Regular expressions

- The set \( RE \) of regular expressions is given by:
  \[
  r \in RE ::= a \mid \varepsilon \mid \emptyset \mid r_1 r_2 \mid r_1 + r_2 \mid r^* \quad a \in \Sigma
  \]

- \( L : RE \to 2^{\Sigma^*} \) is given inductively by
  \[
  L(a) := \{a\} \quad L(r_1 r_2) := \{xy \mid x \in L(r_1) \text{ and } y \in L(r_2)\}
  \]
  \[
  L(\emptyset) := \emptyset \quad L(r_1 + r_2) := L(r_1) \cup L(r_2)
  \]
  \[
  L(\varepsilon) := \{\varepsilon\} \quad L(r^*) := \{x_1 x_2 \cdots x_n \mid n \geq 0 \text{ and } x_i \in L(r)\}
  \]

- \( x \) matches \( r \) means \( x \in L(r) \)

**Exercise:** Are the following regular expressions equivalent (i.e. have the same language?)

- \( L((01+10)^k) = \{01, 0101, 010101, \ldots\} \)

- \( L((01)^{\ast}) = \{0, 1, 01, 101, 10101, \ldots\} \)

- \( L((10)^{\ast}) = \{0, 1, 01, 101, 010101, \ldots\} \)

- \( L((01+10)^{\ast}) = \{01, 0101, 010101, \ldots\} \)

- \( L((01)^{\ast}) = \{0, 01, 0101, 010101, \ldots\} \)

- \( L((10)^{\ast}) = \{1, 01, 0101, 010101, \ldots\} \)

- \( L((01)^{\ast} + (10)^{\ast}) = \{0, 1, 01, 0101, 010101, \ldots\} \)

- \( L((01+10)^{\ast}) = \{01, 0101, 010101, \ldots\} \)
Kleene's theorem

**Defn:** A language $L$ is **regular** if there is a **regular expression** $r \in RE$ with $L = L(r)$.

**Claim (Kleene's theorem):** The following are equivalent:

- $L$ is DFA-recognizable
- $L$ is NFA-recognizable
- $L$ is regular

**Proof outline:**

![Proof diagram]
Converting RE to NFA

\[ r \in \text{RE} ::= a | \varepsilon | \emptyset | n \cdot 2 | n + 2 | r^* \quad a \in \Sigma \]

**Claim:** For all \( r \in \text{RE} \), there exists an NFA \( N \) with \( L(N) = L(r) \).

**Proof:** By structural induction on \( r \).

\[
\begin{align*}
\text{let } P(r) & \text{ be the statement } L(N) = L(r) \\
\text{let } P(a) & \text{ be true for all } a \in \Sigma \\
P(\varepsilon) & \\
P(\emptyset) & \\
P(n \cdot 2) & \\
P(n + 2) & \\
P(r^*) & \\
\end{align*}
\]

\[
\begin{align*}
\text{assume } & \exists N_1 \text{ with } L(N_1) = L(r_1) \\
& \exists N_2 \text{ with } L(N_2) = L(r_2) \\
\text{and } & \exists N \text{ with } L(N) = L(r_1 \cdot r_2) \\
\text{and } & L(N) = L(r_1) \cup L(r_2) \\
\end{align*}
\]

Remaining details are exercises.
Converting NFA to RE

**Claim:** For all NFA \( N \), there exists a regular expression \( r \) with \( L(N) = L(r) \).

**Proof idea:**

Generalized NFA has re. on the transitions

\[ ab^*c \]

\[ ab^*a \]
Beyond regular languages (and beyond 2800)

Turing Machine:
like a DFA, two new abilities:
1. Can overwrite its input
2. Can move to left or right

a→ move left, write b

(!) Turing machines can simulate any program or computer.