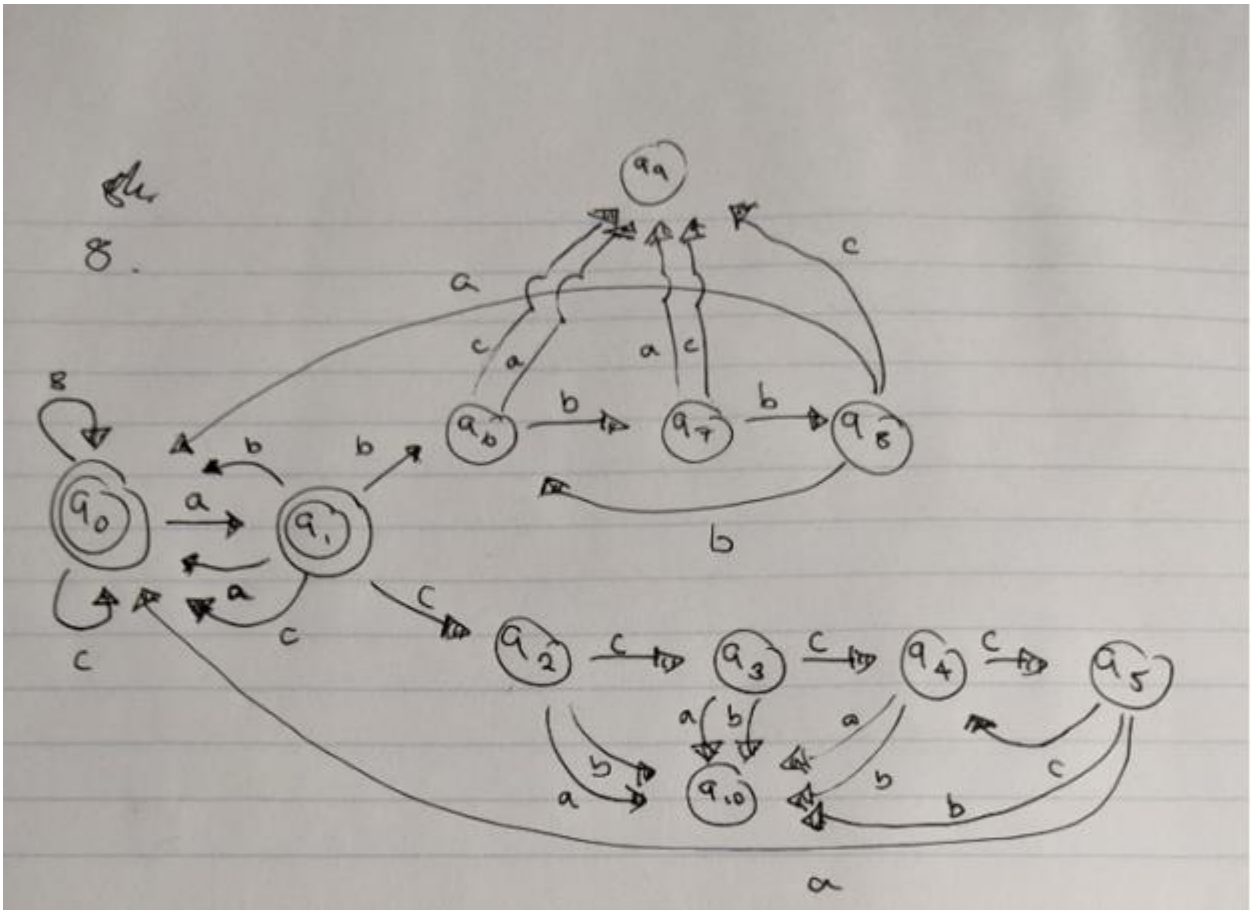
1. **Define**
    a. Unsolvable problem – A problem is considered unsolvable if it cannot by solved by a Turing machine. For example, a problem halting
    b. Decision problem – A decision problem is a problem that you can answer either YES or NO to
2. A problem is intractable when there is no efficiency algorithm to solve them, the algorithms will always not be efficient, for example, The Tower of Hanoi as it will always have a worst-case running time that is at least $O(2^n-1)$. The gist of if it is that it is said to be intractable if the problems time complexity is worse than polynomial time
3. A Turing machine is a model of computation that defines an abstract machine which works with a strip of tape according the different states. Said tape is of infinite length, which is divided into cells, similar to that of a linked list where it reads a symbol and replaces it with another. It is also important to study the models of computation to know how we come to an output given an input.
4. An FSM main characteristic is that it has a finite amount of memory and cannot obtain more memory whereas the Pushdown Automaton has a stack memory that is unbounded. The similarity is that they both use a set of states to reach an output given an input. Finite automata basically allows for regular languages whereas PDA's can be constructed for CFG
5. This is partially true as taking the cardinality of S1 and S2 gives the total number of elements within each set. However, there may be scenario such that S1 Union S2 have similar elements so | S1 + S2 | May not be the same as |S1| + |S2|. So False, they must be compared first to look for elements that may appear in both sets S1 and S2

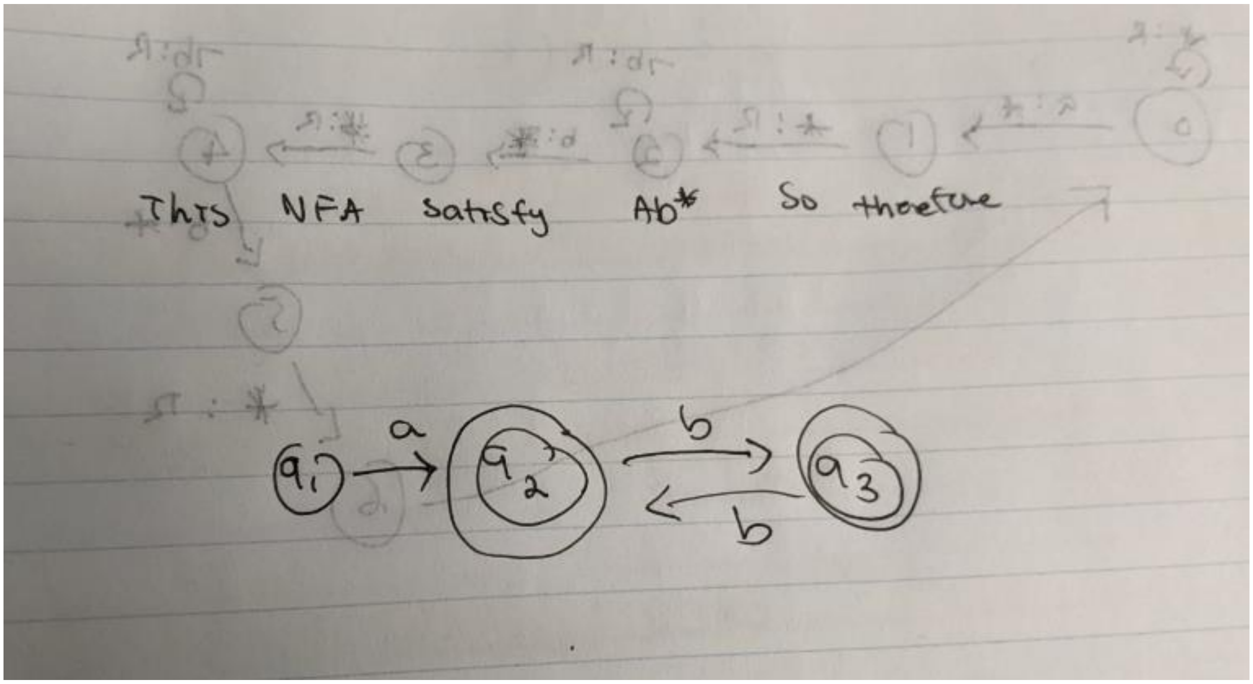
6. The set of natural numbers is infinite for example {2,4,6,8….} where each number increases by two. All sets that can be put into a bijective relation to the natural number is said to be infinite so FALSE
7. Give examples
    a. {0,1} This is countable and is finite, there is a very visible end that we can see here therefore it is countable finite
    b. {…,-2,-1,0,1,2, ...} the set of all real numbers is not countable as there is just an infinite amount of numbers and no end
    c. {0,-2,2,-4,4…} one-to-one with set of natural numbers therefore it is countable and infinite. It is only stated as countable because it is one-to-one

8.



String examples : abbbbbbbbbab, acccccccabc

9.



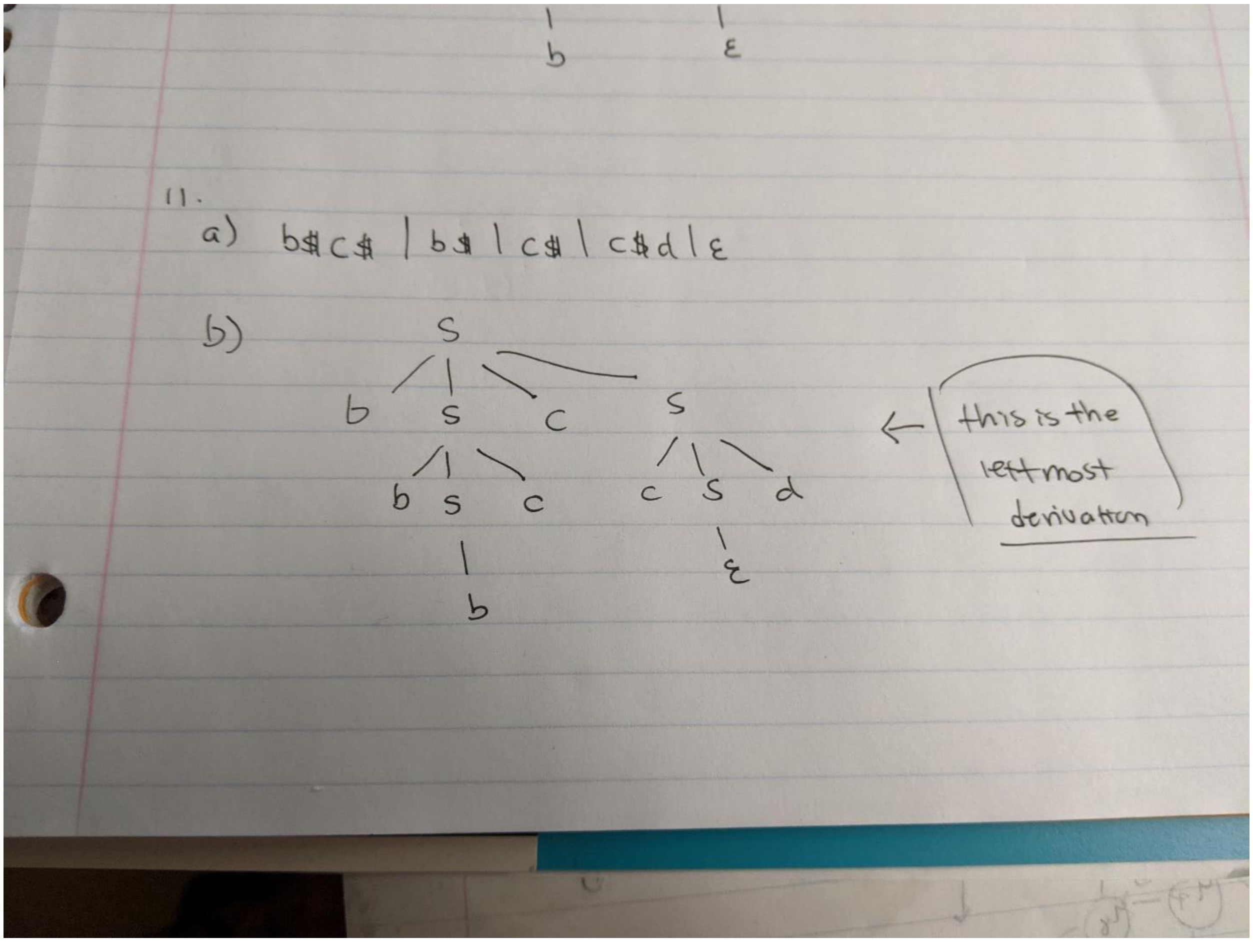
10.

S -> xyxB

B -> yyxB | C

C -> x

11.



12. The way to multiply two binary numbers is to follow the basic principles of multiplication and shifting digits but we are only working with 0 and 1's so we have to follow this truth table

0 * 0 = 0

0 * 1 = 0

1 * 0 = 0

1 * 1 = 1

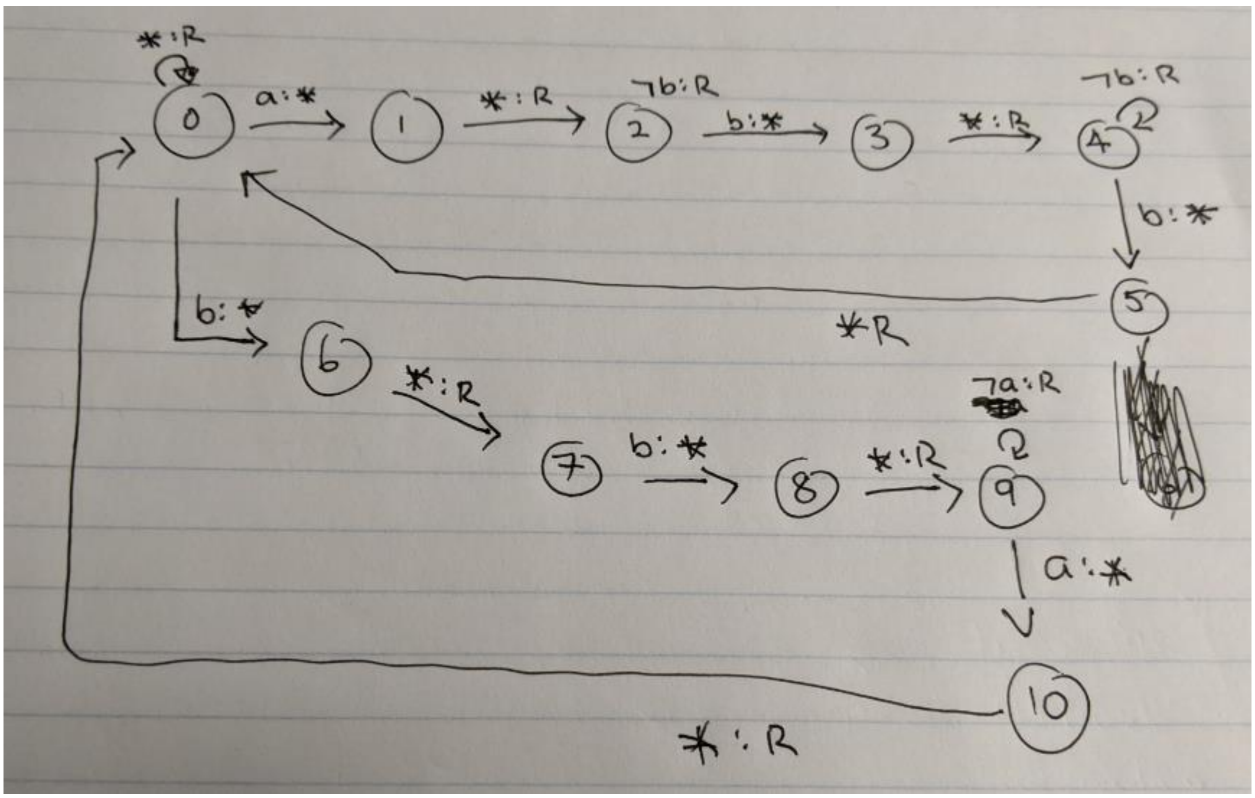
Knowing this, how a Turing machine would work is it would read two binary values in question, and run the digits through the truth table from above and return the corresponding value while following the basic principles of multiplication of regular numbers

13. Examples

    a. {0,1,2} where f(n) = n mod 3

    b. f(x) = x^2 where x is greater than 0

14. Turing Machine



b.

i. w = abb

0 ->1 read “a” write * | 1->2 read “*” go RIGHT | 2 -> 3 if not “b” then keep going RIGHT otherwise if “b” write * | 3 -> 4 read * go RIGHT| 4 ->

5 if not “b” then keep going RIGHT otherwise if “b” write * | 5 -> 0 read * go RIGHT|

ii. w = baa

0 -> 6 read “b” write *| 6 -> 7 read “*” go RIGHT| 7 -> 8 read “b” go RIGHT | 8->9 read “*” go RIGHT| if not “a” go RIGHT otherwise

Read “a” write *| 9->10 read “*” go RIGHT

c. The time needed this will be O(3n) such that for every a there is 2 b such and the space complexity will be O(6n) because you will write a * for every letter you read

15. 3 problems

a. The traveling salesperson problem is a list of cities, you are to find the shortest and most efficient route a person must take given a list of destinations that he/she musts go through. The reason why this problem is special is because there is no quick solution, and the complexity of the problem increases every time you add more destinations to the list of destinations. This can be seen as an NP-complete(np hard) problem.
b. The satisfiability problem is the question if there exists a truth assignment that makes the function true, like NOT NOT b. The uniqueness of this problem is that it is NP complete
c. The similarity of each problem is that they are both NP-complete such that there is no efficient method of solving them