

Homework 5

Problem 1: [10 points]

For each of the following statements, decide whether it is true or false. If true, write a letter "T"; if false, write a letter "F". Either way, also give a brief explanation (1-3 sentences).

Hint: Do not forget to include the explanations...

(a) Let the alphabet $\Sigma = \{a, b\}$. The regular expressions $(a^*b)^*a^*$ and $(a + b)^*$ represent the same language.

Answer: True

Explanation: The regular expression $(a^*b)^*a^*$ generates all strings consisting of as and bs , as a^* represents zero or more as , and b appears in between as needed. Similarly, $(a + b)^*$ represents all strings over $\{a, b\}$. Since both generate the same set of strings, they represent the same language.

(b) The CFG:

$$S \rightarrow aSb \mid bSa \mid a \mid b \mid \epsilon$$

Decides the language $\{w \in \{a, b\}^* \mid w \text{ is a palindrome}\}$

Answer: True

Explanation: This CFG generates palindromes over the alphabet $\Sigma = \{a, b\}$. The rules aSb and bSa ensure that the string begins and ends with matching symbols, while the terminal rules a , b , and ϵ handle single characters and the empty string. Therefore, the CFG correctly defines the language of palindromes.

(c) Let the alphabet $\Sigma = \{a\}$. There exists a DFA with only two states that decide the language $\{a\}$, i.e., the set of a single string " a ". (Note that the problem mentions a DFA, not an NFA.)

Answer: False

Explanation: A DFA for the language $\{a\}$ requires three states: one start state, one accepting state for a single a , and one trap state for inputs longer than or different from a . With only two states, the DFA cannot distinguish between strings of different lengths or handle invalid inputs correctly.

(d) Let the alphabet $\Sigma = \{a\}$. There exists a Turing machine that decides the language $\{a^n \mid n \text{ is a multiple of } 3\}$.

Answer: True

Explanation: A Turing machine can decide this language by counting the number of as modulo 3. It can move through the input tape, marking every third a as it counts. If the tape ends exactly after marking a multiple of three as , it accepts; otherwise, it rejects. This process is within the capability of a Turing machine.

(e) In the execution of a Turing machine, once it enters an accepting state, the execution halts.

Answer: True

Explanation: The Turing machine halts once it reaches an accepting or rejecting state. Although the TM may never halt if for example it has an infinite loop.

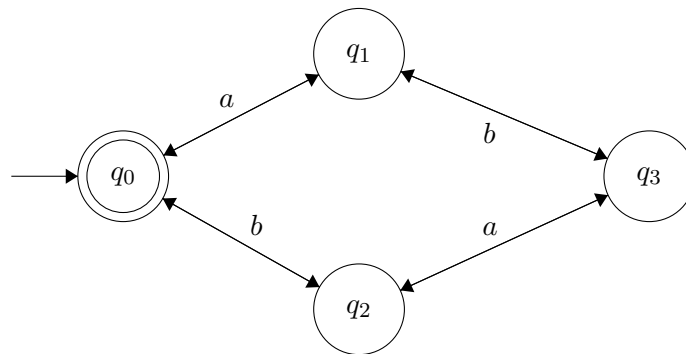
Problem 2: [10 points]

(b): Let the alphabet $\Sigma = \{a, b\}$. Construct a Turing machine that decides the language $L = \{w \mid w \text{ contains an even number of } a\text{'s and an even number of } b\text{'s}\}$. (For example, $aaaaaa$, $abaaab$ and ϵ are in L , while $abbaa$ is not.) Your Turing machine must be in the form of a table similar to the examples in lectures 23 and 24; in particular, do not draw a transition graph.

Hint: L is actually a regular language, and it's not too hard to come up with a DFA that decides it. Then turn this DFA into a Turing machine. The answer should be much simpler than the Turing machines we saw in class. Make sure that the format is correct (whenever in doubt, check lectures 23 and 24).

Solution:

Equivalent DFA:



Turing Machine:

State	Read	Do
even <i>as</i> and <i>bs</i> (starting state)	a	Write X; Tape head →; Go to "odd <i>as</i> even <i>bs</i> "
	b	Write X; Tape head →; Go to "odd <i>bs</i> even <i>as</i> "
	#	Tape head →
	□	go to accept (string finished with even <i>as</i> and <i>bs</i>)
odd <i>as</i> even <i>bs</i>	a	Write X; Tape head →; Go to "even <i>as</i> and <i>bs</i> "
	b	Write X; Tape head →; Go to "odd <i>bs</i> odd <i>as</i> "
	□	go to reject (string finished with uneven <i>as</i>)
odd <i>bs</i> even <i>as</i>	a	Write X; Tape head →; Go to "odd <i>bs</i> odd <i>as</i> "
	b	Write X; Tape head →; Go to "even <i>as</i> and <i>bs</i> "
	□	go to reject (string finished with uneven <i>bs</i>)
odd <i>bs</i> odd <i>as</i>	a	Write X; Tape head →; Go to "odd <i>bs</i> even <i>as</i> "
	b	Write X; Tape head →; Go to "odd <i>as</i> even <i>bs</i> "
	□	go to reject (string finished with uneven <i>as</i> and <i>bs</i>)
Accept		
Reject		

Explanation:

This Turing machine is very simple and operates in a single pass of the string and can also be simply demonstrated with a DFA which I gave above. It does this by reading each letter from the string, crossing it out, and transitioning to the corresponding state. It continues until the string is empty, at which point the final state determines whether to accept or reject. This simplicity mirrors the behavior of the DFA and avoids extensive tape searching.