Solutions

October 26, 2014

1 Exercise 5

A grammar has a cycle if there is a reachable and productive non-terminal A such that $A \Rightarrow^{+} A$, i.e. it is possible to derive the non-terminal A from A by a nonempty sequence of production rules.

Show that if a grammar has a cycle, then it is not LL(1).

1.1 Solution

We know that LL(1) grammars are not ambiguous. Consider a left most derivation D that contains A. Let D be of the form $S \Rightarrow^{*} xA\beta \Rightarrow^{*} w$, where x is a (possibly empty) sequence of terminals and β is a sentential form (sequence of terminals and non-terminals). Note that such a derivation must exist as A is both reachable (from the start symbol) and productive . Since $A \Rightarrow^{+} A$, we can construct another left most derivation by replacing A by the chain $A \Rightarrow^{+} A$. Formally, $S \Rightarrow^{*} xA\beta \Rightarrow^{+} xA\beta \Rightarrow^{*} w$. Therefore, there exists two different left most derivation and hence two different parse trees for w. This implies that the grammar is ambiguous and hence not LL(1).

2 Exercise 7

This exercise is quite difficult. It is completely optional for you to read and understand the following proof. We would certainly not be asking questions as difficult as this in the quiz.

Show that the language $L = \{a^l b^m \mid l > m\}$ which is defined by the grammar

$$\begin{aligned} S &\rightarrow aS \mid P \\ P &\rightarrow aPb \mid a \end{aligned}$$

cannot have an LL(1) grammar.

2.1 Solution

Say we have an LL(1) grammar G recognizing L. Without loss of generality, assume that G only has reachable and productive non-terminals. Since the language is infinite, the grammar has at least one "recursive" non-terminal N, i.e, $N \Rightarrow^* \alpha N \beta$, where α and β are sentential forms which is a (possibly empty) sequence of terminals and non-terminals. Moreover, there exists a recursive non-terminal A such that $A \Rightarrow^* \alpha A \beta$ and $\alpha \Rightarrow^* a^k$ for some $k > 0$. Otherwise, it is easy to show that the number of a's has to be bounded in every string generated by the grammar.

Case (i): β is empty i.e, $A \Rightarrow^* \alpha A$, or β only derives empty string i.e, $\beta \Rightarrow^* w$ implies $w = \epsilon$.

Consider a derivation D of a string $a^l b^m$, $l > 0$ that uses the production $A \Rightarrow \alpha A \beta$. Note that there has to be one such derivation since A is a reachable non-terminal. Let ρ be the prefix of the derivation before the "last application" of $A \Rightarrow \alpha A \beta$. That is, let D be $S \Rightarrow^* \rho A \delta \Rightarrow \rho \alpha A \beta \delta \Rightarrow^* a^l b^m$, where there is no other application of $A \Rightarrow \alpha A \beta$ after the one shown. By assumption, β is empty or it can derive only empty string. Therefore, D is of the form $S \Rightarrow^* \rho \alpha A \delta \Rightarrow^* a^l b^m$.

We know that α derives a non-empty sequence of a's i.e, $\alpha \Rightarrow a^k$, $k > 0$. Hence, ρ can only derive (a possibly empty) sequence of a's. Otherwise, if $\rho \Rightarrow^* a^l b^i$, $i > 0$ then we can derive a string that does not belong to the language as $\rho\alpha \Rightarrow^* a^l b^i a^k$, where i, k are positive integers. Therefore, D has to be of the form $S \Rightarrow^* a^j A \delta \Rightarrow^* a^l b^m$, $j > 0$. We now show that $j = l$. If $j < l$ then the derivation D would again have to apply the production $A \Rightarrow \alpha A \beta$, since it is the only alternative of A that can start with a (note that G is LL(1)). But, by assumption, ρ is the prefix before the last application of $A \Rightarrow \alpha A \beta$. Hence, there could be no more application of the production beyond a^j in the derivation D. Therefore $j < l$ is not possible. Hence, $S \Rightarrow^* a^l A \delta \Rightarrow^* a^l b^m$.

Now, consider the (partial) derivation $D' : S \Rightarrow^* a^l A \delta \Rightarrow a^l \alpha A \delta \Rightarrow^* a^{l+k} A \delta$, $k > 0$. (The sentential form β is omitted in D' as it either empty or it can only derive ϵ). Since $a^{l+k} b^l \in L$ and the grammar G is LL(1), $a^{l+k} b^l$ has to be derivable through $a^{l+k} A \delta$. That is, $S \Rightarrow^* a^{l+k} A \delta \Rightarrow^* a^{l+k} b^l$. Hence, $A\delta \Rightarrow^* b^l$.

Using this fact in derivation D, we get $S \Rightarrow^* a^l A \delta \Rightarrow^* a^l b^l$. But, $a^l b^l \notin L$ (note that $l > 0$). Hence, when β is empty or when it can only derive ϵ, we obtain a contradiction.

Case (ii): β is non-empty and it derives a non-empty string. That is, $\beta = N_1 N_2 \cdots N_n$ and $\beta \Rightarrow^* w$ s.t. $|w| > 0$.

Claim 1: Both A and β are nullable i.e, $\beta \Rightarrow^* \epsilon$ and $A \Rightarrow^* \epsilon$.

As in the previous case, consider a derivation D of a string $a^l b^m$ that uses the production $A \Rightarrow \alpha A \beta$. By the same argument presented earlier, we can deduce that D has to be of the form $S \Rightarrow^* a^l A \beta \delta \Rightarrow^* a^l b^m$. Since $a^l \in L$ and the grammar G is LL(1), a^l has to be derivable through $a^l A \beta \delta$. Therefore, $S \Rightarrow^* a^l A \beta \delta \Rightarrow^* a^l$. This implies that both A and β are nullable.

Claim 2: $first(\beta) = \{b\}$.

By the above claim, A is nullable. If $a \in first(\beta)$ then $first(A) \cap follow(A) \neq \emptyset$ which violates the LL(1) property. Therefore, $first(\beta) \subseteq \{b\}$. By assumption, β can derive a non-empty string. Hence, $first(\beta) \neq \emptyset$. Therefore, $first(\beta) = \{b\}$.

Now, let's come back to the proof of the main statement. Given $\beta = N_1 N_2 \cdots N_n$. Since β is nullable, each of the N_i's are nullable. By the definition of $follow$, $follow(A) \subseteq follow(N_i)$ for each $1 \leq i \leq n$ as every N_i is nullable. Hence, $b \in follow(N_i)$ for all

$1 \leq i \leq n$ as $b \in follow(A)$. Since $b \in first(\beta)$, there exists a j such that $b \in first(N_j)$. Therefore, $first(N_j) \cap follow(N_j) = \{b\}$ and N_j is nullable. This violates the LL(1) property and hence is a contradiction.

Since we get a contradiction in both cases where β is empty and is non-empty, there cannot exist an LL(1) grammar G for L