FOCS - Recitation 9

May 6, 2018

Contents

1 A function maps 1, 2, 3, 4 to a, b, c, d, e. How many such functions are:
  1.1 a Injective?
  1.2 b Surjective?
  1.3 c Bijective?

2 Problem 22.15

3 Problem 23.11
  3.1 a For $L = \{1, 10\}$, give the strings of length at most 6 in $L^*$.
  3.2 For $L$ in (a), prove that $1110111 \in L^*$ and that $11100111 \notin L^*$.

4 Problem 24.5
  4.1 b The DFA accepts quarters and dispenses a coke each time the machines gets 3 quarters.
  4.2 c The DFA accepts quarters and dimes and dispenses candy when the balance is at least 50c.
  4.3 c) The DFA accepts the language $L = \{00010, 10111\}$.
  4.4 d) The DFA accepts the language $L = \{\text{strings containing 101 as a substring}\}$.

5 Problem 25.5
  5.1 c) Strings of the form $0^{3n}$ for $n \geq 0$
  5.2 d) Strings of the form $0^n1^{3n+1}$ for $n \geq 0$

6 Problem 26.3(i)
1 A function maps 1, 2, 3, 4 to a, b, c, d, e. How many such functions are:

1.1 a Injective?

Solution:
An injective function(one-one) is formed if every element in the domain maps to a unique element in the codomain. We have 4 elements in the domain and 5 elements in the codomain. So if for the first element we have 5 choices, second one has 4, 3 one has 3 and the 4th one has 2. So we have $5 \times 4 \times 3 \times 2 = 120$ such functions.

1.2 b Surjective?

Solution:
This cannot exist as by the definition, we need that every element in the codomain needs to be mapped. So we need $|\text{codomain}| \leq |\text{domain}|$. But in our case, we will always have one element in the codomain $\text{(B)} = |5|$ unmapped. Hence we have zero surjective functions possible.

1.3 c Bijective?

Solution:
Since the number of elements in the domain and codomain are not equal, we cannot form a mapping where all elements in the codomain are mapped. So we have zero bijective functions.

2 Problem 22.15

Let $B$ be the finite binary strings, and $T$ the finite ternary strings (strings whose characters are 0, 1, or 2). Show that $|B| = |T|$. 

Solution:
Firstly to prove this we essentially need to show that there is a bijective mapping between sets $B$ and $T$. We notice that the set of $B$ and $T$ are countably infinite as their elements can be listed, we now need to show that there is a unique mapping between every element in $B$ and $T$. Strings in $T$ are composed of elements from the alphabet set $\sigma = \{0, 1, 2\}$. Each of these elements can be represented by binary strings $0 \mapsto 00$, $1 \mapsto 01$, $2 \mapsto 10$. Any linear combination of these values can also hence be represented uniquely by a binary combination, hence we can say that there exists a unique mapping between every element in $T$ with $B$. And since we have already established that $B$ and $T$ are countably infinite. We can say that there is a bijective mapping, hence $|B| = |T|$

3 Problem 23.11

For a language $L$, recall that $L^*$ contains all strings obtained by concatenating zero or more strings in $L$ (concatenation of zero strings is $\epsilon$ and concatenation of one string is the string itself); $\sigma^*$ is the set of all binary strings.

3.1 a For $L = \{1, 10\}$, give the strings of length at most 6 in $L^*$

Solution:
$L^*$ is formed by the concatenation of $L$ with itself zero or more times, and it is interesting to note that $n = 0$ as well, which indicates that the empty string $\epsilon$ is also valid. Let us now enumerate the possible strings given the language $L = \{1, 10\}$

$n = 0 - \epsilon$
$n = 1 - 1$
$n = 2 - 11, 10$
$n = 3 - 111, 110, 101$
$n = 4 - 1111, 1101, 1010, 1011$
$n = 5 - 11111, 11011, 10101, 10111, 11010, 11011, 11101, 11110, 11111, 11110, 110110$

3.2 For $L$ in (a), prove that $110111 \in L^*$ and that $11100111 \notin L^*$.

Solution:
One way to prove this is, we can write out the CFG for this language and prove that the second string can never be produced by this CFG.
CFG(Rightmost derivation): $S \rightarrow S1|S10|\epsilon$
This CFG clearly shows that if there is a zero it always follows a 1, and a 0 can never follow another 0. ie 0 always co-occurs with a 1. 
So string 1110111 cannot be produced by the above grammar, another way to put is that this string has no derivation in the language \( L^* \).
To prove membership of string 1110111, we can show a derivation that gets us to this value.
\[ S \rightarrow S \rightarrow S1 \rightarrow S11 \rightarrow S111 \rightarrow S10111 \rightarrow S110111 \rightarrow S1110111 \rightarrow \epsilon 110111 \rightarrow 1110111 \]

4 Problem 24.5
Give DFAs for the following computing problems.

4.1 b The DFA accepts quarters and dispenses a coke each time the machines gets 3 quarters.
Solution:
If a DFA accepts only quarters, then all the state reachable will be in increments of 25. Also another point to note is that to be able to dispense a coke, there needs to be at least 75c in the account, so we need a minimum of three states.

4.2 c The DFA accepts quarters and dimes and dispenses candy when the balance is at least 50c
Solution:
In this DFA the only acceptable transitions are of dimes(10c) and quarters(25c), we only dispense when we see at least 50c. So we consider all denominations leading upto 50c as possible states.
4.3 c). The DFA accepts the language $L = \{00010, 10111\}$.

Solution:
There are only two valid strings in this language.

4.4 d). The DFA accepts the language $L = \{\text{strings containing 101 as a substring}\}$.

Solution:
This problem has the restriction that a string has to contain 101 as a contiguous sequence within a string. This implies that we need a set of three states to be able to measure this. Point to note is that we can also have just these values alone, as a string is also a substring of itself.

5 Problem 25.5

Construct CFGs for the following:

5.1 c). Strings of the form $0^{3n}$ for $n \geq 0$

Solution:
This contains all strings formed by the concatenation of 3 zeros. So the CFG has to be able to represent the production rules to produce this language $L$ of three zeroes. $S \rightarrow 000S|\epsilon$

5.2 d). Strings of the form $0^n1^{3n+1}$ for $n \geq 0$

Solution:
This represents the language $L$ of all strings who have $n$ zeroes, but are followed by $3n + 1$ as many 1s. Since this seems like a string which allows has to end with at least one, we can use two set of production rules here.
$S \rightarrow S_1 1$
$S_1 \rightarrow 0S_1 111|\epsilon$
6 Problem 26.3(i)

(i) \( L = \{0^{2^n} | n \geq 0 \} \)

(ii) Give pseudocode of a Turing Machine to solve the problem

(Part ii, iii from a student’s solution)

TM Pseudocode:
1. From start of string, mark every other 0. If reach ⊥ when a mark should be placed, reject (odd number of zeros). Else, go to step 2.
2. Go to start of string
3. Find the odd-numbered marked 0’s and unmark them. If reach ⊥ when a 0 should be unmarked, reject. Else, go to step 4.
4. Move left until reach * or marked 0. If reach * first, accept. If reach marked 0 first, go back to step 2.

(ii) Give machine code for each module in your pseudocode

module 1
\{q0\}{0}\{q1\}\{R\}
\{q0\}{⊥}\{step2\}\{}
\{q1\}{0}\{q0\}\{V\}{R}\}
\{q1\}{⊥}\{E\}\{}

module 2 \{step2\}\{0\}\{step2\}\{L\}
\{step2\}\{āLU\}\{step3\}\{}

module 3 \{step3\}\{0\}\{step3\}\{R\}
\{step3\}\{Mark0\}\{q3\}\{0\}{R}\}
\{step3\}{⊥}\{step4\}\{}
\{q3\}\{Mark0\}\{step3\}\{R\}
\{q3\}{⊥}\{E\}\{}
\{q3\}\{0\}\{q3\}\{R\}

module 4 \{step4\}\{0, ⊥\}\{step4\}\{L\}
\{step4\}\{Mark0\}\{step2\}\{L\}
\{step4\}\{āLU\}\{Accept\}\{}

(iii) Combine your modules to form a Turing Machine to solve the problem (machine code).

1. States: \{q0, q1, E, step2, step3, q3, step4, Accept\}
2. Symbols: \{0, *, Mark0\}
3. Machine level transition instruction \{q0\}\{0\}\{q1\}\{R\}
\{q0\}{⊥}\{step2\}\{}
\{q1\}{0}\{q0\}\{Mark0\}\{R\}
\{q1\}{⊥}\{E\}\{}
\{step2\}\{, 0\}\{step2\}\{L\}
\{step2\}\{āLU\}\{step3\}\{}
\{step3\}\{0\}\{step3\}\{R\}
\{step3\}\{Mark0\}\{q3\}\{0\}\{R\}
\{step3\}{⊥}\{step4\}\{}
\{q3\}\{Mark0\}\{step3\}\{R\}
\{q3\}{⊥}\{E\}\{}
\{q3\}\{0\}\{q3\}\{R\}
\{step4\}\{0, ⊥\}\{step4\}\{L\}
\{step4\}\{Mark0\}\{step2\}\{L\}
\{step4\}\{āLU\}\{Accept\}\{}

5