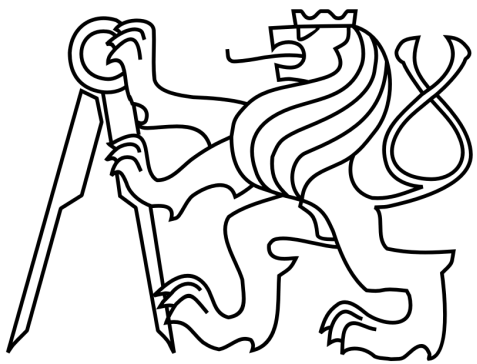
Formální Metody a Specifikace (LS 2011)
Přednáška 5:
Operační sémantika programů

Stefan Ratschan

Katedra číslicového návrhu
Fakulta informačních technologií
Ceské vysoké učení technické v Praze

18. březen 2011



Sémantika programů

- Vstup: $a \in \mathcal{A}_n$, $k \in \mathbb{N}$
- Výstup:
  - **T** pokud existuje $i \in \{1, \ldots, n\}$ tak, že $a[i] = k$,
  - **F**, jinak.

```
1: i ← 1
2: if i ≤ n then
3:     if a[i] = k then
4:         return T
5:     i ← i + 1
6:     goto 2
7: return F
```

Správnost:

$$\models T \Rightarrow \left[\forall x, x' \,.\, [I(x) \wedge x' = f(x)] \Rightarrow O(x, x')\right]$$

Jak můžeme najít takovou funkci f popisující chování programu?

To je: f pak popisuje význam/sémantiku programu.

Stav programů

Čítač programu + hodnoty všech proměnných

Předpokládáme program který má

- l řádků, a
- proměnné ve množině V tak, že každá proměnná v má typ T_v.

Stav je valuací která

- přiřazuje speciální proměnné pc prvek množiny $\{1, \ldots, l\}$, a
- každé proměnné $v \in V$ prvek množiny T_v.

Množina všech stavů S.

Vývoj stavu

Každý krok v programu
  vezme určitý stav $s \in S$ a
    počítá další stav $s' \in S$.

Tím pádem píšeme $s \to s'$.

Relaci $\to \subseteq S \times S$ nazýváme *přechodovou relací* (transition relation)

Proč nepoužíváme funkci místo relace?

Nedeterminismus!
  Výsledek toho že něco nevíme, nebo nechceme modelovat

Např.:

- Nevíme rychlost jednotlivých vláken, vstup uživatelů, výsledky čidel
- Nechceme modelovat algoritmus generátoru náhodných čísel, zaokrouhlení čísel s pohyblivou čárkou atd.

Přechodová relace: Přiřazení

Pokud $s(pc)$ odkazuje na řádek s přiřazením $v \leftarrow t$:

$s \rightarrow s'$ přesně když

$$\mathcal{I}, s \circ \pi(s') \models pc' = pc + 1 \wedge v' = t \wedge \bigwedge_{u \in V, u \neq v} u' = u$$

přičemž

- pro valuaci s, $\pi(s)$ je valuací která přiřazuje stejné hodnoty do čárkovaných proměnných,
- pro valuace s, s' s disjunktními množinami proměnných, $s \circ s'$ je valuací tak, že

$$(s \circ s')(v) = \begin{cases} \text{s(v), pokud } v \text{ je proměnou valuace s, } a \\ \text{s'(v), pokud } v \text{ je proměnou valuace s'.} \end{cases}$$

- $\mathcal{I}$ je interpretací, která dává všem operacím v termu t příslušný význam($a[i] \leftarrow x$: zkrátka pro $a \leftarrow write(a, x, i)$).

Příklad

$$s = \{pc \mapsto 5, a \mapsto [4,5,6,7,8], k \mapsto 7, i \mapsto 2\}$$

$$s' = \{pc \mapsto 6, a \mapsto [4,5,6,7,8], k \mapsto 7, i \mapsto 4\}$$

$$\pi(s') = \{pc' \mapsto 6, a' \mapsto [4,5,6,7,8], k' \mapsto 7, i' \mapsto 4\}$$

$$s \circ \pi(s') = \{pc \mapsto 5, pc' \mapsto 6, a \mapsto [4,5,6,7,8], a' \mapsto [4,5,6,7,8], k \mapsto 7, k' \mapsto 7, i \mapsto 2, i' \mapsto 4\}$$

$$s \circ \pi(s') \not\models pc' = pc + 1 \wedge i = i + 1 \wedge a' = a \wedge k' = k$$

Ale pro

$$s' = \{pc \mapsto 6, a \mapsto [4,5,6,7,8], k \mapsto 7, i \mapsto 3\}$$

$$s \circ \pi(s') \models pc' = pc + 1 \wedge i = i + 1 \wedge a' = a \wedge k' = k$$

Přechodová relace: Řídicí struktury

- Pokud $s(pc)$ odkazuje na řádek **goto** r:

  $s \to s'$ přesně když

  $$\mathcal{I}, s \circ \pi(s') \models pc' = r \wedge \bigwedge_{u \in V} u' = u$$

- Pokud $s(pc)$ odkazuje na řádek **if** P **then**:

  $s \to s'$ přesně když

  $$\mathcal{I}, s \circ \pi(s') \models [P \Rightarrow pc' = pc + 1] \wedge [\neg P \Rightarrow pc' = l] \wedge \bigwedge_{u \in V} u' = u$$

  přičemž l je číslem řádku po konci **if-then** bloku.

- Další řídicí struktury: kombinace

Přechodová relace: Vedlejší účinky

Pokud $s(pc)$ odkazuje na řádek **input** v:

$s \to s'$ přesně když

$$\mathcal{I}, s \circ \pi(s') \models pc' = pc + 1 \wedge \bigwedge_{u \in V, u \neq v} u' = u$$

pokud $u = v$?

Nedeterminismus

output v?

Přechodová relace: souhrn

$s \to s'$ přesně když

$$\mathcal{I}, s \circ \pi(s') \models \Phi_P$$

přičemž Φ_P (*přechodová podmínka*) je formule tvaru

$$\bigwedge_{i \in \{1,\ldots,l\}} pc = i \Rightarrow \Phi_{P,i}$$

přičemž $\Phi_{P,i}$ je formule odpovídající řádku i v programu P.

Tudíž: máme formuli predikátové logiky prvního řádu
které popisuje jednotlivé kroky programu.

Celkový programový průběh

Program může dělat libovolný počet kroků podle $\rightarrow$:

$r \rightarrow^* r'$ přesně když

existuje posloupnost $s_1, \ldots, s_n$ tak, že $r = s_1 \rightarrow \cdots \rightarrow s_n = r'$

Pro libovolnou relaci $\rightarrow$,

$\rightarrow^*$ nazýváme *tranzitivním uzávěrem* relace $\rightarrow$

Pro jednoduchost předpokládáme že vstup a výstup je celý stav

(obvykle pro vstupní stav s, $s(pc) = 1$)

(Operační) sémantika programu:

Funkce která pro s (vstupní stav programu)

počítá s' (výstupní stav programu) tak, že

- $s'(pc)$ obsahuje **return**,
- $s \rightarrow^* s'$.

Proč operační? varianta na základě podmínek (constraint-based)

Použití sémantiky

Pro libovolný program P,
píšeme $[\![P]\!]$ pro příslušnou sémantiku.

Pozor: Nedeterminismus: definice $[\![P]\!]$ není jednoznačná!

Program který

- pro určitý program P, vstupní stav s
- počítá $[\![P]\!](s)$

se jmenuje interpret.

Program který

- pro určitý program P,
- počítá program Q tak, že pro vstupní stav s, $Q(s) = [\![P]\!](s)$

se jmenuje překladač.

Použití sémantiky

Definicí přechodové relace jsme přesně specifikovali,
jak se interpret, překladač má chovat.

Pro určité programové jazyky to takto funguje.

Bohužel jsou i jiné programové jazyky kde

- existuje jen neformální popis chování, a
- jediná přesná definice chování je sám interpret/překladač:

  $[\![P]\!](s) := s'$ tak,
  že s' je výstupní stav programu P pro vstupní stav s.