2.2
Regular Expressions
Regular Expressions

A way to denote regular languages

- simple patterns to describe related strings
- useful in
  - text search (editors, Unix/grep, emacs)
  - compilers: lexical analysis
  - compact way to represent interesting/useful languages
  - dates back to 50’s: Stephen Kleene
    who has a star names after him.
Inductive Definition

A regular expression $r$ over an alphabet $\Sigma$ is one of the following:

**Base cases:**
- $\emptyset$ denotes the language $\emptyset$
- $\epsilon$ denotes the language $\{\epsilon\}$
- $a$ denote the language $\{a\}$.

**Inductive cases:** If $r_1$ and $r_2$ are regular expressions denoting languages $R_1$ and $R_2$ respectively then,
- $(r_1 + r_2)$ denotes the language $R_1 \cup R_2$
- $(r_1 \cdot r_2) = r_1 \cdot r_2 = (r_1 r_2)$ denotes the language $R_1 R_2$
- $(r_1)^*$ denotes the language $R_1^*$
Inductive Definition

A regular expression $r$ over an alphabet $\Sigma$ is one of the following:

**Base cases:**
- $\emptyset$ denotes the language $\emptyset$
- $\epsilon$ denotes the language $\{\epsilon\}$.
- $a$ denote the language $\{a\}$.

**Inductive cases:** If $r_1$ and $r_2$ are regular expressions denoting languages $R_1$ and $R_2$ respectively then,
- $(r_1 + r_2)$ denotes the language $R_1 \cup R_2$
- $(r_1 \cdot r_2) = r_1 \cdot r_2 = (r_1 r_2)$ denotes the language $R_1 R_2$
- $(r_1)^*$ denotes the language $R_1^*$
<table>
<thead>
<tr>
<th>Regular Languages</th>
<th>Regular Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅ regular</td>
<td>∅ denotes ∅</td>
</tr>
<tr>
<td>{ε} regular</td>
<td>ε denotes {ε}</td>
</tr>
<tr>
<td>{a} regular for a ∈ Σ</td>
<td>a denote {a}</td>
</tr>
<tr>
<td>R₁ ∪ R₂ regular if both are</td>
<td>r₁ + r₂ denotes R₁ ∪ R₂</td>
</tr>
<tr>
<td>R₁ R₂ regular if both are</td>
<td>r₁ • r₂ denotes R₁ R₂</td>
</tr>
<tr>
<td>R* is regular if R is</td>
<td>r* denote R*</td>
</tr>
</tbody>
</table>

Regular expressions denote regular languages — they explicitly show the operations that were used to form the language.
For a regular expression \( r \), \( L(r) \) is the language denoted by \( r \). Multiple regular expressions can denote the same language!

**Example:** \( (0 + 1) \) and \( (1 + 0) \) denote same language \( \{0, 1\} \)

- Two regular expressions \( r_1 \) and \( r_2 \) are equivalent if \( L(r_1) = L(r_2) \).
- Omit parenthesis by adopting precedence order: \( \ast \), concatenate, \( + \).

**Example:** \( r^\ast s + t = ((r^\ast)s) + t \)

- Omit parenthesis by associativity of each of these operations.
  **Example:** \( rst = (rs)t = r(st) \), \( r + s + t = r + (s + t) = (r + s) + t \).

- Superscript \( + \). For convenience, define \( r^+ = rr^\ast \). Hence if \( L(r) = R \) then \( L(r^+) = R^+ \).
- Other notation: \( r + s \), \( r \cup s \), \( r|s \) all denote union. \( rs \) is sometimes written as \( r \cdot s \).
For a regular expression $r$, $L(r)$ is the language denoted by $r$. Multiple regular expressions can denote the same language!

**Example:** $(0 + 1)$ and $(1 + 0)$ denote same language $\{0, 1\}$

Two regular expressions $r_1$ and $r_2$ are equivalent if $L(r_1) = L(r_2)$.

Omit parenthesis by adopting precedence order: $\ast$, concatenate, $\oplus$.

**Example:** $r^\ast s + t = ((r^\ast)s) + t$

Omit parenthesis by associativity of each of these operations.

**Example:** $rst = (rs)t = r(st)$, $r + s + t = r + (s + t) = (r + s) + t$.

Superscript $\oplus$. For convenience, define $r^\oplus = rr^\ast$. Hence if $L(r) = R$ then $L(r^\oplus) = R^\oplus$.

Other notation: $r + s$, $r \cup s$, $r|s$ all denote union. $rs$ is sometimes written as $r \cdot s$. 

Notation and Parenthesis
For a regular expression $r$, $L(r)$ is the language denoted by $r$. Multiple regular expressions can denote the same language!

**Example:** $(0 + 1)$ and $(1 + 0)$ denote same language $\{0, 1\}$

Two regular expressions $r_1$ and $r_2$ are equivalent if $L(r_1) = L(r_2)$.

Omit parenthesis by adopting precedence order: $\ast$, concatenate, $+$. 

**Example:** $r^*s + t = ((r^*)s) + t$

Omit parenthesis by associativity of each of these operations.

**Example:** $rst = (rs)t = r(st)$, $r + s + t = r + (s + t) = (r + s) + t$.

Superscript $+$. For convenience, define $r^+ = rr^*$. Hence if $L(r) = R$ then $L(r^+) = R^+$.

Other notation: $r + s$, $r \cup s$, $r | s$ all denote union. $rs$ is sometimes written as $r \cdot s$. 

Har-Peled (UIUC)  
CS374  
Fall 2020  
16 / 27
Notation and Parenthesis

For a regular expression $r$, $L(r)$ is the language denoted by $r$. Multiple regular expressions can denote the same language!

**Example:** $(0 + 1)$ and $(1 + 0)$ denote same language $\{0, 1\}$

Two regular expressions $r_1$ and $r_2$ are equivalent if $L(r_1) = L(r_2)$.

Omit parenthesis by adopting precedence order: $\ast$, concatenate, $\oplus$.

**Example:** $r \ast s + t = ((r \ast s) + t$

Omit parenthesis by associativity of each of these operations.

**Example:** $rst = (rs)t = r(st)$, $r + s + t = r + (s + t) = (r + s) + t$.

Superscript $\ast$. For convenience, define $r^+ = rr^\ast$. Hence if $L(r) = R$ then $L(r^+) = R^+$.

Other notation: $r + s$, $r \cup s$, $r | s$ all denote union. $rs$ is sometimes written as $r \cdot s$. 
For a regular expression $r$, $L(r)$ is the language denoted by $r$. Multiple regular expressions can denote the same language!

**Example:** $(0 + 1)$ and $(1 + 0)$ denote same language \{0, 1\}

Two regular expressions $r_1$ and $r_2$ are **equivalent** if $L(r_1) = L(r_2)$.

Omit parenthesis by adopting precedence order: $\ast$, concatenate, $\oplus$.

**Example:** $r^\ast s + t = ((r^\ast)s) + t$

Omit parenthesis by associativity of each of these operations.

**Example:** $rst = (rs)t = r(st)$, $r + s + t = r + (s + t) = (r + s) + t$.

Superscript $\oplus$. For convenience, define $r^\oplus = rr^\ast$. Hence if $L(r) = R$ then $L(r^\oplus) = R^\oplus$.

Other notation: $r + s$, $r \cup s$, $r | s$ all denote union. $rs$ is sometimes written as $r \bullet s$. 
For a regular expression \( r \), \( L(r) \) is the language denoted by \( r \). Multiple regular expressions can denote the same language!

**Example:** \((0 + 1)\) and \((1 + 0)\) denote same language \(\{0, 1\}\)

Two regular expressions \( r_1 \) and \( r_2 \) are equivalent if \( L(r_1) = L(r_2) \).

Omit parenthesis by adopting precedence order: \(*\), concatenate, \(+\).

**Example:** \( r^*s + t = ((r^*)s) + t \)

Omit parenthesis by associativity of each of these operations.

**Example:** \( rst = (rs)t = r(st) \), \( r + s + t = r + (s + t) = (r + s) + t \).

**Superscript** \(+\). For convenience, define \( r^+ = rr^* \). Hence if \( L(r) = R \) then \( L(r^+) = R^+ \).

**Other notation:** \( r + s \), \( r \cup s \), \( r|s \) all denote union. \( rs \) is sometimes written as \( r \cdot s \).
Skills

- Given a language $L$ “in mind” (say an English description) we would like to write a regular expression for $L$ (if possible)
- Given a regular expression $r$ we would like to “understand” $L(r)$ (say by giving an English description)
Skills

- Given a language $L$ “in mind” (say an English description) we would like to write a regular expression for $L$ (if possible).
- Given a regular expression $r$ we would like to “understand” $L(r)$ (say by giving an English description).
THE END

... 

(for now)