Homework 1

14d. Find a grammar for $\Sigma = \{a, b\}$ that generates the set of all strings with at least three a's.

Solution: The following is a suitable grammar:

$$S \rightarrow AaAaAaA$$

$$A \rightarrow aA|bA|\lambda$$

This grammar is guaranteed to produce a string with at least three a's, while providing the flexibility for inserting any number of a's and b's anywhere in the string.

14e. Find a grammar for $\Sigma = \{a, b\}$ that generates the set of all strings that start with an a and end with a b.

Solution: The following is a suitable grammar:

$$S \rightarrow aA$$

$$A \rightarrow aA|bA|b$$

The start state starts the string with an a. We then move to state A, where we generate any number of a's or any number of b's. The string can only terminate with a b, fulfilling the second condition. The following grammar would also work:

$$S \rightarrow aAb$$

$$A \rightarrow aA|bA|\lambda$$

Again, all strings would start with an a and end with a b, while generating any number of a's and b's in the middle.

17b. Let $\Sigma = \{a, b\}$. Find a grammar that generates:

$$L_2 = \{a^{3n}b^{2n} : n \geq 2\}$$

Solution: We start by generating some sample strings:

- *aaaaaabbbb*
- *aaaaaaaaabbbbbb*

A few things become clear. First, we need to generate the a's before we generate the b's. Second, while we generate a's we generate 3 a's at a time, and while we generate b's we generate 2 b's at a time. Therefore, an acceptable grammar would look like this:

$$S \rightarrow aaaaaaAbbbb$$

$$A \rightarrow aaaAbb|\lambda$$

The start state guarantees the minimum string *aaaaaabbbb*, while state A extends the string to all n.

17c. Let $\Sigma = \{a, b\}$. Find a grammar that generates:

$$L_3 = \{a^{n+3}b^n : n \geq 2\}$$

Solution: We start again by generating some sample strings:

- *aaaaabb*
- *aaaaaabbb*

Essentially we need to ensure that the string contains three more a's than b's at all times. We can accomplish this by first generating the three a's, then by generating an equal numbers of a's and b's. A suitable grammar would look like this:

$$S \rightarrow aaaaaAbb$$

$$A \rightarrow aAb|\lambda$$

Another suitable grammar would look like this:

$$S \rightarrow aaaAbb$$

$$A \rightarrow aAb|ab$$

which removes the λ step but is otherwise identical.

22. Show that the grammar $G = (\{S\}, \{a, b\}, S, P)$, with productions

$$S \to SS|SSS|aSb|bSa|\lambda$$

and

$$S \to SS|aSb|bSa|\lambda$$

are equivalent.

Solution: We need to show that the two grammars generate the same language to prove their equivalence. The first language generates all strings with an equal number a's and b's. The second language also does this. In fact, we can make the second grammar look exactly like the first grammar. Let S go to SS. Then let the second S go to SS again. At this point we have the production SSS, which is the only state represented in the first grammar but not in the second. Therefore, these two grammars must be equivalent.

12.