LEX regular expressions

LEX REGULAR EXPRESSIONS. A LEX regular expression is a word made of

- text characters (letters of the alphabet, digits, ...)
- operators : `" \ { } [ ] ^ $ < > ? . * + | () /`

Moreover

- An operator can be used as a text character if it preceeded with the escape operator (backslash).
- The quotation marks indicate that whatever is contained between a pair of quotes is to be taken as text characters. For instance

  ```
  xyz"++"
  ```

  matches the string `xyz++`.

A CHARACTER CLASS is a class of characters specified using the operator pair `[ ]`. The expression

```
[ab]
```

matches the string `a` or `b`.

Within square brackets most operators are ignored except the three special characters `\ - ^` are which used as follows

(a)
the escape character `\` as above,

(b)
the minus character `-` which is used for ranges like in

```
digit       [0-9]
```

(c)
the *hat* character `^` as first character after the opening square bracket, it is used for complemented matches like in

```
NOTabc      [^abc]
```

OPTIONAL EXPRESSIONS. The `?` operator indicates an optional element of an expression. For instance

```
ab?c
```

matches either `ac` or `abc`.

REPEATED EXPRESSIONS. Repetitions of patterns are indicated by the operators `*` and `+`.

- The pattern `a*` matches any number of consecutive `a` characters (including zero).
- The pattern `[a-z]+` is any positive number of consecutive lower-case alphabetic characters.

Hence we can recognize identifiers in a typical computer language with

```
[A-Za-z][A-Za-z0-9]*
```

Repetitions can also be obtained with the pair operator {}.

- If {} encloses numbers, it specifies repetitions. For instance `a{1,5}` matches 1 to 5 repetitions of `a`.
- **Note** that if {} encloses a name, this name should be defined in the definition section. Then LEX substitutes the definition for the name.

ALTERNATING. The operator `|` indicates alternation. For instance

```
(ab|cd)
```

matches the language consisting of both words `ab` and `cd`.

GROUPING. Parentheses are used for grouping (when not clear). For instance

```
(ab|cd+)?(ef)*
```

denotes the language of the words that are either empty or

- optionally starts with
  - `ab` or
  - `c` followed by any positive number of `d`
- and continues with any number of repetition of `ef`

Another example: an expression specifying a real number

```
-?(([0-9]+)|([0-9]*\.[0-9]+)([eE][-+]?[0-9]+)?)
```

where `\.` denotes a literal period.

CONTEXT SENSITIVITY. LEX provides some support for contextual grammatical rules.

- If `^` is the first character in an expression, then this expression will only be matched at the beginning of a line.
- If `$` is the last character in an expression, then this expression will only be matched at the end of a line.
- If *r* and *s* are two LEX regular expressions then *r*/*s* is another LEX regular expression.
  - It matches *r* if and only if it is followed by an *s*.
  - It is called a *trailing context*.
  - After use in this context, *s* is then returned to the input before the action is executed. So the action only sees the text matched by *r*
  - *Left context* is handled by means of *start conditions* which we will talk about later.

Marc Moreno Maza
2004-12-02