Regular expressions
\iffalse\rightleftharpoons\fi
Regular languages
Sipser 1.3 (pages 63–76)

Last time...
Regular expressions

• Definition 1.52:
  Say that $R$ is a regular expression if $R$ is
  1. $a$ for some $a$ in the alphabet $\Sigma$
  2. $\varepsilon$
  3. $\emptyset$
  4. $(R_1 \cup R_2)$, where $R_1$ and $R_2$ are regular expressions
  5. $(R_1 \cdot R_2)$, where $R_1$ and $R_2$ are regular expressions
  6. $(R_1^*)$, where $R_1$ is a regular expression

Regular expressions and NFAs

• Theorem 1.54: A language is regular if and only if some regular expression describes it.
• Proof ($\Leftarrow$)
  1. If $a \in \Sigma$, then $a$ is regular.
  2. $\varepsilon$ is regular.
  3. $\emptyset$ is regular.
  4. If $R_1$ and $R_2$ are regular, then $(R_1 \cup R_2)$ is regular.
  5. If $R_1$ and $R_2$ are regular, then $(R_1 \cdot R_2)$ is regular.
  6. If $R_1$ is a regular, then $(R_1^*)$ is regular.
Going forward

• Theorem 1.54: A language is regular if and only if some regular expression describes it.

• (⇒)

Diagram:
- Regular expression
- 3-state DFA
- 2-state GNFA
- 3-state GNFA
- 4-state GNFA
- 5-state GNFA
Proof

DFA to GNFA...

- Step 1: Add a unique start state with an $\varepsilon$ jump to the original one
- Step 2: Add a unique accept state with $\varepsilon$ jumps from the previous accept states
- Step 3: Convert multiple labels to $\cup$
- Step 4: Add $\emptyset$ jumps for any transition that's missing
Induction step: rip a state

A simple example