Notation

- \( \Sigma \): finite non-empty set of symbols, e.g., \( \{ a, b \} \)
- \( \Sigma^k \): concatenation of symbols of \( \Sigma \), \( k \) times, \( k \geq 0 \)
  - e.g., \( \Sigma^3(\{a, b\}) \), \( (a, b, a, bb, ab, ba) \), …
  - \( \Sigma = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup ... \)
- string \( \in \Sigma \), e.g., \( ababa \)
- \( |\text{string}| \): length of string, e.g., \( |ababa| = 6 \)
- \( s^R \): reversal of string \( s \), e.g., \( ababa^R = ababab \)
- \( s, t \): concatenation of strings \( s \) and \( t \)
- language \( \subseteq \Sigma \), e.g., \( L_{ab} = \{ \omega \in \Sigma | \omega \text{ has odd # of } a \} \)

Language Operations

- \( L_1 \cup L_2 = \{ \omega \in \Sigma^* | \omega \in L_1 \text{ or } \omega \in L_2 \} \)
  - e.g., \( L_{ab} \cup L_{aa} = \{ \omega \in \Sigma^* | \omega \text{ has odd number of } a \text{'s or does not end with } a \} \)
- \( L_1 \cap L_2 = \{ \omega \in \Sigma^* | \omega \in L_1 \text{ and } \omega \in L_2 \} \)
  - e.g., \( L_{aa} \cap L_{ab} = \{ \omega \in \Sigma^* | \omega \text{ has odd number of } a \text{'s and does not end with } a \} \)
- \( L_1 - L_2 = \{ \omega \in \Sigma^* | \omega \in L_1 \text{ and } \omega \notin L_2 \} \)
  - e.g., \( L_{aa} - L_{ab} = \{ \omega \in \Sigma^* | \omega \text{ has odd number of } a \text{'s and ends with } a \} \)
- \( \overline{L_1} = \{ \omega \in \Sigma^* | \omega \notin L_1 \} \)
  - e.g., \( \overline{L_{ab}} = \{ \omega \in \Sigma^* | \omega \text{ ends with } a \} \)
- \( L_1^R = \{ \omega \in \Sigma^* | \omega^R \in L_1 \} \)
  - e.g., \( L_{ab}^R = \{ \omega \in \Sigma^* | \omega \text{ do not start with } a \} \)

Regex

- so far, we have explicitly seen
  - RL can be shown by FSA
  - RL can be shown by set description
- another way to define RL is by:
  - Regular Expressions
    - aka regex, RE
**RL: formal definition** (revisit)

- Let $\Sigma$ be the alphabet:
  - the empty set, $\emptyset$, is a RL
  - the set $\{a\}$ is a RL
  - for each $a \in \Sigma$, the set $\{a\}$ is a RL
  - if $L_1$ and $L_2$ are regular languages, then
    - union: $L_1 \cup L_2$ is a RL
    - concatenation: $L_1 \cdot L_2$ is a RL
    - Kleene star: $L_1^*$ is a RL
- No other RL over $\Sigma$ exists.

**$L(r)$ is defined by structural induction**

- **basis step:**
  - if $r$ is a regex defined by basis step of the definition,
    - $L(\emptyset)$ is a RL
    - $L(\varepsilon)$ is a RL
    - $L(a)$, for any $a \in \Sigma$, is a RL
- **inductive step:**
  - if $r_1$, $r_2$ are regex's defined by ind step of the definition,
    - $L(r_1 + r_2) = L(r_1) \cup L(r_2)$ is a RL
    - $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$ is a RL
    - $L(r_1^*) = L(r_1)^*$ is a RL

**regex examples (96)**

- Assume $\Sigma = \{0,1\}$
  - $\emptyset$, $\varepsilon$, 0, 1, 0+1, 00, 01, 10, 11, 000, 111,
  - $L((0+1)^*)$
  - $L(0^*)$
  - $L((10)^*)$
  - $L(10^*)$

**regex examples**

**notes**

- Example 83: (revisit)
- Example 84: (revisit)
- Example 85: (revisit)
Example 97

- Prove $L_{B6} = L(r_{B6})$ where
  - $L_{B6} = \{ \omega \in \{0,1\}^* | \omega \text{ starts and ends with different bits} \}$
  - $r_{B6} = 0.(0+1)^*1 + 1.(0+1)^*0$

regex identities

- communitativity of union:
- associativity of union:
- associativity of concatenation:
- left distributivity:
- right distributivity:
- identity for union:
- identity for concatenation:
- annihilator for concatenation:
- idempotence of Kleene star:

NFA, DFA, regex

NFA, DFA, regex

notes