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 there may exist words $w \in T^*$ produced by a unique parse tree.

   (b) If G_1, G_2 are unambiguous grammars and $L(G_1) \cap L(G_2) = \emptyset$, then it is possible to construct an unambiguous grammar G such that $L(G) = L(G_1) \cup L(G_2)$.

   (c) Suppose $G_i = (N_i, T, R_i, S_i), i = 1, 2$, where $N_1 \cap N_2 = \emptyset$. Let $G = (N_1 \cup N_2 \cup \{S\}, T, R_1 \cup R_2 \cup \{S \to S_1 S_2\})$ (where we assume that $S \notin N_1 \cup N_2$). (By the way, $L(G) = L(G_1)L(G_2)$.) If G_1 and G_2 are unambiguous, then so is G.

   (d) Suppose that for each $w \in T^*$ and $A \in N - \{S\}$ there exists at most one leftmost derivation which produces w from A. Suppose also that S does not appear on the right-hand side of any rule, namely for every $A \to \alpha \in R$ (with any $A \in N$) we have $\alpha \in (N \cup T - \{S\})^*$. Then G is unambiguous.

5. (a) Suppose G is defined by the rules

   $S \to Aab \mid Aba \mid c\,,$

   $A \to SAc \mid AAb \mid ada\,,$

   and we employ the algorithm discussed in class for eliminating left-recursion (with $A_1 = S, A_2 = A$). Then the grammar we obtain contains exactly eight rules.

(b) Call a grammar *indirectly semi-recursive* (for the purposes of this question) if there exist $A \in N$ and $\alpha, \beta \in (N \cup T)^*$ such that $A \overset{+}{\Longrightarrow} \alpha A \beta$. If $L(G)$ is infinite then there does not exist a grammar, equivalent to G, that is not indirectly semi-recursive.

(c) Denote by $l(G)$ the sum of lengths of the right-hand sides of all rules in R, namely:

$$l(G) = \sum_{A \to \alpha \in R} |\alpha|.$$

For a grammar G with neither ε-productions nor unit productions, let $\mathsf{Chomsky}(G)$ be the grammar in Chomsky Normal Form, equivalent to G, constructed according to the algorithm presented in class. Then there exists a function $f : \mathbf{N} \to \mathbf{N}$ (where $\mathbf{N}$ denotes the set of positive integers) such that $l(\mathsf{Chomsky}(G)) \le f(l(G))$ for every grammar with neither ε-productions nor unit productions nor useless letters.

(d) Suppose all rules in R are of one of the three forms $A \to b$, $A \to bC$, and $A \to Bc$. Then a slight modification of the CYK algorithm yields a parsing algorithm that works in time $O(n^2)$ for words on length n.

6. (a) The number of stages (not including stage 0) in the algorithm presented in class for computing the set FIRST is at most $|T|$.

(b) For $A \in N$, denote by $\mathsf{DIRECT\_FOLLOW}(A)$ the set of all letters $X \in N \cup T$ for which R includes a rule of the form $B \to \alpha A X \beta$ for some $B \in N$ and $\alpha, \beta \in (N \cup T)^*$. Then:

$$\mathsf{FOLLOW}(A) = \bigcup_{X \in \mathsf{DIRECT\_FOLLOW}(A)} \mathsf{FIRST}(X).$$

(c) If R includes the four rules $A \to aB$, $A \to \varepsilon$, $B \to aA$, and $B \to \varepsilon$ (in addition to other rules), then the grammar is not $LL(1)$.

(d) If R includes the rules $A \to SBS$ and $A \to SbS$ (in addition to other rules), and $L(G)$ is infinite, then G is not $LL(k)$ for any $k \ge 1$.

Solutions

4. (a) Ambiguity means that there are words that can be produced by more than one parse tree. However, there may still be words produced by a single tree. For example, the grammar defined by the rules

$S \to A \mid a \mid b\,,$

$A \to a\,,$

is ambiguous since the word a can be produced in two essentially different ways, namely $S \Longrightarrow a$ and $S \Longrightarrow A \Longrightarrow a$. However, there is clearly a unique parse tree for the word b.

As a more interesting example, one may consider the grammar with **if-then** and **if-then-else**, discussed in class. The grammar is ambiguous, yet words without any occurrence of **else** are produced by a unique parse tree, as are words with an equal number of occurrences of **if** and **else**.

(b) The classical construction of a grammar that accepts the language $L(G_1) \cup L(G_2)$ (obtained by an addition of a new start symbol S and the rules $S \to S_1$ and $S \to S_2$) is easily seen to provide an unambiguous grammar in our case.

(c) Suppose G_1 and G_2 are defined by the rules

$S_1 \to a \mid ab\,,$

and

$S_2 \to a \mid ba\,,$

respectively. Then the word aba is produced in G in two ways: $S \Longrightarrow S_1S_2 \Longrightarrow aS_2 \Longrightarrow aba$, and $S \Longrightarrow S_1S_2 \Longrightarrow abS_2 \Longrightarrow aba$. Thus, while both G_1 and G_2 are clearly unambiguous, G is ambiguous.

(d) The grammar defined by the rules

$S \to A \mid a\,,$

$A \to a\,,$

satisfies the required condition, yet it is obviously ambiguous as the word a may be obtained in two ways.

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

5. (a) The rules for S do not have left-recursion. For A, we first need to replace the rule $A \to SAc$ by rules whose right-hand side does not start with the letter S. We obtain the following rules for A:

$$A \to AabAc \mid AbaAc \mid cAc \mid AAb \mid ada\,.$$

Now we need to get rid of the direct left-recursion we still have for A. We add a new non-terminal A', and instead of the five above rules for A obtain the rules

$$A \to cAcA' \mid adaA'\,,$$
$$A' \to abAcA' \mid baAcA' \mid AbA' \mid \varepsilon\,.$$

The new grammar has altogether nine rules.

(b) Consider any parse tree corresponding to a grammar that is not indirectly semi-recursive. Take any path from the root to a leaf of the tree. The assumed property of the grammar implies that all nodes along the way have distinct labels. Hence the height of the tree is bounded by $|N|$. It follows that $L(G)$ is finite.

(c) In the process of passing from G to $\mathsf{Chomsky}(G)$, we first add a rule $C_a \to a$ for each terminal a. Next we replace each rule of the form $A \to B_1B_2\dots B_k$ with right-hand side of length $k \geq 3$ by $k-1$ rules with right-hand sides of length 2 each. It follows that $l(\mathsf{Chomsky}(G)) \leq |T| + 2l(G)$. Since all letters are useful, each terminal appears on the right-hand side of at least one rule, and consequently $|T|$ is bounded above by $l(G)$. It follows that $l(\mathsf{Chomsky}(G)) \leq 3l(G)$, so that we may take f as the function given by $f(m) = 3m$.

(d) We proceed as in the CYK algorithm. This time, the question for which non-terminals A and subwords $a_ia_{i+1}\dots a_j$ of the given input word $a_1a_2\dots a_n$ we have $A \overset{*}{\Longrightarrow} a_ia_{i+1}\dots a_j$ reduces to a bounded number of questions of the same form, with $a_ia_{i+1}\dots a_j$ replaced by one of the subwords $a_ia_{i+1}\dots a_{j-1}$ and $a_{i+1}a_{i+2}\dots a_j$, each of length $j-i$. Thus, for each $k \leq n$, we answer the above question for words of length k in time $O(k)$. Going over all k up to n, we complete the work in time $O(n^2)$.

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

6. (a) At the first stage, we find that a terminal a belongs to the FIRST set of a non-terminal A if $A \to \alpha a \beta \in R$ for some α, β with $\mathsf{Nullable}(\alpha)$. At the second stage we find the same if $A \to \alpha B \beta \in R$, where $\mathsf{FIRST}(B)$ is known to include the terminal a from the preceding stage and $\mathsf{Nullable}(\alpha)$, and so forth. It follows that the number of stages is bounded above by $|N|$. This bound cannot be reduced in general, as the grammar defined by the rules

$S \to A_1\,,$
$A_1 \to A_2\,,$
$\ldots$
$A_{m-2} \to A_{m-1}\,,$
$A_{m-1} \to a \mid \varepsilon\,,$

shows.

(b) In the proposed equality, indeed every letter belonging to the right-hand side belongs to the left-hand side too. In fact, let $a \in \mathsf{FIRST}(X)$, say $X \overset{*}{\Longrightarrow} \gamma a \delta$ for some $\gamma, \delta \in (N \cup T)^*$ with $\mathsf{Nullable}(\gamma)$, where $X \in \mathsf{DIRECT\_FOLLOW}(A)$. Since all letters are useful, for suitable $\alpha', \beta' \in (N \cup T)^*$ we have

$$S \overset{*}{\Longrightarrow} \alpha' B \beta' \Longrightarrow \alpha' \alpha A X \beta \beta' \overset{*}{\Longrightarrow} \alpha' \alpha A \gamma a \delta \beta \beta' \overset{*}{\Longrightarrow} \alpha' \alpha A a \delta \beta \beta',$$

so that $a \in \mathsf{FOLLOW}(A)$.

However, the inverse inclusion is false in general. If $\mathsf{Nullable}(X)$, then elements of $\mathsf{FIRST}(\beta)$ also belong to $\mathsf{FOLLOW}(A)$. For example, consider the grammar defined by the rules

$S \to ABC\,,$
$A \to a\,,$
$B \to b \mid \varepsilon\,,$
$C \to c\,.$

One verifies easily that $\mathsf{FOLLOW}(A) = \{b, c\}$, while

$$\bigcup_{X \in \mathsf{DIRECT\_FOLLOW}(A)} \mathsf{FIRST}(X) = \{b\}.$$

(c) It is easy to see that the grammar defined by the rules

$$
\begin{aligned}
&S \to A\,,\\
&A \to aB \mid \varepsilon\,,\\
&B \to aA \mid \varepsilon\,.
\end{aligned}
$$

is $LL(1)$.

(d) Let $k \geq 1$ be arbitrary and fixed. We claim that G is not $LL(k)$. Since $L(G)$ is infinite, we may find a word $u \in L(G)$ with $|u| \geq k$. Consider a sequence of derivations of the form

$$S \stackrel{*}{\Longrightarrow} vA\beta \Longrightarrow vSBS\beta \stackrel{*}{\Longrightarrow} vuBS\beta \stackrel{*}{\Longrightarrow} vuw$$

for some $v, u, w \in T^*$ and $\beta \in (N \cup T)^*$. Now suppose we have to parse the input word vuw. After getting the sentential form $vA\beta$, the next k letters of the input that we have to match are the first k letters of u. Clearly, both rules $A \to SBS$ and $A \to SbS$ are equally suitable to be used at this stage (as far as the next k letters of the input are concerned). Hence G is not $LL(k)$.

Thus, only (d) is true.