
Concrete Syntax of Clite (from Tucker and Noonan, 2007)

$$\begin{aligned} \text{Program} &\rightarrow \text{int main } () \{ \text{Declarations Statements} \} \\ \text{Declarations} &\rightarrow \{ \text{Declaration} \} \\ \text{Declaration} &\rightarrow \text{Type Identifier } [[\text{Integer}]] \{ , \text{Identifier } [[\text{Integer}]] \} \\ \text{Type} &\rightarrow \text{int} \mid \text{bool} \mid \text{float} \mid \text{char} \\ \text{Statements} &\rightarrow \{ \text{Statement} \} \\ \text{Statement} &\rightarrow ; \mid \text{Block} \mid \text{Assignment} \mid \text{IfStatement} \mid \text{WhileStatement} \\ \text{Block} &\rightarrow \{ \text{Statements} \} \\ \text{Assignment} &\rightarrow \text{Identifier } [[\text{Expression}]] = \text{Expression}; \\ \text{IfStatement} &\rightarrow \text{if } (\text{Expression}) \text{Statement } [\text{else Statement}] \\ \text{WhileStatement} &\rightarrow \text{while } (\text{Expression}) \text{Statement} \\ \\ \text{Expression} &\rightarrow \text{Conjunction } \{ \mid \mid \text{Conjunction} \} \\ \text{Conjunction} &\rightarrow \text{Equality } \{ \&\& \text{Equality} \} \\ \text{Equality} &\rightarrow \text{Relation } [\text{EquOp Relation}] \\ \text{EquOp} &\rightarrow == \mid != \\ \text{Relation} &\rightarrow \text{Addition } [\text{RelOp Addition}] \\ \text{RelOp} &\rightarrow < \mid < = \mid > \mid > = \\ \text{Addition} &\rightarrow \text{Term } \{ \text{AddOp Term} \} \\ \text{AddOp} &\rightarrow + \mid - \\ \text{Term} &\rightarrow \text{Factor } \{ \text{MulOp Factor} \} \\ \text{MulOp} &\rightarrow * \mid / \mid \% \\ \text{Factor} &\rightarrow [\text{UnaryOp}] \text{Primary} \\ \text{UnaryOp} &\rightarrow - \mid ! \\ \text{Primary} &\rightarrow \text{Identifier } [[\text{Expression}]] \mid \text{Literal} \mid \\ &\quad (\text{Expression}) \mid \text{Type } (\text{Expression}) \\ \\ \text{Identifier} &\rightarrow \text{Letter } \{ \text{Letter} \mid \text{Digit} \} \\ \text{Letter} &\rightarrow \text{a} \mid \text{b} \mid \dots \mid \text{z} \mid \text{A} \mid \text{B} \mid \dots \mid \text{Z} \\ \text{Digit} &\rightarrow 0 \mid \dots \mid 9 \\ \text{Literal} &\rightarrow \text{Integer} \mid \text{Boolean} \mid \text{Float} \mid \text{Char} \\ \text{Integer} &\rightarrow \text{Digit } \{ \text{Digit} \} \\ \text{Boolean} &\rightarrow \text{true} \mid \text{false} \\ \text{Float} &\rightarrow \text{Integer} . \text{Integer} \\ \text{Char} &\rightarrow ' \text{ASCIIChar} ' \end{aligned} \tag{1}$$

Abstract Syntax of Clite (from Tucker and Noonan, 2007)

Program = *Declarations* decpart; *Statements* body;
Declarations = *Declaration**
Declaration = *VariableDecl* | *ArrayDecl*
VariableDecl = *Variable* v; *Type* t
ArrayDecl = *Variable* v; *Type* t; *Integer* size
Type = int | bool | floatmidchar
Statements = *Statement**
Statement = *Skip* | *Block* | *Assignment* | *Conditional* | *Loop*
Skip =
Block = *Statements*
Conditional = *Expression* test; *Statement* thenbranch, elsebranch
Loop = *Expression* test; *Statement* body
Assignment = *VariableRef* target; *Expression* source
Expression = *VariableRef* | *Value* | *Binary* | *Unary*
VariableRef = *Variable* | *ArrayRef* (2)
Binary = *Operator* op; *Expression* term1, term1
Unary = *Operator* op; *Expression* term
Operator = *BooleanOp* | *RelationalOp* | *ArithmeticOp*
BooleanOp = && | ||
RelationalOp = == | != | < | < = | > | > =
ArithmeticOp = + | - | * | /
UnaryOp = ! | -
Variable = *String* id
ArrayRef = *String* id; *Expression* index
Value = *IntValue* | *BoolValue* | *FloatValue* | *CharValue*
IntValue = *Integer* intValue
BoolValue = *Boolean* floatValue
FloatValue = *Float* boolValue
CharValue = *Char* charValue