Parsing

Homework 4 (Deduction-based Parsing), due 18 May 2020

Laura Kallmeyer

SS 2020, Heinrich-Heine-Universität Düsseldorf

Question 1 (Top-down Parsing with deduction rules)

Consider again the CFG from last week's homework: G with $N = \{S, X\}, T = \{a\}$, start symbol S and productions

$$S \to aSa \,|\, X, X \to aX \,|\, a$$

and the input $w = aaa$.

1. *Give **all** items the parser generates for this input. We assume, however, that items $[\alpha, i]$ with the length of α being greater than $n - i$ are not allowed, i.e., the rules block them (as specified on slide 18).*

   For every item, indicate the rule that was used to deduce this item and indicate the antecedent items of this rule.

2. *How does the parser know whether $w = aaa$ is in the language generated by the grammar?*

Solution:

1.

id	item	rule	antecedent item
1.	$[S, 0]$	axiom	
2.	$[aSa, 0]$	predict S_1	from 1.
3.	$[X, 0]$	predict S_2	from 1.
4.	$[Sa, 1]$	scan	from 2.
5.	$[aX, 0]$	pred. X_1	from 3.
6.	$[a, 0]$	pred. X_2	from 3.
7.	$[Xa, 1]$	pred. S_2	from 4.
8.	$[X, 1]$	scan	from 5.
9.	$[\varepsilon, 1]$	scan	from 6.
10.	$[aa, 1]$	pred. X_2	from 7.
11.	$[aX, 1]$	pred. X_1	from 8.
12.	$[a, 1]$	pred. X_2	from 8.
13.	$[a, 2]$	scan	from 10.
14.	$[X, 2]$	scan	from 11.
15.	$[\varepsilon, 2]$	scan	from 12.
16.	$[\varepsilon, 3]$	scan	from 13.

2. The parser found a goal item, item 16.

Question 2 (Unger deduction rules for GNF and for CNF)

1. *Consider the Unger Parser for CFGs in Greibach Normal Form.*

   Give the deduction rules for the Unger Parser for CFGs in GNF where the predictions are constrained by the condition that the first lefthand side element (a terminal) has to match the next input symbol. In this case, a predict item for this terminal is not necessary, one can immediately generate the completed item (dot on the right). If the terminal is the only element in the lefthand side, we generate only the completed item, otherwise one completed item and predict items for all other lefthand side elements.

   For example, assuming that we have productions $A \to aSB$ and $A \to c$ and an input abc, we should be able to

- *deduce* $[a\bullet, 0, 1]$, $[\bullet S, 1, 2]$ *and* $[\bullet B, 2, 3]$ *from* $[\bullet A, 0, 3]$ *in a single step, due to the first production, and*
- *deduce* $[c\bullet, 2, 3]$ *from* $[\bullet A, 2, 3]$ *in a single step, due to the second production.*

2. *Now assume that we are dealing only with grammars in Chomsky Normal Form. The rules of the Unger parsing are then as follows:*

   Predict: $\dfrac{[\bullet A, i, k]}{[\bullet B, i, j], [\bullet C, j, k]} \quad A \to BC \in P, i < j < k$

   Scan: $\dfrac{[\bullet A, i, i+1]}{[A\bullet, i, i+1]} \quad A \to w_{i+1} \in P$

   Complete: $\dfrac{[\bullet A, i, k], [B\bullet, i, j], [C\bullet, j, k]}{[A\bullet, i, k]} \quad A \to BC \in P$

   What is the time complexity of this algorithm in the length of the input string, i.e., of the fixed recognition problem? Explain your answer by estimating the maximal number of different rule applications possible for a given CFG in CNF.

Solution:

1. Predict_scan: $\dfrac{[\bullet A, i_0 - 1, j]}{[a\bullet, i_0 - 1, i_0], [\bullet A_1, i_0, i_1], \ldots, [\bullet A_k, i_{k-1}, i_k]} \quad \begin{array}{l} A \to aA_1 \ldots A_k \in P \\ w_{i_0} = a, j = i_k, i_m < i_{m+1} \text{ for } 0 \le m < k \end{array}$

   This replaces the original scan and predict. Complete remains as the original rule.

2. Predict: maximal $|P| \cdot n \cdot n \cdot n$ different instances that can be applied.

   Scan: maximally $|N| \cdot n$

   Complete: like predict.

   Therefore, we have a total of $\le 2 \cdot |P|n^3 + |N| \cdot n \le cn^3$ different rule applications where c is a constant depending on the grammar. Consequently, the time complexity of the fixed recognition problem is $\mathcal{O}(n^3)$.