

Syntaks og semantik

Semantikopgave

Uli Fahrenberg

Foråret 2007

1 Indledning

Kurset ‘Syntaks og semantik’ følges af to opgaver, en syntaks- og en semantikopgave. Formålet med opgaverne er at give jer et par større eksempler på praktiske anvendelser, og at træne jeres evne til at formulere jer på skrift i et teoretisk emne.

Til opgaverne skal nemlig afleveres en skriftlig besvarelse, gruppevist. Deadline for aflevering af denne semantikopgaves besvarelse er

mandag 14/5 kl. 12:00 (dansk sommertid)

Herefter får grupperne hinandens besvarelser udleveret og skal kommentere dem, hvorefter jeg selv vil kommentere besvarelsen og kommentarerne.

Syntaks- og semantikopgavens besvarelser samt kommentarer indgår som en del af pensum til eksamen.

De grupper der har kurset som PE-kursus kan vælge at erstatte semantikopgaven med en tilsvarende opgave der har relation til gruppens projekt. Det skal i så fald aftales med mig.

Til sidst en tak til Hans Hüttel, forfatteren til sidste års semantikopgaver.

2 Udtryksprocedurer (funktioner)

Betragt **Bof**, en variant af sproget **Bip** med *udtryksprocedurer* (‘funktioner’) som i f.eks. C og Pascal i stedet for procedurer. En udtryksprocedure har én call-by-value-parameter. En udtryksprocedure kan kaldes i aritmetiske udtryk, og den værdi den returnerer, er slutværdien af en alias-variabel med samme navn som udtryksproceduren. Man kan altså nu erklære f.eks. udtryksproceduren **f** som angivet herunder:

```
function f (x) is
  begin
    var i:= 1;
    f := x + i
  end
```

Man kan så anvende **f** f.eks. i

```
y := f(17) + 4
```

Her vil **y** få tildelt værdien 22. Opgaven er nu at lave en komplet big-step operationel semantik for **Bof**. Antag rent statiske scoperegler. Den abstrakte syntaks for **Bof** er denne:

- Vi tilføjer en syntaktisk kategori af udtryksprocedure-navne, **Fnavne**, hvor f betegner et vilkårligt element i **Fnavne**. Vi antager at $\mathbf{Fnavne} \subseteq \mathbf{Var}$.
- Vi tilføjer en syntaktisk kategori af udtryksprocedure-erklæringer **Erk_F**, hvor D_F betegner et vilkårligt element i **Erk_F** og ε betegner den tomme erklæring

Opbygningsreglerne for **Kom** og **Aud** er ændret til

$$\begin{aligned} S &::= \dots \mid \mathbf{begin} \ D_V \ D_F \ S \ \mathbf{end} \\ a &::= \dots \mid f(a) \end{aligned}$$

og opbygningsreglen for **Erk_F** er

$$D_F ::= \mathbf{function} \ f(x) \ \mathbf{is} \ S \ \mathbf{end}; D_F \mid \varepsilon$$

3 Målet med denne opgave

Målet med denne opgave er at give en fuldstændig definition af big-step-semantikken for **Bof**. Besvarelsen skal rumme en komplet operationel semantik for sproget i samme format som beskrivelsen af **Bip** i bogens appendiks.

Dette at indføre udtryksprocedurer får en del konsekvenser. Her er derfor nogle vink:

- Når man evaluerer et aritmetisk udtryk, kan det nu indebære en *sideeffekt* ved at man udfører en kommando, nemlig kroppen af en funktion. Transitionerne i big-step-semantikken for **Aud** må derfor være på en anden form end $env_V, sto \vdash a \rightarrow_a v$.
- En variabel-erklæring **var** $x := a$ indebærer at vi evaluerer et aritmetisk udtryk. Det samme er tilfældet, når vi evaluerer et boolsk udtryk. Er der andre steder, hvor der kan ske noget nyt af samme grund?
- Det bliver nødvendigt at definere hvad funktions-environments er!
- Find ud af hvor man skal gemme sin alias-variabel, når en funktion bliver kaldt.
- Og find ud af, hvordan vi finder værdien igen.