Answer to Scientific Theory in Informatics Computation: Home Exercises 1

Welemhret Welay Baraki

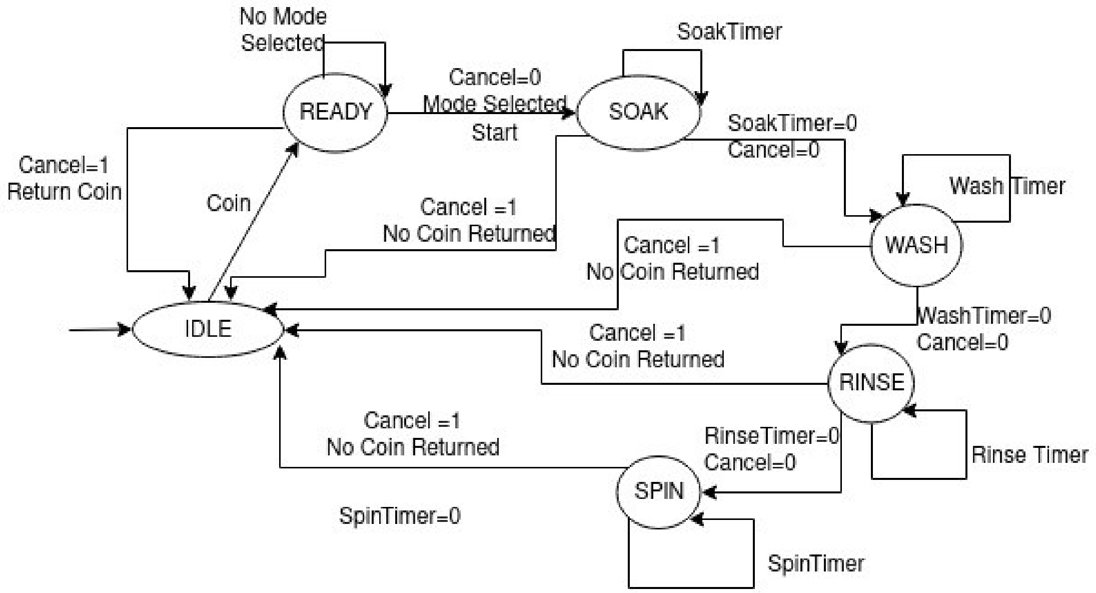
University of Skovde -Data Science(Masters Program)

20 October 2020

1. Design either a DFA or NDFA to control a washing machine. You may include failure states if you wish.

   Answer:

   The design to Control the washing machine is illustrated as follows:



2. In the slides is a Turing machine for recognising a language B = {w#w | w ∈ {0, 1}* } Show that this machine will recognise 10#10.

   Answer:

   Show the steps Either Diagrammatically or Description

   10#10

1	0	#	1	0	
x	0	#	1	0	
x	0	#	x	0	
x	x	#	x	0	
x	x	#	x	x	**Accept**

The word 10#10 is a valid word and the machine accepts.

3. Consider the following Cod: Find Time Complexity and Order of Time Complexity of the following algorithm.

   Answer:

```
int i, j, n;        --------------------3
n = 100;        --------------------1
while (i < n) {------------------n
  b[i] = 0;-----------------------n-1
  i++;  -------------------------n-1
```

   I am assuming that i starts from 0, there is no explicit declaration and Initialization on that.

   $T(n) = n-1+n-1 +n +1 +3 =3n+2$

   Since $n=100$, so $T(n) =3*100+2 =302$

   $T(n) =302$

   Since the n is not an arbitrary size so that the order of Time complexity of the algorithm is $O(1)$.

4. Recursive Function: Time Complexity and the Order of Time Complexity of the Algorithm.

   Answer:

```
void f(int n) {  ---------------------------------1
    if (n>0) { ------------------------------------n+1
        cout << n << " ";------------------------n
        f(n-1);------------------------------------n
    }
}
```

   Time Complexity of the above algorithm is:

   $T(n) =1+ n+1+ n+ n =3n+2$

   Order of Time Complexity of the above algorithm is:

   Since the Time complexity is of Linear running time i.e $T(n) =3n+2$, the Order of Time Complexity of the algorithm is $O(n)$/Linear Running time.

5. The traveling sales person problem is a classic optimization problem. With respect to this problem please discuss the complexity classes P, NP, NP-Complete and NP-Hard.

   Answer:

   P: the set of all decision problems solvable in polynomial time on a deterministic Turing machine, i.e., on a computer as we know it. Traveling sales man problem can't be a P problem, since the destination and the possible trip is not easily verified.

   NP: is the set of all decision problems solvable in polynomial time on a nondeterministic Turing machine. In Traveling Sales Person problem Selecting the optimal next leg in the Traveling salesman's trip is an NP problem. For all decision points in the Traveling Salesman problem the person can, in constant time, make a perfect guess on the best choice.

   NP-Hard: B is NP-Hard if it satisfies: B is in NP and Every A in NP is not reducible polynomial time to B. Traveling Salesman Problem is

   NP-Complete: B is NP-complete if it satisfies: B is in NP/NP Hard and Every A in NP is polynomial time reducible to B. The salesman has to visit each one of the cities starting from a certain one and returning to the same city. The challenge of the problem is that the traveling salesman wants to minimize the total length of the trip.

6.

   a. As examples of D&Q algorithms, what is the fundamental difference between merge sort and A?

   b. As sorting methods, what are the similarities and differences between merge sort and A?

   Answer:

The difference between the two algorithms is in the second(A) algorithm the pivot is selected and removed so there is an increase in memory and time complexity. In the second it only selects a pivot and rearranges elements which is less time and space complexity than the algorithm A.

A is not a sorting algorithm and hence is not a recursive so that It will not fully sorted the algorithm. The merge sort can fully sort the algorithm. The two algorithms are similar they both pick a pivot and rearange elements accordingly. Therefore, both algorithms use divide and conquer Algoritm design method.

7. Recursive algorithm of Equation: $fac(n)=\begin{cases} 1, n=0 \\ \dfrac{fac(n+1)}{n+1}, n\geq 1 \end{cases}$

Answer

```
factNonNegative(n)
{
    If (n==0) 1
    else (n)*factNonNegative (n-1)
}
```

8. Suppose that we have found that the word "Rolex" occurs in 250 of 2000 messages known to be spam and in 5 of 1000 messages known not to be spam. Estimate the probability that an incoming message containing the word "Rolex" is spam, assuming that it is equally likely that an incoming message is spam or not spam. If our threshold for rejecting a message as spam is 0.9, will we reject such messages?

Answer:

$$P(\text{Spam}/\text{Rolex}) = \frac{P(\text{Rolex } |S)}{P(\text{Rolex}/ \text{ Spam}) + p(\text{Rolex}|□\text{Spam})}$$

$$= \frac{0.125}{0.125 + 0.005}$$

$$= 0.962$$

9. In a communication channel a zero or a one is transmitted. The probability that a zero is transmitted is 0.1. Due to noise in the channel, a zero can be received as one with probability 0.01, and a one can be received as a zero with probability 0.05. If you receive a zero, what is the probability that a zero was transmitted? If you receive a one what is the probability that a one was transmitted?

Answer

t0: Transmitted bit is 0

t1: Transmitted bit is 1

r0: Received bit is 0

r1: Received bit is 1

Given

$P(t_0) = 0.1$ $\qquad$ $P(r_0 / t_1) + P(r_1 / t_1) = 1$

$P(t_1) = 0.9$ $\qquad$ $P(r_1 / t_1) = 1 - P(r_0 / t_1) = 1 – 0.01 = 0.99$

$P(r_0 / t_1) = 0.01$

$P(r_1 / t_0) = 0.05$

$$P(t_0 / r_0) = \frac{P(r_0 / t_0) P(t_0)}{P(r_0)}$$

$= 0.95(0.1) /0.104$

$= 0.913$

$$P(t_1/ r_1) = \frac{P(r_1 / t_1) P(t_1)}{P(r_1)}$$

$= 0.99(0.9) /0.896$

$= 0.994$

$P(r_1 / t_0) + P(r_0 / t_0) = 1$

$P(r_0 / t_0) = 1 - P(r_1 / t_0) = 1 - 0.05 = 0.95$

Total probability rule ________

$P(r_0) = P(r_0 / t_0) P(t_0) + P(r_0 / t_1) P(t_1)$

$= 0.95(0.1) + 0.01(0.9) = 0.104$

$= 0.104$

$P(r_1) = P(r_1 / t_1) P(t_1) + P(r_1 / t_0) P(t_0)$

$= 0.99(0.9) + 0.05(0.1)$

$= 0.891 + 0.005 = 0.896$