Regular expressions

In arithmetic, we use $+$, $\times$ to build up expressions e.g.

$$(5+3)\times 4$$

Similarly, we can use regular operations to build up regular expressions: e.g. $(\{\alpha\}\cup\{\beta\})\{\alpha\}^*$

For simplicity: $(\alpha\cup\beta)\alpha^*$

Priority: star $>$ catenation $>$ union

Regular expressions

Defn. Call R a *regular expression* if R is

1) $\{\sigma\}$ where $\sigma \in \Sigma$ (or simply σ)
2) $\{\epsilon\}$ (or simply ϵ)
3) $\emptyset$
4) $(R_1 \cup R_2)$, where R_1, R_2 are regular expressions
5) $(R_1 \circ R_2)$, where R_1, R_2 are regular expressions
6) (R_1^*), where R_1 is a regular expression

Write $L(R)$ for the language of regular expression R

Examples

$\Sigma = \{0, 1\}$

1) $R = 0^*10^*$
$L(R) =$

2) $R = (0 \cup \epsilon)1^*$
$L(R) =$

3) $R = (\Sigma\Sigma)^*$
$L(R) =$

4) $R = \Sigma^*001\Sigma^*$
$L(R) =$

Examples

1) $R \cup \emptyset =$

2) $R\emptyset =$

3) $R\epsilon =$

4) $\emptyset^* =$

Examples

$\Sigma = \{0, 1\}$

1) $L = \{s \in \Sigma^* \mid s \text{ contains at least one } 0\}$
Design R s.t. $L(R) = L$
$R =$

2) $L = \{s \in \Sigma^* \mid s \text{ starts and ends with the same symbol}\}$
Design R s.t. $L(R) = L$
$R =$

3) $L = \{\epsilon, 0, 1, 01\}$
Design R s.t. $L(R) = L$
$R =$

Equivalence of regular expressions and regular languages

Theorem. R is a regular expression iff $L(R)$ is a regular language

Conclusion:

Regular expressions $\Leftrightarrow$ regular languages $\Leftrightarrow$ NFA $\Leftrightarrow$ DFA

Let's prove: if R is a regular expression, then $L(R)$ is a regular language.

Equivalence of regular expressions and regular languages

Proof: Let R be a regular expression.

Then by defn. we have 6 cases:

1) $R = \sigma$, where $\sigma \in \Sigma$
2) $R = \epsilon$
3) $R = \emptyset$
4) $R = R_1 \cup R_2$, where R_1, R_2 are regular expressions
5) $R = R_1 \circ R_2$, where R_1, R_2 are regular expressions
6) $R = R_1^*$, where R_1 is a regular expression

Equivalence of regular expressions and regular languages

1) $R = \sigma$, where $\sigma \in \Sigma$

2) $R = \epsilon$

3) $R = \emptyset$

Equivalence of regular expressions and regular languages

4) $R = R_1 \cup R_2$, where R_1, R_2 are regular expressions
5) $R = R_1 \circ R_2$, where R_1, R_2 are regular expressions
6) $R = R_1^*$, where R_1 is a regular expression

Example

Convert $R = (\alpha\beta \cup \alpha)^*$ to an NFA

Example

Convert $R = (\alpha \cup \beta)^* \alpha\beta\alpha$ to an NFA

Equivalence of regular expressions and regular languages

Theorem. R is a regular expression iff $L(R)$ is a regular language

We have just proved: if R is a regular expression, then $L(R)$ is a regular language.

The reverse direction: if $L \subseteq \Sigma^*$ is a regular language then there exists a regular expression R s.t. $L(R) = L$

(The proof needs a concept called generalized NFA; see Sipser's book, p.70)