[a4paper,11pt]article [T1]fontenc [utf8x]inputenc 0mm 1mm

The Language SW BNF-converter

document

$\epsilon$ [1]1 [1]$\langle 1 \rangle$ ::= | [1]1 [1]1 [1]1

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

*Literals Integer literals Int are nonempty sequences of digits.

String literals String have the form "$x$", where $x$ is any sequence of any characters except " unless preceded by 6.

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

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

Envar literals are recognized by the regular expression '$' '_' letter (letter | $digit$)$*$

Symvar literals are recognized by the regular expression '&' letter (letter | $digit$ | '$\_'$)$*$

SubId literals are recognized by the regular expression '' letter (letter | $digit$ | '$\_'$)$*$

Id literals are recognized by the regular expression letter (letter | $digit$ | '$\_'$)$*$

ValidImport literals are recognized by the regular expression '$\{'(letter \mid digit \mid `\_' \mid `.' \mid `/') * `\}'$

Date literals are recognized by the regular expression digit digit digit digit ('$-$' digit digit)* 'T' (':' | $digit$ | '$\text{-}'$)$*$

*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:

tabularlll ; {}

*Comments Single-line comments begin with #.

*The syntactic structure of SW

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

tabularlll Valide ValidCFG

tabularlll ValidSW ListStm

tabularlll Stm DataFlow

tabularlll ListStm

tabularlll Subdef SubId {$ListSubnet$}

tabularlll Subnet Hermt

tabularlll ListSubnet

tabularlll ExtPortIn Proc Prt Larrow Tab

tabularlll ExtPortOut Tab Larrow Prt Proc

tabularlll Tab Numval

tabularlll DataFlow Proc Prt Larrow Prt Proc

tabularlll Larrow < TypeDef Buffsize $-$

tabularlll Rarrow $-$ TypeDef Buffsize >

tabularlll TypeDef Symvalu

tabularlll Buffsize Numval

tabularlll Hermt Symvalu Comp ListArgument

tabularlll Symvalu Symval

tabularlll Proc ( Symvalu Comp ListArgument )

tabularlll Prt Numval

tabularlll Comp Symval

tabularlll ModPath / Symval /

tabularlll RemPath  ValidImport Symval
tabularlll Argument  Stringval
tabularlll ListArgument
tabularlll Numassgn  Numvar = Numval
tabularlll Strassgn  Stringvar = Symval
tabularlll SymAssgn  Symvar = Symval
tabularlll Numval  Integer
tabularlll Stringval  String
tabularlll Symval  Symvar
tabularlll ValidCFG  StreamWork: ListEntry
tabularlll Entry  KeyVal
tabularlll ListEntry
tabularlll KeyVal  KeyName Integer
tabularlll KeyName  Symval :