CSL253 - Theory of Computation

Tutorial 7

Team Members

1. Gaurav Kumar - 12340790
2. Rahul Razz - 12341690
3. Om Anand - 12341510

Question 16

Decidability of PREFIX-FREEREX

Given a regular expression R over alphabet Σ, determine whether the language recognized by R is prefix-free, i.e., no accepted word can be extended to another accepted word.

Solution for Question 16

Theorem

Theorem 1. *The problem PREFIX-FREEREX is decidable.*

Proof. Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA equivalent to R. Define a finite directed graph whose vertices are the states in Q and whose edges are the transitions of D.

Observe that a violation of the prefix-free property occurs exactly when there is an accepting state $p \in F$ from which one can reach, via one or more edges, another accepting state $q \in F$.

Since this is a reachability question on a finite graph, it can be decided by standard graph search (e.g. depth-first search).

Decision Algorithm.

Algorithm 1 Decide prefix-freeness via reachability

Require: Regular expression R

Ensure: YES if prefix-free, NO otherwise

```
Construct DFA D = (Q, Σ, δ, q_0, F) for R
Initialize S ← ∅
for all p ∈ F do
    Perform graph search from p in the transition graph of D
    if an accepting state q ∈ F is reached in at least one step then
        Add p to S
    end if
end for
if S = ∅ then
    return YES
else
    return NO
end if
```

Correctness and Termination. All operations (DFA construction, graph search) occur on the finite set Q, so the procedure halts. Reachability in a directed graph is decidable, and exactly captures whether an accepting state can reach another, witnessing a prefix violation. Hence PREFIX-FREEREX is decidable.

□

Question 17

Decidability of AMBIGNFA

Determine whether a given NFA N is ambiguous, i.e., accepts some string via two distinct computation paths.

Solution for Question 17

Graph-Theoretic Proof Idea

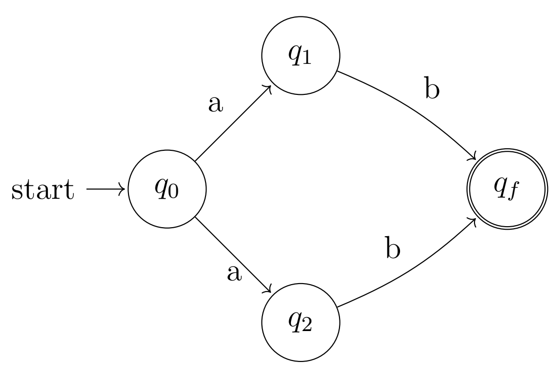
Model the NFA as a directed graph:

- **States:** nodes
- **Transitions:** directed, labeled edges

BFS Traversal with Parent Marking

- Start BFS from initial state.
- For each visited state:
  - Track its parent(s) (who reached it)
  - Mark the parent with a dot (•)
- On reaching a final state:
  - Check how many distinct parents it has.
  - If ≥ 2 parents are marked, the NFA is ambiguous.
- Since BFS explores level-wise, all paths consuming same length strings are processed together.

Example NFA



Traversal Illustration

- Start at q_0
- On `a`: move to q_1 and q_2
- Mark q_0 as parent of both.
- On `b`:
  - $q_1 \to q_f$ (parent: q_1)
  - $q_2 \to q_f$ (parent: q_2)
- Now, q_f has two distinct parents (q_1 and q_2)
- Mark both q_1 and q_2 with dots
- Since q_f has two parents $\to$ NFA is ambiguous

Decidability Argument

- BFS is guaranteed to terminate (finite graph).
- For each final state, check number of distinct parents.
- If any final state has ≥ 2 parents for the same string (same BFS level), NFA is ambiguous.
- Since marking and parent tracking are finite and computable, this procedure is decidable.

Therefore, $AMBIGNFA$ is decidable.