Decision algorithms for Regulor Language / Decision properties for Regular language.

There are six properties which are decidable.

1. Emptiness and non-emptyness
2. finiteness and infiniteness
3. Membership
4. Equality

1. Emptiness and non-emptyness

Algorithm

Step 1: select the state that cannot be reached from the initial states and delete them.

Step 2: If the resulting machines contains at least one final states, so then the finite automata accepts the non-empty Language.

Step 3: If the resulting machine is free from final states then finite automata accepts the empty Language.

2. Finiteness and infiniteness

Step1: Select the states that cannot be reached from the initial state and delete them.

Step2: Select the state from which we cannot reach the final states and delete them (remove dead state).

Step3: If the resulting machine contains the loop or cycles then the finite automata accepts infinite language.

Step4: If the resulting machine donot contains loops or cycles then the finite automata accepts finite language.

3. Membership

- Membership is a property to verify an arbitrary string is accepted by a finite automaton or not i.e it a member of a language or not.

- let M is a finite automata that accepts some strings over an alphabet, and let 'w' be any string defined over the alphabet. if there exist a transition path in M, which starts at initial state and ends in anyone of final state then w is a member of M otherwise 'w' is not a member of M.

4. <u>Equality</u>

- two Finite state automata M_1 and M_2 is said to be equal if and only if they accept the same Language.
- Minimize the Finite state automata and the minimal DFA will be unique.