Decision Problems for Regular Languages

Lecture 14
Section 4.2

Robb T. Koether

Hampden-Sydney College

Mon, Sep 26, 2016

Outline

Decision Problems

Definition (Decision Problem)

A decision problem is a question that has a yes-or-no answer.

Definition (Decidable)

A decision problem is decidable if there is an algorithm for it that will produce the correct yes-or-no answer for every instance of the problem.

Outline

The Acceptance Problem for DFAs

Definition (The Acceptance Problem for DFAs)

Given a DFA M and a string w, does M accept w?

The Acceptance Problem for DFAs

- To decide the problem, we let w be the input to M and see whether we end up in an accepting state.

The Acceptance Problem for DFAs

- To decide the problem, we let w be the input to M and see whether we end up in an accepting state.
- Therefore, the Acceptance Problem for DFAs is decidable.

Outline

The Emptiness Problem for DFAs

Definition (The Emptiness Problem for DFAs)

Given a DFA M, is the language of M empty. That is, does M reject every word in Σ^*?

The Emptiness Problem for DFAs

- The strategy is to do a breadth-first search of the state diagram for an accept state, starting from the start state.

The Emptiness Problem for DFAs

- If the start state is an accept state, then the answer is “no.”

The Emptiness Problem for DFAs

- If the start state is an accept state, then the answer is “no.”
- If not, then mark the start state as “inspected.”

The Emptiness Problem for DFAs

- If the start state is an accept state, then the answer is “no.”
- If not, then mark the start state as “inspected.”
- Then inspect every state that is reachable in one transition from the start state and is not yet marked.

The Emptiness Problem for DFAs

- If the start state is an accept state, then the answer is “no.”
- If not, then mark the start state as “inspected.”
- Then inspect every state that is reachable in one transition from the start state and is not yet marked.
- If any is an of them is an accept state, then the answer is “no.”

The Emptiness Problem for DFAs

- If the start state is an accept state, then the answer is “no.”
- If not, then mark the start state as “inspected.”
- Then inspect every state that is reachable in one transition from the start state and is not yet marked.
- If any is an of them is an accept state, then the answer is “no.”
- If not, then mark them as inspected.

The Emptiness Problem for DFAs

- If the start state is an accept state, then the answer is “no.”
- If not, then mark the start state as “inspected.”
- Then inspect every state that is reachable in one transition from the start state and is not yet marked.
- If any is an of them is an accept state, then the answer is “no.”
- If not, then mark them as inspected.
- Continue in this manner with the states that are reachable from the marked states in one transition and that have not yet been marked.

The Emptiness Problem for DFAs

- This procedure will eventually terminate when
  - It reaches an accept state, in which case the answer is “no.”
  - It can reach only states that are already marked, none of which are accept states, in which case the answer is “yes.”

The Emptiness Problem for DFAs

- This procedure will eventually terminate when
  - It reaches an accept state, in which case the answer is “no.”
  - It can reach only states that are already marked, none of which are accept states, in which case the answer is “yes.”
- Therefore, the Emptiness Problem for DFAs is decidable.

Outline

The Equivalence Problem for DFAs

Definition (The Equivalence Problem for DFAs)

Given two DFAs M_1 and M_2, do they have the same language? That is, does $L(M_1) = L(M_2)$?

The Equivalence Problem for DFAs

- The strategy is to follow the algorithm to build the DFA M whose language is

$$\left(L(M_1) \cap \overline{L(M_2)}\right) \cup \left(\overline{L(M_1)} \cap L(M_2)\right).$$

The Equivalence Problem for DFAs

- The strategy is to follow the algorithm to build the DFA M whose language is

$$\left(L(M_1) \cap \overline{L(M_2)}\right) \cup \left(\overline{L(M_1)} \cap L(M_2)\right).$$

- Then solve the Emptiness Problem for M.

The Equivalence Problem for DFAs

- The strategy is to follow the algorithm to build the DFA M whose language is

$$\left(L(M_1) \cap \overline{L(M_2)}\right) \cup \left(\overline{L(M_1)} \cap L(M_2)\right).$$

- Then solve the Emptiness Problem for M.
  - If $L(M) = \varnothing$, then $L(M_1) = L(M_2)$.

The Equivalence Problem for DFAs

- The strategy is to follow the algorithm to build the DFA M whose language is

$$\left(L(M_1) \cap \overline{L(M_2)}\right) \cup \left(\overline{L(M_1)} \cap L(M_2)\right).$$

- Then solve the Emptiness Problem for M.
  - If $L(M) = \varnothing$, then $L(M_1) = L(M_2)$.
  - If $L(M) \neq \varnothing$, then $L(M_1) \neq L(M_2)$.

The Equivalence Problem for DFAs

- The strategy is to follow the algorithm to build the DFA M whose language is

$$\left(L(M_1) \cap \overline{L(M_2)}\right) \cup \left(\overline{L(M_1)} \cap L(M_2)\right).$$

- Then solve the Emptiness Problem for M.
  - If $L(M) = \varnothing$, then $L(M_1) = L(M_2)$.
  - If $L(M) \neq \varnothing$, then $L(M_1) \neq L(M_2)$.
- Therefore, the Equivalence Problem for DFAs is decidable.

Outline

- Other decision problems for DFAs:
  - Decide whether $L(M) = \Sigma^*$.
  - Decide whether M accepts any string of even length.
  - Decide whether $L(M)$ is infinite.
  - Decide whether $L = L^*$.
  - Decide whether $L = L^R$.
  - Decide whether $L(M_1) \subseteq L(M_2)$.

Outline

Assignment

Assignment

- Section 4.2 Exercises 1, 2, 4, 6, 9, 14, 17, 18.