CHAPTER 9.

THE SIMULA SYNTAX.

9.1 The SIMULA Reference Language.

In the present chapter the syntax of a SIMULA reference language is defined. The language is somewhat more general than SIMULA as described in the preceding chapters. The concepts concerned are SIMULA blocks, activities, and sets.

The syntax is given as an extension of the ALGOL 60 syntax. Syntactic classes not defined here are understood to be those defined in the ALGOL 60 report. The following syntactic classes are understood to be redefined. The notation <as in the report> refers to the right hand side of the corresponding definition in the ALGOL 60 report. The section numbers are those of the report.

Section 3. Expressions.

<expression>::=<as in the report>|<element expression>|
<set designator>

Section 3.4.1. Boolean expressions.

<relation>::=<as in the report>|
<element expression>=<element expression>|
<element expression>≠<element expression>

Section 4.1.1. Statements.

<unlabelled basic statement>::=<as in the report>|<SIMULA
statement>

Section 4.6.1. For statements.

<for list element>::=<as in the report>|<element expression>|
<element expression> while <Boolean ex-
pression>

Section 5. Declarations.

<declaration>::=<as in the report>|<activity declaration>

Section 5.1.1. Type declarations.

<type>::=<as in the report>|element|set

The following additional basic symbols are introduced.

<SIMULA basic symbol>::= SIMULA|activity|element|set|
activate|reactivate|at|delay|prior|
before|after|inspect|extract|
when|otherwise|new|none

9.2 Expressions.

9.2.1 Element expressions.

<activity identifier>::=<identifier>
<process designator>::=<activity identifier><actual parameter part>|
new<activity identifier><actual parameter part>
<simple element expression>::= none|<variable>|<function designator>|
<process designator>|<activity identifier>
<element expression>::=<simple element expression>|
<if clause><simple element expression>
else<element expression>

9.2.2 Set designators.

<simple set designator>::=<variable>
<set designator>::=<simple set designator>|
<if clause><simple set designator>
else<set designator>

9.3 SIMULA statements.

<SIMULA statement>::=<SIMULA block>|<scheduling statement>|
<connection statement>

9.3.1 SIMULA blocks.

<SIMULA block>::= SIMULA<unlabelled block>

9.3.2 Scheduling statements.

<activator>::= activate|reactivate
<activation clause>::=<activator><element expression>
<simple timing clause>::= at<arithmetic expression>|
delay<arithmetic expression>
<timing clause>::=<simple timing clause>|
<simple timing clause>prior
<scheduling clause>::=<empty>|<timing clause>|
before<element expression>|
after<element expression>
<scheduling statement>::=<activation clause><scheduling clause>

9.3.3 Connection statements.

<connector>::= inspect|extract
<inspection clause>::=<connector><element expression>
<connection block>::=<statement>
<connection clause>::= when<activity identifier>do<connection block>
<simple connection part>::=<connection clause>|
<simple connection part><connection clause>
<connection part>::=<simple connection part>|
<simple connection part>otherwise<statement>
<connection statement>::=<inspection clause><connection part>|
<connector><process designator>do<connection block>|
<scheduling statement><connection part>|
<activator><process designator><scheduling clause>
do<connection block>

A symbol "when" or "otherwise" refers back to the nearest <inspection clause> or <scheduling statement>. This is sufficient to avoid ambiguity if a <connection block> is itself a <connection statement>.

9.4 Activity Declarations.

<activity heading>::=<activity identifier><formal parameter part>;
<value part><specification part>
<activity body>::=<statement>
<activity declaration>::= activity<activity heading><activity body>

9.5 Syntax Restrictions.

SIMULA, as defined in CHAPTERS 2-6 has the following syntax restrictions compared to the reference language.

1. The SIMULA basic symbols are only available within a SIMULA block, with the exception of the symbol "SIMULA".

2. SIMULA blocks must not be nested.

3. Activity declarations are only permitted in a SIMULA block head.

4. The parameter mechanism of an activity declaration is restricted to <type>, array, and <type>array parameters.

5. The <value part> of an <activity heading> is omitted. (Call by value is assumed for <type> parameters, call by name otherwise.)

6. Assignment statements of the form
<variable>:=<set designator>
are not permitted.

7. The set procedure concept is excluded.

Obvious and possibly fruitful extensions can be made to SIMULA by removing some of the above restrictions. The removal of others would require a careful analysis of the semantics involved. The semantics of the unrestricted SIMULA reference language can be defined in more than one way.