Context-Free Grammar

Sipser Page 130

2.4 Give context free grammars that generate the following languages. In all parts $\sum = \{0, 1\}$

a) $L = \{w|w \text{ starts and ends with the same symbol}\}$

$$S_0 \to 0S_10|1S_11|\varepsilon$$

$$S_1 \to 0S_1|1S_1|\varepsilon$$

c) $L = \{w||w| \text{ is odd}\}$

$$S_0 \to 0S_1|1S_1$$

$$S_1 \to 00S_1|01S_1|10S_1|11S_1|\varepsilon$$

e) $L = \{w|w = w^R\}$**, i.e. w is a palindrome**

$$S_0 \to 1S_01|0S_00|0|1|\varepsilon$$

Sipser Page 132

2.21 Let $\sum = \{a, b\}$. Give a CFG generating the language of strings with twice as many a's as b's. Prove that your grammar is correct.

$$S_0 \to S_1\text{aab}|\text{aS}_1\text{ab}|\text{aaS}_1b|\text{aabS}_1|S_1\text{aba}|\text{aS}_1\text{ba}|\text{abS}_1a|\text{abaS}_1|$$

$$S_1\text{baa}|\text{bS}_1\text{aa}|\text{baS}_1a|\text{baaS}_1$$

$$S_1 \to S_0|\varepsilon$$

Proof by induction.

Smallest strings possible are: $x_0 \in \{\text{aab}, \text{aba}, \text{baa}\}$ all of which have $N_A(x_0) = 2N_B(x_0)$, where $N_A(x)$ gives the number of a's in string x and $N_B(x)$ gives the number of b's in string x.

Assume $N_A(x_n) = 2N_B(x_n)$ holds.

Show if n is true n+1 is also true, where x_{n+1} is string x_n with substring $s \in \{\varepsilon, \text{aab}, \text{aba}, \text{baa}\}$ inserted.

Subsequent insertions of S_1 into strings produced, either add 0 a's and 0 b's or 2 a's and 1 b.

Case when 0 a's and 0 b's are inserted.

$$N_A(x_{n+1}) = N_A(x_n) + 0$$

$$N_B(x_{n+1}) = N_B(x_n) + 0$$

$$N_A(x_{n+1}) = 2N_B(x_{n+1})$$

Case when 2 a's and 1 b are inserted.

$$N_A(x_{n+1}) = N_A(x_n) + 2$$
$$N_B(x_n + 1) = N_B(x_n) + 1$$
$$N_A(x_n) + 2 = 2(N_B(x_n) + 1)$$
$$N_A(x_n) = 2N_B(x_n)$$

Therefore all strings generated using the grammar contain twice as many a's as b's.

Study Guide 2

3. Find context free grammara for each of the following languages

a) $L = a^i b^j c^k | i = j + k\}$

$$S_0 \to \mathrm{aS_0}c | S_1$$
$$S_1 \to \mathrm{aS_1}b | \varepsilon$$

b) $L = \{a^i b^j c^k | j = i + k$

$$S_0 \to \mathrm{aS_1bS_2} | S_1\mathrm{bS_2}c | \varepsilon$$
$$S_1 \to \mathrm{aS_1}b | \varepsilon$$
$$S_2 \to \mathrm{bS_2}c | \varepsilon$$

c) $L = \{a^i b^j c^k | i \neq j + k\}$

$\{a^i b^j c^k | i \neq j + k\} = \{a^i b^j c^k | i > j + k\} \bigcup \{a^i b^j c^k | i < j + k\}$

$\{a^i b^j c^k | i > j + k\}$

$$S_1 \to \mathrm{aS_2}$$
$$S_2 \to \mathrm{aS_2} | \mathrm{aS_2}c | S_3$$
$$S_3 \to \mathrm{aS_3} | \mathrm{aS_3}b | \varepsilon$$

$\{a^i b^j c^k | i < j + k\}$

$$S_4 \to S_5 c$$
$$S_5 \to S_5 c | \mathrm{aS_5}c | S_6$$
$$S_6 \to S_6 b | \mathrm{aS_6}b | \varepsilon$$

$\{a^i b^j c^k | i \neq j + k\}$

$$S_0 \to S_1 | S_4$$

e) $L = \{a^i b^j c^k | j = i \text{ or } j = k\}$

$\{a^i b^j c^k | j = i\}$

$$S_1 \to \mathrm{aS_2bS_3} | \varepsilon$$
$$S_2 \to \mathrm{aS_2}b | \varepsilon$$
$$S_3 \to \mathrm{cS_3} | \varepsilon$$

$\{a^i b^j c^k | j = k\}$

$$S_4 \to S_5\mathrm{bS_6}c | \varepsilon$$
$$S_5 \to \mathrm{cS_5} | \varepsilon$$
$$S_6 \to \mathrm{bS_6}c | \varepsilon$$

$\{a^i b^j c^k | i > j = i \text{ or } j + k\}$

$$S_0 \to S_1 | S_4$$

Context Free Grammars - week 5/6-ish

2.8 - Show that CFLS are closed under union, concatenation, and star

1. Closure under union - show that $\forall L_1, L_2 \in \text{CFL}\,, L_1 \cup L_2 \in \text{CFL}$

   Let the start variables for L_1 and L_2 be S_1 and S_2 respectively; then we can define a grammar for their union as follows.

   $$S \rightarrow S_1 | S_2$$

   By definition this will generate any string generated by S_1 or by S_2 (or both), which is the union of the two langauges.

2. Closure under concatenation - show that $\forall L_1, L_2 \in \text{CFL}, \{w_1 w_2 : w_1 \in L_1 \wedge w_2 \in L_2\} \in \text{CFL}$

   Using a similar argument we can define a grammar for the concatenation thus:

   $$S \rightarrow S_1 S_2$$

   By definition this will generate any string consisting of a string from L_1 followed by a string from L_2, which is the concatenation of the two languages.

3. Closure under star - show that $\forall L_1 \in \text{CFL}, L_1^* \in \text{CFL}$

   Once again we have to implement star using the primitives of a CFG; let the start symbol for L_1 be S_1. Then we can define the following grammar:

   $$S \rightarrow S_1 S | \epsilon$$

   This will generate zero or more strings from L_1, which is the definition of star.

2.10 - Give a CFG for $A = \{a^i b^j c^k | (i = j \vee j = k) \wedge i, j, k \geqslant 0\}$ etc

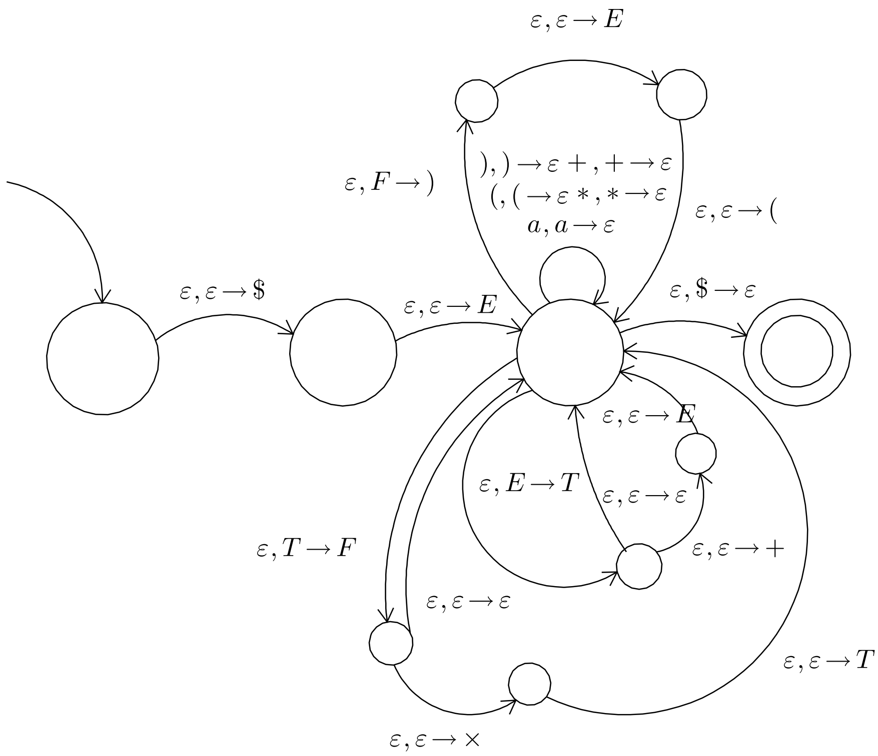
It's easiest to divide the grammar into two parts, one for when $i = j$ and another for when $j = k$

$$\begin{array}{rcl} A_{i=j} & \rightarrow & E_{ab} X_1 \\ E_{ab} & \rightarrow & a\,E_{ab}\,b | \varepsilon \\ X_1 & \rightarrow & c\ X_1 | \varepsilon \\ & \text{and} & \\ A_{j=k} & \rightarrow & X_2\,E_{bc} \\ E_{bc} & \rightarrow & b\,E_{bc}\,c | \varepsilon \\ X_2 & \rightarrow & a\,X_2 | \varepsilon \end{array}$$

Then any string which is from one of these two languages is generated by $A \rightarrow A_{i=j} | A_{j=k}$, which covers everything in the intended set and nothing outside it. This grammar is ambiguous because any string in which $i = j = k$ is generated by both $A_{j=k}$ and $A_{i=j}$. An interesting question is whether it could be disambiguated... (exercise for the reader there)

2.12 - Convert the CFG given in Ex 2.1 to a PDA

$$\begin{array}{rcl} E & \rightarrow & E + T | T \\ T & \rightarrow & T \times F | F \\ F & \rightarrow & (E) | a \end{array}$$


$\varepsilon, \varepsilon \rightarrow E$
$\varepsilon, F \rightarrow)$
$),) \rightarrow \varepsilon \; +, + \rightarrow \varepsilon$
$(, (\rightarrow \varepsilon \; *, * \rightarrow \varepsilon$
$a, a \rightarrow \varepsilon$
$\varepsilon, \varepsilon \rightarrow ($
$\varepsilon, \varepsilon \rightarrow \$$
$\varepsilon, \varepsilon \rightarrow E$
$\varepsilon, \$ \rightarrow \varepsilon$
$\varepsilon, \varepsilon \rightarrow E$
$\varepsilon, E \rightarrow T$
$\varepsilon, \varepsilon \rightarrow \varepsilon$
$\varepsilon, \varepsilon \rightarrow +$
$\varepsilon, T \rightarrow F$
$\varepsilon, \varepsilon \rightarrow \varepsilon$
$\varepsilon, \varepsilon \rightarrow T$
$\varepsilon, \varepsilon \rightarrow \times$

2.5.b. For the following grammar G find an equivalent CFG in Chomsky normal form that generates the language L(G) \ {ε}.

<S → S(S) | ε>

1. Add new start variable

S_0->S
S → S(S) | ε

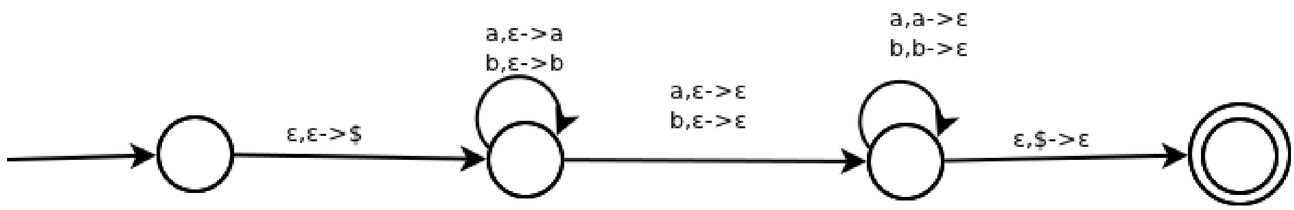
2. Remove A->ε rules.

S_0->S
S->S(S)|(S)|S()|()

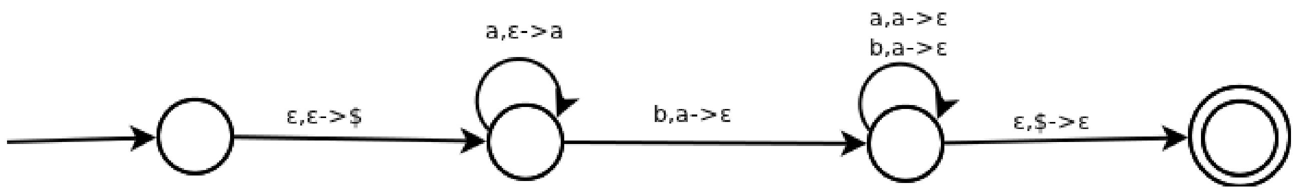
3. Change all rules to the form A->BC

S_0->S
S->SS_1
S1->S_4S_2
S2->SS_5
S->S_1
S->SS_3
S_3->S_4S_5
S_4->(
S_5->)

3.2.a. The language of all odd-length palindromes over the alphabet {a, b}. Recall that w ∈ {a, b}∗ is a palindrome if it reads the same backwards, that is w = w^R.



3.2.c. The language $\{a^N x \mid N \geq 0, \text{and} x \in \{a, b\}^*, \text{and} \mid x \mid \leq N\}$.



3.a. Deterministic push-down automata for the language (over {a, b}) {x | Na (x) = Nb (x)} (recall that Na (x) means the number of occurences of a in x).