Warm Up

Question 1. Give a regular expression for the language $L_1 = \{w \in \{a,b\}^* | \ |w|_b \geq 2\}$.

ANSWER:
Forcing two b's, is as easy as :

$$(a+b)^* b(a+b)^* b(a+b)^*.$$

Question 2. Give a grammar for the language $L_2 = \{a^i bc^j \mid i,j \geq 0\}$.

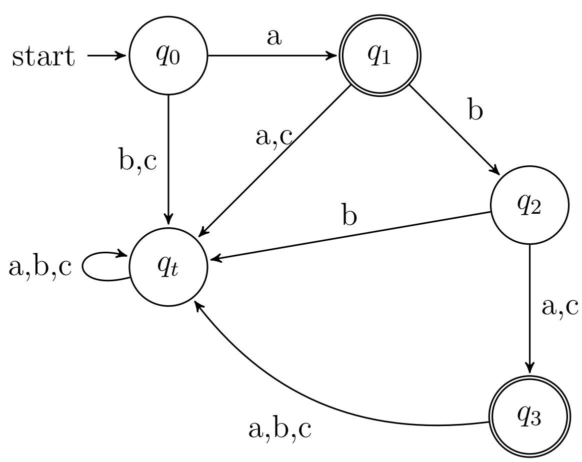
ANSWER:
Given the grammar $G = (V,T,S,P)$ such that all
productions are in the form of either Aa or A for all $A \in V$ and $a \in T$.

So a grammar for L_2 is :

$$S \to Sc|Ab$$
$$A \to Aa|\lambda$$

Question 3. Give a minimal DFA for the language $L_3 = \{a, aba, abc\}$.

ANSWER:



Minimal just means a DFA with the least number of states, since we need a trap state, 3 states to process ab and then 1 more to process aba and abc each, the minimum is 5 states.

Turing Machines

A one-way Turing Machine is defined exactly as the standard Turing Machine except the tape head can move right and stay, but cannot move left.

Question 4. Give a Language L_1 that can be decided by a one-way Turing Machine.

ANSWER:
What does only being able to move right take away from a TM? What other automaton that we have studied have similar properties? Going left essentially lets you keep track of things, if you go right, you have to deal with the input as is and cant save anything because you can only keep going right to the next input character. What about the language $L((a + b)*)$? A one-way TM can definitely accept that.

Question 5. Give a Language L_2 that can not be decided by a one-way Turing Machine.

ANSWER:
Essentially any language that needs to keep track of anything, so counting and harder.

$$L_{cnt} = \{a^i b^i | i \geq 0\}.$$

L_{cnt} definitely can not be accepted by a one-way TM.

Question 6. Give a characterization of languages that can be decided by one way Turing Machines. Justify your answer.

ANSWER:
We touched on this a little bit above. Since going right only is essentially the same as traversing the input with out memory , and stay allows for λ-transitions, A right only TM is essentially an NFA, so the class of languages that one way TM's can decide is **Regular languages**.

Languages

1. Give a language $L_1 \in CF \backslash REG$
   ANSWER:
$$L_1 = \{a^i b^i | i \geq 0\}$$

2. Give a language $L_2 \in DECIDABLE \backslash CF$
   ANSWER:
$$L_2 = \{ww | w \in \{a, b\}^*\}$$

3. Give a language $L_3 \in RECOGNIZABLE \backslash DECIDABLE$
   ANSWER:
$$L_3 = HALT$$

Question 7. Classify each language, chose exactly one. None means the language is not RECOGNIZABLE.

	Question	CF	DEC	RECOG	None
1	$\{\langle M \rangle \mid$ TM M accepts exactly three strings. $\}$				$\ddot\smile$
2	$\{\langle M, w, i \rangle \mid M$ is TM and $w^i \notin L(M)\}$				$\ddot\smile$
3	$\{a^i \mid$ There is some TM that loops forever on $a^i\}$	$\ddot\smile$			
4	$\{a, b\}^* \backslash \{a^i b^i a^i \| i \geq 0\}$	$\ddot\smile$			
5	$\{ww^R w^R w \| w \in \{a, b\}^*\}$		$\ddot\smile$		
6	$\{a^i \mid i$ is not divisible by 7.$\}$	$\ddot\smile$			
7	$\{\langle w_1, w_2 \rangle \mid$ either $w_1 = \lambda$ or $w_2 = \lambda\}$		$\ddot\smile$		
8	$\{\langle M \rangle \| L(M) = \Sigma^*$ and M is a TM. $\}$				$\ddot\smile$
9	$\{\}^*$	$\ddot\smile$			

Reductions

Let $L_a = \{a^i | i \text{ is odd}\}$ and consider the language

$$\textbf{NotOddAs} = \{\langle M\rangle | L(M) \not\subset L_a \text{ and } M \text{ is a TM. }\}$$

Question 8. Show that the language **NotOddAs** is recognizable.

ANSWER:
To show NotOddA's is in $RECOG$ we need to give a TM M' that accepts it. Given some input M, let
M' accept M by:

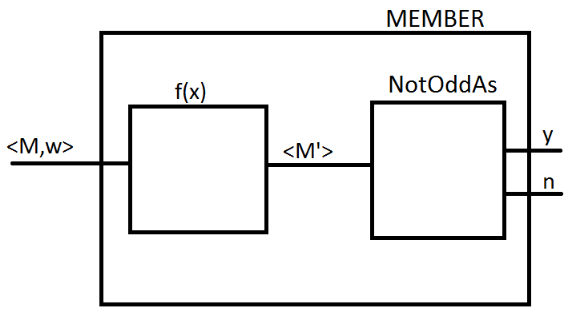
1. Emulate M on all possible inputs w in some order.
2. If M accepts w, check if w has the form a^i for some odd i.
3. Accept if $w \neq a^i$ for some odd i

Proof of correctness:
To prove correctness we assume that some element M is in the language NotOddA's, and prove that M' would accept. We do not care what happens if $M \notin$ NotOddA's, so we do not need to prove anything regarding that case.

Assume $M \in$ NotOddA's. Since $L(M) \not\subset L_a$, some $w \in L(M)$ exists such that $w \neq a^i$ for some odd i. So M' will emulate M on w at some point and M will halt. We can check if a string is of some form such as a^i in finite computable time, so M' would see that w is not of the form a^i and accept M.

Question 9. Show that the language **NotOddAs** is undecidable by reducing any one of the two languages **Member** or **Halt**, to **NotOddAs**.

ANSWER:
To prove NotOddAs is undecidable we reduce a undecidable language to NotOddAs. We will use **Member**. Picture set up is below:



We need to satisfy the two conditions:

1. If $\langle M, w\rangle \in$ **Member** then $\langle M'\rangle \in$ **NotOddA's**.
2. If $\langle M, w\rangle \notin$ **Member** then $\langle M'\rangle \notin$ **NotOddA's**.

with our computable reduction algorithm $f(x)$. Here is the reduction:

$F =$ "On input $< M, w >$
-Given M and w, let M' be the TM that:
-For some input x check if $x = a^i$. (i odd)
-If yes, then M' accepts x.
-If no, then M' simulates M on w and if M accepts w then M' accepts x."

Note: This is pretty weird. Assume that $w \in L(M)$ and try picking different examples of x, like $x = abab, x = abbb, x = aa, x = aaa$ What gets accepted? (you should find that everything does). Now assume that $w \notin L(M)$, what does M' accept? You should find that only a^i for odd i.

Proof of correctness:
(Not necessary for you guys to do but gives insight on why the reduction works)

We want to prove both conditions stated above. So lets prove condition 1, its an if-then statement, so assume the if and prove the then.

Assume $\langle M, w\rangle \in$ **Member**. This means that M accepts w. So the last line in the description of M' will always accept for any x and $L(M') = \Sigma^*$. Since $\Sigma^* \not\subset L_a$, we get $\langle M'\rangle \in$ **NotAllA's**.

No we prove the second condition:

Assume $\langle M, w\rangle \notin$ **Member**. This means that M does not accept w. So the last line in the reduction will never accept anything for M' and $L(M') = \{a^i | i \text{ odd}\}$ by the first if statement in the description of M'. Since $\{a^i | i \text{ odd}\} \subset L_a$, we get $\langle M'\rangle \notin$ **NotAllA's**.

So we have reduced Member to NotOddA's, and since Member is undecidable, NotOddA's is undecidable.