Inf2A: *Tutorial Exercise 6*

Anderson, Hutchins-Korte

Week 7 (2 Nov – 6 Nov 2009)

If possible, you should attempt to answer these questions before your tutorial.
During the tutorial your tutor will discuss your answers with you, and give assistance and feedback.

Pumping Lemma for CFLs

The use of the Pumping Lemma for CFLs is similar to that for FSAs. Usually we are looking to establish that some language is not context-free by deriving a contradiction the assumption the language is context-free. Review Kozen Lecture 22, pp 148-156.

Recall the statement of the Pumping Lemma *For every CFL* L*, there exists a constant* $k \geq 0$ *such that every* $z \in L$ *such that the length of* z *is greater than or equal to* k *we can find strings* u, v, w, x, y *such that* $z = uvwxy$*,* $vx \neq \varepsilon$*, the length of* vwx *is less than or equal to* k*, and for all* $i \geq 0$ $uv^i wx^i y \in L$

Decide whether the following languages are context-free and provide a justification of your decision

1. $\{a^m b^n a^m b^n \mid n, m \geq 0\}$
2. Suppose $b(n) \in \{0,1\}^*$ is the binary representation of the natural number n, and $rev(t)$ is the reverse of the string t (i.e. the last letter is first, second last second, ...) is the language $\{b(n)\#b(n+1) \mid n \geq 0\}$ context-free? *(Due to Kozen).*
3. $\{b(n)\#rev(b(n+1)) \mid n > 0\}$ *(Also due to Kozen)*
4. $\{a^n b^n a^n \mid n \geq 0\}$

Calculating First and Follow sets

In this section you are asked to calculate First_k and Follow_k sets for a series of grammars. What follows is a brief summary of the approach. For both First_k and Follow_k sets there are a variety of ways of techniques for calculating them. Here we construct successive approximations to the sets taking the empty set as the initial approximation.

Calculating First Sets

Recall the definition of First_k, for a particular grammar $G = (N, \Sigma, P, S)$:

$$\text{First}_k(\alpha) = \{t \in \Sigma^* \mid \alpha \rightarrow_G^* t\beta \text{ and } \|t\| = k \text{ or } \alpha \rightarrow_G^* t \text{ and } \|t\| \leq k\}$$

In order to be able to calculate $\text{First}_k(\alpha)$ for any $\alpha \in (N \cup \Sigma)^*$ we just need to calculate $\text{First}_k(A)$ for each nonterminal A.

Suppose we want to calculate First_2 for each nonterminal in the grammar:

$$S' \rightarrow \textit{term}\$\$$$

$$\textit{term} \rightarrow \textit{braces} \mid \textit{parens} \qquad \textit{braces} \rightarrow \{\textit{list}\} \qquad \textit{list} \rightarrow \textit{term rest}$$

$$\textit{parens} \rightarrow (\textit{list}) \qquad \textit{list} \rightarrow \mathsf{x}\ \textit{rest}$$

$$\textit{rest} \rightarrow , \textit{list} \mid \varepsilon$$

Then we construct a table where the successive columns are approximations to the First_k sets for the nonterminals. The initial approximation is that all the First_k sets are empty. Suppose a nonterminal A of G has m productions: $A \rightarrow \alpha_1 \mid \ldots \mid \alpha_m$ then to get the next approximation of $\text{First}_k(A)$ we set it to:

$$\text{First}_k(A) = \text{First}_k(\alpha'_1) \cup \ldots \cup \text{First}_k(\alpha'_m)$$

where α'_i is a set of strings derived from α_i by substituting the current approximation for $\text{First}_k(B)$ for each occurrence of each nonterminal B in α_i. The following table illustrates this by successively approximating First_2 for the nonterminals of our example grammar.

Iteration	0	1	2	3	4	5	6	7
S'	∅	∅	∅	∅	∅	{x (x	{x (x	{x (x {{ {{ ({ ((
term	∅	∅	∅	∅	{x (x	{x (x	{x (x {{ {{ ({ ((	{x (x {{ {{ ({ ((
braces	∅	∅	∅	{x	{x	{x	{x {{ {{	{x {{ {{
parens	∅	∅	∅	(x	(x	(x	(x ({ ((	(x ({ ((
list	∅	∅	x	x	x x,	x {x (x x,	x {x (x x,	x {x (x x, {{ {{ ({ ((
rest	∅	ε	ε	ε ,x	ε ,x	ε ,x ,{ ,(	ε ,x ,{ ,(	ε ,x ,{ ,(

Calculating Follow Sets

Calculating Follow_k sets is a similar process to that for First_k sets. We begin by taking ∅ as the first approximation to $\text{Follow}_k(A)$ then we generate next approximation to $\text{Follow}_k(A)$ by considering each occurrence $B_i \rightarrow \alpha_i A \beta_i$ of A on the Right Hand Side of a rule and computing the next approximation of $\text{Follow}_k(A) = \bigcup_i \text{First}_k(\text{First}_k(\beta_i)\text{Follow}_k(B_i))$ using our calculated First_k sets and the current approximation of the Follow_k sets. The table below illustrates this by calculating Follow_1 for our example grammar. Note, we do not compute the Follow set for our new top symbol S' because it cannot be followed by any symbol.

Iteration	0	1	2	3
S'				
term	∅	$,	$,	$, })
braces	∅	∅	$,	$, })
parens	∅	∅	$,	$, })
list	∅	})	})	})
rest	∅	∅	})	})

Compute the First_1 and Follow_1 sets for the nonterminals of following grammars (assume upper case letters are non-terminals and lower-case letters are terminals):

5.

$$S' \rightarrow S\$ \qquad S \rightarrow SAB \mid SBC \mid \varepsilon$$

$$A \rightarrow aAa \mid \varepsilon \qquad B \rightarrow bB \mid \varepsilon \qquad C \rightarrow cC \mid \varepsilon$$

6.

$$S' \to S\$ \qquad S \to ABS \mid BCS \mid \varepsilon$$
$$A \to AA \mid a \qquad B \to bB \mid \varepsilon \qquad C \to cC \mid c$$

7.

$$S' \to B\$ \qquad B \to T \vee B \mid T \mid \{B \Rightarrow B\ ;\ B\}$$
$$T \to F \wedge T \mid F \qquad F \to (B) \mid i$$

Checking if a grammar is LL(n)

8. For each of the above grammars:
   (a) Check whether the grammar is LL(1).
   (b) If the grammar is not LL(1), first check whether it is ambiguous. If it is ambiguous, give a sentence with multiple parses in the language generated by the grammar.
   (c) Finally, attempt to change the grammar so it is LL(1) and still generates the same language as the original grammar.

9. Is the following grammar LL(k) for some k? If it is then find the least k for which it is LL(k):

$$S' \to S\$\$\$\$\$ \qquad S \to AC \mid aaB$$
$$A \to aA \mid \varepsilon \qquad B \to bB \mid d \qquad C \to b \mid cC$$

Constructing LL(1) parse tables

For each of the following grammars, construct an LL(1) parse table.

10.

$$S' \to S\$ \qquad S \to NV$$
$$N \to jN \mid n \qquad V \to WN$$
$$W \to aW \mid v$$

11.

$$S' \to S\$ \qquad S \to ABS \mid BCS \mid \varepsilon$$
$$A \to aaA \mid a \qquad B \to bbbB \mid \varepsilon \qquad C \to cC \mid c$$