Homework 9

Monday, November 25, 2024 3:58 PM

1. $Some_{DFA} = \{ \langle A \rangle \mid A$ is a DFA and $L(A)$ is not empty and $L(A)$ not equal to $\Sigma^* \}$

DFA A_1 s.t. $L(A_1) \mathrel{!=} \Sigma^*$ is decidable. Let M_1...

- Start with start state
- Mark every state that can be reached by transition
- If any of the reached/marked states are not accept states (implies some combination of input cannot be accepted $\Rightarrow L(A_1) \mathrel{!=} \Sigma^*$), then accept. Else, reject.

DFA A_2 s.t. $L(A_2) \mathrel{!=} \emptyset$ decidable. Let M_2...

- Start with start state
- Mark all reachable states by using transitions
- If a state is reached, while there are still unmarked states, that is an accept state (implying an accept state can be reached, and $L(A_2) \mathrel{!=} \emptyset$), then accept and halt.
- If all states are marked, implying previous condition was never satisfied, then reject and halt.

$L(A_1)$, $L(A_2)$ decidable

$L(Some_{DFA}) = L(A_1) \cap L(A_2)$

$L(A_1) \cap L(A_2)$ decidable $\Rightarrow L(Some_{DFA})$ decidable (closed under intersection)

Let $M_{someDFA}$...

- Start at start state
- Mark states reachable by transition
- Check if one marked state is an accept state (satisfies $L(A) \mathrel{!=} \emptyset$) and one marked state is not an accept state (satisfies $L(A) \mathrel{!=} \Sigma^*$), accept and halt
- If all states marked, implying prev condition not true, then reject and halt.

Thus, $M_{\text{some DFA}}$ decides some_{DFA}

2. $Alot_{RE} = \{ \langle A \rangle \mid A$ is RegEx and $L(A)$ is infinite $\}$

RegEx B is regular

B^* is regular and infinite, $L(B^*)$ infinite

- Reg. languages closed under star

Thus, $L(A) = L(B^*)$, where $L(A)$ is infinite

$\Rightarrow A$ regular $\Rightarrow A$ decidable

Let DFA A_1 accept RegEx A.

Why? RegEx has equivalent DFA

Let M_{AlotRE} ...

- Given k, the states in A_1, let A_2 be a DFA s.t. $L(A_2) = \{ w \mid |w| > k \}$
- Let DFA A_3 s.t. $L(A_3) = L(A_1) \cap L(A_2)$, which accepts $w \in L(A_1)$ s.t. $|w| > k$, implying a loop had to be made $\Rightarrow$ accepts infinite
- Simulate AlotRE on A_3
- If $A_3 \,!\!= \emptyset$, accept and halt, else, reject and halt

Thus, M_{AlotRE} decides $Alot_{RE}$

3. $\text{Complimentary}_{RE, DFA} = \{ \langle A, B \rangle \mid A$ is RegEx and B is DFA s.t. $L(A) \cup L(B) = \Sigma^*$ and $L(A) \cap L(B) = \emptyset \}$

A RegEx $\Rightarrow L(A)$ regular

B DFA $\Rightarrow L(B)$ regular

$L(A) \cup L(B)$ regular, $L(A) \cap L(B)$ regular

$\Rightarrow \text{complimentary}_{RE, DFA}$ regular

Why? closed under $\cup$ and $\cap$

Let C be the equivalent DFA of A

Let $M_{\text{complimentary } RE, DFA}$ ...

- Start on start state of C, and simulate on C
- If C accepts, simulate on B
  - If B accepts, reject and halt
  - If B rejects, accept and halt

- If B rejects, accept ...
  (maintains that input $w \notin L(A) \cap L(B)$)
- If C rejects, simulate on B
  - If B accepts, accept and halt
  - If B rejects, reject and halt

Thus, $M_{\text{complimentary RE, DFA}}$ decides $\text{complimentary}_{\text{RE, DFA}}$

4. $ALL_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \Sigma^*\}$

   A is DFA ⇒ L(A) is regular ⇒ L(A) decidable

   Let M_{ALLDFA} ...

   - Start at A's start state
   - Mark all states that can be reached by a transition
   - If a state is marked and is not an accept state, halt and reject, else, accept and halt
     - Reaching a nonaccept state means some input $\in \Sigma^*$ is not accepted

   Thus, M_{ALLDFA} decides ALL_{DFA}.

5. $N_{\varepsilon CFG} = \{\langle G \rangle \mid G \text{ is a CFG and } G \text{ does not generate } \varepsilon\}$

   G is CFG ⇒ L(G) is CFL ⇒ L(G) is decidable

   Let $M_{N\varepsilon CFG}$ ...

   - G_1 is CNF of G
   - Mark all variables that generate ε
   - Mark all variables that generate 2 marked variables
   - If start variable S is marked, reject and halt, else accept and halt
     - $S \to \varepsilon$ will be marked by first step

   Thus, $M_{N\varepsilon CFG}$ decides $N_{\varepsilon CFG}$

6. 4.6 b) No. There does not exist $a \in X$ st. $f(a) = b$ $\forall b \in Y$.
   Namely for $8, 9, 10 \in Y$.

   ... 1-to-1 is false.

c) NO. Onto is false. Thus, ...

e) Yes

f) Yes

7. $U = \{ \langle A, B, C \rangle \mid A, B, C$ are DFAs and $|L(A)| = |L(B)| + |L(C)| \}$

Let M_U ...

- Let D be a DFA that accepts all strings of length $> n$, where n is states in A
- Let E be a DFA s.t. $L(E) = L(D) \cap L(A)$
- Simulate $L(A)$ on E. If E is empty, halt and reject. Else...
    - Create same scheme and E_1 for DFA B. If E_2 not empty, accept and halt. Else...
    - Repeat for E_3 and C. If E_3 not empty, accept and halt. Else, halt and reject.

Why? If $x_1 = x_2 + x_3$ where x_1 is finite, we cannot determine what x_1, the size of $L(A)$, is. However, if $x_1 = \infty$, then $\infty = x_2 + x_3$ is true if either $x_2 \mid x_3$ is ∞.

Thus, M_U decides U

8. $A = \{ \langle R \rangle \mid R$ is regEx s.t. at least one w s.t. 111 is a substring$\}$

R is regEx $\Rightarrow$ $L(R)$ is regular $\Rightarrow$ decidable

Let M_A...

- Let R_1 be all strings w/ subset 111. Convert R and R_1 to DFA D and D_1. Let $D_2 = L(D) \cap L(D_1)$
- Simulate $L(R)$ and $L(R_1)$ on D_2 ...
    - Mark all states. If there is an accept state marked, then halt and accept. Else reject and halt.

Why? $L(D) \cap L(D_1) \neq \emptyset$ means at least one w in $D \approx R$ contains 111

Thus, M_A decides A

9. $E_{PDA} = \{ \langle P \rangle \mid P$ is PDA and $L(P) = \emptyset \}$

P PDA $\Rightarrow$ G CFG $\Rightarrow$ L(G) is CFL $\Rightarrow$ L(G) is decidable

Let M_{EPDA}:

- Convert P to a CFG in CNF
- Mark all terminals ε and variables that lead to terminals including ε
- Mark variables that lead to 2 marked
- If S is marked, halt and reject, else halt and accept.
  - If S is marked, means an output of fully terminals can be derived, and L(G) not empty

10. $L_{useless} = \{ \langle P \rangle \mid$ PDA P that has no useless states $\}$

Let $M_{Luseless}$

- Mark all possible states that can be reached by transition
- If a state is unmarked, halt and reject. Else, halt and accept.

Thus, $M_{Luseless}$ decides $L_{useless}$