Justin Li
Dr. Krehbiel
CSCI 161
10/22/20

HW 5

1) L = { x1#x2 | x1, x2 ∈ 1∗, x1 != x2 }
   CFG:
   S -> 1S | #T
   T -> 1T | ε

2) L = { uv | u ∈ (0 ∪ 1)∗ , v ∈ (0 ∪ 1)∗ 1(0 ∪ 1)∗ , |u| ≥ |v| }
   CFG:
   S -> AB
   A -> 0A | 1A | ε
   B -> C1D
   C -> C0 | C1 | ε
   D -> 0D | 1D | ε

3) a) This language begins with a recursive definition, allowing repetitions of its initial state, all of which are to be replaced by a variable T. T then is recursively defined, generating the terminals a and b at least once, which can be repeatedly produced on its left and right sides.

   b) String: abaaabbbaabb
   2 Leftmost Derivations:
   S => SS => SSS => TSS => abSS => abTS => abaTbS => abaaTbbS => abaaabbbS => abaaabbbT => abaaabbbaTb => abaaabbbaabb
   S => SS => TS => abS => abSS => abTS => abaTbS => abaaTbbS => abaaabbbS => abaaabbbT => abaaabbbaTb => abaaabbbaabb

   c) Steps for Conversion to CNF:
   1) Remove Start Recursion
      S0 -> S
      S -> SS | T
      T -> aTb | ab
   2) Consolidate ε rules
      (N/A)
   3) Collapse Unit Rules

S_0 -> SS | aTb | ab
S -> SS | aTb | ab
T -> aTb | ab

4) Decompose Long Rules
S_0 -> SS | aU | ab
S -> SS | aU | ab
T -> aU | ab
U -> Tb

5) Clean Up Terminals
S_0 -> SS | AU | AB
S -> SS | AU | AB
T -> AU | AB
U -> TB
A -> a
B -> b

4) 2) R = (R1 ◦ R2)
This is equivalent to the CFG that consists of all rules and variables from each of G_1 and G_2 with a new start variable S and one additional rule S -> S_1S_2. Equivalence is because S derives a string derived from both S_1 and S_2; the presence of each rule in the 2 underlying grammars ensures that any strings derivable by G_1 and G_2 are derivable in the new grammar, and there are no other rules to derive any other strings.

3) R = (R_1*)
This is equivalent to the CFG that consists of all rules and variables from G_1 with one additional rule $S_1 \rightarrow S_1S_1$. Equivalence is because S_1 is a string that is derived from another S_1; the presence of the rule in the underlying grammar ensures that any string derivable by G_1 is still derivable in the new grammar, and there are no other rules to derive any other strings.