Homework 7 - Computer Science 461 **Name:** ________________________

Due Monday, March 24.

1. Use the algorithm we discussed in class (see the notes from Wed, March 5) to convert the following context-free grammar to Chomsky normal form:

$$
\begin{aligned}
&S \to ASA \mid A \mid \epsilon \\
&A \to aa \mid \epsilon
\end{aligned}
$$

Solution:

Add new start variable. $S_0 \to S \mid \epsilon$.

Remove epsilon rules. $S \to AS \mid SA \mid ASA \mid A$ and $A \to aa$.

Remove unit rules. $S \to AS \mid SA \mid ASA \mid aa$.

Add terminal variables. $A \to T_aT_a$ and $T_a \to a$.

Break up long rules. $S \to AS \mid SA \mid AU \mid aa$ where $U \to SA$.

So the grammar in Chomsky normal form is:

$$
\begin{aligned}
S_0 &\to S \mid \epsilon \\
S &\to AS \mid SA \mid AU \mid T_aT_a \\
A &\to T_aT_a \\
U &\to SA \\
T_a &\to a
\end{aligned}
$$

2. Give a detailed written description (but not a state diagram) of a Turing machine that accepts the following language.

$$L = \{w \in \{a, b\}^* : w \text{ has an equal number of } a\text{'s and } b\text{'s}\}.$$

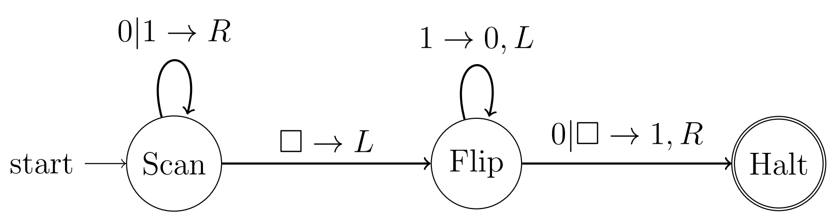
Solution: Option 1. Use a section of the tape to store an encoding of an integer which is initially 0. Move left to right through the input reading either a or b. If you get an a, then replace with an x and then move to the integer and increase it by one. If you get a b, replace it with an x and decrease the integer by 1. When there are no more as and bs, move to the integer and accept if it is zero and reject otherwise.

Option 2. Reading left to right, if you see an a, then replace it with an x, then move right until you see a b. If you find one replace it with an x too, but if you get to a blank first, reject. If you see a b replace it with an x, then more right until you get to an a. Replace it with an x, or reject if you get to a blank first. If you see an x ignore it, and move right.

After this, move all the way to the left and then repeat. If you get to a blank without seeing either an a or b, then accept.

3. A `binary-incrementer` is a function that reads a binary number from a tape, and replaces it with the binary number that is one greater. So 111 becomes 1000, for example. Draw a state diagram for a Turing machine that evaluates the `binary-incrementer` function. Hint: You should only need a few states.

Solution:



4. If you have a Turing machine that computes the `binary-incrementer` function, explain how you could create a Turing machine that reads a string of n 1's, and replaces it with the binary integer that represents n. For example 1111 would become 100 since 100 represents $n = 4$ in binary. You don't need to draw a state diagram, but explain in detail how you would incorporate the `binary-incrementer` machine into your new Turing machine.

Solution: Add a separator symbol and a 0 to left of input. The read the input right to left replacing each read 1 with a blank and then incrementing the binary number to the left of the separator. Repeat until there is nothing right of the separator, then remove the separator.

5. Let Σ be an alphabet, and let $L \subset \Sigma^*$ be a language. If L is decidable, prove that its complement $\overline{L}$ is also decidable.

Solution: Let D be a decider for L. Create a new TM E that accepts w if D rejects w and rejects w when D accepts w. Then E is a decider for $\overline{L}$.

6. Why doesn't the same argument show that the complement of an acceptable language is acceptable?

Solution: If we only have a recognizer R for L, then R may loop forever on some strings. So we wouldn't know whether that string is in L or in $\overline{L}$.