Oversættere, ugeopgave 2

Anders Bjerg Pedersen (andersbp@me.com)

29. november 2009

Opgave 1

Vi bliver her bedt om at konstruere en entydig kontekstuafhængig grammatik for operatorerne +, -, *, **, + (unær) og - (unær) under anvendelse af variablene x, y og z. Vi opskriver produktionerne startende med de mindst bindende operatorer øverst og anvender metoderne fra afsnit 3.4.1:

$$
\begin{aligned}
E_1 &\rightarrow E_1 + E_2 \quad (venstreassociativ) \\
E_1 &\rightarrow E_1 - E_2 \quad (venstreassociativ) \\
E_1 &\rightarrow E_2 \\
E_2 &\rightarrow E_2 * E_3 \quad (venstreassociativ) \\
E_2 &\rightarrow E_2 / E_3 \quad (venstreassociativ) \\
E_2 &\rightarrow E_3 \\
E_3 &\rightarrow E_4 * * E_3 \quad (højreassociativ) \\
E_3 &\rightarrow E_4 \\
E_4 &\rightarrow +E_4 \quad (venstreassociativ) \\
E_4 &\rightarrow -E_4 \quad (venstreassociativ) \\
E_4 &\rightarrow x \\
E_4 &\rightarrow y \\
E_4 &\rightarrow z
\end{aligned}
$$

Opgave 2 (Exercise 3.14)

a)

Vi anvender metoden fra afsnit 3.12.1 til at fjerne venstrerekursion fra grammatikken. Dette gør vi ved at tilføje nonterminalen E' og dennes produktion $E' \rightarrow \varepsilon$. De resterende transformationer $E \rightarrow E\alpha$ transformeres til $E' \rightarrow \alpha E'$, og til sidst transformeres $E \rightarrow \textbf{num}$ til $E \rightarrow \textbf{num}\ E'$:

$$
\begin{aligned}
E' &\rightarrow \varepsilon \\
E' &\rightarrow E + E' \\
E' &\rightarrow E * E' \\
E &\rightarrow \textbf{num}\ E'
\end{aligned}
$$

b)

Vi anvender metoden i afsnit 3.12.2 og tilføjer en ekstra nonterminal Aux til at beskrive forskellene i produktion 2 og 3 fra a) ovenfor. Vi lader nu blot $E \to EAux$ og lader så Aux have to produktioner, der tager sig af selve postfix-operatorerne:

$$\begin{aligned} E' &\to \varepsilon \\ E' &\to EAux \\ Aux &\to +E' \\ Aux &\to *E' \\ E &\to \textbf{num}\ E' \end{aligned}$$

c)

Vi beregner *Nullable*, *FIRST* og *FOLLOW* ud fra metoderne i afsnit 3.8 og 3.10 (*Nullable* til venstre i skemaet, *FIRST* til højre):

	$Init$	$Iter1$	$Iter2$	$Init$	$Iter1$	$Iter2$
$E' \to \varepsilon$	F	T	T	$\emptyset$	$\emptyset$	$\emptyset$
$E' \to EAux$	F	F	F	$\emptyset$	$\emptyset$	$\{\textbf{num}\}$
$Aux \to +E'$	F	F	F	$\emptyset$	$\{+\}$	$\{+\}$
$Aux \to *E'$	F	F	F	$\emptyset$	$\{*\}$	$\{*\}$
$E \to \textbf{num}\ E'$	F	F	F	$\emptyset$	$\{\textbf{num}\}$	$\{\textbf{num}\}$
E	F	F	F	$\emptyset$	$\{\textbf{num}\}$	$\{\textbf{num}\}$
E'	F	T	T	$\emptyset$	$\emptyset$	$\{\textbf{num}\}$
Aux	F	F	F	$\emptyset$	$\{+,*\}$	$\{+,*\}$

Vi kan nu beregne *FOLLOW* ud fra *Nullable* og *FIRST* og tilføjer produktionen $E'' \to E\$$:

	$FOLLOW$
$E'' \to E\$$	$\{\$\} \subseteq FOLLOW(E)$
$E' \to \varepsilon$	$-$
$E' \to EAux$	$\{+,*\} \subseteq FOLLOW(E),\ FOLLOW(E') \subseteq FOLLOW(Aux)$
$Aux \to +E'$	$FOLLOW(Aux) \subseteq FOLLOW(E')$
$Aux \to *E'$	$-$
$E \to \textbf{num}\ E'$	$FOLLOW(E) \subseteq FOLLOW(E')$
E	$FOLLOW(E) = \{\$,+,*\}$
E'	$FOLLOW(E') = \{\$,+,*\}$
Aux	$FOLLOW(Aux) = \{\$,+,*\}$

d)

Vi konstruerer LL(1)-tabellen som beskrevet i afsnit 3.11.2:

	num	$+$	$*$	\$
E	$E \to \textbf{num}\ E'$	$-$	$-$	$-$
E'	$E' \to EAux$	$E' \to \varepsilon$	$E' \to \varepsilon$	$E' \to \varepsilon$
Aux	$-$	$Aux \to +E'$	$Aux \to *E'$	$-$

Opgave 3

Der menes her først, at hvis en kontekstuafhængig grammatik ikke genkender et tomt input (aka. den tomme streng), kan den transformeres på en sådan måde, at ingen af produktionerne i grammatikken har ε på deres højreside. Man kan med andre ord erstatte eventuelle "tomme" højresider med alternative produktioner uden at ændre grammatikken. Dette udsagn er korrekt, da vi her taler om *kontekstuafhængige* grammatikker. Her vil vi aldrig kunne have en højreside bestående af både terminaler og nonterminaler, og der vil derfor altid være en vej fra startsymbolet til enhver af de andre nonterminaler. Hmmmm...

Den anden påstand er, at hvis en grammatik accepterer den tomme streng, er det nødvendigt at have en produktion fra en nonterminal til den tomme streng (altså $S \to \varepsilon$), men der vil i dette tilfælde ikke være brug for flere ε-højresider end denne. Dette udsagn er ligeledes korrekt, da man (uden at ændre grammatikken) kan lave en transformation der sender startsymbolet S over i en ny nonterminal E, som er den eneste af produktionerne, der kan gå over i ε. De resterende tomme produktioner sendes nu over i E i stedet. Hermed har vi kun én tom produktion, og grammatikken accepterer den tomme streng. En sådan ændret grammatik kan efterfølgende reduceres.

Opgave 4

Regulære udtryk har den fordel, at de kan læses tegn for tegn, samt at de i sig selv er skrevet i en grammatik (som for eksempel kan være defineret som i bogens afsnit 2.2). Så ligesom der kan opstilles produktioner, der beskriver grammatikker, som er udtrykt i regulære udtryk, kan der også opstilles produktioner, der beskriver opbygningen af regulære udtryk generelt. Da regulære udtryk desuden er utvetydige og kan læses fra venstre mod højre, kan vi bruge LL(1)-parsing til at fortolke dem. Tidskompleksiteten er oplagt lineær, da vi læser ét tegn af gangen, indtil vi har nået slutningen af strengen. Hvis parseren anvender tabel-drevet parsing, skal vi for hvert tegn i det regulære udtryk slå op i en todimensionel tabel, hvis størrelse afhænger af de tilladte tegn i grammatikken. Disse opslag kan gøres i konstant tid (?) og bidrager dermed ikke til kompleksiteten, der dermed forbliver lineær.