[CS 235] Assignment 7

Wednesday, November 6, 2024 10:33 PM

Assignment 7

Computer Science 235

Reading. Sections 4.1 and 4.2

1) Consider the problem of determining whether a DFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable.

$EQ_{DFA,REX} = \{ \langle D, R \rangle \mid D$ is a DFA, R is a regular exp. and $L(D) = L(R)\}$

Let D_R be the DFA constructed from the regular expression R. Then, we have $L(D_R) = L(R)$, which means $L(D) = L(D_R)$.

Given that $EQ_{DFA} = \{\langle D, D_R \rangle\}$ is decidable, $EQ_{DFA,REX}$ is also decidable.

2) Let $A\varepsilon_{CFG} = \{\langle G \rangle \mid G$ is a CFG that generates $\varepsilon\}$. Show that $A\varepsilon_{CFG}$ is decidable.

Let M be a TM / algorithm that decides language $A_{\varepsilon CFG}$.

M = "On input $\langle G \rangle$:

1. Convert G to Chomsky normal form → G'.
2. check if any rule of S_0 is an epsilon.
3. If ε, accept $\langle G \rangle$. Otherwise reject $\langle G \rangle$.

Thus, M decides language $A_{\varepsilon CFG}$.

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

Suppose Ψ is countable. Then we must have some enumeration for all the elements of Ψ.

Let A be a binary sequence within Ψ. $A = a_i^1, \ldots, a_i^n$ where $i = 1, 2, \ldots$

Suppose we have a new binary sequence B which is the complement of each element in the sequence A.

Then, we have $B = b_i^1, \ldots, b_i^n$ where $i = 1, 2, \ldots$

However, $B \cap A = \emptyset$, which means $B \notin \Psi$, a contradiction.

Hence, Ψ is uncountable.

4) A ***useless state*** in a pushdown automaton is never entered on any input string. Consider the problem of determining whether a pushdown automaton has any useless states. Formulate this problem as a language and show that it is decidable.

Let $L = \{\langle P \rangle \mid P$ is a PDA w/ ≥ 1 useless state$\}$

Let M be a TM / algorithm that decides L.

M = "On input $\langle P \rangle$:

1. Check if each $q \in p$ is accessible from the start state.

M = "On input <P>:

1. Check if each $q \in P$ is accessible from the start state.
2. Begin DFS from the start state.
3. For each $q \in P$, check if a state has a sequence of transitions from q to an accept state.
4. Perform reverse DFS from all accept states.
5. If $q \notin$ visited states from steps 2 or 4, q is a useless state.
6. If the set of useless states is not empty, accept <P>. Otherwise, reject.

Hence $\exists$ an algorithm M to decide L.

5) Let $A = \{<R, S> \mid R$ and S are regular expressions and $L(R) \subseteq L(S)\}$. Show that A is decidable.

M = "On input <R,S>:

1. Construct C s.t. $L(C) = L(A) \leq L(B)$.
2. Construct E_{DFA} $L(E) \neq L(S) \cap L(R)$.
3. If $\langle C \rangle \in E_{DFA}$, accept. Otherwise, reject.

M accepts A. Hence, A is decidable.

6) Let R be a regular expression and let G be a CFG. Show that the problem of determining whether there exists a string that is generated by both R and G is decidable (note, we are not checking if a specific string w is generated by both R and G but rather if there exists any string that is generated by both R and G).

M = "On input <R, G>:

1. Convert R to a CFG G_R.
2. Construct a new CFG, G' that generates $L(G_R) \cap L(G)$.
3. Check if $L(G)$ is $\emptyset$.
4. If $L(G)$ is non-empty, accept. Otherwise, reject

Here the decidability of a string w can be decided by M (it is decidable).