Closure properties of regular languages

Deepak D’Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

08 August 2011
Outline

1. Closure under boolean ops
2. Induction
3. NFA’s
Closure properties

- Class of Regular languages is closed under
  - Complement, intersection, and union.
  - Concatenation, Kleene iteration.
- Non-deterministic Finite-state Automata (NFA) = DFA.
Closure under complementation

- Idea: Flip final states.
- Formal construction:
  - Let $A = (Q, s, \delta, F)$ be a DFA over alphabet $A$.
  - Define $B = (Q, s, \delta, Q - F)$.
  - Claim: $L(B) = A^* - L(A)$.

Proof of claim

- $L(B) \subseteq A^* - L(A)$.
  - $w \in L(B) \implies \hat{\delta}(s, w) \in (Q - F)$.
    $\implies \hat{\delta}(s, w) \notin F$
    $\implies w \notin L(A)$
    $\implies w \in A^* - L(A)$.

- $L(B) \supseteq A^* - L(A)$.
Closure under intersection

Product construction. Given DFA’s $A = (Q, s, \delta, F)$, $B = (Q', s', \delta', F')$, define product $C$ of $A$ and $B$:

$$C = (Q \times Q', (s, s'), \delta'', F \times F'),$$

where $\delta''((p, p'), a) = (\delta(p, a), \delta'(p', a))$.

Product construction example
Correctness of product construction

Claim: \( L(C) = L(A) \cap L(B) \). 

Proof of claim \( L(C) = L(A) \cap L(B) \). 

1. \( L(C) \subseteq L(A) \cap L(B) \). 
   - \( w \in L(C) \implies \hat{\delta}''((s, s'), w) \in F \times F' \). 
   - \( (\hat{\delta}(s, w), \hat{\delta}'(s', w)) \in F \times F' \) (by subclaim) 
   - \( \hat{\delta}(s, w) \in F \) and \( \hat{\delta}'(s', w) \in F' \) 
   - \( w \in L(A) \) and \( w \in L(B) \) 
   - \( w \in L(A) \cap L(B) \).

2. \( L(C) \supseteq L(A) \cap L(B) \). 

Subclaim: \( \hat{\delta}''((s, s'), w) = (\hat{\delta}(s, w), \hat{\delta}'(s', w)) \).
Closure under union

- Follows from closure under complement and intersection since

\[ L_1 \cup L_2 = \overline{L_1} \cap \overline{L_2}. \]
Closure under union

- Follows from closure under complement and intersection since
  \[ L_1 \cup L_2 = \overline{L_1 \cap L_2}. \]

- Can also do directly by product construction: Given DFA’s \( \mathcal{A} = (Q, s, \delta, F) \), \( \mathcal{B} = (Q', s', \delta', F') \), define \( \mathcal{C} \):
  \[ \mathcal{C} = (Q \times Q', (s, s'), \delta'', (F \times Q') \cup (Q \times F')) \]
  where
  \[ \delta''((p, p'), a) = (\delta(p, a), \delta(p', a)). \]
Principle of Mathematical Induction

- \( \mathbb{N} = \{0, 1, 2 \ldots\} \)
- \( P(n) \): A statement \( P \) about a natural number \( n \).
- Example:
  - \( P(n) = \) “\( n \) is even.”
  - \( P_1(n) = \) “Sum of the numbers 1 \ldots \( n \) equals \( n(n + 1)/2 \).”
  - \( P_2(n) = \) “For all \( w \in A^* \), if length of \( w \) is \( n \) then \( \hat{\delta}''((s, s'), w) = (\hat{\delta}(s, w), \hat{\delta}'(s', w)) \).”

Principle of Induction

If a statement \( P \) about natural numbers
- is true for 0 (i.e. \( P(0) \) is true), and,
- is true for \( n + 1 \) whenever it is true for \( n \) (i.e. \( P(n) \implies P(n + 1) \))
then \( P \) is true of all natural numbers (i.e. “For all \( n \), \( P(n) \)” is true).
Proof of subclaim

Exercise: Prove the Subclaim:

$$\hat{\delta}''((s, s'), w) = (\hat{\delta}(s, w), \hat{\delta}'(s', w)).$$

using induction.
Nondeterministic Finite-state Automata (NFA)

- Allows multiple start states.
- Allows more than one transition from a state on a given letter.

A word is accepted if there is **some** path on it from a start to a final state.
Example NFA’s

NFA for “contains $abb$ as a subword”
NFA definition

- Mathematical representation of NFA
  \[ \mathcal{A} = (Q, S, \Delta, F), \text{ where } S \subseteq Q, \text{ and } \Delta : Q \times A \rightarrow 2^Q. \]
  Define relation \( p \xrightarrow{w} q \) which says there is a path from state \( p \) to state \( q \) labelled \( w \).
  \[ p \xrightarrow{\epsilon} p \]
  \[ p \xrightarrow{ua} q \text{ iff there exists } r \in Q \text{ such that } p \xrightarrow{u} r \text{ and } q \in \Delta(r, a). \]
  Define \( L(\mathcal{A}) = \{ w \in A^* \mid \exists s \in S, f \in F : s \xrightarrow{w} f \} \).

- NFA \rightarrow DFA: Subset construction
  Example: determinize NFA for “contains abb.”
  Formal construction
  Correctness
Closure under concatenation and Kleene iteration

- Concatenation of languages:
  \[ L \cdot M = \{ u \cdot v \mid u \in L, \ v \in M \}. \]

- Kleene iteration of a language:
  \[ L^* = \{ \epsilon \} \cup L \cup L^2 \cup L^3 \cup \cdots, \]
  where
  \[ L^n = L \cdot L \cdots L \text{ (n times)}, \]
  \[ = \{ w_1 \cdots w_n \mid \text{each } w_i \in L \}. \]