl vor avea variabilele după execuția programului.

În consecință, o valoare concretă (o mulțime concretă de stări) este aproximată printr-un element din domeniul abstract ales (un element din laticea semnelor).