Subset Construction

Introduction

- DFA and NFA both represent regular languages
- Equivalent DFA using subsets of NFA states

Outline

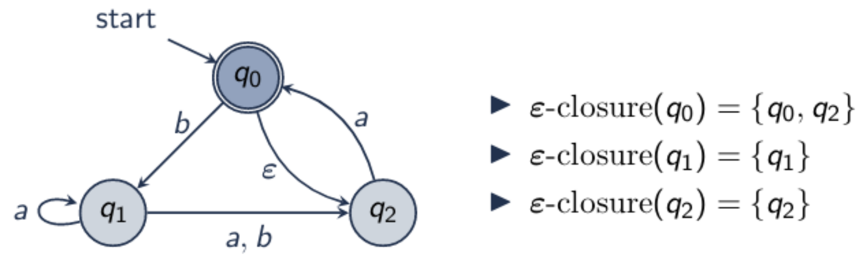
- ϵ Closures
  - ϵ-closure
  - move-ϵ-closure
- NFA Simulation
- Subset Construction Algorithm

ϵ Closures

ϵ-closure(q): states reachable from q on ϵ transitions

move-ϵ-closure: states reachable from state q after reading symbol σ

ϵ-closure



- ε-closure$(q_0) = \{q_0, q_2\}$
- ε-closure$(q_1) = \{q_1\}$
- ε-closure$(q_2) = \{q_2\}$

Figure 1: Example of epsilon closures on a NFA

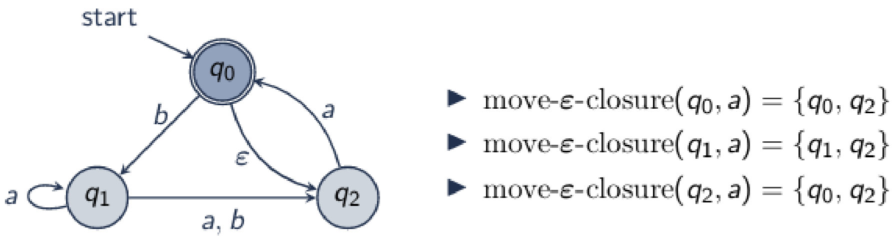
Idea:

- Input: initial set of states
- Output: set of states (the closure) reachable only on ϵ transitions
- Approach: recursively visit states and "accumulate" the ϵ-closure

  Base case: Visiting a state already contained in the ϵ-closure (set of states)

  Recursive case: Add the current state and recurse on all its ϵ-neighbors

move-ϵ-closure



- ▶ move-ε-closure$(q_0, a) = \{q_0, q_2\}$
- ▶ move-ε-closure$(q_1, a) = \{q_1, q_2\}$
- ▶ move-ε-closure$(q_2, a) = \{q_0, q_2\}$

Figure 2: Example of move epsilon closure on a NFA

Idea:

- Input: initial state set Q and input symbol σ
- Output: set of states reachable from Q after reading σ (any number of ϵ-transitions are allowed)
- Approach:
  1. Find ϵ-closure of initial set Q
  2. Find set of states after reading the symbol σ
  3. Find ϵ-closure of resulting set
- Note: move first and then do epsilon

NFA Simulation

- Input: NFA N and Input String w
- Outline: Does N accept the input string w
- Algorithm Outline:
  1. Compute ϵ-closure of start state as the current state (subset)
  2. While the input string contains more symbols:
     2.1 Read the next input symbol
     2.2 Compute the move-ϵ-closure from the current state (subset)
     2.3 Update the current state (subset)
  3. At the end of the input string:
     If the current state (subset) contains an accept state, return accept
     Otherwise, return reject

Subset Construction Algorithm

Subset Construction: NFA to DFA

- Input: NFA
- Output: Equivalent DFA
- Algorithm Outline: Recursively visit subsets, beginning with ϵ-closure of the start state
  1. If the current subset was already visited, return
  2. For each symbol σ in alphabet Σ, find move-ϵ-closure from the current subset on σ
  3. Add the resulting subsets and edges, then recursively visit

[LINK]: Converting NFA to DFA ez mode

Practical usage of DFAs VS NFAs

- If we have an NFA:
  1. Simulate NFA by tracking all possible current states (subsets)
  2. Convert NFA to DFA by constructing the subsets of NFA state set