Regular Expressions

A regular expression describes a language using three operations.

Regular Expressions

A ***regular expression*** (RE) describes a language.

It uses the three ***regular*** operations, called ***union/or***, ***concatenation*** and ***star***.

Brackets (and) are used for grouping, just as in normal math.

Union

The symbol $+$ means ***union*** or ***or***.

Example:

`0` $+$ `1`

means either a zero or a one.

Concatenation

The ***concatenation*** of two REs is obtained by writing the one after the other.

Example:

$$(\mathtt{0}+\mathtt{1})\,\mathtt{0}$$

corresponds to $\{\mathtt{00}, \mathtt{10}\}$.

$$(\mathtt{0}+\mathtt{1})\,(\mathtt{0}+\varepsilon)$$

corresponds to $\{\mathtt{00}, \mathtt{0}, \mathtt{10}, \mathtt{1}\}$.

Star

The symbol $*$ is pronounced star and means zero or more copies.

Example:

$$\texttt{a}^*$$

corresponds to any string of a's: $\{\varepsilon, \texttt{a}, \texttt{aa}, \texttt{aaa}, \ldots\}$.

$$(\texttt{0} + \texttt{1})^*$$

corresponds to all binary strings.

Example

An RE for the language of all binary strings of length at least 2 that begin and end in the same symbol.

$$\mathtt{0}(\mathtt{0}+\mathtt{1})^*\mathtt{0} + \mathtt{1}(\mathtt{0}+\mathtt{1})^*\mathtt{1}$$

Note ***precedence*** of regular operators: *star* always refers to smallest piece it can, *or* to largest piece it can.

Example

Consider the regular expression

$$((\mathtt{0}+\mathtt{1})^*\mathtt{1}+\varepsilon)\,(\mathtt{00})^*\,\mathtt{00}$$

This RE is for the set of all binary strings that end with an even nonzero number of $\mathtt{0}$'s.

Note that different language to:

$$(\mathtt{0}+\mathtt{1})^*\,(\mathtt{00})^*\,\mathtt{00}$$

Regular Operators for Languages

If one forms RE by the ***or*** of REs R and S, then result is union of R and S.

If one forms RE by the ***concatenation*** of REs R and S, then the result is all strings that can be formed by taking one string from R and one string from S and concatenating.

If one forms RE by taking the ***star*** of RE R, then the result is all strings that can be formed by taking any number of strings from the language of R (possibly the same, possibly different), and concatenating.

Regular Operators Example

If language L is $\{\texttt{ma}, \texttt{pa}\}$ and language M is $\{\texttt{be}, \texttt{bop}\}$, then

$L + M$ is $\{\texttt{ma}, \texttt{pa}, \texttt{be}, \texttt{bop}\}$;

LM is $\{\texttt{mabe}, \texttt{mabop}, \texttt{pabe}, \texttt{pabop}\}$; and

L^* is $\{\varepsilon, \texttt{ma}, \texttt{pa}, \texttt{mama}, \ldots, \texttt{pamamapa}, \ldots\}$.

Notation: If Σ is some alphabet, then Σ^* is the set of all strings using that alphabet.

An RE for Decimal Numbers

English: “Some digits followed maybe by a point and some more digits.”

RE:

$$(\texttt{-} + \varepsilon)\, \texttt{D}\, \texttt{D}^* \,(\varepsilon + \texttt{.}\, \texttt{D}^*)$$

where D stands for a digit.

Kleene's Theorem

Kleene's Theorem. *There is an FA for a language if and only there is an RE for the language.*

Proof (to come) is algorithmic.

Regular language is one accepted by some FA or described by an RE.

Applications of REs

- Specify piece of programming language, e.g. real number. This allows automated production of ***tokenizer*** for identifying the pieces.
- Complex search and replace.
- Many UNIX commands take regular expressions.

Practice

Give an RE for each of the following three languages:

1. All binary strings with at least one `0`
2. All binary strings with at most one `0`
3. All binary strings starting and ending with `0`

Solutions to Practice

1. $(\texttt{0}+\texttt{1})^*\texttt{0}(\texttt{0}+\texttt{1})^*$
2. $\texttt{1}^*+\texttt{1}^*\texttt{01}^*$
3. $\texttt{0}(\texttt{0}+\texttt{1})^*\texttt{0}+\texttt{0}$

In each case several answers are possible.

Summary

A regular expression (RE) is built up from individual symbols using the three Kleene operators: union ($+$), concatenation, and star (*). The star of a language is obtained by all possible ways of concatenating strings of the language, repeats allowed; the empty string is always in the star of a language.