Theory of Computation, CSCI 438 spring 2022
Ambiguity in grammars and more examples, pg. 107-108, Feb. 16

Exercises 2.6b, 2.15, 2.16 and 2.17

2.6 b The complement of the language $\{a^n b^n \mid n \geq 0\}$.

Answer developed by students in the class and this appears to be unambiguous:

$S \rightarrow bX \mid Xa \mid aSb$

$X \rightarrow aX \mid bX \mid \varepsilon$

My answer is MUCH longer:

Break this into cases:

Case 1: $a^n b^m$ where $n<m$

Case 2: $a^n b^m$ where $m<n$

Case 3: b appears previous to an a

The following grammar derives the language, but allows ambiguity.

$S \rightarrow C_1 \mid C_2 \mid C_3$

$C_1 \rightarrow aC_1 b \mid C_1 b \mid b$

$C_2 \rightarrow aC_2 b \mid aC_2 \mid a$

$C_3 \rightarrow X\ b\ a\ X$

$X \rightarrow aX \mid bX \mid \varepsilon$

Here is an unambiguous grammar for the language.

$S \rightarrow C_1 \mid C_2 \mid C_3$

$C_1 \rightarrow aC_1 b \mid B$

$B \rightarrow bB \mid b$

$C_2 \rightarrow aC_2 b \mid A$

$A \rightarrow aA \mid a$

$C_3 \rightarrow A'b\ a\ X$

$A' \rightarrow aA' \mid \varepsilon$

$X \rightarrow aX \mid bX \mid \varepsilon$

2.15 Give a counterexample to show that the following construction fails to prove that the class of context-free languages is closed under star. Let A be a CFL that is generated by the CFG $G = (V, \Sigma, R, S)$. Add the new rule $S \to SS$ and call the resulting grammar G'. This grammar is supposed to generate A*.

Consider the language L= {a}. Note that $L^*=\{ \varepsilon, a, aa, aaa, \ldots\}$

A grammar of L is $G=(\{S\}, \{a\}, \{(S\to a)\}, S)$.

The construction gives: $G'=(\{S\}, \{a\}, \{(S\to a), (S\to SS)\}, S)$.
However $\mathcal{L}(G)$ does not contain ε so the construction doesn't produce a grammar for L^*.
(There are other problems with the construction as well.)

2.16 Show that the class of context-free languages is closed under the regular operations, union, concatenation, and star.

Given context-free languages L_1, described by the grammar $G_1=(V_1, \Sigma, R_1, S_1)$ and L_2, described by the grammar $G_2=(V_2, \Sigma, R_2, S_2)$

Union:
The grammar $G_{union} = (V_1 \cup V_2, \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{ (S\to S_1), (S\to S_2) \}, S)$ describes $L_1 \cup L_2$ so context-free languages are closed under the union operator.
(In this case S was added to the variables and the rules $S\to S_1 \mid S_2$ were added to the rules.)

Concatenation:
The grammar $G_{concat} = (V_1 \cup V_2, \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{ (S\to S_1 S_2) \}, S)$ describes $L_1 \circ L_2$ so context-free languages are closed under the concatenation operator.
(In this case S was added to the variables and the rule $S\to S_1 S_2$ was added to the rules.)

Star:
The grammar $G_{star} = (V_1 \cup \{S\}, \Sigma, R_1 \cup \{ (S\to \varepsilon), (S\to S_1 S_1) \}, S)$ describes L_1^* so context-free languages are closed under the star operator.
(In this case S was added to the variables and the rule $S\to \varepsilon \mid S_1 S_1$ was added to the rules.)

2.17 Use the results of Exercise 2.16 to give another proof that every regular language is context- free, by showing how to convert a regular expression directly to an equivalent context-free grammar.

Given a regular language L. It has been proven that every regular language has a regular expression that describes it. Let r be a regular expression that describes L. That is, let r be such that $\mathcal{L}(r)=L$. Since r is a regular expression it must be of the form:

- a where $a \in \sum$,
- ε ,
- Φ ,

or be formed recursively from two regular expressions r_1 and r_2 are regular expressions

- $r_1 \cup r_2$,
- $r_1 \circ r_2$,
- $r_1{}^*$

For r = a where $a \in \sum$, a grammar for $\mathcal{L}(a)$ is G = ({S}, Σ, { (S, a) }, S)

For $r = \varepsilon$, a grammar for $\mathcal{L}(\varepsilon)$ is G = ({S}, Σ, { ($S \rightarrow \varepsilon$) }, S)

For $r = \Phi$, a grammar for $\mathcal{L}(\varepsilon)$ is G = ({S}, Σ, { }, S)

The results of exercise 2.16 show that context-free languages are closed under union, concatenation and star-closure, thus every regular language is context-free.