Regular expressions

- used for describing string patterns, e.g.

\[(0 \cup 1)0^* \quad (\{0\} \cup \{1\}) (\{0\})^*\]

\[(0 \cup 1)^* \rightarrow (\{0\} \cup \{1\})^* \quad \text{all strings over \{0,1\}}\]

Strings starting with 0 or 1 and followed by an arbitrary # 0's
Regular expressions

Formal definition:

R is a **regular expression** if R is one of the following:

1. a for some \( a \in \Sigma \),  
   \[ \{0, 1\} \]
2. \( \epsilon \)
3. \( \emptyset \)
4. \( (R_1 \cup R_2) \), where \( R_1, R_2 \) are regular expressions
5. \( (R_1 \cdot R_2) \), where \( R_1, R_2 \) are regular expressions
6. \( (R_1)^* \), where \( R_1 \) is a regular expression.

Note: this type of definition is called a **recursive/inductive definition** (i.e. the definition is a recursive algorithm).
Regular expressions

For convenience: $R^+ = RR^*$

Examples: give regular expressions for the following languages:

- $\{ w \in \{0,1\}^* \mid w$ contains the substring 001 $\}$
  
  $(0u1)^* 0.0.1(0u1)^* \quad \text{priority of operations: } *, \cdot, \cup$

- $\{ w \in \{0,1\}^* \mid w$ does not contain two consecutive 0's $\}$
  
  $(01u1)^* (0u1)$

- $\{ w \in \{0,1\}^* \mid |w|$ is divisible by 2 or 3 $\}$
  
  $(0u1)(0u1)^* \cup (0u1)(0u1)(0u1)^*$

  OR

  $(0u1)^* \cup (0u1)^3 \cup (0u10u11)^* \cup (0u1)^3 \cup (0u10u11)^* \cup (0u1)^3 \cup \ldots \cup 111$

- $\{ w \in \{0,1\}^* \mid |w| < 4 \}$
  
  $(0u1u1)^3 = \varepsilon \cup 0u1 \cup 00u1 \cup 01u10u11 \cup 000u \ldots \cup 111$
Examples: let $R$ be any regular expression

- $R \cdot \emptyset = \emptyset$
- $R \cdot \{ \varepsilon \} = R$
- $\emptyset^* = \varepsilon$
- $\{ \varepsilon \}^* = \varepsilon$

The language defined by $R$ is denoted $L(R)$. We’ll often abuse notation and use $R$ to denote the language $L(R)$. 
Thm 1.54: A language is regular iff some regular expression describes it.

Lemma 1.55: Given a regular expression R, there exists a FA M such that $L(M) = L(R)$.

Lemma 1.60: Given a FA M, there exists a regular expression R such that $L(R) = L(M)$. 
Lemma 1.55: Given a regular expression $R$, there exists a FA $M$ such that $L(M) = L(R)$. 

by induction (structural, i.e. on the structure of $R$):

**BASE CASES:**
1) $a \in \Sigma$ then we construct $M$: 

\[ \rightarrow \circ \rightarrow \circ \checkmark \]

2) $\varepsilon$ 

\[ \rightarrow \circ \checkmark \]

3) $\emptyset$ 

\[ \rightarrow \circ \checkmark \]

**IND. CASES:**
4) if $R = R_1 \cup R_2$ then, by IH, we have NFAs $M_1$ and $M_2$ for $R_1$ and $R_2$ 

by Thm 1.45 we get an NFA for $R_1 \cup R_2$ 

\[ \rightarrow \circ \rightarrow \circ \checkmark \]

5) if $R = R_1 \cdot R_2$ then Thm 1.47 

\[ \rightarrow \circ \rightarrow \circ \checkmark \]

6) if $R = R_1^*$ Thm 1.49 

\[ \rightarrow \circ \rightarrow \circ \checkmark \]
Equivalence of reg. expr. and FA’s

**Lemma 1.60**: Given a FA $M$, there exists a regular expression $R$ such that $L(R) = L(M)$.

**Proof idea:**

*Generalized NFA (GNFA)*

- transitions may be marked by reg. expr. (not just $\Sigma \cup \{\varepsilon\}$)
- single accept state that a) has arrows coming in from every other state, b) does not have any outgoing arrows
- start state that a) has arrows to every other state, b) does not have any incoming arrows
- all other states have arrows to all other states
Lemma 1.60: Given a FA $M$, there exists a regular expression $R$ such that $L(R) = L(M)$.

Proof idea:

Generalized NFA (GNFA) $(Q, \Sigma, \delta, q_{start}, q_{accept})$ where all as usual except $\delta: (Q-\{q_{accept}\}) \times (Q-\{q_{start}\}) \rightarrow R$ where $R$ is the set of all regular expressions over $\Sigma$.

Idea: start with a GNFA, remove states one by one and redraw arrows as necessary.

How to get a GNFA:

1) all pink arrows are $\emptyset$
2) new accept, transition b is it from states in $T$ on $b$
3) new start, on $\varepsilon$ to the old start
4) add all missing transitions (arrows) on $\emptyset$
Lemma 1.60: Given a FA $M$, there exists a regular expression $R$ such that $L(R) = L(M)$.

Proof idea:

How to construct an equivalent GNFA with one fewer state?