**Regular expressions**

Regular expressions are expressions denoting certain [languages](http://lara.epfl.ch/w/strings_and_languages). They are precisely those languages that can be built from the singleton languages $\{\epsilon\}$, and $\{a\}$for each $a \in \Sigma$, using operations union ( $\cup$), concatenation ( ${\cdot}$), and iteration ( $*$) on languages.

In regular expressions, $\cup$is typically denoted $+$(sometimes $|$), and the curly braces around singleton sets are omitted, so the language $\{a\}$for $a \in \Sigma$is denoted simply $a$and the language $\{\epsilon\}$by $\epsilon$. The concatenation operator is simply omitted. We next make this more precise. We write $L(r)$to denote the set that denotes the meaning of a regular expression.

A **regular expression** over alphabet $\Sigma$is given by:

* $\emptyset$is a regular expression, $L(\emptyset) = \emptyset$
* $\epsilon$is a regular expression $L(\epsilon) = \{\epsilon\}$
* if $a \in \Sigma$, then $a$is a regular expression $L(a) = \{a\}$
* if $r_1$and $r_2$are regular expressions, so are $r_1 r_2$, $r_1 + r_2$, and $r_1*$; we define $L(r_1 r_2) = L(r_1) \cdot L(r_2)$, $L(r_1 + r_2) = L(r_1) \cup L(r_2)$, $L(r_1*) = L(r_1)^*$
* every regular expression is obtained by a finite number of applications of the rules above.

**Example:** let $\Sigma = \{a,b,r\}$. The regular expression

(bar)\*(a+b)\*

denotes the language $\{ (bar)^n w \mid n \geq 0, w \in \{a,b\}^* \}$.

**Example:** Integer constants (such as 3123, 9123, 1389123) that do not begin with digit zero are given by (using ‘|’ notation instead of ‘+’):

(1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)\*

**Example:** let $\Sigma = \{ b, e, r, c, o, k \}$

(be\*r + coke)\*

denotes the set of strings that include:

br, ber, beer, beeer, ... , coke, brcoke, ... , berbrcokecokebeeer

Notes:

* we can introduce shorthand r+ to stand for rr\*
* we can allow non-recursive shorthands, for example

posDigit = 1|2|3|4|5|6|7|8|9

digit = 0|posDigit

intConst = posDigit digit\*