

The Language SW

BNF-converter

June 17, 2019

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 SW

Identifiers

Identifiers $\langle Ident \rangle$ are unquoted strings beginning with a letter, followed by any combination of letters, digits, and the characters `_ ' ,` reserved words excluded.

Literals

Numvar literals are recognized by the regular expression `'%'<letter>(<letter> | <digit>)*`

Stringvar literals are recognized by the regular expression `'$'<letter>(<letter> | <digit>)*`

Symvar literals are recognized by the regular expression `'&'<letter>(<letter> | <digit> | ' _ ')*`

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 SW are the following:

There are no reserved words in SW.

The symbols used in SW are the following:

$$\begin{array}{lcl} ; & ' & \{ \\ \} & < & - \\ - & (&) \\ . & = & \end{array}$$

Comments

Single-line comments begin with #.

Multiple-line comments are enclosed with {# and #}.

The syntactic structure of SW

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.

$$\begin{aligned} \langle ValidSW \rangle & ::= \langle ListStm \rangle \\ \\ \langle Stm \rangle & ::= \langle S-tream \rangle \\ & \quad | \langle Numassgn \rangle \\ & \quad | \langle Strassgn \rangle \\ & \quad | \langle Hermt \rangle \\ & \quad | \langle Subdef \rangle \\ \\ \langle ListStm \rangle & ::= \epsilon \\ & \quad | \langle Stm \rangle ; \langle ListStm \rangle \\ \\ \langle Subdef \rangle & ::= ' \langle Ident \rangle \{ \langle ListSubnet \rangle \} \\ \\ \langle Subnet \rangle & ::= \langle Hermt \rangle \\ & \quad | \langle S-tream \rangle \\ & \quad | \langle ExtPortIn \rangle \\ & \quad | \langle ExtPortOut \rangle \\ \\ \langle ListSubnet \rangle & ::= \epsilon \\ & \quad | \langle Subnet \rangle ; \langle ListSubnet \rangle \\ \\ \langle ExtPortIn \rangle & ::= \langle Snk \rangle \langle Arrow \rangle \langle Tab \rangle \end{aligned}$$

$$\begin{aligned}
\langle \text{ExtPortOut} \rangle & ::= \langle \text{Tab} \rangle \langle \text{Arrow} \rangle \langle \text{Srce} \rangle \\
\langle \text{Tab} \rangle & ::= \langle \text{Numval} \rangle \\
& \quad | \quad \langle \text{Symval} \rangle \\
\langle \text{S-tream} \rangle & ::= \langle \text{Snk} \rangle \langle \text{Arrow} \rangle \langle \text{Srce} \rangle \\
& \quad | \quad \langle \text{S-tream} \rangle \langle \text{Prt} \rangle \langle \text{Arrow} \rangle \langle \text{Srce} \rangle \\
\langle \text{ListS-tream} \rangle & ::= \epsilon \\
& \quad | \quad \langle \text{S-tream} \rangle \\
& \quad | \quad \langle \text{S-tream} \rangle ; \langle \text{ListS-tream} \rangle \\
\langle \text{Arrow} \rangle & ::= < \langle \text{Buffsize} \rangle - \\
\langle \text{Buffsize} \rangle & ::= \langle \text{Numval} \rangle \\
& \quad | \quad \epsilon \\
\langle \text{Srce} \rangle & ::= \langle \text{Prt} \rangle \langle \text{Proc} \rangle \\
\langle \text{Snk} \rangle & ::= \langle \text{Proc} \rangle \langle \text{Prt} \rangle \\
\langle \text{Hermt} \rangle & ::= \langle \text{Ident} \rangle \langle \text{Comp} \rangle \langle \text{ListArgument} \rangle \\
& \quad | \quad \langle \text{Ident} \rangle \langle \text{ListArgument} \rangle \\
& \quad | \quad _ \langle \text{Comp} \rangle \langle \text{ListArgument} \rangle \\
& \quad | \quad _ \langle \text{ListArgument} \rangle \\
\langle \text{Proc} \rangle & ::= (\langle \text{Ident} \rangle \langle \text{Comp} \rangle \langle \text{ListArgument} \rangle) \\
& \quad | \quad (\langle \text{Ident} \rangle \langle \text{ListArgument} \rangle) \\
& \quad | \quad (_ \langle \text{Comp} \rangle \langle \text{ListArgument} \rangle) \\
& \quad | \quad (_ \langle \text{ListArgument} \rangle) \\
\langle \text{Prt} \rangle & ::= \langle \text{Numval} \rangle \\
& \quad | \quad \langle \text{Numval} \rangle . \langle \text{Symval} \rangle \\
& \quad | \quad \langle \text{Symval} \rangle . \langle \text{Numval} \rangle \\
& \quad | \quad \langle \text{Symval} \rangle \\
& \quad | \quad \epsilon \\
\langle \text{Comp} \rangle & ::= \langle \text{Ident} \rangle \\
& \quad | \quad \langle \text{Ident} \rangle . \langle \text{Ident} \rangle \\
& \quad | \quad ' \langle \text{Ident} \rangle \\
\langle \text{Argument} \rangle & ::= \langle \text{Stringval} \rangle \\
\langle \text{ListArgument} \rangle & ::= \epsilon \\
& \quad | \quad \langle \text{Argument} \rangle \langle \text{ListArgument} \rangle \\
\langle \text{Numassgn} \rangle & ::= \langle \text{Numvar} \rangle = \langle \text{Numval} \rangle
\end{aligned}$$

$$\langle \textit{Strassgn} \rangle ::= \langle \textit{Stringvar} \rangle = \langle \textit{Stringval} \rangle$$

$$\begin{array}{l} \langle \textit{Numval} \rangle ::= \langle \textit{Integer} \rangle \\ | \quad \langle \textit{Numvar} \rangle \end{array}$$

$$\begin{array}{l} \langle \textit{Stringval} \rangle ::= \langle \textit{String} \rangle \\ | \quad \langle \textit{Stringvar} \rangle \end{array}$$

$$\begin{array}{l} \langle \textit{Symval} \rangle ::= \langle \textit{Symvar} \rangle \\ | \quad \langle \textit{Ident} \rangle \end{array}$$