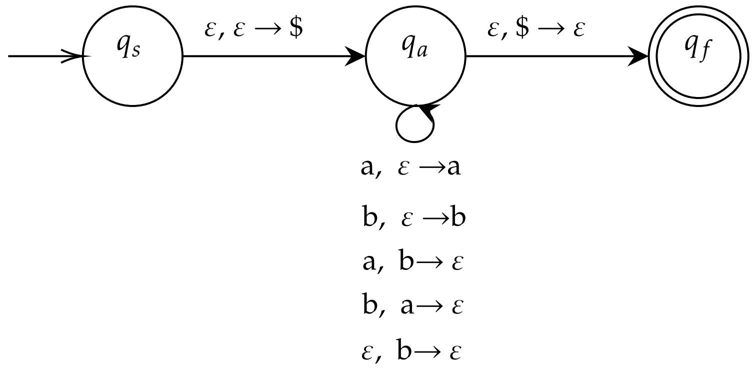
Name: Kevin Zhang

Problem 1.



The components are q_s, q_a, and q_f. q_s is the start state, and simply puts a dollar on the stack, which will get used later to determine when we reached the bottom of the stack. q_a does two things: match a with b as they show up, and discard any extra bs at the end. Once we've reached the bottom of the stack (indicated by the dollar), we can safely transition to q_f and accept. q_a's first four transitions are to deal with scenarios in which the a shows up before the b (or vice versa), or multiple as shows up before the next b (or vice versa). After everything is read, if every b is used to wipe out a a (and vice versa), the stack should end up with only bs, or empty (with dollar). The last transition of q_a is to discard excess bs.

Problem 2.

Left empty – submitted online

Problem 3.

(a) The transitions of M are:

$(s\ 0)\ (s\ 1\ S)$
$(s\ 1)\ (q_{accept}\ 0\ L)$
$(s\ \text{␣})\ (q_{reject}\ 1\ S)$

(b) M accepts all non-empty strings. If the string starts with 1, it replaces the 1 with a 0. Otherwise, it converts 0 to 1, and then back to 0. In both cases, M end in q_{accept} with the head to the left of the first cell. M rejects when the string is empty (and writes a 1, but that's not as important as rejecting.)

(c) M is a decider. M never looks beyond the first character. If the first character is ␣, then it rejects. If the first character is 0, then it flip-flops to 1, back to 0, and then it accepts. If the first character is 1, then it converts it to 0, and accepts. In all cases, M halts.

Problem 4.

1. A and B are DFAs, which makes $L(A)$ and $L(B)$ regular languages
2. $L(A)$ and $L(B)$ have pumping lengths p_A and p_B that satisfy the pumping lemma (they are both regular languages).
3. Pick p to whichever p_A or p_B is greater. The Pumping Lemma states that in a regular language L, there is a pumping length p such that $w \in L$, where $|w| \geq p$ and $w = xyz$, where $y \neq \varepsilon$. Then $w_{pumped} = xy^n z$ where $n \in \mathbb{N}$ and $w_{pumped} \in L$.
4. This means that after length p, all words in a language L can be pumped from some word of length p. This applies to both $L(A)$ and $L(B)$.
5. We picked p to be the greater of p_A and p_B. So after length p, all words in $L(A) \subseteq L(B)$. This means that the words $|w| \leq p$ are the only ones we have to test.
6. DFAs accept only if a word is in a regular language, so, we can test the DFAs for strings of length $k \leq p$.

Problem 5.

(a) Let ODD-SELF = { ⟨ M ⟩ — M is a Turing machine that halts on input ⟨ M ⟩ after an odd number of steps }

Assume ODD-SELF is decidable by Turing Machine T.

We can use diagonalization to construct a D like so.

M_i halts on $\langle M_i \rangle$ after odd steps	M_0	M_1	$\dots$
$\langle M_0 \rangle$	yes	$\dots$	$\dots$
$\langle M_1 \rangle$	$\dots$	no	$\dots$
$\vdots$	$\dots$	$\dots$	$\ddots$
D	no	yes	$\dots$

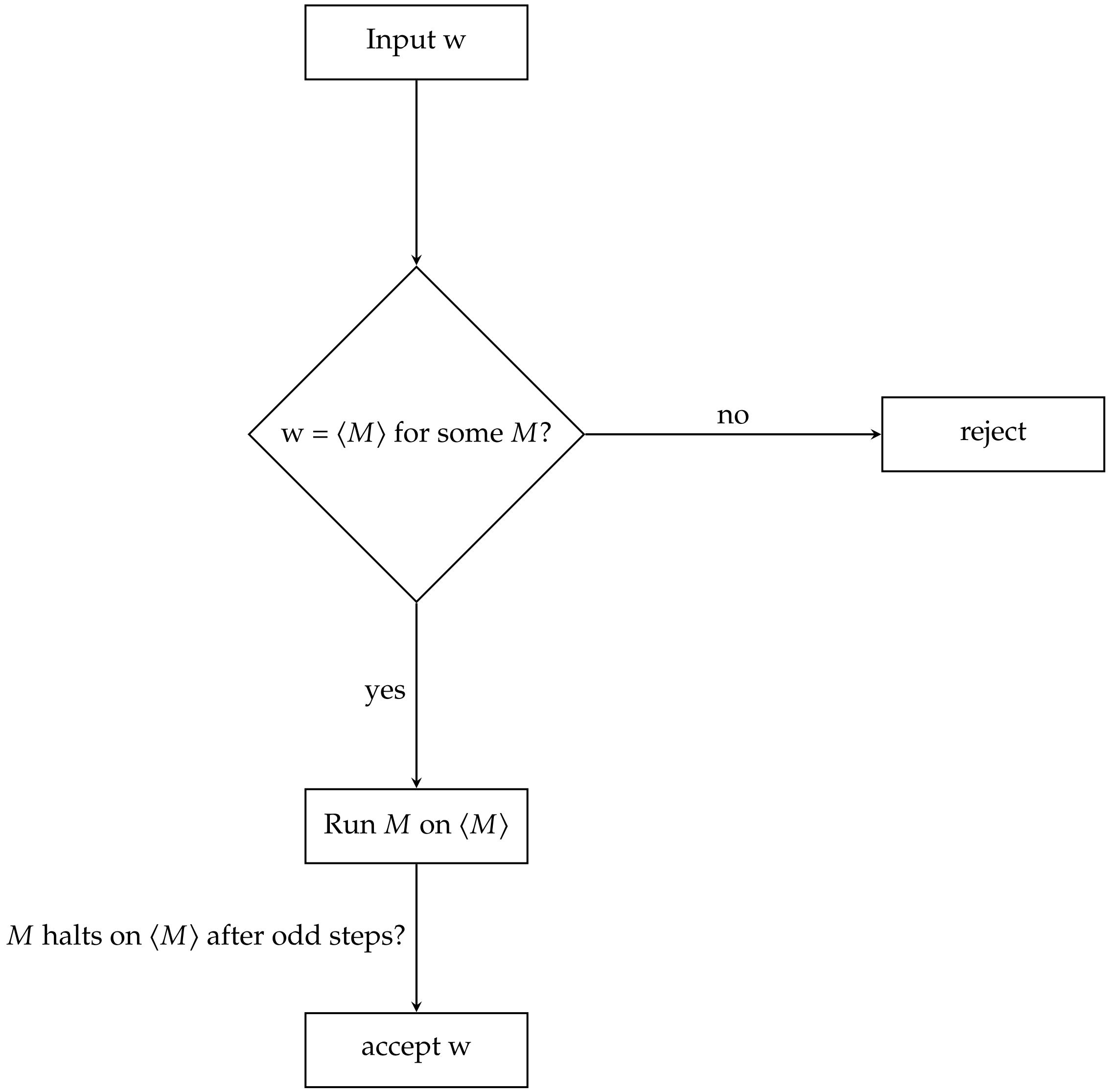
This will create a $D \neq M_1, M_2, M_3, \ldots$. We can program D by doing the following:

(a) On input $w \in \Sigma^*$, if w is not encoding, do anything

(b) If w is encoding, then run T on w, and add one more step (such as a stay-put). This will accept if T reject, reject if T accept.

This creates a contradiction, because if we run D on $\langle D \rangle$, we end up with a situation such that both D accepts and rejects, which means that T is contradictory.

(b) Yes. We can build a recognizer as follows:



(c) The language ODD-HALT is not decidable. Assume (for contradiction), that ODD-HALT is decidable. Then, we can write a machine that takes $\langle M, \langle M \rangle \rangle$, where w is now $\langle M \rangle$. This means that if ODD-HALT is decidable, then ODD-SELF must be decidable. But, we've proven that ODD-SELF is not deciable. Thus, ODD-HALT is not decidable.

Problem 6.

(a) Undecidable. There are some languages such that $1011 \in L(M_1)$. There are some languages such that $1011 \notin L(M_2)$. As such, the property is non-trivial, and is a property of the languages of M_1 and M_2. Therefore, we can use Rice's theorem to show that this is undecidable.

(b) Undecidable. There are some langauges that accept all input, so $L(M_1) = \Sigma^*$. There are also some languages that will reject some input, so $L(M_2) \neq \Sigma^*$. As such, the property is non-trivial, and the property is a property of the languages. Therefore, we can use Rice's theorem to show that this is undecidable.

(c) Undecidable. There are some langauges that reject all input, so $L(M_1) = \varnothing$. There are also some languages that will accept some input, so $L(M_2) \neq \varnothing$. As such, the property is non-trivial, and the property is a property of the languages. Therefore, we can use Rice's theorem to show that this is undecidable.

Problem 7.

Yes. To put it another way, Self-A_{TM} can be written as
$\{ \langle M \rangle — M$ is a TM and $\langle M \rangle \in L(M)\}$.

This is the same as testing for 1011 in problem 6a, and since this property is non-trivial and is property of the language of M, we can use Rice's theorem on this question.