Decision Properties of Regular Languages

Introduction

Decision Property

- A decision property is a (Boolean) question about a language
  - Is the language empty?
  - Is the language a subset of another language?
- Algorithm on a representation
- Applications in verification

Decision properties:

- A boolean operation on a language
  - Unary: $\mathrm{F} : \mathrm{S} \rightarrow \mathbb{B}$
  - Binary: $\mathrm{T} : \mathrm{S} \times \mathrm{S} \rightarrow \mathbb{B}$

Outline

- Membership
- Empty Set and Empty String
- Equivalence and Subset
- Language Size

Membership

Input: String w and the language as Finite Automaton A

Output: Is w in the language of A?

$\mathrm{w} \in L(\mathrm{A})$?

Solution:

1. Simulate A on w
2. If A accepts w, then $\mathrm{w} \in L(\mathrm{A})$
3. Otherwise, A does not accept w, so $\mathrm{w} \notin L(\mathrm{A})$

Empty Set and Empty String

Emptiness

Input: Finite Automaton A

Output: Is the language of A empty?

$L(\mathrm{A}) = \emptyset$?

Solution:

1. Search nodes reachable from the start state q_0
2. If an accept state $\mathrm{q} \in \mathrm{F}$ is reachable, $L(\mathrm{A})$ is not empty
3. Otherwise, no accept states are reachable, so the language is empty

Empty String

Input: Finite Automaton A

Output: Is the empty string in language of A?

$\epsilon \in L(\mathrm{A})$

Solution:

1. Search nodes reachable from the start state q_o following only ϵ transitions
2. If an accept state $\mathrm{q} \in \mathrm{F}$ is reachable using only ϵ transitions, then the empty string ϵ is in $L(\mathrm{A})$
3. Otherwise, no accept states are reachable on ϵ, so the ϵ is not in $L(\mathrm{A})$

Empty Set VS Empty String

Empty Set:

- Set with zero members
- $\emptyset = \{\}$
- $|\emptyset| = 0$
- $\emptyset \neq 0$
- $L(\emptyset) = \emptyset = \{\}$

Empty String

- String of length zero
- $\epsilon = ()$
- $|\epsilon| = 0$
- $\epsilon \neq 0$ and $\epsilon \neq \emptyset$ and $\epsilon \neq \mathrm{e}$
- $L(\epsilon) = \{\epsilon\} = \{()\}$

Equivalence and Subset

Input: Finite Automata L and M

Output: Do L and M recognize the same language?

L(L) = L(M)

Solution: Is there a string accepted by only one of L or M

- Idea: run L and M in "parallel"
- But: we don't want to actually simulate on both DFA (What if we are comparing L(aa*) and L(a*a)?)
- Instead: Compute the Product DFA of L and M

Product DFA

Input: Finite Automata

L = (Q_L, Σ, δ_L, q_{0L}, F_L)

M = (Q_M, Σ, δ_M, q_{0M}, F_M)

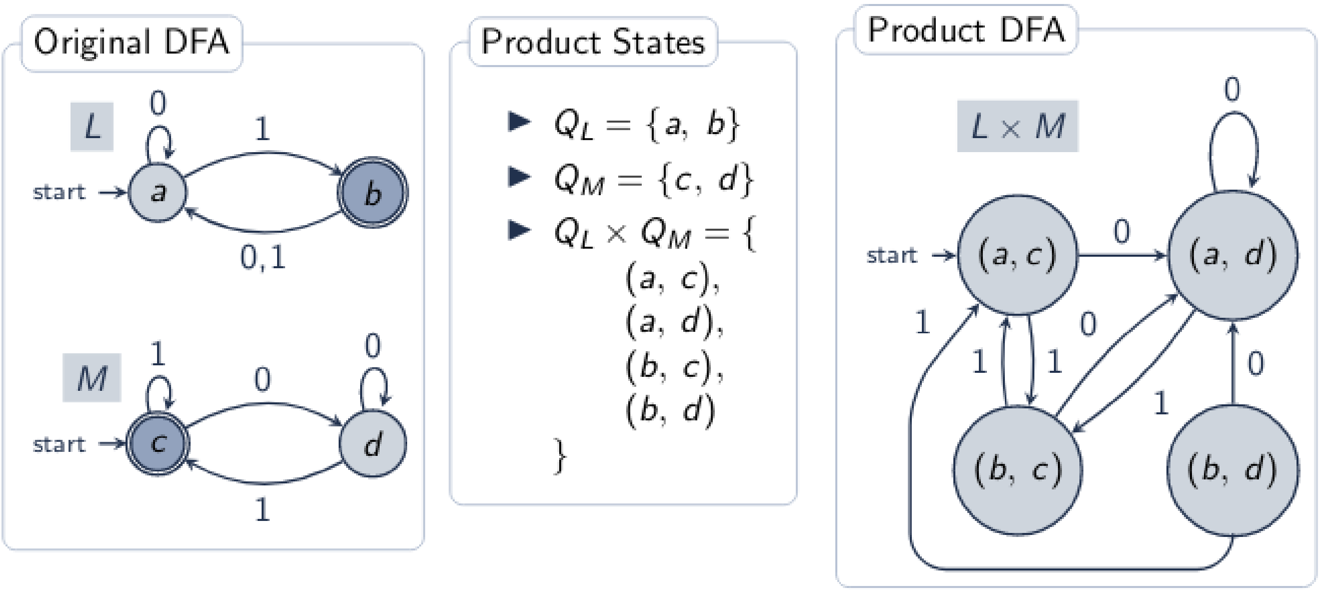
Output: Product Automaton P = (Q_P, Σ, δ_P, q_{0P}, F_P)

Solution

- States: $Q_P = Q_L \times Q_M$
- Start State: $q_{0P} = (q_{0L}, q_{0M})$
- Transition Function $\delta_p((l,m),\sigma) = (\delta_L(l,\sigma),\delta_M(m,\sigma))$
- Accept States: depend on usage

Transition, basically for each pair for a given sigma, see what the first item goes to, see what the second item goes to, and then the two items it transitions to is the new pair state you transition to.

Example:



Product States

- $Q_L = \{a, b\}$
- $Q_M = \{c, d\}$
- $Q_L \times Q_M = \{$
  $(a, c),$
  $(a, d),$
  $(b, c),$
  (b, d)
  $\}$

Solution to Equivalence

For 2 languages L and M, after using the product DFA of both:

- Accept only when one of L and M accept:

  $\mathrm{F_P} = \{(l, \mathrm{m}) \mid (l \in \mathrm{F_L}) \oplus (\mathrm{m} \in \mathrm{F_M})\ \}$

  - (l, m) = Product state
  - $(l \in \mathrm{F_L})$ = L accepts
  - $(\mathrm{m} \in \mathrm{F_M})$ = M accepts

- Equivalent when product DFA is empty:

  $(L(\mathrm{L}) = L(\mathrm{M}) \leftrightarrow (\emptyset = L(\mathrm{P}))$

Subset

Input: Finite Automata

$\mathrm{L} = (\mathrm{Q_L}, \Sigma, \delta_\mathrm{L}, \mathrm{q_{0L}}, \mathrm{F_L})$

$\mathrm{M} = (\mathrm{Q_M}, \Sigma, \delta_\mathrm{M}, \mathrm{q_{0M}}, \mathrm{F_M})$

Output: $L(\mathrm{L}) \subseteq L(\mathrm{M})$

Solution: Product DFA

- $\mathrm{P} = \mathrm{L} \times \mathrm{M}$
- $\mathrm{F_P}= \{(l \in \mathrm{Q_L}, \mathrm{m} \in \mathrm{Q_M})|(l \in \mathrm{F_L}) \wedge (\mathrm{m} \notin \mathrm{F_M})\}$
  - $(l \in \mathrm{Q_L}, \mathrm{m} \in \mathrm{Q_M})$ = product state
  - $(l \in \mathrm{F_L})$ = accepted by L
  - $(\mathrm{m} \notin \mathrm{F_M})$ = not accepted by M
- $(L(\mathrm{L}) \subseteq L(\mathrm{M})) \equiv (\emptyset = L(\mathrm{P}))$
- equivalent: $(L(\mathrm{L}) \subseteq L(\mathrm{M})) \equiv (\emptyset = L(\mathrm{L}) \cap L(\mathrm{M}))$

Strict Subset

Strict subset $(L(\mathrm{L}) \in L(\mathrm{M}))$ is a decidable property of regular languages

Proof by construction:

- Input: Finite Automata
- Output: $L(\mathrm{L}) \subset L(\mathrm{M})$

Language Size

Given DFA A and string $\sigma \in L(\mathrm{A})$

If the string is larger than the number of transitions, what does this tell us about the language? (INFINITE HEHE)

Infiniteness

Input: Finite Automaton A

Output: Is the language of A infinite?

$|L(\mathrm{A})| = \infty$?

Solution: Doe the transition graph contain a cycle that:

- Is reachable from the start state
- Can reach an accept state
- Contains a non-epsilon transition