Learning Outcomes

By the end of this session you should be able to...

- define regular expressions (RegExpr)
- build an NFA accepting the same language of a RegExpr
Regular expressions, I

Given \( \Sigma \), the set of regular expressions is defined inductively.

**Base cases:**

- **Every** \( \sigma \in \Sigma \) is a regular expression
  - \( L(\sigma) \overset{\text{def}}{=} \{ \sigma \} \)
- \( \epsilon \) is a reg. exp.
  - \( L(\epsilon) \overset{\text{def}}{=} \{ \epsilon \} \)
- \( \emptyset \) is a reg. exp.
  - \( L(\emptyset) \overset{\text{def}}{=} \emptyset \)
Regular expressions, II

Operations:

- If $\alpha, \beta$ are reg. exp. then $\alpha + \beta$ is a reg. exp.
  
  - $L(\alpha + \beta) \overset{\text{def}}{=} L(\alpha) \cup L(\beta)$
  
  - some authors write $\alpha | \beta$ instead of $\alpha + \beta$

- If $\alpha, \beta$ are regular expressions then $\alpha \beta$ is a regular expression
  
  - $L(\alpha \beta) \overset{\text{def}}{=} L(\alpha)L(\beta)$

- If $\alpha$ is a regular expression then $\alpha^*$ is a regular expression
  
  - $L(\alpha^*) \overset{\text{def}}{=} L(\alpha)^*$
Examples

- We say that a string $s$ matches a regular expression $\alpha$ whenever $s \in L(\alpha)$.
  - no strings match $\emptyset$ (since $L(\emptyset) = \emptyset$);
  - strings that end in $b$ match $(a + b)^*b$;
  - $L((a + b)^*b) = L((a + b)^*)L(b) = L(a + b)^*L(b)$
    $$= (L(a) \cup L(b))^*\{b\} = \{a, b\}^*\{b\}$$
    $$= \{x_1 \ldots x_ny \mid n \in \mathbb{N}, x_i \in \{a, b\}, y \in \{b\}\}$$
  - strings of even length match $((a + b)(a + b))^*$;
  - $abba$ and the empty string match $abba + \epsilon$;
Kleene’s Theorem

Theorem.

- If $\alpha$ is a regexp then $L(\alpha)$ is a regular language
- If $L$ is a regular language then $L = L(\alpha)$ for some regexp $\alpha$

- In other words, finite automata and regular expressions describe the same languages!
Proving Kleene’s Theorem

• We know that DFAs, NFAs and $\varepsilon$NFAs accept precisely the regular languages;

• To prove Kleene’s theorem we will:
  1. show how to convert a regular expression to a $\varepsilon$NFA;
  2. show how to convert any NFA to a regular expression
Reg exp to finite automaton

• The set of regular expressions is built inductively:
  o $\sigma \in \Sigma$, $\epsilon$ and $\emptyset$ are the base cases;
  o $+$, $\cdot$ and $-\ast$ are the operations.

• So to prove that $\forall \alpha. L(\alpha)$ is regular we need to prove that:
  o $L(\sigma)$, $L(\epsilon)$ and $L(\emptyset)$ are regular.
  o if $L(\alpha)$, $L(\beta)$ are regular then so is $L(\alpha + \beta)$;
  o if $L(\alpha)$, $L(\beta)$ are regular then so is $L(\alpha\beta)$;
  o if $L(\alpha)$ is regular then so is $L(\alpha^\ast)$.
Base cases

- \( L(\sigma) \) is regular;

- \( L(\epsilon) \) is regular;

- \( L(\emptyset) \) is regular;
• Want to show that $L(\alpha + \beta) \overset{\text{def}}{=} L(\alpha) \cup L(\beta)$ is regular;

• Can assume $L(\alpha)$ and $L(\beta)$ are regular;

• We’ve proved in a tutorial that regular languages are closed under union. Hence $L(\alpha + \beta)$ is regular.
Concatenation

- Want to show that $L(\alpha \beta) \overset{\text{def}}{=} L(\alpha)L(\beta)$ is regular;
- Can assume $L(\alpha)$ and $L(\beta)$ are regular;
- Let $M_1$ and $M_2$ be NFAs for $L(\alpha)$ and $L(\beta)$ respectively. We construct a new $\epsilon$NFA $M_3$:

  ![Diagram](image)

  - as final states of $M_3$ take those of $M_2$. It is easy to show that $L(M_3) = L(M_1)L(M_2) = L(\alpha)L(\beta)$. 
Kleene star

• Want to show that $L(\alpha^*)$ is regular;
• Can assume $L(\alpha)$ is regular;
• Let $M$ be an NFA for $L(\alpha)$. We construct a new $\epsilon$NFA $M'$:

![Diagram of NFA]

• as final states of $M'$ take the singleton $\{0\}$. It is easy to show that

$$L(M') = L(M)^* = L(\alpha)^* = L(\alpha^*).$$
Reg exp to finite automaton (recap)

• The set of regular expressions is built inductively:
  o $\sigma \in \Sigma$, $\epsilon$ and $\emptyset$ are the base cases;
  o $+$, $\cdot$ and $-\ast$ are the operations.

• So to prove that $\forall \alpha. L(\alpha)$ is regular we have shown that:
  o $L(\sigma)$, $L(\epsilon)$ and $L(\emptyset)$ are regular.
  o if $L(\alpha)$, $L(\beta)$ are regular then so is $L(\alpha + \beta)$;
  o if $L(\alpha)$, $L(\beta)$ are regular then so is $L(\alpha\beta)$;
  o if $L(\alpha)$ is regular then so is $L(\alpha^\ast)$. 
Learning Outcomes

You should be able to . . .

• . . . *define* regular expressions (RegExpr)

• . . . *build* an NFA accepting the same language of a RegExpr