

## Fig. 2.7 *Clite* grammar: lexical level

*Identifier*  $\rightarrow$  *Letter* { *Letter* | *Digit* }

*Letter*  $\rightarrow$  a | b | ... | z | A | B | ... | Z

*Digit*  $\rightarrow$  0 | 1 | ... | 9

*Literal*  $\rightarrow$  *Integer* | *Boolean* | *Float* | *Char*

*Integer*  $\rightarrow$  *Digit* { *Digit* }

*Boolean*  $\rightarrow$  true | false

*Float*  $\rightarrow$  *Integer* . *Integer*

*Char*  $\rightarrow$  ' ASCII Char '

(ASCII Char is the set of ASCII characters)

## Fig. 2.7 *Clite* Grammar: Statements

*Program*  $\rightarrow$  *int* *main* ( ) { *Declarations* *Statements* }

*Declarations*  $\rightarrow$  { *Declaration* }

*Declaration*  $\rightarrow$  *Type* *Identifier* [ [ *Integer* ] ] { , *Identifier* [ [ *Integer* ] ] } ;

*Type*  $\rightarrow$  *int* | *bool* | *float* | *char*

*Statements*  $\rightarrow$  { *Statement* }

*Statement*  $\rightarrow$  ; | *Block* | *Assignment* | *IfStatement* | *WhileStatement*

*Block*  $\rightarrow$  { *Statements* }

*Assignment*  $\rightarrow$  *Identifier* [ [ *Expression* ] ] = *Expression* ;

*IfStatement*  $\rightarrow$  *if* ( *Expression* ) *Statement* [ *else* *Statement* ]

*WhileStatement*  $\rightarrow$  *while* ( *Expression* ) *Statement*

## Fig. 2.7 *Clite* Grammar: Expressions

$Expression \rightarrow Conjunction \{ \mid \mid Conjunction \}$

$Conjunction \rightarrow Equality \{ \&\& Equality \}$

$Equality \rightarrow Relation [ EquOp Relation ]$

$EquOp \rightarrow == \mid !=$

$Relation \rightarrow Addition [ RelOp Addition ]$

$RelOp \rightarrow < \mid <= \mid > \mid >=$

$Addition \rightarrow Term \{ AddOp Term \}$

$AddOp \rightarrow + \mid -$

$Term \rightarrow Factor \{ MulOp Factor \}$

$MulOp \rightarrow * \mid / \mid \%$

$Factor \rightarrow [ UnaryOp ] Primary$

$UnaryOp \rightarrow - \mid !$

$Primary \rightarrow Identifier [ [ Expression ] ] \mid Literal \mid ( Expression ) \mid$   
 $Type ( Expression )$