Language 2: Abstract Syntax

<program> ::= **program <id> is** <block>

Program ← Program(program, **id** , Block)

<block> ::= <declseq> **begin** <com-seq> **end**

Block ← Block(Declseq , ComSeq)

<declseq> ::= <decl> **|** <declseq> **,** <decl>

DeclSeq ← Decl$^+$ SB **,**

<decl> ::= <type> **<id> | const** <type> **<id> =** <expr>

Decl ← var(type, **id**) **|** const(type, **id** , Expr))

<com-seq> ::= <com> **|** <com-seq> **;** <com>

ComSeq ← Com$^+$ SB **;**

<com> ::= **if** <expr> **then** <com-seq> **end if | <id> :=** <expr> **| write** <expr> **| read <id>**

Com ← IfThen (Expr , ComSeq) **|** Assign(**id** , Expr) **|** Write(Expr) **|** Read(**id**)

<expr> ::= **num | true | false | character**

<type> ::= **int | bool | char**