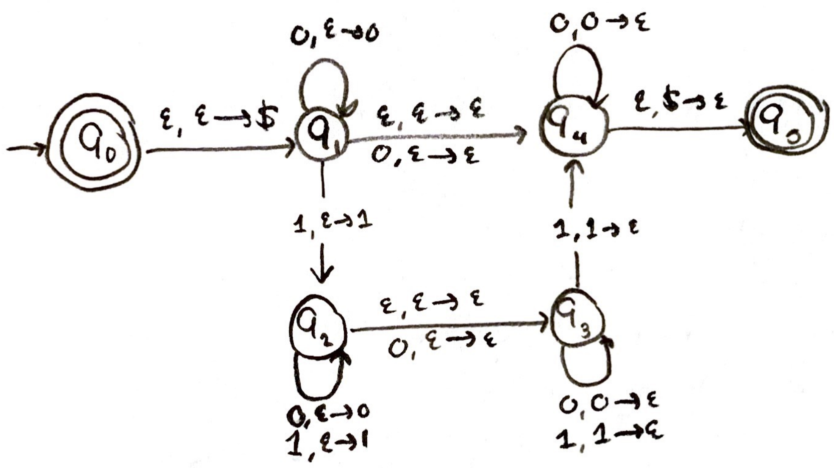
S → AAA|B

Jonathan Tso

HW 3

CS301

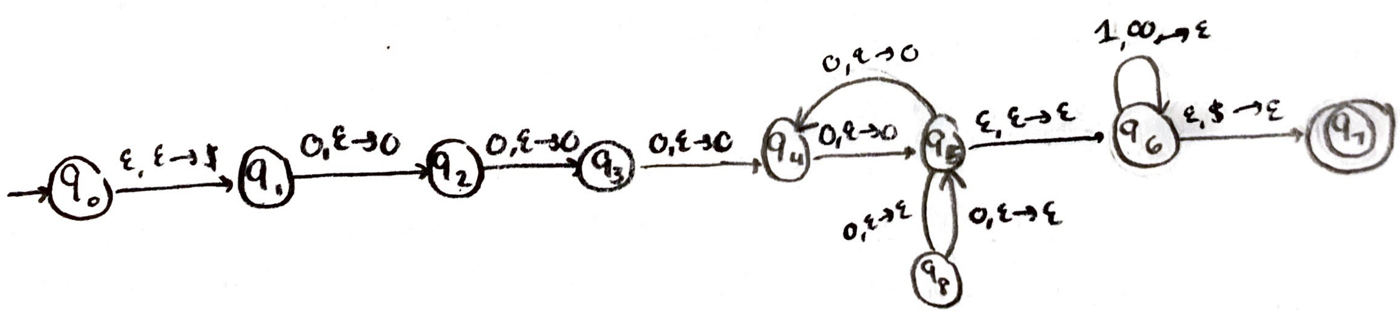
1. a)



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

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

b)



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

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

$A \to aA \mid B$
$B \to \varepsilon$

Problem #2

Start

$S_0 \to S$
$S \to AAA \mid B$
$A \to aA \mid B$
$B \to \varepsilon$

Bin

$S_0 \to S$
$S \to AA_1 \mid B$
$A_1 \to AA$
$A \to aA \mid B$
$B \to \varepsilon$

DEL

$S_0 \to S$
$S \to AA_1 \mid \varepsilon$
$A_1 \to AA$
$A \to aA \mid \varepsilon$
$\cancel{B \to \varepsilon}$

$S_0 \to S \mid \varepsilon$
$S \to AA_1 \mid aA_1 \mid A_1 \mid a$
$A_1 \to AA \mid aA \mid a$
$A \to aA \mid a$

Unit

$S_0 \to AA_1 \mid aA_1 \mid A_1 \mid \varepsilon \mid a$
$S \to AA_1 \mid aA_1 \mid A_1 \mid a$
$A_1 \to AA \mid aA \mid a$
$A \to aA \mid a$

$S_0 \to AA_1 \mid aA \mid Aa \mid a \mid \varepsilon$
$S \to AA_1 \mid aA \mid a \mid A_1$
$A_1 \to AA \mid aA \mid a$
$A \to aA \mid a$

Term

$S_0 \to AA_1 \mid XA \mid AX \mid X \mid \varepsilon$
$S \to AA_1 \mid XA \mid X \mid AA$
$A_1 \to AA \mid XA \mid X$
$A \to XA \mid X$
$X \to a$

3. a. $L = \{0^n \mid n \text{ is a prime}\}$

$S = 0^p$

Consider string $S = \underbrace{0}_{v}0\underbrace{0}_{y}00$ where $n = 5$, that is part of the language.

Case 1:

When we do this, we see that if $v = 0^p$ and $y = 0^p$ that it holds $|vy| > 0$.
However, in no possible way can we make it such that $|vxy| \leq p$ so there is contradiction that a CFG describes it.

Case 2:

If we choose either v or y to be 0^p while the other is ε (and x is ε) then we retain that $|vy| > 0$. We also retain that $|vxy| \leq p$. However, if we pump it such that $y^i = y^2$, we get $S = 000\underline{000}$ and have 6 0's, which is not part of the language. Therefore, L is not context free.

3. b. $L = \{a^n b^n c^i \mid n \leq i \leq 2n\}$

Let us approach this by contradiction. Suppose language L is a CFL. Then, there must be a CFG that describes it such that we can break down an arbitrary s into uv^ixy^iz, where we have $|vy| > 0$ and $|vxy| \leq p$ per dfn. Also, we know i is in the range of n to $2n$.

Let us consider the string $a^p b^p c^p$ where we have $i = n$

We can then look into the following cases

Case 1. v and y only contain one letter, such as $v = a^p, y = b^p$. If we look at this, we can see that $|vy| > 0$ and it is possible that $|vxy| \leq p$ (consider if v and y were just a, b of a string aaabbbcccccc). However, when we pump v and y such that uv^ixy^iz where $i > 2n$ then we force the number of a's and b's to be greater than n, such that the number of c's does not fall within the expected range and a contradiction is formed.

Case 2. v and y only contain one letter, and one of them contains c, such as $v = c^p$. If we were to pump v such that uv^ixy^iz where $xyz = \varepsilon$, then by pumping beyond $i=2$ will cause the number of c's to go out of bounds where $c^p > c^{2n}$ and then cause a contradiction.

Case 3. v and y <u>or</u> v or y contain more than 1 letter. This poses a problem because if we pump any such combination $v = a^ib^i$ or $v = b^ic^i$ then we would lose our ordering where all a's must precede all b's and then come c's. Thus, this would cause a contradiction.

By looking at the possible cases and seeing that for values of i we escape the language, by pumping lemma and contradiction, this language is not a CFL.

4a. Prove that CFL are not closed under set difference

Let us prove by contradiction. First, we assume that language L_1 and language L_2 are both context-free. When we are looking at set difference, we look for $L_1 - L_2$. Now, consider that L_2 is a subset of L_1. $L_1 - L_2$ is then effectively what is not in L_2, and part of L_1. ~~This effectively then breaks down into~~ We then are looking at the intersection of L_1 and $\overline{L_2}$. ~~Provided that the intersection~~ Per course notes, we have already proven that the intersection and complement are both not closed over CFL's. Therefore, we have a contradiction and as a result, the set difference is not closed under set difference.

4.b.

Consider the language of L that can accept the string 100#001.

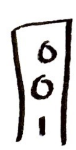
Let us first assume that a PTA and NPDA are equally expressive. Then, any language that can be decided by a PDA must also be decidable by a PTA.

Let L be $\{1^n \# 0^m \mid n=1, m=2\}$

Here, we can see that the above string 001#100 will be a part of the language. Additionally, we can see that a PDA can be generated to decide this. For instance, if we see a 1, put a 1 on the stack, and if we see a 0, put a 0 on the stack. When we reach the #, reverse and see if the stack top matches our value. Effectively, this works for

$X \# X^R$.

The nature of a FIFO PTA prevents this from being decidable. Once we reach the #, we will have the following: But then we read from the 1. Therefore,



we also can't simply say if we see a 0, remove 1 from stack and vice versa because on our stack we have 2 0's and only 1 1 in our remainder of string.

Because of this, the PTA cannot solve any $X \# X^R$ and thus, by contradiction and example, they cannot solve the same languages as an NPDA.

5.

a) DPDA $\subseteq$ 2-stack DPDA

True - the 2DPDA must be at least as expressive as the DPDA, meaning that it will meet the requirement ~~that~~ ~~the~~ ~~~~ ~~the~~ that DPDA is a subset (since ~~since~~ subset can be the whole set).

b) True

Since we can deterministically look at a language, an additional stack will not provide any new benefit so they both are the same expressivity. Therefore it is also a subset.

c) False

The second stack does not help us in determining all possible states an NFA can.

d) True.

DPDA are subsets of NPDA.