NFA and regex

- the Boolean algebra of languages
- non-deterministic machines
- regular expressions
The intersection of two regular languages is regular

Run both machines in parallel?

Build one machine that simulates two machines running in parallel!

Keep track of the state of each machine.
The intersection of two regular languages is regular.
The intersection of two regular languages is regular
The intersection of two regular languages is regular
The regular languages $A \subseteq \Sigma^*$ form a Boolean Algebra

• Since they are closed under intersection and complement.
Two examples

<table>
<thead>
<tr>
<th>×2</th>
<th>×2 + 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Input sequence is accepted if it ends with a zero.

Even binary numbers

Input sequence is accepted if it ends with a one.

Odd binary numbers
Three examples

Which binary numbers are accepted?

\[
\begin{array}{cccc}
\times 2 & \times 2 + 1 \\
\text{mod 3} & 0 & 1 \\
0 & 0 & 1 \\
1 & 2 & 0 \\
2 & 1 & 2 \\
\end{array}
\]
By three or not by three?

![Diagram](image)

- Divisible by three
- Not divisible by three
The complement of a regular language is regular

If $A \subseteq \Sigma^*$ is recognised by $M$ then $\overline{A} = \Sigma^* \setminus A$ is recognised by $\overline{M}$ where $\overline{M}$ and $M$ are identical except that the accepting states of $\overline{M}$ are the non-accepting states of $M$ and vice-versa.
The intersection of two regular languages is regular

divisible by 6
≡
divisible by 2
and
divisible by 3
The intersection of two regular languages is regular

Run both machines in parallel?

Build one machine that simulates two machines running in parallel!

Keep track of the state of each machine.
The intersection of two regular languages is regular
The intersection of two regular languages is regular
The intersection of two regular languages is regular.
The regular languages $A \subseteq \Sigma^*$ form a Boolean Algebra

- Since they are closed under intersection and complement.
Is there a regular expression for every FSM?
Is there a regular expression for every FSM?

$L_1 = \varepsilon$

$L_0 = \varepsilon$

$L_2 = \varepsilon$
Is there a regular expression for every FSM?

$L_1 = L_0 \ a \mid L_2 \ c$

$L_0 = \varepsilon$

$L_2 = L_1 \ b$
Is there a regular expression for every FSM?

$L_1 = L_0 \; a \mid L_2 \; c$

$\quad = a \mid L_1 \; bc$

$L_0 = \varepsilon$

$L_2 = L_1 \; b$
Is there a regular expression for every FSM?

$L_1 = L_0 \ a \ I \ L_2 \ c$

$L_1 = a \ I \ L_1 \ bc$

$L_1 = a \ I \ (bc)^*$

$L_0 = \varepsilon$

$L_2 = L_1 \ b$
Arden’s Lemma

If $R$ and $S$ are regular expressions then the equation

$$X = R \mid X \cdot S$$

has a solution $X = R \cdot S^*$

If $\epsilon \notin L(S)$ then this solution is unique.
Is there a regular expression for every FSM?

Let $L_i$ be the language accepted if $i$ is the accepting state.

$L_0 = \varepsilon$
$L_1 = L_0a$
$L_2 = L_1b \lor L_0c$

$L_2 = L_0a \lor L_1 \lor L_0c$
$L_2 = \varepsilonab \lor \varepsilonc$
$L_2 = ab \lor c$
Is there a regular expression for every FSM?

\[ L_1 = L_2 b \]
\[ L_2 = L_3 b | L_1 a \]
\[ L_3 = \varepsilon | L_1 b \]
\[ = \varepsilon | L_2 b b \]
\[ L_2 = (\varepsilon | L_2 b b) b | L_2 b a \]
\[ = b | L_2 b b b | L_2 b a \]
\[ = b | L_2 (b b b | b a) \]
Is there a regular expression for every FSM?

\[ L_2 = b \lor L_2 (b b b \lor b a) \]

\[ L_2 = b (b b b \lor b a)^* \]

\[ L_1 = L_2 b = b (b b b \lor b a)^* b \]

\[ L_3 = \varepsilon \lor L_2 b b = \varepsilon \lor b (b b b \lor b a)^* bb \]
Arden’s Lemma

If $R$ and $S$ are regular expressions then the equation

$$X = R \mid XS$$

has a solution $X = RS^*$

If $\varepsilon \not\in L(S)$ then this solution is unique.

$L_2 = b \mid L_2 (b b b b b a)$

$L_2 = b (b b b b b a)^*$
regular expressions

- any character is a regexp
  - matches itself
- if R and S are regexps, so is RS
  - matches a match for R followed by a match for S
- if R and S are regexps, so is R|S
  - matches any match for R or S (or both)
- if R is a regexp, so is R*
  - matches any sequence of 0 or more matches for R
- The algebra of regular expressions also includes elements 0 and 1
  - 0 matches nothing; 1 matches the empty string
regular expressions denote regular sets

- any character a is a regexp
  - \{<a>\}
- if R and S are regexs, so is RS
  - \{ r s \mid r \in R \text{ and } s \in S \}
- if R and S are regexps, so is RIS
  - R \cup S
- if R is a regexp, so is R*
  - \{ r^n \mid n \in \mathbb{N} \text{ and } r \in R \}
- 0 \mid S = S = S 0
  - \emptyset \text{ empty set}
- 1 \mid S = S = S 1
  - \{<>\} singleton empty sequence:

https://en.wikipedia.org/wiki/Kleene_algebra