The Language XLE$_R ULES$

BNF-converter

October 25, 2016

This document was automatically generated by the *BNF-Converter*. It was generated together with the lexer, the parser, and the abstract syntax module, which guarantees that the document matches with the implementation of the language (provided no hand-hacking has taken place).

The lexical structure of XLE$_R ULES$

Identifiers

Identifiers ⟨*Ident*⟩ are unquoted strings beginning with a letter, followed by any combination of letters, digits, and the characters _ ', reserved words excluded.

Literals

Reserved words and symbols

The set of reserved words is the set of terminals appearing in the grammar. Those reserved words that consist of non-letter characters are called symbols, and they are treated in a different way from those that are similar to identifiers. The lexer follows rules familiar from languages like Haskell, C, and Java, including longest match and spacing conventions.

The reserved words used in XLE$_R ULES are the following$:

There are no reserved words in XLE$_R ULES$.

The symbols used in XLE$_R ULES are the following$:

1

```
-->   =   .
*     (   )
{     }   :
;     ^   !
@     $   |
```

Comments

There are no single-line comments in the grammar.
Multiple-line comments are enclosed with /* and */, " and ".

The syntactic structure of $\mathbf{XLE}_R ULES$

Non-terminals are enclosed between $\langle$ and $\rangle$. The symbols ::= (production),
| (union) and ϵ (empty rule) belong to the BNF notation. All other symbols
are terminals.

$\langle GRAMMAR \rangle$ ::= $\langle ListRULE \rangle$

$\langle RULE \rangle$ ::= $\langle LHS \rangle$ --> $\langle ListRHS \rangle$ $\langle RULEES \rangle$
  | $\langle LHS \rangle = \langle ListRHS \rangle$ $\langle RULEES \rangle$

$\langle RULEES \rangle$ ::= .

$\langle LHS \rangle$ ::= $\langle Ident \rangle$

$\langle SYMBOL \rangle$ ::= $\langle Ident \rangle$
  | $\langle Ident \rangle$ *

$\langle RHSSYMBOL \rangle$ ::= $\langle Ident \rangle$
  | $\langle Ident \rangle$ *

$\langle RHS \rangle$ ::= $\langle RHSSYMBOL \rangle$
  | ($\langle RHSSYMBOL \rangle$)
  | { $\langle ListORHS \rangle$ }
  | $\langle RHSSYMBOL \rangle$: $\langle ListSCHEM \rangle$
  | $\langle RHSSYMBOL \rangle$: $\langle ListSCHEM \rangle$; $\langle ListRHS \rangle$
  | ($\langle RHSSYMBOL \rangle$: $\langle ListSCHEM \rangle$)

$\langle ORHS \rangle$ ::= $\langle RHSSYMBOL \rangle$
  | $\langle AT \rangle$ ($\langle RHSSYMBOL \rangle$)

$\langle SCHEM \rangle$ ::= $\langle LEFTSCHEMA \rangle = \langle RIGHTSCHEMA \rangle$
  | $\langle LEFTSCHEMA \rangle$

$\langle UP \rangle$ ::=   `^`

$\langle DOWN \rangle$ ::=   `!`

$\langle AT \rangle$ ::=   `@`

$\langle LEFTSCHEMA \rangle$ ::= $\langle UP \rangle$
       | $\langle DOWN \rangle$
       | `(` $\langle UP \rangle$ $\langle SYMBOL \rangle$ `)`
       | `(` $\langle DOWN \rangle$ $\langle SYMBOL \rangle$ `)`
       | $\langle DOWN \rangle$ `$` `(` $\langle UP \rangle$ $\langle SYMBOL \rangle$ `)`
       | `(` $\langle UP \rangle$ $\langle SYMBOL \rangle$ `{` $\langle ListSORHS \rangle$ `}` `)`
       | `(` $\langle DOWN \rangle$ $\langle SYMBOL \rangle$ `{` $\langle ListSORHS \rangle$ `}` `)`

$\langle SORHS \rangle$ ::= $\langle SSYMBOL \rangle$

$\langle SSYMBOL \rangle$ ::= $\langle Ident \rangle$

$\langle RIGHTSCHEMA \rangle$ ::= $\langle SYMBOL \rangle$
       | $\langle DOWN \rangle$

$\langle ListORHS \rangle$ ::= ϵ
    | $\langle ORHS \rangle$
    | $\langle ORHS \rangle$ `|` $\langle ListORHS \rangle$

$\langle ListSORHS \rangle$ ::= ϵ
    | $\langle SORHS \rangle$
    | $\langle SORHS \rangle$ `|` $\langle ListSORHS \rangle$

$\langle ListRHS \rangle$ ::= ϵ
    | $\langle RHS \rangle$ $\langle ListRHS \rangle$

$\langle ListSCHEM \rangle$ ::= ϵ
    | $\langle SCHEM \rangle$ $\langle ListSCHEM \rangle$

$\langle ListRULE \rangle$ ::= ϵ
    | $\langle RULE \rangle$ $\langle ListRULE \rangle$

$\langle ListSYMBOL \rangle$ ::= ϵ
    | $\langle SYMBOL \rangle$ $\langle ListSYMBOL \rangle$