Final #1

Mark all correct answers in each of the following questions.

Unless stated otherwise, $G = (N, T, R, S)$ is a context-free grammar without useless letters.

4. (a) If G is ambiguous, then $L(G)$ includes at most finitely many words having a unique parse tree.

   (b) There exists an algorithm which, for every non-terminal A and terminal a, determines whether there exists a word $w \in T^*$, containing at least one occurrence of a, such that $A \overset{*}{\Longrightarrow} w$.

   (c) Let $G' = (N, T, R', S)$, where $R' = \{A \to \alpha^R : A \to \alpha \in R\}$. (Here α^R is the word consisting of the same letters as α, but in the opposite order.) Then G' is unambiguous if and only if G is such.

   (d) The grammar defined by the rules
   $S \to SaSa \mid bb$
   is unambiguous.

5. (a) If there exist two non-terminals A, B such that $\text{Nullable}(A) = \text{Nullable}(B) = \text{true}$, then G is not $LL(1)$.

   (b) The grammar defined by the rules
   $S \to a^2Sb^2 \mid a^5b^5 \mid b^5a^5$
   is $LL(3)$, but not $LL(2)$.

   (c) The grammar defined by the rules

   $$S \to AS \mid A,$$

$A \to aAb \mid ab,$

is not $LL(k)$ for any k.

(d) Let $w_1, w_2, \ldots, w_m$ be distinct words of length 10 in T^*. Then the grammar defined by the rules
$S \to w_1 S \mid w_2 S \mid \ \ldots \ \mid w_m S \mid \varepsilon$
is $LL(k)$ for sufficiently large k.

6. (a) The grammar defined by the rules
$S \to a^3 S b^3 \mid a^9 b^9 \mid b^9 a^9$
is $LR(1)$, but not $LR(0)$.

(b) If $S \to SaS, S \to SbS \in R$ (in addition to other rules) and $ca^3b^2c \in LC(S)$, then $LC(S)$ includes over 10^{30} strings of length 207.

(c) If for every word $w = \sigma_1\sigma_2 \ldots \sigma_n \in L(G)$ there exists only one subword $w' = \sigma_i\sigma_{i+1} \ldots \sigma_j$ for which there exists a non-terminal A with a rule $A \to w'$, and this non-terminal is unique, then G is $LR(0)$.

(d) Denote (for the purposes of this question):

$$RC(A) = \{\gamma \in (N \cup T)^* : S' \overset{*}{\underset{r}{\Longrightarrow}} \beta A \gamma\}, \qquad A \in N.$$

Then $RC(A)$ is regular for each $A \in N$.

Solutions

4. (a) It is possible for all words except for one to have a unique parse tree. In fact, start from any unambiguous grammar G. Add to N a new non-terminal A and to R the rules $S \to A$ and $A \to w$, where w is any word in $L(G)$. For the new grammar, w is the only word with two parse trees.

(b) Let M be the set of non-terminals from which one can produce a word including an occurrence of a. Initialize M as the set of all non-terminals for which there exists a rule whose right-hand side includes an a. Next add to M all non-terminals for which there exists a rule whose right-hand side includes a non-terminal already in M. Continue in the same way until at some stage no non-terminals have been added to M. The set M we have at this stage is the required set.

(c) Every sequence of leftmost derivations for G gives rise to a corresponding sequence of rightmost derivations for G', as follows. To the sequence

$$S \underset{l}{\Longrightarrow} \gamma_1 \underset{l}{\Longrightarrow} \gamma_2 \underset{l}{\Longrightarrow} \dots \underset{l}{\Longrightarrow} \gamma_n = w \in T^*$$

there corresponds the sequence

$$S \underset{r}{\Longrightarrow} \gamma_1^R \underset{r}{\Longrightarrow} \gamma_2^R \underset{r}{\Longrightarrow} \dots \underset{r}{\Longrightarrow} \gamma_n^R = w^R \in T^*.$$

The correspondence is clearly one-to-one. Hence, G has more than one sequence of leftmost derivations producing a word $w \in T^*$ if and only if G' has more than one sequence of rightmost derivations producing the word w^R. Consequently, either both grammars are ambiguous or both are unambiguous.

(d) One way to see that the grammar is unambiguous is to notice that it is what might be called $RR(1)$. Namely, the last letters of the right-hand sides of the two rules in this grammar are distinct terminals. Thus, similarly to the case of $LL(1)$ grammars, when trying to parse top-down, but using rightmost derivations, we have at each stage a unique possible way to continue.

In fact, one may connect it to the preceding question. The grammar obtained by "reversing" all rules is obviously an $LL(1)$ grammar, and in particular unambiguous, and therefore so is the given grammar.

Another way to see it is related to bottom-up parsing. First note by induction that no word in $L(G)$ may contain three consecutive b's. Hence, given a word in $\{a, b\}^*$, we may first reduce each occurrence of the block bb to S. Now we have a word in $\{a, S\}^*$, to be reduced to S. Notice that, if this word has been produced from S, then it contains some occurrences of the block $SaSa$. Moreover, by induction on the number of derivations, we see that the last occurrence of this block must have been obtained by deriving an S, and hence we may reduce this block back to S. Continuing to reduce in this way, we finally get to S (or to another word which contains no occurrences of the block $SaSa$, which means that the original word does not belong to $L(G)$).

Thus, (b), (c) and (d) are true.

5. (a) In general, to know if a grammar is $LL(1)$ or not, we need to consider the rules relating to the same non-terminal. The reason is that we need to decide at each stage in the parse which rule to use for a given non-terminal. The "competitors" are the various rules for the same non-terminal, and not the various non-terminals.

   For example, the grammar defined by the rules

   $S \to AB$,
   $A \to aA \mid \varepsilon$,
   $B \to bB \mid \varepsilon$,

   is clearly $LL(1)$ even though all non-terminals are nullable.

   (b) The grammar is certainly not even $LL(5)$. In fact, consider the sequences of derivations

$$S \Longrightarrow a^2Sb^2 \Longrightarrow a^4Sb^4 \Longrightarrow a^6Sb^6 \Longrightarrow a^{11}b^{11},$$

   and

$$S \Longrightarrow a^5b^5.$$

   The words produced in both cases have the same first 5 letters, yet already the first derivation was distinct.

   Let us show for completeness that G is $LL(6)$. In fact, every sequence of derivations yielding a word in $L(G)$ applies the first rule for S a certain number of times, and then either the second or the third rule once. Thus, suppose we have applied the first rule n times by now. The word we have is $a^{2n}Sb^{2n}$, and we need to get to some word of the form $a^{2n}u$. If the first letter of u is b, we must apply now the third rule. If the prefix of length 6 of u is a^5b, we must apply the second rule. Otherwise, we need to apply the first rule. Thus, at each point during the parse, the following 6 letters suffice in order to determine the rule to be applied.

(c) From the non-terminal A, one can produce any word of the form a^nb^n with $n \geq 1$, and only such words. It follows that $L(G)$ consists of concatenations of such words. The grammar is unambiguous since the number of a^nb^n blocks in a word determines the number of times the rule $S \to AS$ should be applied. Namely, if the number of blocks is r, then this rule should be applied $r-1$ times, followed by the rule $S \to A$, to yield A^r, from which can produce words made of r blocks.

However, the grammar is not $LL(k)$ for any k. The reason is that in the beginning we need to apply the rule $S \to AS$ or $S \to A$ depending on whether the number of a^nb^n blocks in the input word is greater than 1 or equal to 1, respectively. Thus, for any k, the two words a^kb^k and a^kb^kab, that have the same prefix of length k (and even $2k$), require a different rule to be applied for them at the first stage, so that k letters are not always sufficient to determine which rule to apply.

(d) The grammar is $LL(10)$. In fact, at each point during the parse, we already produced a word $w_{i_1}w_{i_2}\dots w_{i_n}$, where $1 \leq i_j \leq m$ for each j, and need to produce from it the word $w_{i_1}w_{i_2}\dots w_{i_n}u$ for a certain u. If $u = \varepsilon$ then we apply the rule $S \to \varepsilon$ to finish the parsing. Otherwise, the prefix of length 10 of u must be one of the w_j's, so that we must apply the rule $S \to w_jS$ with that j.

We mention in passing that G may be $LL(k)$ for some $k < 10$. In fact, if no two of the words w_j have the same prefix of length k, then G is $LL(k)$. Thus, for the minimal k possessing this property, G is $LL(k)$ but not $LL(k-1)$.

Thus, (c) and (d) are true.

6. (a) It is easy to verify that

$$L(G) = \{a^{3n}b^{3n} : n \geq 3\} \cup \{a^{3n}b^9a^9b^{3n} : n \geq 0\}.$$

Let us show that the grammar is $LR(1)$, but not $LR(0)$. Suppose first that the input word w is of the form $a^{3n}b^9a^9b^{3n}$. We need to shift $3n+18$ places, reduce the b^9a^9 to S, and then n consecutive times shift 3 places and reduce a^3Sb^3 to S. Thus, we know when to reduce long before we have shifted enough to be able to actually reduce. In fact, after reading the prefix $a^{3n}b^9a$ we know that the word must be $a^{3n}b^9a^9b^{3n}$. Only after shifting 8 more places can we reduce for the first time. Then, after shifting each time 3 more places we can reduce. (Thus, up to this point we may well suspect that the grammar is $LR(0)$.)

Now let the input be of the form $a^{3n}b^{3n}$. This time we have to shift $3n+9$ places, reduce a^9b^9 to S, and then continue as before. However, at the point where have to make the first reduction we are still in doubt. In fact, the prefix we have read up to this point, namely $a^{3n}b^9$, is also a prefix of the word $a^{3n}b^9a^9b^{3n}$. Hence we are not sure yet that we need to reduce. Only after shifting one place, thus having read the word $a^{3n}b^{10}$, do we know that the input word must be $a^{3n}b^{3n}$, and can make all further reductions right when we have read all letters participating in the reduction.

Let us also show that the grammar is not $LR(0)$ using the context sets. We have

$$LC(S) = LC(S') \cup LC(S) \cdot \{a^3\},$$

and therefore $LC(S) = \{a^3\}^*$. Consequently:

$$\begin{array}{lcl} LR(0)\text{-}C(S \to a^3Sb^3) & = & \{a^3\}^*\{a^3Sb^3\}, \\ LR(0)\text{-}C(S \to a^9b^9) & = & \{a^3\}^*\{a^9b^9\}, \\ LR(0)\text{-}C(S \to b^9a^9) & = & \{a^3\}^*\{b^9a^9\}. \end{array}$$

Since the word $a^9b^9 \in LR(0)\text{-}C(S \to a^9b^9)$ is a prefix of the word $a^9b^9a^9 \in LR(0)\text{-}C(S \to b^9a^9)$, the grammar is not $LR(0)$.

(b) We have

$$LC(S) \supseteq LC(S') \cup LC(S) \cdot \{Sa\} \cup LC(S) \cdot \{Sb\}.$$

Since $ca^3b^2c \in LC(S)$, this implies

$$LC(S) \supseteq \{ca^3b^2c\} \cdot \{Sa, Sb\}^*.$$

Thus, the set of strings of length 207 belonging to $LC(S)$ contains all words of the form $ca^3b^2c\alpha_1\alpha_2\ldots\alpha_{100}$, where each α_i is either Sa or Sb. The number of words of this type is

$$2^{100} = (2^{10})^{10} > 1000^{10} = 10^{30}.$$

We mention in passing that $LC(S)$ contains, according to the data in the question, all the language $\{aS, bS, ca^3b^2c\}^*$, and in particular words of length 207 other than the ones noted above.

(c) The given property implies only that, when given an input word, there is just one possibility of reducing it. However, the condition implies nothing about the following steps. For example, consider the grammar defined by the rules
$S \rightarrow aS \mid a^2S \mid b$.
Clearly, $L(G) = \{a\}^*\{b\}$. For any word $a^nb \in L(G)$, the only reduction possible initially is of the b to S. However, later it is not clear at each step whether we should reduce aS to S or a^2S to S. In fact, the grammar is even ambiguous.

(d) Consider the grammar defined by the rules

$S \rightarrow AB$,

$A \rightarrow \varepsilon$,

$B \rightarrow aBb \mid \varepsilon$.

One verifies straightforwardly that

$$RC(A) = \{a^nb^n : n \geq 0\} \cdot \{\varepsilon, B\},$$

which is not regular.

Thus, (a) and (b) are true.