

Interpretare abstractă

11 mai 2004

Analiza programelor. Curs 3

Marius Minea

Interpretare abstractă: Introducere

O metodă pentru definirea unei *semantici abstracte* a unui program, care poate fi utilizată pentru a analiza programul și a produce informații despre comportamentul său în execuție. [Cousot & Cousot '77]

Inspirată din:

- analiza fluxului de date
 - systematizează noțiunile de proprietăți analizate și caracteristicile fundamentale ale metodelor de analiză
- semantică denotațională
 - formalizează noțiunea de *aproximare* a semanticii unui program (corectitudinea e nedecidabilă \Rightarrow avem nevoie de aproximare)

Interpretare abstractă

3

Interpretare abstractă

4

Exemplu simplu: abstracția semn

Considerăm un limbaj care conține doar întregi și înmulțire: $e ::= i \mid e * e$
Definim semantica printr-o funcție care dă valoarea unei expresii:

$\mu : Exp \rightarrow Int, \mu(i) = i, \mu(e_1 * e_2) = \mu(e_1) * \mu(e_2)$

Definim o funcție de abstracție $\alpha : Exp \rightarrow \{-, 0, +\}$:

Întâi pentru întregi: $\alpha(i) = \begin{cases} - & \text{dacă } i < 0 \\ 0 & \text{dacă } i = 0 \\ + & \text{dacă } i > 0 \end{cases}$

și pentru expresii prin tabelul:

| | | | |
|---|---|---|---|
| - | - | 0 | + |
| 0 | + | 0 | - |
| + | 0 | 0 | 0 |
| + | - | 0 | + |

La fel se poate introduce operatorul minus unar:

| | | | |
|---|---|---|---|
| - | - | 0 | + |
| - | + | 0 | - |

Analiza programelor. Curs 3

Marius Minea

Abstracția semn: continuare

Introducem operația de adunare \Rightarrow abstracția nu mai e precisă

| | | | |
|---|---|---|---|
| + | - | 0 | + |
| - | - | - | + |
| 0 | - | 0 | + |
| + | + | + | + |

Apar valori care nu pot fi determinate precis (se pierde precizia)

\Rightarrow e necesară introducerea lui $T = \{-, 0, +\}$

\Rightarrow apare din nou noțiunea de *lattice*

Interpretare abstractă

5

Interpretare abstractă

6

Abstracția semn: continuare

Similar, la introducerea operatorului de împărțire $/$:
nu putem împărți la zero.

| | | | | |
|---|---|---|---|---|
| / | - | 0 | + | T |
| - | + | ⊥ | - | T |
| 0 | 0 | ⊥ | 0 | 0 |
| + | - | ⊥ | + | T |
| T | T | ⊥ | T | T |

și extindem toate operațiile și la \perp : $\perp \text{ op } x = x \text{ op } \perp = \perp$

Analiza programelor. Curs 3

Marius Minea

Interpretarea abstractă: considerații fundamentale

Semantica unui program: un model matematic (formal) al tuturor comportamentelor posibile ale unui sistem de calcul care execută acel program, în interacțiune cu orice mediu posibil [Cousot]

Semantica unui limbaj de programare: definește semantica oricărui program

Abordare generală: semantica unui program poate fi definită ca soluție a unei *ecuații de punct fix* (corespunzătoare iterației).

\Rightarrow toate semantica unui program pot fi organizate într-o ierarhie, după nivelul lor de *abstracție*.

Analiza programelor. Curs 3

Marius Minea

Exemple de abstracții

- secvențe de execuție (finite sau infinite): descriu în fiecare punct valoarea variabilelor din program
- semantică operațională: descrie comportamentul la nivel de stări și tranziții (ca și automat)
- semantică denotațională: descrie rezultatul execuției (incl. neterminare)
- semantică naturală: ca mai sus, ignoră aspectul neterminării

Analiza programelor. Curs 3

Marius Minea

Abstractizare și concretizare

Constă în:

- un domeniu concret D și un domeniu abstract A , legate printr-o *conexiune Galois*:
- o funcție de abstracție $\alpha : D \rightarrow A$
- o funcție de concretizare $\gamma : A \rightarrow \mathcal{P}(D)$ (asociază fiecărei valori abstracte o mulțime de valori concrete)
- a.î. $\forall x \in \mathcal{P}(D), x \subseteq \gamma(\alpha(x))$ și $\forall a \in A, a = \alpha(\gamma(a))$
- (abstracția urmată de concretizare introduce aproximare; concretizarea urmată de abstracție e exactă)

Aceasta e proprietatea fundamentală care exprimă corectitudinea abstracției (*soundness*): ea poate genera valori suplimentare, dar niciodată nu va omite valori \Rightarrow e o abstracție conservatoare.

Analiza programelor. Curs 3

Marius Minea

Abstracții de punct fix

Semantica unui program e dată în general de o ecuație de punct fix (datorată ciclurilor din program)

- am văzut deja ecuații de punct fix pentru analiza de flux de date

Avem:

- punctul fix al funcției f în domeniul concret
- punctul fix al funcției $f^\#$ în domeniul abstract
- concretizarea punctului fix al lui $f^\#$

Atunci, $\text{lfp } f \sqsubseteq \gamma(\text{lfp } f^\#)$

Analiza programelor. Curs 3

Marius Minea

Exemple efective (numerice) de abstracții

- abstracția semn
- abstracția pe intervale
- abstracția poliedrală (înfășurătoarea convexă a valorilor)
- abstracția octogonală (ecuații de forma $\pm x \leq c, \pm y \leq c, \pm x \pm y \leq c$)
- abstracția modulo un număr, punctual sau pe intervale
- abstracții nerelaționale (ex. abstracția carteziană): calculează abstracția individual pentru fiecare variabilă, neținând cont de corelare
- abstracții relaționale: păstrează corelarea între variabile

Analiza programelor. Curs 3

Marius Minea

Abstracții ale funcțiilor

Fiind dată o funcție (transformare) în programul $f : D \rightarrow D$ în programul concret, ce corespondent are aceasta în programul abstract ?

Răspuns: $f^\# = \alpha \circ f \circ \gamma$ Pentru un element abstract $x \in A$:

- producem întâi mulțimea de valori concrete $\gamma(x)$
- aplicăm funcția concretă f fiecărei valori, obținând o mulțime de valori (concrete)
- abstracționăm mulțimea de valori concrete într-o valoare abstractă (cu obișnuita posibilă pierdere de precizie)

Analiza programelor. Curs 3

Marius Minea

Probleme de punct fix și terminare

Problemă: dacă lanțul ascendent în calculul de punct fix are lungime infinită (lattice e de înălțime infinită în raport cu funcția/transformarea dată), nu se poate determina un punct fix într-un număr finit de pași

```
while (x >= 0)
  x = x + 1;
```

Ex. analiza codului de mai sus chiar cu intervale.

Pe de altă parte, "ghicind" intervalul $[0, \infty)$ e clar că acesta e punct fix (și punctul fix minimal).

Analiza programelor. Curs 3

Marius Minea

Lărgire și restrângere (widening și narrowing)

Permite eliminarea lanțurilor de lungime infinită la calculul de punct fix, în doi pași:

- la început, o aproximare mai grosieră a lanțului ascendent, care conduce la convergență mai rapidă (în număr finit de pași)
- apoi, o revenire la precizie mai bună, printr-un lanț de aproximări descendente, spre punctul fix propriu-zis