Programmeerimiskeelte semantika

Varmo Vene
Arvutiteduse Instituut
Tartu Ülikool

EMAIL: varmo@cs.ut.ee
WWW: http://www.cs.ut.ee/~varmo/SEM00/
LIST: ati.funprog@lists.ut.ee

Kursuse plaan

- Operatsiooniline semantika
  - Loomulik semantika
  - Struktuurne semantika
  - Realisatsiooni korrektsus
- Denotatiivne semantika
  - Doomenite teooria
  - “Otsene” semantika
  - Programmide staatiline analüüs
- Aksiomaatiline semantika
  - Programmide verifitseerimine

Kirjandus

1. H.R.Nielson, F.Nielson. *Semantics with Applications: A Formal Introduction*. Wiley, 1992.
2. G.Winskel. *The Formal Semantics of Programming Languages: An Introduction*. MIT, 1993.
3. D.A.Schmidt. *The Structure of Typed Programming Languages*. MIT, 1994.

Programmeerimiskeeled

Notatsioon arvutile käskude esitamiseks (näit. Pascal, mingi rakendusprogrammi sisendkäsud, . . .)

- **Süntaks**
  - keele lausete "väljanägemine" ja struktuur
- **Semantika**
  - keele lausete tähendus
- **Pragmaatika**
  - keele kasutatavus (näit. rakendusvaldkonnad, efektiivsus, . . .)

Keele realisatsioon

Koosneb kahest peamisest osast:

- **Parser**
  - Kontrollib programmi vastavust süntaksile, teisendab programmi sisekujule
- **“Väärtustaja”**
  - Väärtustab sisendprogrammi sellele vastavaks väljundiks (so. annab programmile tähenduse — semantika)

Realisatsioon kuulub keele pragmaatika valdkonda.

Väärtustamine

- **Interpreteerimine**
  - Programmi täitmine
  - Interpretaator *defineerib* programmi tähenduse
- **Kompileerimine**
  - Programmi teisendamine ekvivalentseks programmiks mingis teises keeles (näit. masinkoodis)
  - Kompilaator *säilitab* programmi tähenduse

Formaalsed spetsifikatsioonid

- **Süntaks:** BNF (Backus-Naur Form)
  - Vastavus BNF'i ja parserite vahel
  - Sisend parserite generaatorile (näit. yacc)
- **Semantika:** ??
  - Täpne standard realiseerimiseks
  - Vahend keele disainiks ja analüüsiks
  - Sisend kompilaatorite generaatorile

Semantikat on palju raskem kirjeldada ("semantika on kõik see, mida ei saa kirjeldada BNF abil").

Operatsiooniline semantika

- Keel on defineeritud interpretaatori (abstraktse masina) abil
- Iga konstruktsioon on defineeritud üleminekureegli abil
- Programmi tähendus on interpretaatori olekute jada

  $$P : S_1 \to S_2 \to \ldots \to S_n,$$

  kus P on programm ja S_i interpretaatori olek

Aksiomaatiline semantika

- Keel on defineeritud loogiliste aksioomide ja tuletusreeglide abil
- Igale konstruktsioonile vastab aksioom
- Ei defineeri mitte programmi tähendust vaid omadusi

  $$\{A\}P\{B\}$$

  kus P on programm ning A ja B vastavalt eel- ja järel tingimused
- Tõestatavad omadused ei tarvitse kirjeldada programmi unikaalselt

Denaotatiivne semantika

- Programmi tähendus on matemaatiline objekt
- Igale konstruktsioonile seatakse väärtustusfunktsiooni abil vastavusse tema tähendus (denotatsioon)

  $$\mathcal{S}[\![P]\!] = D$$

  kus P on programm, $\mathcal{S}$ väärtustusfunktsioon ja D denotatsioon
- Programmi tähendus on selle osade tähenduste kompositsioon
- On abstraktsem kui operatsiooniline semantika (puuduvad arvutussammud), kuid konkreetsem kui aksiomaatiline semantika (tähendus ilmutatud kujul)

Binaararvude semantika

- Süntaktilised kategooriad:

$$n \in \mathbf{Num} \qquad \text{binaararvud}$$

- Abstraktne süntaks:

$$n \ ::= \ 0 \mid 1 \mid n\,0 \mid n\,1$$

- Semantiline funktsioon: $\mathcal{N} : \mathbf{Num} \to \mathbf{Z}$

$$\begin{aligned} \mathcal{N}[\![0]\!] &= 0 \\ \mathcal{N}[\![1]\!] &= 1 \\ \mathcal{N}[\![n\,0]\!] &= 2 \star \mathcal{N}[\![n]\!] \\ \mathcal{N}[\![n\,1]\!] &= 2 \star \mathcal{N}[\![n]\!] + 1 \end{aligned}$$

Avaldiste semantika

- Süntaktilised kategooriad:

$$\begin{array}{ll} n \in \mathbf{Num} & \text{arvud} \\ x \in \mathbf{Var} & \text{muutujad} \\ a \in \mathbf{AExp} & \text{aritmeetilised avaldised} \end{array}$$

- Abstraktne süntaks:

$$a \ ::= \ n \mid x \mid a_1 + a_2 \mid a_1 \star a_2 \mid a_1 - a_2$$

- Muutujate semantika annab keskond (olek)!

$$\mathbf{State} = \mathbf{Var} \to \mathbf{Z}$$

Avaldiste semantika

- Semantiline funktsioon: $\mathcal{A} : \mathbf{AExp} \to (\mathbf{State} \to \mathbf{Z})$

$$
\begin{array}{lcl}
\mathcal{A}[\![n]\!]s & = & \mathcal{N}[\![n]\!] \\
\mathcal{A}[\![x]\!]s & = & s\,x \\
\mathcal{A}[\![a_1 + a_2]\!]s & = & \mathcal{A}[\![a_1]\!]s + \mathcal{A}[\![a_2]\!]s \\
\mathcal{A}[\![a_1 \star a_2]\!]s & = & \mathcal{A}[\![a_1]\!]s \star \mathcal{A}[\![a_2]\!]s \\
\mathcal{A}[\![a_1 - a_2]\!]s & = & \mathcal{A}[\![a_1]\!]s - \mathcal{A}[\![a_2]\!]s
\end{array}
$$

Lihtne imperatiivne keel — While

- Süntaktilised kategooriad

$n \in \mathbf{Num}$	arvud
$x \in \mathbf{Var}$	muutujad
$a \in \mathbf{AExp}$	aritmeetilised avaldised
$b \in \mathbf{BExp}$	tõeväärtusavaldised
$S \in \mathbf{Stm}$	laused

Lihtne imperatiivne keel — While

- Abstraktne süntaks

$$
\begin{array}{rcl}
a & ::= & n \mid x \mid a_1 + a_2 \mid a_1 \star a_2 \mid a_1 - a_2 \\
b & ::= & \texttt{true} \mid \texttt{false} \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \neg(b) \mid b_1 \wedge b_2 \\
S & ::= & x := a \mid \texttt{skip} \mid S_1 ; S_2 \\
 & \mid & \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \mid \texttt{while}\ b\ \texttt{do}\ S
\end{array}
$$

- Näide:

$$y := 1;\ \texttt{while}\ \neg(x = 1)\ \texttt{do}\ (y := y \star x;\ x := x - 1)$$