Regular Expressions and their Languages

- Recursion
- Regular Languages
- Regular Expressions
- Examples
- Formalism
- Equivalence
Recursive Definitions

**in general**

1. building blocks

2. rules for combining
   - the building blocks and
   - the results of previous combining

3. rule out everything else
Recursive Definition

Example: Propositional Logic

building blocks:  T, F;  p, q, r, ... ; , , ...

rules for combining

( , , ), ...

“results of previous combining: use of , and .

rule out everything else
Regular Languages:  
a recursive definition

1. $\emptyset$ and $\{\square\}$ are regular languages.

   $\square \square \square : \{a\}$ is a regular language.

2. If $L$ is regular, $L^*$ is regular.

   If $L_1$ and $L_2$ are regular, then $L_1L_2$ and $L_1\sqcup L_2$ are regular.

3. No other languages over $\square$ are regular.
Regular Expressions and Regular Languages

Regular expressions ...

... *represent* regular languages

... look like abbreviations for them

Preliminary examples:

b is the RE for \{b\}.

a+b is the RE for \{a\} \text{ or } \{b\} = \{a,b\}.
The Languages
that R.E.s Represent

1. Building Blocks:

 \[ \emptyset \] is the RE for \( \{\} \).

 \[ \emptyset \] is the RE for \( \emptyset \).

 \[ \cdot \cdot \cdot \cdot \cdot : \] \[ \cdot \] is the RE for \( \{\cdot\} \)

2. Combining

 \[ r_1 r_2 \] is the RE for \( L(r_1) L(r_2) \)

 \[ r_1+ r_2 \] is the RE for \( L(r_1) \cdot L(r_2) \)

 \[ r^* \] is the RE for \( [L(r)]^* \)

3. Nothing else is an RE
Three Roles for a Symbol

If $\Sigma = \{a,b,c\}$, $b$ is ...

- a *symbol*
- a *string* of length 1
- the *RE* for the language \{b\}. 
### Some REs and their Languages in Extensional Form

<table>
<thead>
<tr>
<th>Regular Expression</th>
<th>Corresponding Regular Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>a+bc</td>
<td>{a, bc}</td>
</tr>
<tr>
<td>a(b+c)</td>
<td>{ab, ac}</td>
</tr>
<tr>
<td>(a+b)(a+c)(L+a)</td>
<td>{aa, ac, ba, bc, aaa, aca, baa, bca}</td>
</tr>
<tr>
<td>a*(b+cc)</td>
<td>{b, cc, ab, acc, aab, aacc, aaab, aaacc,...}</td>
</tr>
<tr>
<td>a+bb*</td>
<td>{a, b, bb, bbb, bbbbb, bbbbb,...}</td>
</tr>
<tr>
<td>(a+bb)*</td>
<td>{L, a, bb, aa, abb, bba, bbbbb, aaa,...}</td>
</tr>
<tr>
<td>a<em>b</em></td>
<td>{L, a, b, aa, ab, bb, aaa, aab, abb, bbb,...}</td>
</tr>
</tbody>
</table>
Find the RE

Find an RE for all strings over $\Sigma = \{a, b\}$.

$$(a+b)^*$$

$$(b+a)^*$$

$$(a*b^*)^*$$
Find the RE

Find an RE for all strings of b’s with at least two b’s.

- It’s an infinite language, so we need closure.
- For “at least 0” use $b^*$.
- Increase all lengths by 2, by adding $bb$.
- $b^*bb$ is a correct answer.
- $bbb^*$ and $bb^*b$ are also correct.
- The star applies only to the b right before it.
Find the RE

Find an RE for all strings of a given length over \( \mathcal{A} = \{a,b\} \).

Length 0: \( \varepsilon \)

Length 1: \( a+b \) or \( b+a \)

Length 3: \( (a+b)(a+b)(a+b) = (a+b)^3 \)

Length \( k \): \( (a+b)^k \)

Length 0 to \( k \): \( (a+b+\varepsilon)^k \)
Find the RE

Find REs for string sets over \( \square = \{a,b\} \) with this many b’s:

- exactly 2 b’s
- at least 2 b’s
- an even number of b’s
- an odd number of b’s
Find the RE

Exactly two b’s: \(a^*ba^*ba^*\).

At least 2 b’s: \((a+b)^*b(a+b)^*b(a+b)^*\)
\[a^*ba^*b(a+b)^*\]

Even: Why not \((a^*ba^*ba^*)^*\) ??
Find the RE

Even:  *Why not*  \((a^*ba^*ba^*)^*\) ?

• It’s wrong, … since it disallows all of \(L(a^*)\).
• It’s inelegant, … needlessly repeating \(a^*\)

Even:  \(a^*(ba^*ba^*)^*\)

Odd:  \(a^*ba^*(ba^*ba^*)^*\)
Describe the Language

Describe the language of the RE, \[ +b+bb+bbb\]b*.  

\[ bbb\]b* represents the strings of 3 or more b’s.

The rest of the expression takes care of lengths 0, 1 and 2, giving the set of all strings of b’s.

Thus the given regular expression simplifies to \[ b\]b*.  

A description of the language is

“the set of all strings of zero or more b’s.”
Describe the Language

Describe the language of this RE:

\[0 + (1+2+3+\ldots+9)(0+1+2+\ldots+9)^*\]

“The set of decimal representations of the nonnegative integers, without leading zeroes.”
Specify the Language

Specify with mathematical notation the language of this RE:

\(((a+b)(a+b))^*\)

Answer:

\(\{x \in \Gamma^* \mid |x| \text{ is even}\},\)

where \(\Gamma = \{a,b\}\)
Diagrams can Express Language Operations

Concatenation

- Connect accepting state of \(d_1\) by \(\rightarrow\) to start of \(d_2\).
- Use start state of \(d_1\) and accepting states of \(d_2\).

Union

- New start state connected by \(\rightarrow\) to starts of \(d_1\) and \(d_2\)
- Accepting states of \(d_1\) and \(d_2\) all still accept.

Closure

- New start state is the lone accepting state,
- Connect it by \(\rightarrow\) to start of \(d\)
- Connect each accepting state of \(d\) to it by \(\rightarrow\)
## Diagrams for the Language Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{ab}</code>:</td>
<td>a → b → c</td>
</tr>
<tr>
<td><code>{a}·{b}</code>:</td>
<td>a → b → c</td>
</tr>
<tr>
<td><code>{a}*:</code></td>
<td>a → b → c</td>
</tr>
<tr>
<td><code>{a}·{b}</code>:</td>
<td>a → b → c</td>
</tr>
<tr>
<td><code>{ab}·{ac}</code>:</td>
<td>a → b → c</td>
</tr>
</tbody>
</table>
After Eliminating \( \sqcap \)s

\[
\begin{align*}
\{ab\}: & \quad \rightarrow a \quad b \quad \rightarrow b \\
\{a\sqcap\{b\}: & \quad \rightarrow a \quad b \quad \rightarrow b \\
\{a\}^*: & \quad \rightarrow a \quad \rightarrow a
\end{align*}
\]
Equivalence

- **RE to FA** by combining diagrams

- **FA to RE** by
  - transition merging
  - state removal

- **Re**s and **FAs** *represent the same languages.*