Mini-Pascal Grammar

Start 🡪 **program id (** identifier\_list **) ;** declarations subprogram\_declarations compound\_statement **.**

Identifier\_list 🡪 **id** more\_identifiers

More\_identifiers 🡪 **,** identifier\_list | <empty>

Declarations 🡪 **var** identifier\_list **:** type **;** declarations | <empty>

Type 🡪 standard\_type | **array [num .. num] of** standard\_type

Standard\_type 🡪 **integer** | **real**

Subprogram\_declarations 🡪 subprogram\_declaration **;** subprogram\_declarations | <empty>

Subprogram\_declaration 🡪 subprogram\_head declarations compound\_statement | <empty>

Subprogram\_head 🡪 **function id** arguments **:** standard\_type **;** | **procedure id** arguments **;**

Arguments 🡪 **(** parameter\_list **)** | <empty>

Parameter\_list 🡪 identifier\_list **:** type more\_parameters | <empty>

More\_parameters 🡪 **;** parameter\_list | <empty>

Compound\_statement 🡪 **begin** statement\_list **end**

Statement\_list 🡪 statement more\_statements | <empty>

More\_statements 🡪 **;** statement\_list | <empty>

Statement 🡪 variable **assignop** expression | procedure\_statement | compound\_statement | **if** expression **then** statement **else** statement | **while** expression **do** statement

Variable 🡪 **id** index

Index 🡪 **[** expression **]** | <empty>

Procedure\_statement 🡪 **id** rest\_of\_procedure

Rest\_of\_procedure 🡪 **(** expression\_list **)** | <empty>

Expression\_list 🡪 expression more\_expressions

More\_expressions 🡪 **,** expression\_list | <empty>

Expression 🡪 simple\_expression rest\_of\_expression

Rest\_of\_expression 🡪 **relop** simple\_expression | <empty>

Simple\_expression 🡪 sign term rest\_of\_simple\_expression

Rest\_of\_simple\_expression 🡪 **addop** term | <empty>

Sign 🡪 **+** | **-** | <empty>

Term 🡪 factor rest\_of\_term

Rest\_of\_term 🡪 **multop** term | <empty>

Factor 🡪 procedure\_statement | **num** | **(** expression **)** | **not** factor