Logică și structuri discrete

Relații. Funcții parțiale

Marius Minea
marius@cs.upt.ro

http://www.cs.upt.ro/~marius/curs/lsd/

21 octombrie 2013

Relații în lumea reală și informatică

Noțiunea matematică de relație modelează legătura dintre două elemente (posibil de tip diferit)

- relații umane: copil , părinte , prieten
- relații cantitative : egal, mai mic

Transpuse în informatică:

- rețele sociale : “prieten”, “follow”, “în cercuri”, etc.

O relație între elementele aceleiași mulțimi definește un *graf*
$\Rightarrow$ relațiile sunt o noțiune cheie în teoria grafurilor

Definiția unei relații

O relație binară R între două mulțimi A și B e o submulțime a produsului cartezian $A \times B$: $\quad R \subseteq A \times B$

Notăm: $(x, y) \in R$, sau uneori $x \; R \; y$ (x e în relație cu y)

Generalizat, putem avea o relație n-ară care e o mulțime de n-tupluri (din produsul cartezian a n mulțimi).

O relație e o noțiune *mai generală* decât o funcție:

- $f : A \to B$ asociază *un singur* element $y \in B$ *fiecărui* $x \in A$
- o relație $R \subseteq A \times B$ poate conține
  - mai multe perechi (x, y_1), (x, y_2), ... cu același x
  - sau nicio pereche, pentru anumite valori $x \in A$

În general, o relație nu e o noțiune simetrică: perechea (și produsul cartezian) sunt noțiuni ordonate, $(x, y) \neq (y, x)$.
Există, desigur, relații simetrice (în lumea reală și în matematică)

Reprezentarea unei relații

Explicit, prin mulțimea perechilor
dacă e finită
sau exprimabilă printr-o regulă care leagă elementele:

$$R = \{(x, x^2 + 1) \mid x \in \mathbb{Z}\}$$

În formă de matrice booleană / binară, pentru A, B finite, cu liniile indexate după A, și coloanele după B
$m_{xy} = 1$ dacă $(x, y) \in R$, $m_{xy} = 0$ dacă $(x, y) \notin R$
reprezentare practică dacă A și B nu sunt foarte mari

Printr-o funcție de la A la $\mathcal{P}(B)$

Relația văzută ca funcție

O *relație* $R \subseteq A \times B$ poate fi văzută ca o *funcție* $f_R : A \to \mathcal{P}(B)$ de la A la mulțimea părților lui B:

$$f_R(x) = \{y \in B \mid (x, y) \in R\}$$

Asociază fiecărui x mulțimea (posibil vidă) a tuturor elementelor lui B cu care x e în relație.

Funcții parțiale

Dacă relaxăm condiția ca o funcție să asocieze o valoare *fiecărui* element, obținem o *funcție parțială* $f : A \nrightarrow B$.

Ele sunt utile când domeniul exact al funcției nu e cunoscut (funcții care nu sunt neapărat calculabile în orice punct).

în conjectura $3 \cdot n + 1$, numărul de pași pentru a ajunge la 1

În practică, când domeniul de definiție al funcției e foarte mare sau nelimitat, dar dorim să lucrăm totuși cu o reprezentare finită pentru niște valori de interes

Funcții parțiale în ML

În ML, putem reprezenta funcții parțiale folosind modulul `Map`. parametrizat cu un modul: tipul (ordonat) de definiție al mapării

```
module M = Map.Make(String)
```

creează un modul care poate reprezenta corespondențe între șiruri de caractere și un tip încă neprecizat

Când definim o valoare, adăugând o pereche în mapare, se instanțiază și tipul destinație:

```
let m = M.add "x" 5 M.empty
```

adaugă o mapare de la șirul ”x” la 5 la dicționarul vid
acum, `m` e un *dicționar* de la șiruri la întregi

Căutăm o corespondență în dicționar:
`M.find "x" m` returnează întregul 5
`M.find "y" m` returnează *excepția* `Not_found`

Lucrul cu excepții

O *excepție* este o eroare care întrerupe calculul
dacă nu e *tratată*, se încheie execuția programului
altfel, în tratarea excepției se stabilește ce se face

Sintaxa:

`try` *expresie*
`with` *tipar*
unde *tipar* tratează una sau mai multe excepții și are forma
| *excepție-1* -> *expresie-exceptionala-1*
| *excepție-2* -> *expresie-exceptionala-2*
...

Dacă *expresie* se evaluează normal, ea dă rezultatul;
altfel, dacă apare *expresie-k* se evaluează ramura respectivă

```
fun x m -> try M.find x m
          with Not_found -> 0
```

Excepțiile se propagă, terminând fiecare funcție, până când sunt "prinse" de un bloc de tratare.

Relații cu ajutorul dicționarelor

Am văzut că o *relație* $R \subseteq A \times B$ poate fi privită ca o *funcție* $f_R : A \to \mathcal{P}(B)$ de la A la mulțimea părților lui B:

$$f_R(x) = \{y \in B \mid (x, y) \in R\}$$

Dicționarul va fi atunci de la A la *mulțimi* de elemente din B

```
module M = Map.Make(String) (* dictionar pe siruri *)
module S = Set.Make(String) (* multimea de valori *)

let addpair (x, y) m =
  let oldset = try M.find x m with Not_found -> S.empty
  in M.add x (S.add y oldset) m
(* creeaza dictionar din lista *)
let setmap_of_pairs = List.fold_left addpair M.empty

setmap_of_pairs [("tms", "arad");("tms", "lugoj")];;
```

Proprietăți ale relațiilor

Următoarele proprietăți sunt definite pentru relații binare pe *o* (aceeași) mulțime X: $R \subseteq X \times X$

reflexivă dacă pentru orice $x \in X$ avem $(x, x) \in R$

ireflexivă dacă pentru orice $x \in X$ avem $(x, x) \notin R$

simetrică dacă pentru orice $x, y \in X$,
dacă $(x, y) \in R$ atunci și $(y, x) \in R$

antisimetrică dacă pentru orice $x, y \in X$,
dacă $(x, y) \in R$ și $(y, x) \in R$, atunci $x = y$

tranzitivă, dacă pentru orice $x, y, z \in X$,
dacă $(x, y) \in R$ și $(y, z) \in R$, atunci $(x, z) \in R$

Relații de echivalență

O relație e *de echivalență* dacă e *reflexivă*, *simetrică* și *tranzitivă*

Relația de egalitate e (evident) o relație de echivalență.

Relația de congruență modulo un număr:
$a \equiv b \pmod{n}$ dacă $n \mid a - b$ (divide diferența)

O relație de echivalență pe X definește o *partiție* a lui X
în *clase de echivalență*

$$\hat{x} = \{y \mid (x, y) \in R\}$$

(clasa de echivalență a lui x e mulțimea elementelor aflate în relație cu x)

Relații de ordine (parțială)

O relație e o *ordine parțială* (non-strictă), dacă e *reflexivă*, *antisimetrică* și *tranzitivă*

Exemple:

relația $\leq$ pe (orice fel de) numere (la fel $\geq$)
relația de divizibilitate între întregi
relația de incluziune $\subseteq$ pe mulțimea părților

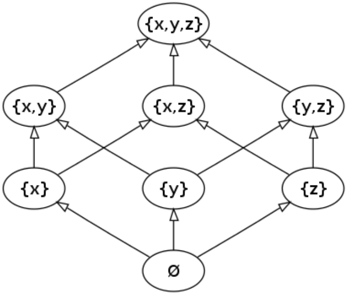
O relație e o *ordine strictă* dacă e *ireflexivă* și *tranzitivă*

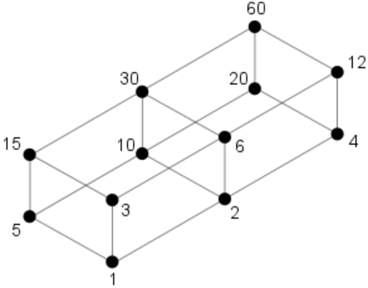
Orice ordine parțială induce o ordine strictă, și reciproc.

Definim: $a \prec b$ dacă $a \preceq b$ și $a \neq b$
Definim: $a \preceq b$ dacă $a \prec b$ sau $a = b$

Latice

O *latice* e o mulțime *parțial ordonată*, în care orice două elemente au un *minorant* și un *majorant*.
(elemente mai mici, respectiv mai mari în ordine decât cele două).
Ex: mulțimea părților unei mulțimi (intersecție, reuniune)
Ex: mulțimea divizorilor unui număr (c.m.m.d.c, c.m.m.m.c)





Imagine: `http://en.wikipedia.org/wiki/File:Hasse_diagram_of_powerset_of_3.svg`

`http://en.wikipedia.org/wiki/File:Lattice_of_the_divisibility_of_60.svg`

Noțiunea de punct fix

$x \in X$ e *punct fix* pentru funcția $f : X \to X$ dacă $f(x) = x$.
(privind f ca o transformare, ea nu îl modifică pe x)

Importanța: putem defini multe prelucrări (recursive/ciclice) ca transformări care se opresc când atingem un *punct fix*

O latice L e *completă* dacă *orice* mulțime $S \subseteq L$ are un cel mai mic majorant (supremum) și un cel mai mare minorant (infimum).

$\Rightarrow$ Luând $S = L$, rezultă că L are un element minim și unul maxim

Teorema de punct fix

Teoremă (Knaster-Tarski):

Fie f o funcție monotonă pe o latice completă.
Atunci mulțimea punctelor fixe a lui f e tot o latice completă.

Corolar: O funcție monotonă pe o latice completă are un cel mai mic punct fix și un cel mai mare punct fix.

se obțin pornind de la 0, f(0), f(f(0)), ... resp. M, f(M), f(f(M)), ...
unde 0 și M sunt elementul cel mai mic respectiv cel mai mare

Punctul fix în ML

Presupunând că există limita șirului x, f(x), f(f(x)) ...
putem scrie simplu

```
let rec fix f x =
  let xp = f x in
  if xp = x then x else fix f xp
```

sau, folosind o funcție interioară cu un singur parametru:

```
let fix f x =
  let rec fixf x =
    let xp = f x in
    if xp = x then x else fixf xp
in fixf
```

Inversa și compunerea de relații

Inversa unei relații $R \subseteq A \times B$ e relația $R^{-1} \subseteq B \times A$, cu $(y, x) \in R^{-1}$ dacă și numai dacă $(x, y) \in R$

$$R^{-1} = \{(y, x) \mid (x, y) \in R\}$$

Fie două relații $R_1 \subseteq A \times B$ și $R_2 \subseteq B \times C$.
Atunci compunerea $R_2 \circ R_1 \subseteq A \times C$ e relația

$$R_2 \circ R_1 = \{(x, z) \mid \text{ există } y \in B \mid (x, y) \in R_1 \text{ și } (y, z) \in R_2\}$$

La fel ca la funcții, scriem $R_2 \circ R_1$ și vedem că pentru $x \in A$ găsim întâi $y \in B$ și apoi $z \in C$.

Se poate vedea simplu că $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$

Pentru relație e de echivalență, $R = R^{-1}$

Dacă R e tranzitivă, $R \circ R \subseteq R$.

Închiderea tranzitivă a unei relații

În general, dacă compunem $R \circ R \ldots \circ R$ de tot mai multe ori, ne așteptăm să obținem tot mai multe perechi în relație.

Închiderea tranzitivă a unei relații $R \subseteq X \times X$ e relația *tranzitivă minimală* R^+ astfel ca $R \subseteq R^+$

Putem calcula R^+ ca fiind $\bigcup_{k=1}^{\infty} R^k = R \cup R^2 \cup \ldots$

Sau, putem defini $f(X) = R \cup (X \circ R)$, atunci $f^n(R) = \bigcup_{k=1}^{n+1} R^k$.

Dar f e monotonă, deci are un punct fix minimal, care e chiar R^+ iar dacă X e finită, ajungem la limită într-un număr finit de pași