Tutorial 11

Formal Language and Automata Theory

March 30, 2023

Question 1

Write the algorithm of a Copy Machine. A copy Machine is a Turing Machine which, when given an input $\#w\#$, where w is a string of a's and b's, makes a copy of w and halts with $\#w\#w\#$.

Question 1

Write the algorithm of a Copy Machine. A copy Machine is a Turing Machine which, when given an input $\#w\#$, where w is a string of a's and b's, makes a copy of w and halts with $\#w\#w\#$.

HINT 1: For copying scan each character one by one and print.

Question 1

Write the algorithm of a Copy Machine. A copy Machine is a Turing Machine which, when given an input $\#w\#$, where w is a string of a's and b's, makes a copy of w and halts with $\#w\#w\#$.

HINT 2: What if each character of the string is printed after encountering the blank symbol?

Question 1

Write the algorithm of a Copy Machine. A copy Machine is a Turing Machine which, when given an input $\#w\#$, where w is a string of a's and b's, makes a copy of w and halts with $\#w\#w\#$.

SOLUTION: $M =$ On input $\#w\#$

1. If the first symbol is an a/b change that into an X/Y
2. Moves right until the first blank symbol
3. When it sees the first blank symbol, it prints a a/b depending on what it read earlier
4. Moves left until it finds an $X/Y$and the process repeats.
5. When no more a or b remains to be copied, the machine prints a $\#$ at the end.
6. Converts all the Xs and Ys into as and bs respectively and halts when it sees the leftmost $\#$ symbol.

Question 2

Design a Turing machine to find 2's complement of a binary number

Question 2

Design a Turing machine to find 2's complement of a binary number

HINT 1: take any binary number (e.g., 1011000) and it's 2's complement. What do you observe?

Question 2

Design a Turing machine to find 2's complement of a binary number

HINT 2: While moving from right to left, don't update anything until you see the first 1. After that reverse the numbers

Question 2

Design a Turing machine to find 2's complement of a binary number
SOLUTION:

1. Here, we need to start from the rightmost end.
2. We will move the R/W head all the way to the right, skipping all the 0s and 1s.
3. On moving right, when we end up on the blank B, then move one step to the left.
4. Then we will move the R/W head to the left skipping all the 0s.
5. When it reaches a 1, then we will skip this 1, and then move one step left.
6. From now, we will make all the 1s to 0s and 0s to 1s.
7. We will repeat it all the way to the left of the string.
8. On moving all the way to the left, we will reach the blank B,
9. then move one step right, such that the R/W head will point to the first character, then we will stop.

Question 3

Show that A Turing machine with doubly infinite tape recognize the class of Turing-recognizable languages

Question 3

Show that A Turing machine with doubly infinite tape recognize the class of Turing-recognizable languages

HINT 1: Can we simulate an ordinary TM with the TM with doubly infinite tape and vice versa?

Question 3

Show that A Turing machine with doubly infinite tape recognize the class of Turing-recognizable languages

HINT 1: Can we simulate an ordinary TM with the TM with doubly infinite tape and vice versa?
HINT 2: Can a 2-tape TM be utilized?

Question 3

Show that A Turing machine with doubly infinite tape recognize the class of Turing-recognizable languages

SOLUTION:

- A TM with doubly infinite tape can simulate an ordinary TM. It marks the left-hand end of the input to detect and prevent the head from moving off of that end.
- To simulate the doubly infinite tape TM by an ordinary TM, we show how to simulate it with a 2-tape TM, which was already shown to be equivalent in power to an ordinary TM. The first tape of the 2-tape TM is written with the input string, and the second tape is blank. We cut the tape of the doubly infinite tape TM into two parts, at the starting cell of the input string. The portion with the input string and all the blank spaces to its right appears on the first tape of the 2-tape TM. The portion to the left of the input string appears on the second tape, in reverse order.

Question 4

Show that Turing machines with left reset recognize the class of Turing-recognizable languages

Question 4

Show that Turing machines with left reset recognize the class of Turing-recognizable languages

HINT 1: Can we simulate the Left Reset TM with an ordinary TM and vice versa?

Question 4

Show that Turing machines with left reset recognize the class of Turing-recognizable languages

HINT 1: Can we simulate the Left Reset TM with an ordinary TM and vice versa?

HINT 2: Can we right shift the whole tape, except the current position?

Question 4

Show that Turing machines with left reset recognize the class of Turing-recognizable languages

SOLUTION:

- Replace the current symbol by the symbol specified by TM with a dot
- Do a left-reset, and moves the contents of each square one position to the right, except for the dotted square, i.e., the “dot” remains in the same tape position
- Once all the squares have been shifted one position to the right, do a second left-reset
- Travel with right-transitions until it reaches the dotted square