Stanley Xiang

Assignment 8

5.1.3 Base case: If a regular expression does not have any operators, it can be expressed in CFG using start symbol G, $G \rightarrow \epsilon$ or $G \rightarrow$ any character.

Inductive step: We must now prove that CFG's can handle the 3 operators of regular languages. (+, ·, *).

Case 1 (+)
For the + operator, we must take either the left or right side argument of the operator. This can be done in CFG using start symbol G, $G \rightarrow G_1 \mid G_2$.

Case 2 (·)
Concatenation is done in CFGs using start symbol G, $G \rightarrow G_1 G_2$.

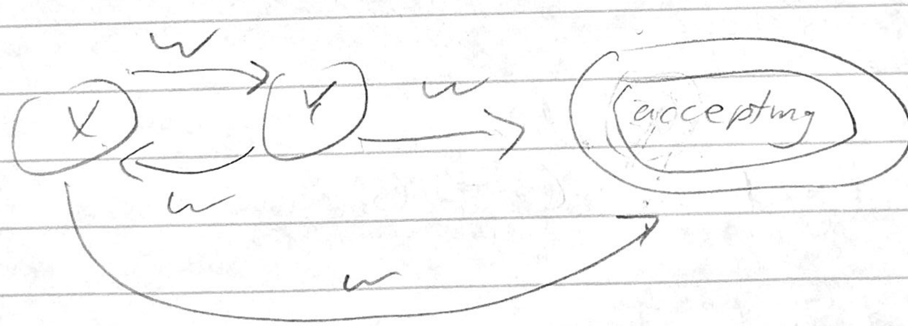
Case 3 (*)
The kleene closure can be represented in CFGs using start symbol G, $G \rightarrow G_i G_i \mid \epsilon$

All regular expressions use these 3 operators, and CFGs are capable of handling them. Therefore Regular languages are context free languages.

5.1.4 a) A right linear CFG is one where each production has at most 1 variable and the variable is on the right side. We can prove that a right liner CFG generates a regular language by representing the CFG as an FSA.

If we have CFG $X \to wY \mid w$
$Y \to wX \mid w$,

Each variable will have its own state in the FSA. If we add more variables, more states will be added. Also, more accepting states are created. A transition function is used to create transitions representing productions.



As seen by above, an FSA was created from a right linear CFG.

b) DFAs can be represented as $D=(Q,\Sigma,\delta,q_0,F)$
Right linear CFGs can be represented as
$G=(V,\Sigma,S,P)$.
V is just the values in Q. Σ appears in both. S is just the start symbol q_0.
P is created by analyzing δ from states q_i to q_k such that $q_i \to q_j q_k$. this enforces right linearity. For each Variable in V, there is a corresponding state in Q.

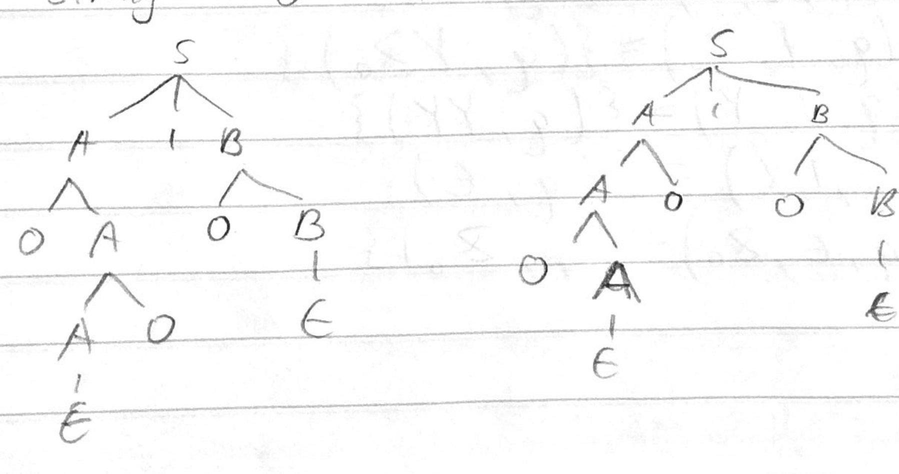
5.4.5 a) There is only one derivation tree for any string with this grammar. A only generates 0's so the first 1 must come from S. S only has 1 production. $A \rightarrow 0A \mid \epsilon$ is not ambiguous because its tree only expands to the right side. The left side is always 0 and the right side can only be A or ϵ. $B \rightarrow 0B \mid 1B \mid \epsilon$ is not ambiguous as well. Its tree only expands to the right side. The left side can either be 0 or 1, and the right side is either B or ϵ. Therefore this grammar is unambiguous, and each string in it only has one tree.

b) $S \rightarrow A1B$

$A \rightarrow 0A \mid A0 \mid \epsilon$

$B \rightarrow 0B \mid 1B \mid \epsilon$

string = 0010



6.2.1 b) $Q = \{q, p\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{X, Z_0\}$

$\delta = \delta$

$q_0 = q$

$Z_0 = Z_0$

$F \subseteq Q = \{p\}$

$\delta(q, 0, Z_0) = \{(q, XZ_0)\}$

$\delta(q, 0, X) = \{(q, XX)\}$

$\delta(q, 1, X) = \{(q, \epsilon)\}$

$\delta(q, \epsilon, X) = \{(p, X)\}$

$\delta(q, \epsilon, Z_0) = \{(p, Z_0)\}$

c) $(\{q, p\}, \{0, 1\}, \{Z_0, X, Y\}, \delta, q, Z_0, \{p\})$

$\delta(q, 0, Z_0) = \{(q, XZ_0)\}$

$\delta(q, 0, X) = \{(q, XX)\}$

$\delta(q, 0, Y) = \{(q, \epsilon)\}$

$\delta(q, 1, Z_0) = \{(q, YZ_0)\}$

$\delta(q, 1, Y) = \{(q, YY)\}$

$\delta(q, 1, X) = \{(q, \epsilon)\}$

$\delta(q, \epsilon, Z_0) = \{(p, Z_0)\}$