CSE303: Theory of Computation, Fall 2021 **Date:** Nov 19

Homework #4

(Due: Dec 4)

Task 1. [30 Points] Convert from CFG to CNF

Convert the CFG for each of the following languages to CNF. You can use the CFGs directly from HW3 solutions (posted on Blackboard) without deriving them again for this task. Please show the transformation step by step (as shown in the class).

(a) **[10 Points]** $L = \{a^i b^j c^k \mid i \geq 0,\ j \geq 0,\ k = 2i + 3j\}$, $\Sigma = \{a, b, c\}$

CFG:

$$\begin{aligned} S &\rightarrow aScc \mid M \mid \epsilon \\ M &\rightarrow bMccc \mid \epsilon \end{aligned}$$

Step 1: Remove start nonterminal from RHS:

$$\begin{aligned} S_0 &\rightarrow S \\ S &\rightarrow aScc \mid M \mid \epsilon \\ M &\rightarrow bMccc \mid \epsilon \end{aligned}$$

Step 2: Remove any productions that lead to ϵ:

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow aScc \mid acc \mid M \\ M &\rightarrow bMccc \mid bccc \end{aligned}$$

Step 3: Add non-terminals for terminals

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow ASCC \mid ACC \mid M \\ M &\rightarrow BMCCC \mid BCCC \\ A &\rightarrow a \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

Step 4: Drop unit rules

$$\begin{aligned} S_0 &\rightarrow ASCC \mid ACC \mid BMCCC \mid BCCC \mid \epsilon \\ S &\rightarrow ASCC \mid ACC \mid BMCCC \mid BCCC \\ M &\rightarrow BMCCC \mid BCCC \\ A &\rightarrow a \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

Step 5: CNF rules:

$$\begin{aligned}
S_0 &\rightarrow N_2N_1 \mid AN_1 \mid N_4N_3 \mid N_5N_1 \mid \epsilon \\
S &\rightarrow N_2N_1 \mid AN_1 \mid N_4N_3 \mid N_5N_1 \\
M &\rightarrow N_4N_3 \mid N_5N_1 \\
N_1 &\rightarrow CC \\
N_2 &\rightarrow AS \\
N_3 &\rightarrow CN_1 \\
N_4 &\rightarrow BM \\
N_5 &\rightarrow BC \\
A &\rightarrow a \\
B &\rightarrow b \\
C &\rightarrow c
\end{aligned}$$

[**10 Points**] $L = \{a^i b^j c^k \mid i \neq j \textbf{ or } j \neq k\}$, $\Sigma = \{a, b, c\}$

CFG:

$$\begin{aligned}
S &\rightarrow S_1 \mid S_2 \\
S_1 &\rightarrow S_1c \mid M \\
M &\rightarrow aMb \mid D \mid E \\
D &\rightarrow aD \mid a \\
E &\rightarrow bE \mid b \\
S_2 &\rightarrow aS_2 \mid N \\
N &\rightarrow bNc \mid E \mid F \\
D &\rightarrow cF \mid c
\end{aligned}$$

Step 1: Remove start nonterminal from RHS (Already satisfied)

Step 2: Remove any productions that lead to ϵ (Already satisfied):

Step 3: Add non-terminals for terminals

$$\begin{aligned}
S &\rightarrow S_1 \mid S_2 \\
S_1 &\rightarrow S_1C \mid M \\
M &\rightarrow AMB \mid D \mid E \\
D &\rightarrow AD \mid A \\
E &\rightarrow BE \mid B \\
S_2 &\rightarrow AS_2 \mid N \\
N &\rightarrow BNC \mid E \mid F \\
D &\rightarrow CF \mid C \\
A &\rightarrow a \\
B &\rightarrow b \\
C &\rightarrow c
\end{aligned}$$

Step 4: Drop unit rules

$$
\begin{aligned}
S &\rightarrow S_1C \mid AMB \mid AD \mid BE \mid AS_2 \mid BNC \mid CF \mid a \mid b \mid c \\
S_1 &\rightarrow S_1C \mid AMB \mid AD \mid BE \mid a \mid b \\
M &\rightarrow AMB \mid AD \mid BE \mid a \mid b \\
D &\rightarrow AD \mid a \\
E &\rightarrow BE \mid b \\
S_2 &\rightarrow AS_2 \mid BNC \mid BE \mid CF \mid b \mid c \\
N &\rightarrow BNC \mid BE \mid CF \\
D &\rightarrow CF \mid c \\
A &\rightarrow a \\
B &\rightarrow b \\
C &\rightarrow c
\end{aligned}
$$

Step 5: CNF rules:

$$
\begin{aligned}
S &\rightarrow S_1C \mid AK \mid AD \mid BE \mid AS_2 \mid LC \mid CF \mid a \mid b \mid c \\
S_1 &\rightarrow S_1C \mid AK \mid AD \mid BE \mid a \mid b \\
M &\rightarrow AK \mid AD \mid BE \mid a \mid b \\
D &\rightarrow AD \mid a \\
E &\rightarrow BE \mid b \\
S_2 &\rightarrow AS_2 \mid LC \mid BE \mid CF \mid b \mid c \\
N &\rightarrow LC \mid BE \mid CF \\
D &\rightarrow CF \mid c \\
K &\rightarrow MB \\
L &\rightarrow BN \\
A &\rightarrow a \\
B &\rightarrow b \\
C &\rightarrow c
\end{aligned}
$$

[**10 Points**] $L = \{a^i b^j c^k d^l \mid i + j = k + l\}$, $\Sigma = \{a, b, c, d\}$

$$
\begin{aligned}
S &\rightarrow S_1 \mid S_2 \mid e \\
S_1 &\rightarrow G_1D_1 \mid A \\
A &\rightarrow E_1D_1 \mid B_1D_1 \mid B \\
B &\rightarrow F_1C_1 \mid B_1C_1 \\
S_2 &\rightarrow H_1D_1 \mid C \\
C &\rightarrow I_1C_1 \mid A_1C_1 \mid B \\
A_1 &\rightarrow a \\
B_1 &\rightarrow b \\
C_1 &\rightarrow c \\
D_1 &\rightarrow d \\
E_1 &\rightarrow B_1A \\
F_1 &\rightarrow B_1B \\
G_1 &\rightarrow A_1S_1 \\
H_1 &\rightarrow A_1S_2 \\
I_1 &\rightarrow A_1C
\end{aligned}
$$

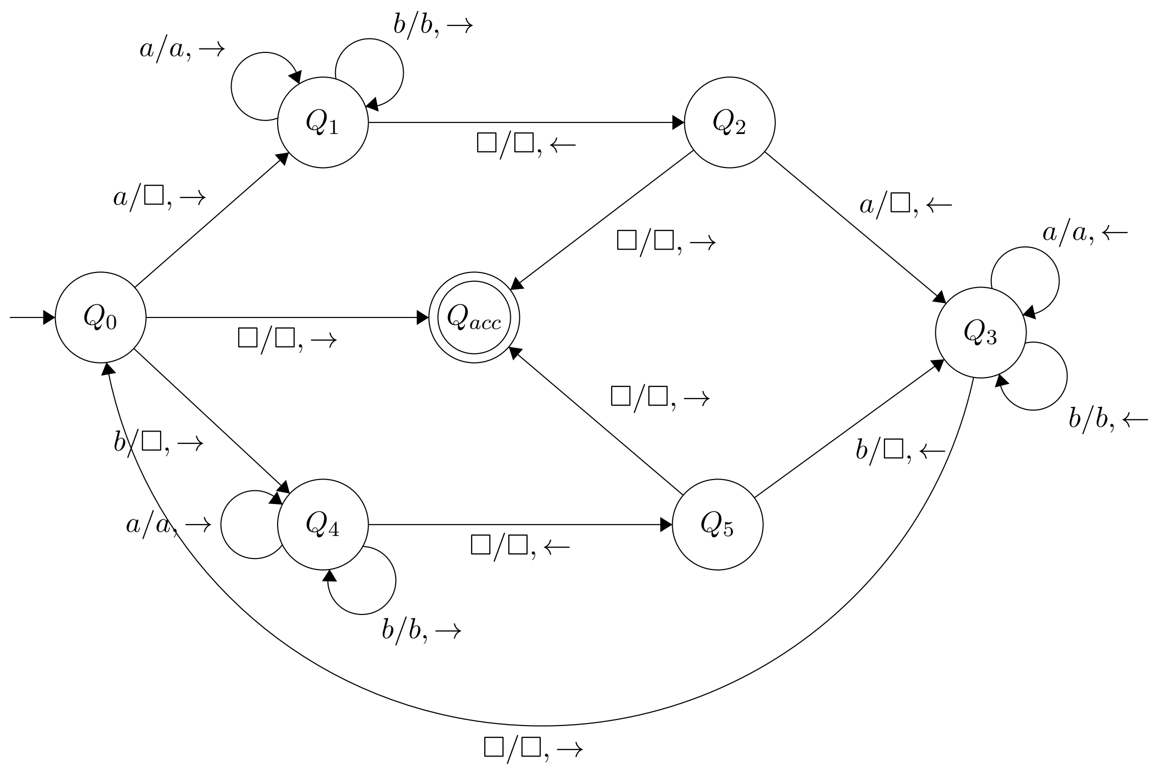
Task 2. [40 Points] Construct TMs

Construct a TM for solving each of the following problems. Write down the TM in the 6-tuple format and draw the state transition diagram showing how it works.

In the solutions, we assume a tape that is infinite on both sides and there are blanks (denoted by squares) on the right and left sides of the strings.

(a) **[10 Points] Check if a string $\in \{a, b\}^*$ (given on the input tape) is a palindrome.**

Solution:
Diagram:



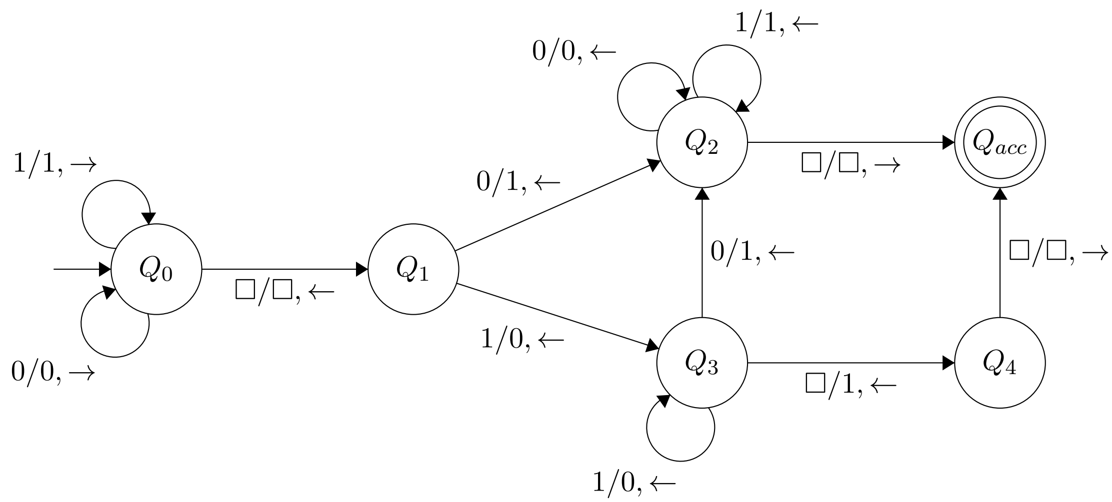
6-Tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$, **where,**

Set of states is	$Q = \{Q_0, Q_1, Q_2, Q_3, Q_4, Q_5, Q_{acc}\}$
Set of strings' symbols is	$\Sigma = \{a, b\}$
Set of tape's symbols is	$\Gamma = \{a, b, \Box\}$
Start state is	$q_0 = Q_0$
Set of halting states is	$H = \{Q_{acc}\}$
Transition function is	

	a	b	$\square$
Q_0	$(Q_1, \square, \rightarrow)$	$(Q_4, \square, \rightarrow)$	$(Q_{acc}, \square, \rightarrow)$
Q_1	$(Q_1, a, \rightarrow)$	$(Q_1, b, \rightarrow)$	$(Q_2, \square, \leftarrow)$
Q_2	$(Q_3, \square, \leftarrow)$	$-$	$(Q_{acc}, \square, \rightarrow)$
Q_3	$(Q_3, a, \leftarrow)$	$(Q_3, b, \leftarrow)$	$(Q_0, \square, \rightarrow)$
Q_4	$(Q_4, a, \rightarrow)$	$(Q_4, b, \rightarrow)$	$(Q_5, \square, \leftarrow)$
Q_5	$-$	$(Q_3, b, \leftarrow)$	$(Q_{acc}, \square, \rightarrow)$

(*b*) **[10 Points] Increment a binary number (given on the input tape) by 1.**

Solution:
Diagram:



6-Tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$**, where,**

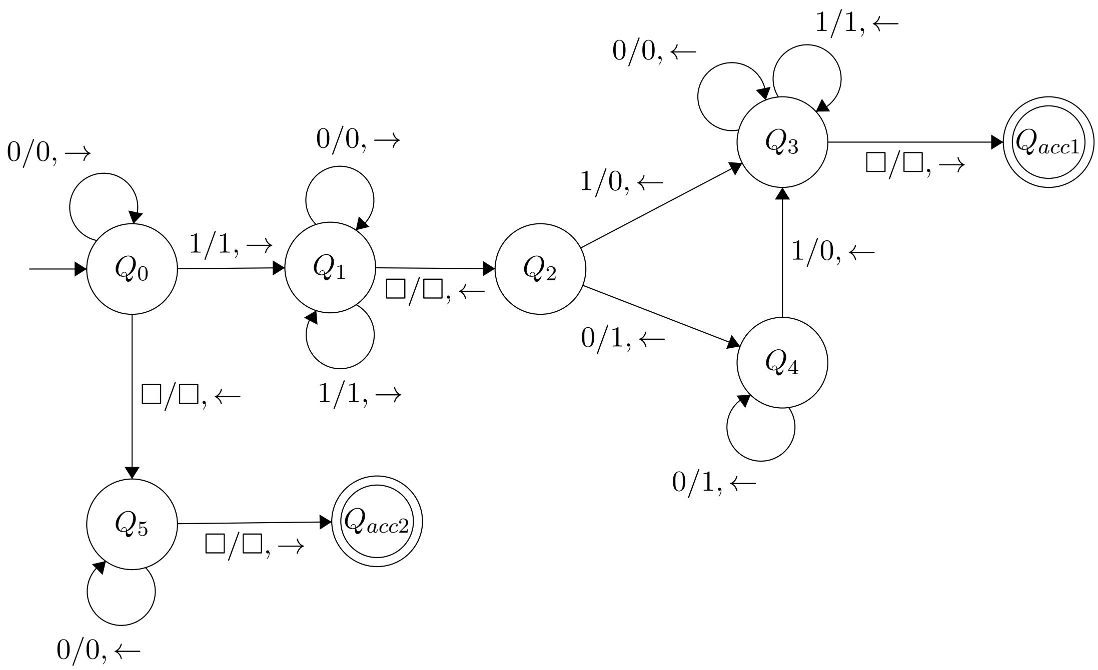
Set of states is	$Q = \{Q_0, Q_1, Q_2, Q_3, Q_4, Q_{acc}\}$
Set of strings' symbols is	$\Sigma = \{0, 1\}$
Set of tape's symbols is	$\Gamma = \{0, 1, \square\}$
Start state is	$q_0 = Q_0$
Set of halting states is	$H = \{Q_{acc}\}$
Transition function is	

δ:

	0	1	$\square$
Q_0	$(Q_0, 0, \rightarrow)$	$(Q_0, 1, \rightarrow)$	$(Q_1, \square, \leftarrow)$
Q_1	$(Q_2, 1, \leftarrow)$	$(Q_3, 0, \leftarrow)$	$-$
Q_2	$(Q_2, 0, \leftarrow)$	$(Q_3, 1, \leftarrow)$	$(Q_{acc}, \square, \rightarrow)$
Q_3	$(Q_2, 1, \leftarrow)$	$(Q_3, 0, \leftarrow)$	$(Q_4, 1, \leftarrow)$
Q_4	$-$	$-$	$(Q_{acc}, \square, \rightarrow)$

(c) **[10 Points] Decrement a binary number (given on the input tape) by 1 (do nothing if the number is 0).**

Solution:
Diagram:



medskip 6-Tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$**, where,**

Set of states is	$Q = \{Q_0, Q_1, Q_2, Q_3, Q_4, Q_5, Q_{acc1}, Q_{acc2}\}$
Set of strings' symbols is	$\Sigma = \{0, 1\}$
Set of tape's symbols is	$\Gamma = \{0, 1, \square\}$
Start state is	$q_0 = Q_0$
Set of halting states is	$H = \{Q_{acc1}, Q_{acc2}\}$
Transition function is	

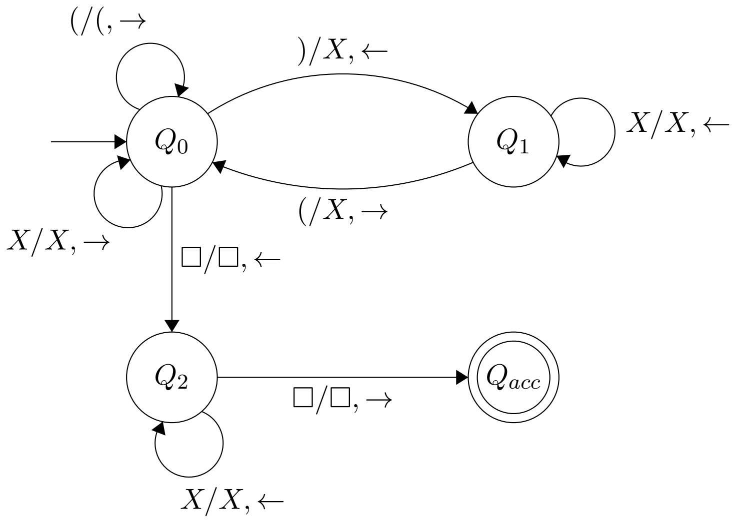
δ**:**

	0	1	$\square$
Q_0	$(Q_0, 0, \rightarrow)$	$(Q_1, 1, \rightarrow)$	$(Q_5, \square, \leftarrow)$
Q_1	$(Q_1, 0, \rightarrow)$	$(Q_1, 1, \rightarrow)$	$(Q_5, \square, \leftarrow)$
Q_2	$(Q_4, 1, \leftarrow)$	$(Q_3, 0, \leftarrow)$	$-$
Q_3	$(Q_3, 0, \leftarrow)$	$(Q_3, 1, \leftarrow)$	$(Q_{acc1}, \square, \rightarrow)$
Q_4	$(Q_4, 1, \leftarrow)$	$(Q_3, 0, \leftarrow)$	$-$
Q_5	$(Q_5, 0, \leftarrow)$	$-$	$(Q_{acc2}, \square, \rightarrow)$

(*d*) **[10 Points] Check if a string of parentheses ($\in \{(,)\}^*$) is balanced.**

Solution:

Diagram:



6-Tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$**, where,**

Set of states is	$Q = \{Q_0, Q_1, Q_2, Q_{acc}\}$
Set of strings' symbols is	$\Sigma = \{\ (\ ,\ )\ \}$
Set of tape's symbols is	$\Gamma = \{\ (\ ,\ )\ ,\ X\ ,\ \square\ \}$
Start state is	$q_0 = Q_0$
Set of halting states is	$H = \{Q_{acc}\}$
Transition function is	

δ**:**

	(	)	X	$\square$
Q_0	$(Q_0, (, \rightarrow)$	$(Q_1, X, \leftarrow)$	$(Q_0, X, \rightarrow)$,	$(Q_2, \square, \leftarrow)$
Q_1	$(Q_0, X, \rightarrow)$	$-$	$(Q_1, X, \leftarrow)$	$-$
Q_2	$-$	$-$	$(Q_2, X, \leftarrow)$	$(Q_{acc}, \square, \rightarrow)$

Task 3. [30 Points] **Compose TMs**

Construct a TM for solving each of the following problems. Give a high level step-by-step description of the algorithm your TM represents and draw a state trnsition diagram showing how it works. You can use TMs you have already constructed under Task 2 (or even Task 3) and/or saw in the class as blackboxes (i.e., subroutines) for solving subproblems.

(a) [**10 Points**] **Add two binary numbers given on the input tape.**

Solution:

Assumption :

we have a # symbol followed binary number, followed by # symbol followed by another binary number

for example #0010#0011

The # symbol is at the start of the tape and also separates the two binary number

Approach :

1.The Idea here is to decrease one from the first number and add one to the second number until the first number becomes zero and then HALT

2. Increment 1 as described in task 2b

3.Decrement 1 as described in task 2c

4. Go back to the first #

(b) [**10 Points**] **Compute the** n**-th Fibonacci number (in binary) given** n **as a binary number on the input tape.**

Solution:

Assumption :

We have a # symbol followed by an input binary number, followed by # symbol, followed by 1, followed by # symbol, followed by 1.

For Example : #0010#1#1

The # symbol is at the start of the tape and also separates the three binary numbers.

Approach :

1. The idea here is to decrement the input number and sum the last two numbers until the input number becomes zero and then HALT.

2. Add 2 Binary Numbers as described in Task 3(a).

3. Decrement as described in Task 2(c).

4. Go back to the first # .

(c) **[10 Points] Multiply two binary numbers given on the input tape.**

Solution:

Assumption :
we have a binary number, followed by # symbol followed by another binary number folowed by #

for example 0010#0011#

Approach :

1.The Idea here is to multiply two numbers x*y by adding x, y times

2. Copy the second number post # symbol used as counter

3. Add the second number to the first using Adder TM created in 3a.

4. Decrement 1 from copied second number using TM created in Task 2c

5. Repeat until the counter is not equal to zero