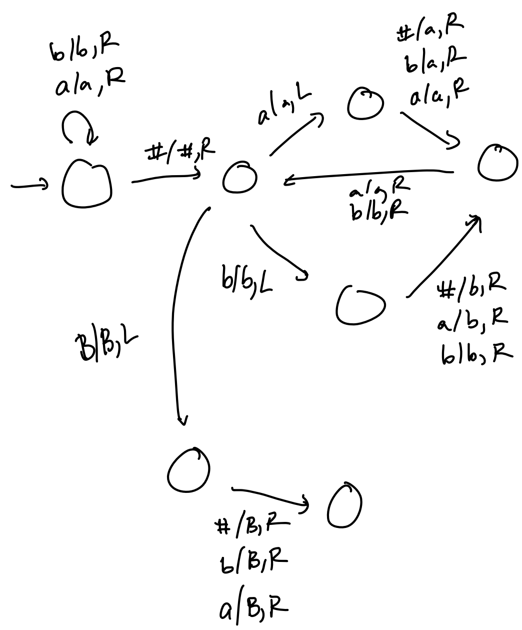
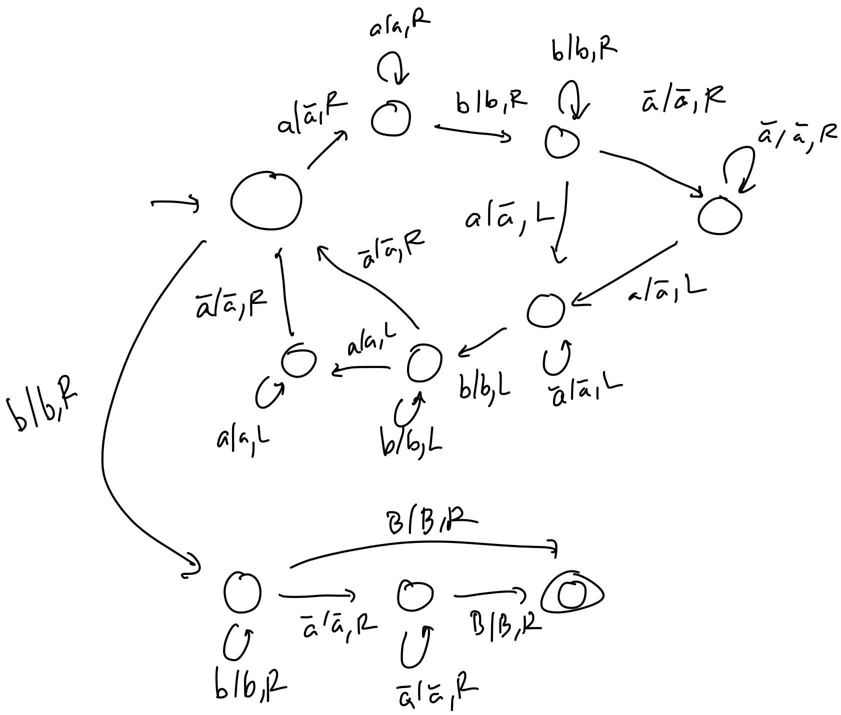
Math 3342 Final Exam

Question 1. (12 points) Please make a Turing Machine which computes the function $f(x\#y) = xy$ where $x, y \in \{a, b\}^*$ and $\#$ is a separator symbol.



Question 2. (12 points) Please make a Turing Machine that accepts the language $\{a^n b^m a^n \mid m > 0\}$.



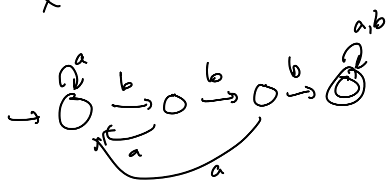
This one was a bit more complicated than I intended!

Question 3. This whole page is about this language:

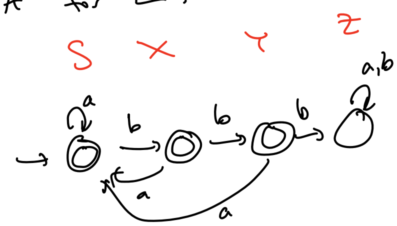
$$L = \{x \in \{a,b\}^* \mid x \text{ never uses 3 consecutive } b\text{'s}\}$$

a) (5 points) Please give a DFA for L. (*Hint:* First make a DFA for the complement, then modify it.)

the compliment:
x uses 3 consec. b's



DFA for L:



b) (5 points) Please give a grammar for L.

Converting my DFA to grammar gives

$S \to aS \mid bX \mid \varepsilon$
$X \to aS \mid bY \mid \varepsilon$
$Y \to aS \mid bZ \mid \varepsilon$
$Z \to aZ \mid bZ$

c) (2 points) For each of the following 4 categories of languages, say if L is in that category or not: regular, context-free, recursive, recursively enumerable. (You don't have to prove anything.)

L is regular (since I made a DFA)

L is all those others too, since "regular" is a subset of all those other categories.

Question 4. (6 points each) Consider a language whose definition has the following format:

$$L = \{a^n b^? c^?\}$$

where the ?s are some integer variable names like n, m, k, etc. The two ? could be the same as each other, or they could be two different symbols.

a) Please give an example of how you could fill in both ?s to make L regular. Say how you would fill in the ?s, and then show that your language is regular.

$L = \{a^n b^m c^k\}$ is regular.

It has a regex $a^* b^* c^*$.

b) Please give an example of how you could fill in both ?s to make L non-regular. Say how you would fill in the ?s, and then show that your language is not regular.

$a^n b^n c^n$ is not regular

Pf let $D_i = \frac{\partial}{\partial a^i} L = \{ a^{n-i} b^n c^n \}$

These are all different, so L is not regular.

Question 5. (5 points each) In each part, give a grammar for the given language:

a) $\{a^n b^k a^{m+n} \mid n, m, k \in \mathbb{N}\}$

$S \to aSa \mid T$

$T \to bT \mid R$

$R \to aR \mid \varepsilon$

b) $\{a^n b^n x x^R \mid x \in \{a,b\}^*, n \in \mathbb{N}\}$

$S \to S_1 S_2$

$S_1 \to a S_1 b \mid \varepsilon$

$S_2 \to a S_2 a \mid b S_2 b \mid \varepsilon$

c) The set of all strings that look like:

$$xxx, xxxxxx, xxx, xx, xxxxx.$$

These are blocks of the letter x separated by commas, with a period at the end. Each block of x's must be nonempty, and there must be at least one block.

$S \to T.$

$T \to B \mid T, B$

$B \to xB \mid x$

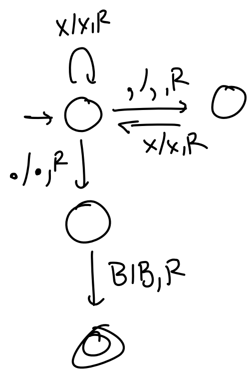
Question 6. (5 points) Only one of the languages above is regular. Decide which one, and give a regular expression for that language.

The last one is regular

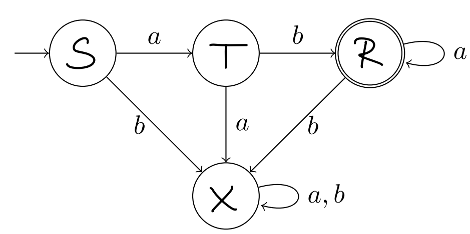
regex: $(xx^*,)^* xx^*.$

Question 7. (8 points) Choose one of the languages from the previous page, and make a Turing Machine which accepts it.

part C is probably easiest:



Question 8. (8 points) Please say what language this DFA accepts, and then prove that all strings in that language are indeed accepted.



$L = \{aba^n\}$

Thm $\delta^*(S, aba^n) = R$ for any n

Pf Induction on n:

base case $n=0$: WTS $\delta^*(S, ab) = R$ which is clear on the diagram

Induction IH: Assume $\delta^*(S, aba^k) = R$, WTS $\delta^*(S, aba^{k+1}) = R$

We have: $\delta^*(S, aba^{k+1}) = \delta^*(\delta^*(S, aba^k), a)$

$\overset{IH}{=} \delta^*(R, a) = R$ Shown!

Question 9. (8 points) Please give a stack machine for the language $\{a^n b a^n b\}$, and show a specific derivation on your stack machine for the string $abab$.

"by hand"

read	pop	push
ε	S	$S_1 S_2$
a	S_1	$S_1 X$
b	S_1	ε
a	X	ε
b	S_2	ε

I'm thinking of a concatenation:

$$(a^n b a^n)(b)$$
$$\uparrow \qquad \uparrow$$
$$S_1 \qquad S_2$$

easier to just convert from CFG

G:
$$S \to S_1 b$$
$$S_1 \to a S_1 a \mid b$$

Stack:

read	pop	push
ε	S	$S_1 b$
ε	S_1	$a S_1 a$
ε	S_1	b
a	a	ε
b	b	ε

Derivation:

$$(abab, S) \mapsto (abab, S_1 b) \mapsto (abab, a S_1 ab)$$
$$\mapsto (bab, S_1 ab) \mapsto (bab, bab)$$
$$\mapsto (ab, ab) \mapsto (b, b) \mapsto (\varepsilon, \varepsilon) \quad \text{accepted!}$$

Question 10. (8 points) My stupid friend once heard something about the Halting Problem, and now he goes around at parties saying stuff like this:

> Because the Halting Problem is unsolvable, it means that whenever you look at a Python program, it's impossible to tell whether or not it will go into an infinite loop.

Briefly explanation why this is not exactly correct, and how you would say it differently so that it is correct.

It's not always impossible to tell if something will halt. In simple cases it will be easy to tell whether or not a given python program will halt.

The unsolvability of HP means it's impossible to have a general procedure that will always identify which programs will halt or not.