

COMP 335: Introduction to Theoretical Computer Science

Assignment 4

Nathan Grenier

November 3, 2024

Fall 2024

1. [20 Points] For each of the following languages, give a context-free grammar (CFG).

(a) (5 Points) $L_a = \{a^n b^m : m, n \geq 0 \text{ and } 2n \leq m \leq 3n\}$

$$S \rightarrow A \mid \lambda$$

$$A \rightarrow aAb bB \mid \lambda$$

$$B \rightarrow b \mid \lambda$$

(b) (5 Points) $L_b = \{a^n b^m c^k : k = 2m + n\}$

$$S \rightarrow A$$

$$A \rightarrow aAC_1 \mid B$$

$$B \rightarrow bBC_2 \mid \lambda$$

$$C_1 \rightarrow c$$

$$C_2 \rightarrow cc$$

(c) (5 Points) $L_c = \{a^n b^m c^k : n = m \text{ or } m \leq k\}$

$$S \rightarrow X \mid Y \mid \lambda$$

$$X \rightarrow A_1 C_1$$

$$A_1 \rightarrow aA_1 B_1 \mid \lambda$$

$$B_1 \rightarrow b$$

$$C_1 \rightarrow cC_1 \mid \lambda$$

$$Y \rightarrow A_2 C_2$$

$$C_2 \rightarrow B_2 C_2 c \mid \lambda$$

$$B_2 \rightarrow b \mid \lambda$$

$$A_2 \rightarrow aA_2 \mid \lambda$$

(d) (5 Points) $L_d = \{w \in \{a, b\}^* : w \neq xx, \text{ for any } x \in \{a, b\}^*\}$

$$S \rightarrow X \mid Y$$

$$X \rightarrow aXa \mid bXb \mid aXb \mid bXa \mid a \mid b$$

$$Y \rightarrow aYa \mid bYb \mid aYb \mid bYa \mid aY_1a \mid bY_2b \mid aY_3b \mid bY_4a$$

$$Y_1 \rightarrow bZb \mid aZb \mid bZa$$

$$Y_2 \rightarrow aZa \mid aZb \mid bZa$$

$$Y_3 \rightarrow aZa \mid bZb \mid aZb \mid \lambda$$

$$Y_4 \rightarrow aZa \mid bZb \mid bZa \mid \lambda$$

$$Z \rightarrow aZa \mid bZb \mid aZb \mid bZa \mid \lambda$$

2. [10 Points] Consider the language $L = \{a^{n+1}b^n : n \geq 0\}$

- (a) (5 Points) Describe in English the complement \bar{L} of L . Your description should specify the types of strings that are in \bar{L} . That is, it is not acceptable to say \bar{L} includes every string over $\{a, b\}$ that is not in L , which is obviously true.

Answer:

The language \bar{L} contains all possible strings except for those where the number of consecutive a 's is one more than the number of consecutive b 's, where a 's always come before b 's. This means that the strings that are in \bar{L} are as follows:

- λ
- Strings starting with b . ($baab$)
- Strings ending with a . ($aaabba$)
- Any string $\{x(ba)y : x, y \in \{a, b\}^*\}$ (including $a^{n+1}(ba)b^n$).
- Strings where all consecutive a 's come before all consecutive b 's and the number of a 's is not one more than the number of b 's ($|a| + 1 \neq |b|$). This can be further broken down into:
 - i. $|b| < |a| + 1 \equiv |b| \leq |a|$
 - ii. $|b| > |a| + 1 \equiv |b| \geq |a| + 2$

- (b) (5 Points) Give a CFG for \bar{L} .

$$S \rightarrow S_1 \mid S_2 \mid bS_3 \mid S_4 \mid S_5 \mid \lambda$$

$$S_1 \rightarrow aS_1b \mid S_1b \mid \lambda$$

$$S_2 \rightarrow aS_2b \mid aS_2 \mid aa$$

$$S_3 \rightarrow aS_3 \mid bS_3 \mid \lambda$$

$$S_4 \rightarrow aS_4 \mid bS_4 \mid a$$

$$S_5 \rightarrow aS_5 \mid S_5b \mid bS_5 \mid S_5a \mid ba$$

3. [15 Points] Let G be the following CFG in which S is the start variable:

$$S \rightarrow AB \mid aB$$

$$A \rightarrow aab \mid \lambda$$

$$B \rightarrow bbA$$

(a) (5 Points) Using the procedure discussed in the class, convert G into an equivalent grammar in Chomsky Normal Form (CNF).

Step 1: Remove λ -productions.

$$A \Rightarrow \lambda$$

$$S \rightarrow AB \mid B \mid aB$$

$$A \rightarrow aab$$

$$B \rightarrow bbA \mid bb$$

Step 2: Remove unit productions.

$$S \Rightarrow B$$

$$S \rightarrow AB \mid aB \mid bbA \mid bb$$

$$A \rightarrow aab$$

$$B \rightarrow bbA \mid bb$$

Step 3: Remove useless productions.

There are none.

Step 4 and 5: Convert to CNF.

$$S \rightarrow AB \mid T_a B \mid T_b V_1 \mid T_b T_b$$

$$V_1 \rightarrow T_b A$$

$$A \rightarrow T_a V_2$$

$$V_2 \rightarrow T_a T_b$$

$$B \rightarrow T_b V_3 \mid T_b T_b$$

$$V_3 \rightarrow T_b A$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

(b) (5 Points) Find an equivalent grammar to G in Greibach Normal Form (GNF).

Answer:

We can start the conversion from the CNF grammar we found in part (a).

Step 1: Make sure every production's RHS starts with a terminal. If not, substitute the variable until the first character is a terminal (or a variable that encodes a terminal).

$$S \rightarrow T_a V_2 B \mid T_a B \mid T_b V_1 \mid T_b T_b$$

$$V_1 \rightarrow T_b A$$

$$A \rightarrow T_a V_2$$

$$V_2 \rightarrow T_a T_b$$

$$B \rightarrow T_b V_3 \mid T_b T_b$$

$$V_3 \rightarrow T_b A$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

Step 2: Substitute the variables that encode terminals that are at the start of the RHS of the productions.

$$S \rightarrow a V_2 B \mid a B \mid b V_1 \mid b T_b$$

$$V_1 \rightarrow b A$$

$$A \rightarrow a V_2$$

$$V_2 \rightarrow a T_b$$

$$B \rightarrow b V_3 \mid b T_b$$

$$V_3 \rightarrow b A$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

- (c) (5 Points) Suppose we modify the original grammar G as follows: remove the λ -production $A \rightarrow \lambda$ and instead add the unit production $A \rightarrow A$. Let us call the resulting grammar G' . Convert G' into CNF, and simplify, if possible. Also describe in English language $L(G')$.

$$G' =$$

$$S \rightarrow AB \mid aB$$

$$A \rightarrow aab \mid A$$

$$B \rightarrow bbA$$

Step 1: Remove λ -productions.

There are none

Step 2: Remove unit productions.

$A \Rightarrow A$ can be removed immediately.

$$S \rightarrow AB \mid aB$$

$$A \rightarrow aab$$

$$B \rightarrow bbA$$

Step 3: Remove useless productions.

There are none.

Step 4 and 5: Convert to CNF.

$$S \rightarrow AB \mid T_a B$$

$$A \rightarrow T_a V_1$$

$$V_1 \rightarrow T_a T_b$$

$$B \rightarrow T_b V_2$$

$$V_2 \rightarrow T_b A$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

The language $L(G')$ accepts only the following 2 string:

i. $abbbaab$

ii. $aabbbbaab$

It can also be described by the following regular expression:

$$r = (aab + a)bbaab$$