Assignment 5

Nishan Dhoj Karki

10/05/2022

1 3.16 c)Show that the collection of Turing-recognizable languages is closed under the operation of star.

Proof by Construction:

Claim: The collection of Turing-recognizable languages is closed under the operation of star.

Proof (by construction):

Let us assume that the language L is Turing-recognizable. If L is Turing-recognizable then there exists a recognizer for L. Let M be such recognizer for L. We will now construct a Turing Machine(TM) M_1 such that it recognizes L^*, which is the star closure of L.
$M_1 =$ "On an input string w:
1. If $w = \epsilon$, accept.
2. For all possible ways to split w, non-deterministically split w into parts such that $w = w_1, w_2..., w_n$ are the sub-strings of w.
3. If M accepts all strings $w_1, w_2..., w_n$, then *accept*.
4. If M does not accept all sub-strings, then *reject*.
5. If M loops, then M_1 also loops.

Here, we have split w in all possible ways into sub-strings and run all such splits on M non-deterministically. If M accepts all the possible sub-string of w from a split, then M_1 will accept. When M rejects any of the sub-string, M_1 also rejects. When M loops, M_1 also loops. Here we have constructed a recognizer for L^*. So, By our construction, we can say that collection of Turing-recognizable languages is closed under the operation of star.

2 4.7) Let $\mathcal{B}$ be the set of all infinite sequences over {0,1}. Show that $\mathcal{B}$ is uncountable using a proof by diagonalization.

Proof by Contradiction: Diagonalization

Claim: $\mathcal{B}$ is uncountable

From Definition 4.14, we have, A set $\mathcal{A}$ is countable if either it is finite or it has the same size as $\mathcal{N}$.

For the sake of obtaining contradiction, let us assume that $\mathcal{B}$ is countable, and a correspondence f exists between $\mathcal{N}$ and $\mathcal{B}$. Let, $f : \mathcal{N} \to \mathcal{B}$ be such correspondence from $\mathcal{N}$ to $\mathcal{B}$.

Let us make an assumption for an illustration, that a correspondence defined in the table below exists.

$n \in \mathcal{N}$	$f(n) \in \mathcal{B}$
1	1010....
2	1001....
3	0011....
.	.
.	.

For it to be a correspondence, f must pair all the members of $\mathcal{N}$ with all the members of $\mathcal{B}$. To obtain a contradiction, Let us try to create a binary sequence z in $\mathcal{B}$, such that it has no correspondence to $\mathcal{N}$. We construct z by selecting the i^{th} digit in z to be different from i^{th} digit of $f(i)$, such that $z \neq f(i)$. We select i^{th} digit z to be $= 1$ if i^{th} digit of $f(i) = 0$ or vice versa. As we continue creating z as described, using diagonalization method we can see that for some z in $\mathcal{B}$ there is no pairing with some $n \in \mathcal{N}$ as n^{th} digit of z is different from n^{th} digit of $f(n)$. Thus, $z \in \mathcal{B}$ does not have any correspondence in $\mathcal{N}$ which contradicts our initial assumption. Hence, $\mathcal{B}$ is uncountable.

3 4.13) Let $A = \{\langle R, S\rangle \mid$ R and S are regular expressions and $L(R) \subseteq L(S)\}$. Show that A is decidable.

Proof by Construction:

Claim: A is decidable

We have R and S regular expressions that generate the regular languages $L(R)$ and $L(S)$ respectively. If $L(R)$ and $L(S)$ are regular languages then they have corresponding DFA's that recognize them. Let M_R and M_S be the corresponding DFA's.
From the definition of M_R and M_S, let us create a DFA M_Q such that it accepts the strings accepted by M_S but not by M_R. Since, $L(R) \subseteq L(S)$, M_S recognizes all the languages that M_R recognizes. Hence, M_Q is not accepting anything, the language of M_Q will be empty. The language $L(Q)$ can be defined as: $L(Q) = L(R) \cap \overline{L(S)}$.
We can deduce from the above conclusions that $L(Q)$ is empty when $L(R) \subseteq L(S)$.

We have,
Theorem 4.4
$E_{DFA} = \{\langle A \rangle \mid A$ is a DFA and $L(A) = \emptyset.\}$
E_{DFA} is a decidable language.

Let us use the above conclusions to build a Turing machine B that decides A.
B = "On input of $\langle R, S \rangle$, where R and S are regular expressions:
1. Construct DFA's M_R and M_S such that they recognize $L(R)$ and $L(S)$ respectively.
2. Construct M_Q such that it recognizes $L(Q)$ such that $L(Q) = L(R) \cap \overline{L(S)}$
3. Run a TM T for E_{DFA} on M_T.
4. If T accepts, accept.
5. If T rejects, reject.

Since E_{DFA} is a decidable language T will always accept or reject.
Here, we have created a decider for A. Hence, A is decidable.