

Traders DSL

Grammar:

$\text{program} \rightarrow \text{declarationList}$

Declarations:

$\text{declarationList} \rightarrow \text{declaration declarationList} \mid \epsilon$

$\text{declaration} \rightarrow \text{envDecl} \mid \text{agentDecl} \mid \text{behaveDecl} \mid \text{varDecl} \mid \text{fieldAssign}$

$\text{envDecl} \rightarrow \text{"env"} \text{ id " {" envBody "}"}$

$\text{agentDecl} \rightarrow \text{"agent"} \text{ id " {" agentBody "}"}$

$\text{behaveDecl} \rightarrow \text{"behave"} \text{ id " {" behaveBody "}"}$

$\text{varDecl} \rightarrow \text{type id " = " expression ";"}$

$\text{fieldAssign} \rightarrow \text{id "." id " = " expression ";"}$

Bodies:

$\text{envBody} \rightarrow \text{varDeclList}$

$\text{agentBody} \rightarrow \text{varDeclList}$

$\text{behaveBody} \rightarrow \text{statementList}$

$\text{varDeclList} \rightarrow \text{varDecl varDeclList} \mid \epsilon$

$\text{statementList} \rightarrow \text{statement statementList} \mid \epsilon$

Statements:

$\text{statement} \rightarrow \text{exprStmt} \mid \text{varDecl} \mid \text{repeatStmt} \mid \text{incaseStmt} \mid \text{primFuncStmt}$

$\text{exprStmt} \rightarrow \text{expression ";"}$

$\text{repeatStmt} \rightarrow \text{"repeat"} \text{ "when"} \text{ expression " {" statementList "}"}$

$\text{incaseStmt} \rightarrow \text{"in"} \text{ "case"} \text{ expression " {" statementList "}" } \text{inothercaseStmt}$

inothercaseStmt \rightarrow "in" "other" "case" expression "{" statementList "}"
 inothercaseStmt | "otherwise" "{" statementList "}" | ϵ

primFuncStmt \rightarrow "print" expression ";" | "move" expression expression ";"
 | "trade" expression expression expression ";" | "find" | "random" expression
 expression

Expressions:

expression \rightarrow logicExpr

logicExpr \rightarrow logicAnd | logicAnd "or" logicExpr

logicAnd \rightarrow equality | equality "and" logicAnd

equality \rightarrow comparison | comparison equalityTail

equalityTail \rightarrow "!=" comparison equalityTail | "==" comparison equalityTail | ϵ

comparison \rightarrow term | term comparisonTail

comparisonTail \rightarrow "<" term comparisonTail | "<=" term comparisonTail
 | ">=" term comparisonTail | ">" term comparisonTail | ϵ

term \rightarrow factor | factor termTail

termTail \rightarrow "+" factor termTail | "-" factor termTail | ϵ

factor \rightarrow unary | unary factorTail

factorTail \rightarrow "*" unary factorTail | "/" unary factorTail | ϵ

unary \rightarrow call | "!" call | "-" call

call \rightarrow primary | id dotTail

dotTail \rightarrow "." idTail | ϵ

idTail \rightarrow id dotTail | listFunc dotTail

listFunc \rightarrow "get" expression | "push" expression | "size" | "pop" | "reverse"

primary \rightarrow "true" | "false" | number | string | "(" expression ")"

Lexical Grammar

$\text{type} \rightarrow \text{"number"} \mid \text{"bool"} \mid \text{"string"} \mid \text{"list"}$