ZÁKLADY MATEMATICKEJ TEÓRIE PROGRAMOV

Časť 3.1: Sémantika programov – základné pojmy

Igor Prívara
Inštitút informatiky a štatistiky

FMFI UK Bratislava, Katedra informatiky

Základy teórie programovania

Programové schémy – abstrakcia programov

- základné pojmy, vlastnosti programových schém
- rozhodnuteľnosť vlastností programových schém
- porovnávanie tried programových schém

Správnosť programov – vzhľadom na špecifikácie

- Floydova metóda a Hoareova metóda
- indukčné techniky použité pri dokazovaní správnosti
- (systematická) konštrukcia správnych programov
- dokazovanie správnosti rekurzívnych programov

Sémantika programov – formálny význam programov

- **princípy operačnej, denotačnej a axiomatickej sémantiky**
- algebraická štruktúra sémantických domén
- formálna definícia denotačného a operačného významu imperatívnych a rekurzívnych programov
- porovnanie denotačného a operačného významu programov

Typy a sémantika – využitie typov pri definícii sémantiky

Formálna sémantika

programovacích jazykov

Neformálny vs. formálny popis sémantiky –

- neformálny prístup – význam programovacích pojmov vysvetľuje v prirodzenom jazyku
- formálny prístup – význam pojmov vyjadruje (definuje) vo formálnom "matematickom" jazyku

Formálny popis významu programov – umožňuje

- definovať význam programu počítačovo nezávislými pojmami
- jednoznačne popísať význam programu
- vytvoriť teórie, podporujúce dokazovanie vlastností programov
- konzistentne definovať význam typovaných aj beztypových jazykov

Rozdiely medzi jazykami logiky a programovania –

- rôzny pohľad na premenné (statický vs. dynamický)
- zložitejšie obory interpretácie resp. sémantické obory
- význam nekonečných objektov (aproximácie)
- program môže "počítať" čiastočnú funkciu (nekonečný výpočet)

Sémantika programov

Syntaktický obor $\mathcal{P}$ – množina správne vytvorených programov v danom jazyku

Sémantický obor $\mathcal{M}$ – množina významov

Sémantika $\mathcal{S}$ – sémantická funkcia

$$\mathcal{S} : \quad \mathcal{P} \quad \mapsto \quad \mathcal{M}$$

- $\mathcal{S}[\![P]\!]$ – denotuje význam programu P vzhľadom na sémantiku $\mathcal{S}$
- sémantika danej triedy programov $\mathcal{P}$ sa dá vyjadriť rôznym spôsobom; rôzne sémantiky sa odlišujú výberom sémantického oboru $\mathcal{M}$

Ekvivalencia programov –
programy P_1, P_2 sú ekvivalentné vzhľadom na sémantiku $\mathcal{S}$ práve vtedy, keď platí $\mathcal{S}[\![P_1]\!] = \mathcal{S}[\![P_2]\!]$

- Ekvivalencia – sémantická vlastnosť
- Rôzne sémantické charakterizácie vedú k rôznym ekvivalenciám!

Porovnanie sémantických definícií

Definícia: sémantika $\mathcal{S}_2 : \mathcal{P} \mapsto \mathcal{M}_2$ je **definovateľná** zo sémantiky $\mathcal{S}_1 : \mathcal{P} \mapsto \mathcal{M}_1$, ak existuje funkcia $\mathcal{F} : \mathcal{M}_1 \mapsto \mathcal{M}_2$ taká, že pre každý program $P \in \mathcal{P}$ platí

$$\mathcal{S}_2[\![P]\!] = \mathcal{F}(\mathcal{S}_1[\![P]\!])$$

- sémantiky $\mathcal{S}_1, \mathcal{S}_2$ sú ekvivalentné ak sú navzájom definovateľné

Definícia: sémantika $\mathcal{S}_1 : \mathcal{P} \mapsto \mathcal{M}_1$ je **podrobnejšia** ako sémantika $\mathcal{S}_2 : \mathcal{P} \mapsto \mathcal{M}_2$ práve vtedy, keď pre ľubovoľné dva programy $P_1, P_2 \in \mathcal{P}$ platí

$$\mathcal{S}_1[\![P_1]\!] = \mathcal{S}_1[\![P_2]\!] \implies \mathcal{S}_2[\![P_1]\!] = \mathcal{S}_2[\![P_2]\!]$$

- $\mathcal{S}_2$ je abstraktnejšia

Lema: Nech $\mathcal{S}_1, \mathcal{S}_2$ sú sémantiky nad syntaktickým oborom $\mathcal{P}$. Potom $\mathcal{S}_2$ je definovateľná z $\mathcal{S}_1$ práve vtedy, keď $\mathcal{S}_1$ je podrobnejšia ako $\mathcal{S}_2$.

Dôkaz:

1. $\mathcal{S}_2$ je definovateľná z $\mathcal{S}_1$: priamo z definície
2. $\mathcal{S}_1$ je podrobnejšia ako $\mathcal{S}_2$:
   - $\mathcal{P}/\mathcal{S}_1$ – rozklad $\mathcal{P}$ vzhľadom na $\mathcal{S}_1$
   - $F_2 : \mathcal{P}/\mathcal{S}_1 \mapsto \mathcal{M}_2$ – predpis $X \mapsto \mathcal{S}_2[\![P]\!]$ pre $P \in X$
   - $F_1 : \mathcal{M}_1 \mapsto \mathcal{P}/\mathcal{S}_1$ – predpis $m \mapsto \{P_1 \in \mathcal{P} : \mathcal{S}_1[\![P_1]\!] = m\}$
   - $\mathcal{F}(\mathcal{S}_1[\![P]\!]) = F_2(F_1(\mathcal{S}_1[\![P]\!])) = \mathcal{S}_2[\![P]\!]$

Kompozičná sémantika

ak program P pozostáva zo zložiek $P_1, \cdots, P_n$ potom jeho význam $\mathcal{S}[\![P]\!]$ je funkciou významov $\mathcal{S}[\![P_1]\!], \cdots, \mathcal{S}[\![P_n]\!]$ jednotlivých zložiek

Induktívna definícia syntaxe jazyka – abstraktná syntax

- elementárne syntaktické objekty (napr. priraďovací príkaz)
- množina **syntaktických konštruktorov** $K_i : \mathcal{P}^n \mapsto \mathcal{P}$

  while_do_od: (BOOL STAT) STAT
  if_then_else_fi: (BOOL STAT STAT) STAT
  _;_: (STAT STAT) STAT
- ľubovolný "zložený" program P v danom jazyku má tvar $P = K(P_1, \cdots, P_n)$

Induktívna definícia sémantiky jazyka –

- charakterizuje sa význam elementárnych syntaktických objektov (priraďovací príkaz)
- ku každému syntaktickému konštruktoru K_i sa definuje zodpovedajúci **sémantický konštruktor** $\mathcal{K}_i : \mathcal{M}^n \mapsto \mathcal{M}$
- sémantika každého zloženého príkazu $P = K(P_1, \cdots, P_n)$ sa definuje **sémantickým pravidlom**

$$\mathcal{S}[\![K(P_1, \cdots, P_n)]\!] = \mathcal{K}(\mathcal{S}[\![P_1]\!], \cdots, \mathcal{S}[\![P_n]\!])$$

Základné prístupy pri popise významu programov

závisia od výberu sémantického oboru $\mathcal{M}$:

- operačná (výpočtová) sémantika,
- denotačná (matematická) sémantika,
- deduktívna (axiomatická) sémantika.

$$\mathcal{S} : \quad \mathcal{P} \quad \mapsto \quad \mathcal{M}$$

Rôzne prístupy odrážajú rôznu úroveň abstrakcie

Technická lema:

$$D_1 \times D_2 \mapsto D_3 \cong D_1 \mapsto (D_2 \mapsto D_3)$$

Intuícia – zafixovaním jedného parametra binárnej funkcie dostaneme unárnu funkciu.

Operačná sémantika

Abstraktný počítač – je definovaný napr.

- množinou stavov (konfigurácií) $S = \{s, s_i, \cdots\}$
- množinou elementárnych inštrukcií $I = \{i, i_k, \cdots\}$
- prechodovou funkciou $q : I \times S \mapsto S$

Sémantický obor – množina výpočtových postupností S^∞,
t.j. postupností stavov (konfigurácií) $s_0, s_1, \cdots, s_n, \cdots$ generovaných abstraktným počítačom v priebehu výpočtu začínajúcom vstupným stavom (konfiguráciou) $s_0 \in S$

Význam programu –
zobrazenie vstupného stavu do zodpovedajúcej výpočtovej postupnosti

Sémantická funkcia –

$$\mathcal{O}(P, s_0) = s_0, s_1, \cdots, s_n(\cdots)$$

$$\mathcal{O} : \mathcal{P} \times S \mapsto S^\infty$$

$$\mathcal{O} : \mathcal{P} \mapsto (S \mapsto S^\infty)$$

Sémantický popis jazyka –

- transformácie stavu, realizované elementárnymi príkazmi jazyka
- pravidlá, podľa ktorých sa z postupnosti stavov (konfigurácií), zodpovedajúcich zložkám príkazov jazyka, vytvoria postupnosti stavov (konfigurácií), zodpovedajúce príkazom jazyka

Denotačná sémantika

Sémantický obor –
priestor funkcií resp. relácií nad množinou stavov

Význam programu –
vstupno–výstupný vzťah vyjadrený funkciou resp. reláciou

Sémantická funkcia –

$$\mathcal{M}(P, s_0) = s_n \qquad \text{(alebo } \perp\text{)}$$

$$\mathcal{M} : \mathcal{P} \times S \mapsto S$$

$$\mathcal{M} : \mathcal{P} \mapsto (S \mapsto S)$$

Sémantický popis jazyka –

- elementárne príkazy vyjadríme ako funkciu (reláciu) definovanú nad oborom stavov
- pre každý syntaktický konštruktor ukážeme, ako sa z funkcií (relácií), zodpovedajúcich zložkám príkazu skonštruuje funkcia (relácia) vyjadrujúca význam príkazu

Intuitívna vlastnosť: $\mathcal{M}$ je definovateľná z $\mathcal{O}$

Deduktívna sémantika

Špecifikačný jazyk –

- jazyk $L = \{p, q, \cdots\}$ na popis vstupno–výstupných podmienok
- najslabšia vstupná podmienka $wp(P, q)$ k výstupnej podmienke q a programu P s vlastnosťou: ak je splnená $wp(P, q)$ a program P sa skončí, musí byť splnená výstupná podmienka q

Sémantický obor – priestor transformátorov predikátov $L \mapsto L$

Význam programu – transformátor predikátov, definujúci napr. $wp(P, q)$ k danej výstupnej podmienke q

Sémantická funkcia –

$$\mathcal{D}(P, p) = q \qquad \text{(napr. } q = wp(P, p))$$

$$\mathcal{D} : \mathcal{P} \times L \mapsto L$$

$$\mathcal{D} : \mathcal{P} \mapsto (L \mapsto L)$$

Sémantický popis jazyka –

- elementárne príkazy sa charakterizujú transformátormi predikátov
- pre každý syntaktický konštruktor ukážeme, ako sa z transformátorov predikátov, zodpovedajúcich zložkám príkazu skonštruuje transformátor predikátov vyjadrujúci význam príkazu

Intuitívna vlastnosť: $\mathcal{D}$ je definovateľná z $\mathcal{M}$

Ekvivalencia pri rôznych sémantikách

Operačná ekvivalencia – $\mathcal{O}[\![P_1]\!] = \mathcal{O}[\![P_2]\!]$

pre každý vstup realizujú programy P_1, P_2 na abstraktnom počítači rovnaké výpočtové postupnosti

Denotačná ekvivalencia – $\mathcal{M}[\![P_1]\!] = \mathcal{M}[\![P_2]\!]$

programy P_1, P_2 realizujú ten istý vstupno–výstupný vzťah (funkciu, reláciu)

Deduktívna "ekvivalencia" – $\mathcal{D}[\![P_1]\!] = \mathcal{D}[\![P_2]\!]$

programy P_1, P_2 definujú ekvivalentné transformátory predikátov, ktoré zobrazujú ľubovolnú výstupnú podmienku do ekvivalentných najslabších vstupných podmienok