

Syntaks og semantik

Syntaksopgave

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 besvarelsen er

mandag 12/3 kl. 12:00 (dansk tid)

Herefter får grupperne hinandens besvarelser udleveret og skal kommentere dem med frist *mandag 2/4 kl. 12:00*. Herefter vil jeg selv 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 syntaksopgaven med en tilsvarende opgave der har relation til gruppens projekt. Det skal i så fald aftales med mig, og deadline for aflevering af *opgavestillingen* er *fredag 16/2 kl. 12:00*.

Der er en delopgave for hver kursusgangs opgaveregning. Afsæt tid til den ved den pågældende øvelsesgang, så I er sikker på at få delopgaven lavet.

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

2 Opgaven

Syntaksopgaven handler om sproget **Sok**¹, der er et meget lille programmeringssprog.

2.1 Uformel syntaks

En uformel beskrivelse af **Sok** er denne:

- Programmer i **Sok** er en følge af globale variabelerklæringer efterfulgt af en følge af metodeerklæringer. Den sidste metode skal hedde **start**. Erklæringer er adskilt med semikolon.
- En variabel erklæres og initialiseres på samme tid. Alle variabler har heltallige værdier. Et eksempel på en erklæring er

```
var x = 7
```

Variabelnavne må bestå af store og små bogstaver og cifre fra 0 til 9. Variable skal starte med et bogstav.

- Heltalskonstanter er følger af cifre, eventuelt med fortegn. En heltalskonstant må ikke starte med cifret 0.
- En metode erklærer man ved at skrive **metode**, metodens navn og en række kommandoer afsluttet med **end** som f.eks.

```
metode subtraktion
    z := x - y ;
    z := (z-3) + x
end
```

- De eneste kommandoer i **Sok** er tilskrivninger og metodekald. Kommandoer er adskilt med semikolon. Der er intet semikolon før **end**.
- En tilskrivning har en variabel på venstresiden og et aritmetisk udtryk på højresiden som f.eks.

```
z := (x *2) + 4
```

I aritmetiske udtryk kan forekomme variabler og numeraler. Aritmetiske udtryk kan opbygges med + og * og der kan sættes parenteser efter behov. Parenteser skal være balancerede på normal vis, så til enhver venstreparentes skal høre en højreparentes.

- En metode kan kalde en anden metode ved brug af det reservede ord **kald** som f.eks. i

```
kald subtraktion;
```

¹Syntaks-opgavens konstruktion.

2.2 Opgaver om Sok

1. Lad L_1 være sproget indeholdende alle gyldige variabelnavne, og L_2 sproget af alle heltalskonstanter. Opskriv på baggrund af den uformelle beskrivelse regulære udtryk der beskriver henholdsvis L_1 og L_2 .
2. Konstruér deterministiske endelige automater der genkender L_1 henholdsvis L_2 .
3. Konstruér på baggrund af den uformelle beskrivelse en NFA som genkender L_1 . Automaten skal have så få tilstande som muligt. Konverter jeres NFA til en DFA. Forklar hvorfor det er vigtigt at forstå hvordan man konverterer en NFA til en DFA.
4. (a) Konstruér en NFA for L_1 ud fra det regulære udtryk for L_1 og sammenlign den med jeres NFA fra opgave 3.
(b) Konstruér et regulært udtryk for L_2 ved at konvertere jeres DFA fra opgave 2. Forklar hvor denne konstruktion kan bruges i oversættelsekonstruktion og i implementation af en teksteditor.
5. Lad L_3 være sproget indeholdende alle aritmetiske udtryk, og bevis at L_3 ikke er et regulært sprog, hvis vi insisterer på at parenteser skal være balancerede.
6. Opskriv en kontekstfri grammatik for hele **Sok** på baggrund af den uformelle beskrivelse. Giv et eksempel på et gyldigt program i **Sok** på mindst 5 linjer og vis ved at konstruere en derivation at programmet er gyldigt ifølge jeres grammatik.
7. Opskriv en grammatik for L_3 som ikke er tvetydig. Konstruér herudfra en PDA som genkender L_3 . Forklar hvad denne PDA er for en algoritme.
8. Bevis at **Sok** ikke er et kontekstfrit sprog, hvis vi insisterer på at en variabel altid skal erklæres, før vi må benytte den.