308

DESIGN AND IMPLEMENTATION OF S-COMPILER INTERPASS

by

Mija Chung Kim

B.A., Seoul National University, 1975
M.A., Seoul National University, 1978

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

Approved by:

Rodney M. Bates

Major Professor

CONTENTS

Chapter 1

INTRODUCTION

In 1983 Spring, an implementation project was held under the direct of Dr. Rodney M. Bates. The goal of the project was to implement a portable SIMULA system(S-Port) on Perkin Elmer interdata machines.

This report describes a design of Interpass which is a main part of the compiler to be used in the future[1] Perkin Elmer portable SIMULA system, and is organized as follows.

Chapter 1 is an overall introduction to the subject matter of this report.

Chapter 2 provides a brief history and some characteristics of SIMULA language.

Chapter 3 introduces S-Port, that is, what this system consists of and what is the main function of each part of the system.

Chapter 4 discusses the overall design of the Perkin Elmer portable SIMULA system.

Chapter 5 illustrates the Syntax/Semantics language(S/SL) which is used to construct the Interpass.

Chapter 6 discusses the design of Interpass by providing with the major data structures and the semantic operations.

---

[1] This system is not available yet.

Chapter 7 shows some intresting problems which we met during the process of interpass design and implementation.

Chapter 8 concludes with a summary of the present status of the Interpass, i.e., what is done up until now, what can be done to make this Interpass design more perfect, and what we contribute to the future work.

Finally there is a reference section and an appendix of S/SL program listing which we implemented.

Chapter 2

SIMULA LANGUAGE

This chapter describes SIMULA language. It is organized in two parts. The first part introduces the historical development of SIMULA[2] and the second part discusses the principal features of SIMULA.

2.1 Development of SIMULA

SIMULA (SIMUlation LAnguage) is a general purpose programming language developed by Ole-Johan Dahl (OJD) and Kristen Nygaard (KN) of the NCC in 1967 under a contract with the Univac Division of Sperry Rand Corporation. The language is a true extension of ALGOL 60, i.e., it contains ALGOL 60 as a subset. However, SIMULA is historically an extended version of the SIMULA I which was also developed by OJD and KN at the NCC during the years of 1961-1964.

The initiating ideas for SIMULA I originated at the Norwegian Defense Research Establishment (NDRE) in the late fifties. At that time, the NDRE started work in the design of a language which could serve the dual purpose of system description and simulation programming. But the major development of this language was carried out at the NCC from the spring of 1961.

---

[2] In this report, SIMULA indicates SIMULA 67. Because SIMULA 67 is the most prevalent version of the language today, and it is commonly called SIMULA.

SIMULA I was initially designed to provide a systems analyst with unified concepts which facilitate the concise description of discrete event systems. In other words, the initial ideas of this language were based upon a mathematically formulated "discrete event network" concept, as evidened by the title of the IFIP 1962 Munich paper, "SIMULA---an extension of ALGOL to the description of discrete event networks".

But as the language developed, the network concept was found to have shortcomings. The language designers of SIMULA I envisaged the important examples of systems which could not naturally be regarded as networks. The result of this was the abandonment of the "network" concept and the introduction of the "process" concept as the basic, unifying concept. That was in February 1964. Language designers no longer regarded a sytem as described by a "general mathematic structure" and instead understood it as a variable collection of interacting processes--each process being present in the program execution, the simulation, as an ALGOL stack. Graphical representaions of simplified(but structurally identical) versions of the program executions were used as models of the systems described by the language.

During 1965 and the beginning of 1966, OJD and KN spent most of the time using and introducing SIMULA I. But they became more and more interested in the possibilities of

SIMULA I as an existing[3] general purpose programming language. That was the starting point of SIMULA idea.

Language designers explored SIMULA I's list structuring and coroutine capabilities, the use of procedure attributes etc. Very soon they realized that important shortcomings existed in the language.

1. The inspect mechanism for remote attribute accessing turned out to be very cumbersome in some situations.

2. When writing simulation programs, processes often shared a number of common properties, both in data attributes and actions, but were structurally different in other respects so that they had to be described by seperate declarations.

In order to solve these problems, Hoare's record class construct was introduced along with the idea of "prefixing". Each of the processes would be a block instance consisting of two layers: a prefix layer containing a successor and predecessor pointer, and other properties associated with two-way list membership, and a main part containing the attributes.

With the gradual development that we described above, SIMULA was born in 1967.

2.2 Principal Features of SIMULA

---

[3] "existing" in the sense that it was available on most of the major computer systems, being used over a long period of time by a substancial number of people throughout the world and having a significant impact upon the development of future programming languages

SIMULA is an extension of ALGOL 60. With some exceptions, the rules of ALGOL 60 have been preserved in SIMULA. The definitions of certain syntactic units, however, have been extended to account for the new concepts of SIMULA.

For example, as in ALGOL, the head of a SIMULA block contains declarations. However, in SIMULA these can be the declarations of classes as indicated by the redefinition of the syntactic unit declaration.

The principal features which convert ALGOL 60 to SIMULA provide the ability to

1. Declare a class,
2. Generate objects of a declared class,
3. Reference variables,
4. Form a hierarchical structure of class declarations.

2.2.1 Class Declaration

The notion of class in SIMULA can be traced back to the notions of block and block instance in ALGOL 60. An ALGOl block is a description of a composite data structure and associated algorithms. When a block is executed, a dynamic instance of that block is generated. The block instance is the composite data structure described by the block and contains the local variables of the block as well as information needed for the dynamic linkage to other blocks. It is possible to have interaction between instances of different blocks and even instances of the same block in the case of recursive procedure call.

The form of a class declaration in SIMULA parallels the form of a procedure declaration in ALGOL 60. A class declaration can have the following form.

```
class A (PA); SA;
begin DA;
    IA
    ; inner ;
    FA
end
```

The identifier A is the name of the class; PA is the formal parameter list of the class A and it contains only parameter names; SA is the list of specifications of the parameters in PA.

The class body begins with a list of declarations DA. The symbols IA and FA represents lists of instructions respectively called initial operations and final operations of the class A. The symbcl "inner" represents a dummy instruction acting as a separator between the initial and final operations.

The quantities passed as parameters in PA or declared in DA are called the attributes cf the class A and hence also attributes of any object of that class.

2.2.2 Object Generation

A central new concept in SIMULA is the "object". An object is a self contained program, having its own local data and actions defined by a class declaration.

The notion of object in SIMULA originated from the notions of record class and record introduced by Hoare and Wirth.

The expression

new A (...)

where A is a class, is an object generator. When encountered, it creates an object which is an instance of class A and starts execution of the initial operations of A. The execution continues until the end of the class body is encountered, at which time the execution terminated.

2.2.3 Reference Variables and Remote Identifiers

Each element of a simulation has attributes whose values determine the state of the element. Since elements are specified in SIMULA by the class declaration, their attributes are naturally represented by the variable which appears in the declaration.

When an object is executed, some of the actions performed involve attributes of the object itself. This is called local access of attributes and the usual rules of ALGOL apply.

The term remote access is used for the accessing of attributes of other objects. In this case a given attribute is not uniquely specified by merely mentioning its attribute identifier. It is also necessary to specify the object to which the considered attribute belongs. A remote access is thus performed in two steps.

1. The selection of a particular object,

2. The determination of the attribute of the selected object.

A new type of variable, the reference variable, is used for referring to objects. A reference to a newly created object is obtained during the execution of an object generator. This reference is analogous to a pointer. The language permits the calculation and assignment of references by introducing special operators

:- reference assignment

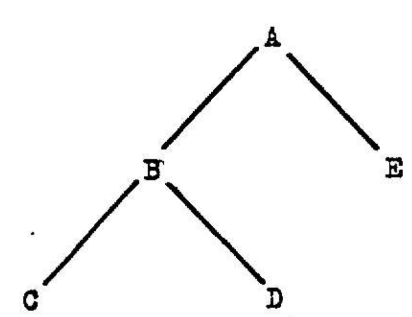
== reference equality

=/= reference inequality

to work with reference variables.

2.2.4 Hierarchical Declarations

A class declaration may be preceded by a prefix which is the name of another class. The prefixed class is now called a subclass of the



Hierarchy of classes and subclasses

prefix. The following is a declaration for the subclass B of the previously declared class A.

```
A class B(PB); SB;
 begin DB;
    IB
    ; inner ;
    FB
end
```

More generally a hierarchy of classes can be introduced by a succession of declarations of subclasses as follows.

```
class A...;
A class B...;
B class C...;
B class D...;
A class E...;
```

A graph corresponding to this hierarchy is an oriented tree. The root of the tree is the class that has no prefix. The prefix sequence of a given class is the sequence of classes encountered on the unique path going from the given class to the root. In the above example the prefix sequence of class C consists of C, B, and A.

A subclass is said to be inner to its prefixes. Thus C is inner to B and A. Conversely B and A are said to be outer to C.

Chapter 3

PORTABLE SIMULA SYSTEM

This chapter provides the information about the portable SIMULA system(S-Port) and discusses the major parts of that system.

3.1 Components of S-Port

S-Port is a portable implementation of the SIMULA language. It is initiated by the Norwegian Computing Center(Oslo) and the Program Library Unit(Edinburgh) in 1979. The system consists of three parts:

1. a portable front-end compiler,
2. a portable run-time support system, and
3. a machine dependent code generator.

The portable front-end compiler translates a source program written in SIMULA into the intermediate language S-code. The compiler itself is written in SIMULA.

The portable run-time support system is used to support the storage allocation, creation and management of objects and activation records, and garbage collection.

Both the front-end compiler and the run-time system will be distributed in the intermediate language S-code. This language is used to transmit an analysed SIMULA program from the front-end compiler to the code generator.

The code generator is a unique machine dependent part of the system and this will produce code for the target machine corresponding to the S-program. This code generator is often called a back-end compiler by contrast with the front-end compiler.

In order to be a complete SIMULA system, S-port is to be supported by this back-end compiler, i.e., code generator. The task of the back-end compiler is to translate the programs in S-code into the machine language of the given machine. So, from now on, we shall call that compiler S-compiler. This compiler must insert necessary links to the operating system of the object machine.

As we described, the Portable SIMULA system consists of three main parts, two language dependent parts and a target dependent part. But, in many cases, the system needs services that can only be provided through the operating system environment, e.g., file handling and interrupt monitoring.

The run-time system requires information about the options selected by the user, the current mode of operation and the environmental conditions applicable to the particular implementation.

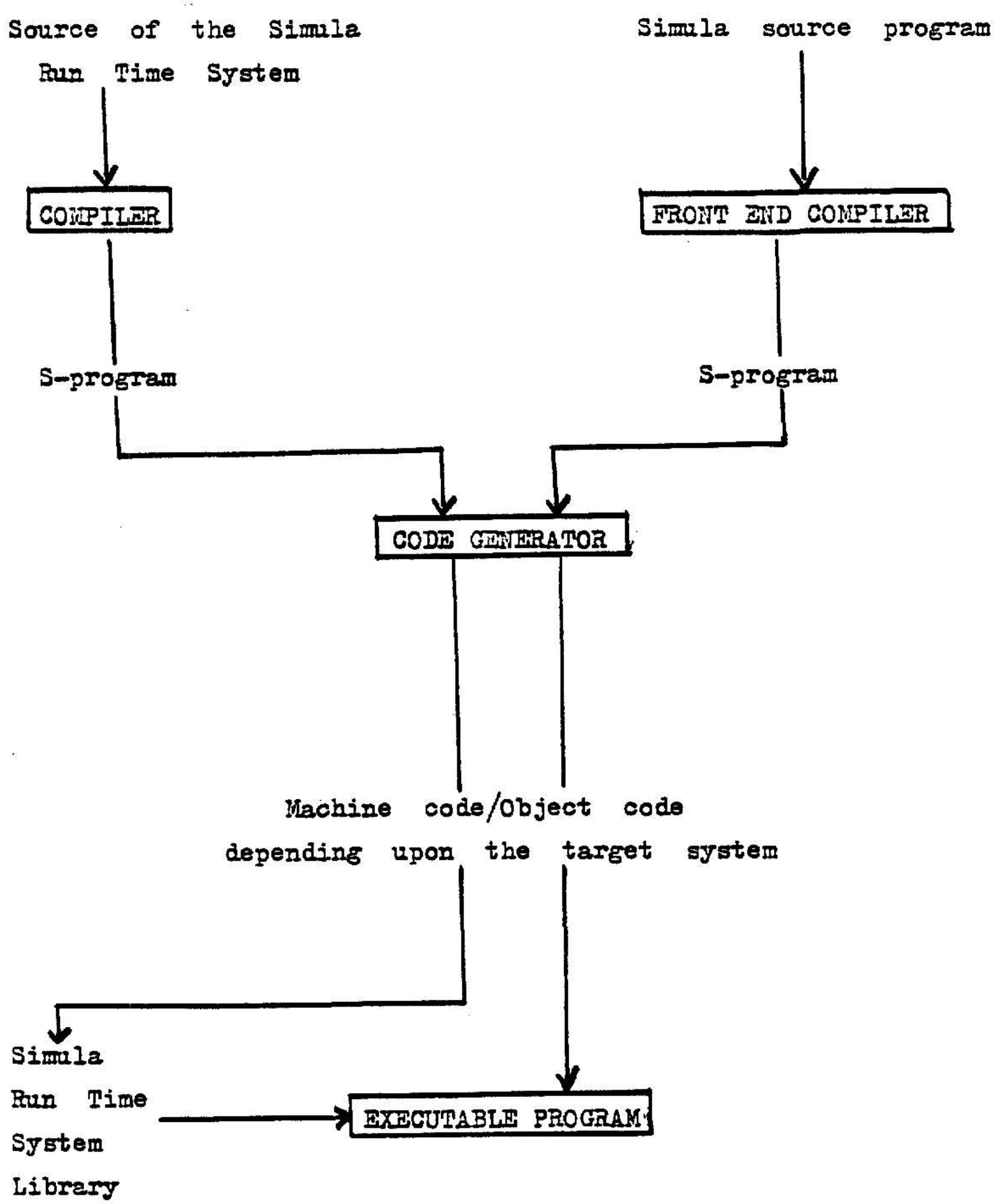
The front-end compiler also requires information about the compile time options selected by the user, and some information which is only known in each implemented system, such as default file names.

The environment interface support package provides system dependent services to the S-Port through Environment Interface. The Environment Interface itself is system independent and is organized according to the S-Code standards.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

General Overview of S-Port



Chapter 4

PERKIN ELMER SIMULA SYSTEM

This chapter describes the overall design of the Perkin Elmer portable SIMULA system. This system is going to be implemented as a complete portable SIMULA system on Perkin Elmer interdata machines. It is to be run under the Unix operating system.

4.1 General Idea

S-Port, which was brought from Norway, provides a portable front-end compiler and a portable run-time support system. However, in order to be a complete portable SIMULA system, it is necessary to develop a machine dependent code generator, i.e., S-Compiler. Also, it is necessary to provide an environment interface support package compatible with the operating system of these machines.

We are going to write S-Compiler which translates the intermediate language S-Code to the 8/32 interdata code. Also we shall provide the Environment Interface which contains utility routines and interfaces to operating system.

Any SIMULA program can be translated into S-Code by the front-end compiler. Then the translated version of S-Code can be compiled by the S-Compiler, linked with the machine code version of the run time system and Environment Interface. This produces the final object code of the original SIMULA program.

4.2 S-Compiler Design

The S-Compiler written for Perkin Elmer machines consists of two parts:

1. a brand new pass Interpass, and
2. a set of modified passes of the Pascal/32 compiler.

4.2.1 Interpass

Interpass is an independent pass, and it translates the token stream* of S-code into the intermediate language of a modified version of Pass 6 of the PASCAL/32 compiler.

Interpass recognizes the operations indicated by the S-code and releases the appropriate tokens for the input stream of modified Pass 6. Unlike the original passes of Pascal/32 compiler, this pass, in certain cases, does not release the tokens at the end of the output stream. Instead through the use of pointers, the tokens are inserted into the middle of the output stream.

Interpass also allocates data space for the constants and for the global or local variables within the respective areas.

This pass uses S/SL as the main controlling language and this S/SL program calls a set of semantic operations coded in Sequential Pascal. S/SL interpreter for the Interpass is also written in Pascal.

4.2.1 Modified passes of Pascal/32 Compiler

---

* Actually, it is a file of integers in binary byte form.

We use passes 6, 7, 8, and 9 of the Pascal/32 compiler which are written in Pascal, and modify especially Pass 6 and Pass 7. This section describes the major functions performed by the passes 1 through 9 and gives an account of modification of Pass 6, Pass 7 and Pass 8.

I. Pascal/32 Compiler

Pascal/32 compiler is an extensive modifications of Hartmann compiler for the concurrent and sequential Pascal languages. It is composed of 9 passes.

Pass 1 of the Pascal/32 compiler is the lexical scanner. It converts the character input to the compiler into the tokens recognized by the parser in Pass 2.

Pass 2 is for syntax analysis. This pass verifies the syntax of the intermediate code generated by Pass 1. The output of this pass is in the form of a polish postfix notation, the operands followed by the operator.

The main function of Pass 3 of the Pascal/32 compiler is name anlysis. This pass applies the language scope rules to convert references to an identifier into the references to the appropriate unique objects in each part of the program.

Pass 4 is for declaration analysis. This pass performs storage allocation and outputs data addressing information along with data type information in the intermediate code.

The main function of Pass 5 is operand analysis. This pass performs operand type checking on all of the operators in the program. This pass applies the operand compatibility rules of the language. It produces output which is effectively the assembler language for a stack machine.

Pass 6 of the Pascal/32 compiler is for program logic. It is designed primarily to perform constant folding, i.e., performing operations on constants at compile time rather than at run time. However, since this is the first pass to build entire procedures as in-memory data structures, it performs several other functions which either perform various other optimizations or rearrange intermediate code to simplify the operation of the later passes.

The main function of Pass 7 of the Pascal/32 compiler is Interdata 8/32 code generation. The input is essentially the instructions of the virtual stack machine output by Pass 5. The actual output from pass 7 is a set of 8/32 instructions in unformatted form.

Pass 8 performs machine dependent code optimization using the peephole technique. This pass makes instruction choices depending upon the length of the instruction. It maps the code labels to the actual machine addresses by generating a table.

Pass 9 is for code optimization. This pass converts the machine code into the format accepted by the linkage loader. It uses the table supplied by Pass 8 to replace the labels with the addresses.

II. Modification of Code Generator Passes

Modification of Pass 6

The data structure used in Pass 6 is a stack of lists of trees. And there's some optimization back to token stream. Each result operand is used only once, so this structure is suitable. The "dup" token of S-Code requires the duplication of its operand. It is not possible to represent this operation by a tree. Hence, Pass 6 is modified to generate graphs rather than trees.

Also, new tokens[5] must be included in the input token stream of thie pass for the language features which are present in S-Code but not in Pascal.

Finally, there's a modification about Pass 6 has to be able to read a mapfile and do random access to the main file containing tokens.

Modification of Pass 7

This pass is designed to hold the descriptors of the runtime operand on the compile time stack as long as they are not computed. The register containing the operand is freed after the computation of the result is complete. In order to process the tokens like "save" and "restore" in S-code, Pass 7 has to be modified to be able to save and restore with the temporary values.

---

[5] For the list of these new tokens, look at S/SL program listing under "pass 6 New Input Tokens".

Also, this pass is for code generation, so it has to generate code for the new tokens introduced by S-Code.

Modification of Pass 8

Right now, Pass 8 eliminates duplicate constants. If same constant value appears twice, then it replaces them with a single copy of that constant. S-Code has initialized variables which behave like constants. For example, S-Code has initial constant values for the variables of the arithmetic type, data address type and instruction type. Pass 8 is therefore modified to include a way of distinguishing constant values and initializing variables, so that the latter do not have duplicates removed.

4.3 Environment Interface

The Environment Interface is used to do things depend on the unix operating system. It is written in language C, since this is the base language of the unix operating system.

This Environment Interface provides the system dependent services for the environment in which Perkin Elmer SIMULA system is to be implemented. It contains routines for the necessary environment switch between the Pascal environment and the C environment, since the run time environment of the Pascal code generator is different from that of unix and SIMULA environment is same as Pascal. This includes saving of stack pointers for both C stack and Pascal stack, and copying of parameters, because the parameter passing mechanisms are also different.

In addition to these, The Environment Interface introduces new functions for the Unix supplied editing services.

Chapter 5

SYNTAX/SEMANTIC LANGUAGE

This chapter serves as an introduction to Syntax/Semantic Language (S/SL) which we use to construct the Interpass. It is organized as follows. We first give general ideas about S/SL, and then the important features of S/SL are given. Also there is a brief explanation about how S/SL is implemented. Finally syntax diagrams[6] are given to illustrate the syntax of S/SL.

5.1 General Concepts

S/SL is a programming language developed at the University of Toronto as a tool for constructing compilers. It is a very modest language, incorporating only the following features: sequences, repetitions and selection of actions(statements); input, matching and output of tokens; output of error signals; subprograms(called rules); and invocation of semantic operations implemented in another language such as Pascal.

S/SL is a language without data or assignment; it is a pure control language. Data can be manipulated only via semantic operations.

---

[6] We use syntax diagrams rather than BNF, because the structure of a program written in S/SL resembles a textual notation for syntax diagrams with statements relating to the output actions and calls to the semantic routines inserted.

An S/SL program behaves like a recursive descent parser with output actions and semantic routine calls included in it.

The semantic routines called by S/SL program can be parameterized or nonparameterized. The parameters passed to the parameterized routines can be only constants since S/SL does not contain any variables. The entire data handling process is carried out through the execution of these routines.

S/SL adheres to a strict data abstraction by seperating the algorithm from the data. The algorithm invokes the semantic routines where the data is manipulated.

5.2 Features of S/SL

Each S/SL program consists of a list of declarations followed by a list of executable rules. Each rule consists of a name, an optional return type and a sequence of actions(statements). Each rule has one of these two forms:

```
name: actions;
name >> type: actions;
```

A rule with the first form is called a procedure rule; a rule with the second form is called a choice rule. These two forms are analogous to procedures and functions in Pascal. In our dialect, the choice rule does not exist.

5.2.1 Declarations in S/SL

We need to make declarations that are used by S/SL rules. In particular we need to specify the set of input tokens, the set of output tokens, the set of error signals, the set of values returned by each choice rule and the set of values to be used as parameters to parameterized semantic ops . Each of these sets is declared by enumerating the names of their members; this is similar to defining enumerated types in Pascal. For example:

```
input: % Input Tokens[7]
  integer
  plus
  ...etc...;
output: % Output Tokens
  Int
  Add
  ...etc...;
error: % Error Signals
  missingSemicolon
  ...etc...;
```

5.2.2 Summary of S/SL Actions

There are eight different actions(statements) in S/SL.

1. The call action: This action is used to invoke a procedure rule. The appearance of @ followed by the name of a procedure rule signifies that the rule is to be called.

---

[7] Here we have shown the convention for comments in S/SL programs, namely, anything to the right of a % on a line is ignored.

2. The return action: The symbol >> in a rule signifies return before reaching the end of the rule. In choice rules, the >> must be followed by a value in the rule's specified retun range, and implicit return via the final semicolon is not allowed. In procedure rules, the return >> is not followed by a value. Usually >> is not used in procedures because return is implicit at the end of the rule.

3. The input(or match) action: The equal sign(=) followed by the appearance of an input token in a rule signifies that the next input should be read and must match the token.

4. The emit token: The appearance of a dot(a period) followed by an output token signifies that the token is to be emitted to the output stream.

5. The error action: The appearance of # followed by an error signal signifies that the error signal is to be emitted to a special error stream. Presumably this stream is used to print appropriate error messages.

6. The cycle action: This action is analogous to the loop structure in the high level languages like Pascal. Each cycle is of the form:

```
        {
          actions
                  }
```

The enclosed actions are repeated until a return(>>) or one of the cycle's exits(>) is encountered.

7. The exit action: The appearance of > signifies exit from the most tightly enclosing cycle.

8. The choice action: The effect of this action is analogous to a case statement in Pascal. The choice action is of the form:

```
[ selector
      | labels:
          actions
      | labels:
          actions
      | *:
          actions
  ]
```

A choice action can be a rule choice, a semantic choice or an input choice. If the selector is the name of an S/SL choice rule then the choice action is a rule choice. In a rule choice, the specified rule is called and then the choice tries to match the returned value to one of the labels. If the selector is the name of a semantic routine then the choice action is considered to be a semantic choice. In a semantic choice, semantic action is called and value it returns is used. If the selector is "=", we have an input choice, which tries to match the current input token to one of the labels. The actions associated with the matched labels are executed; if no label is matched, the final alternative, labelled by the star(*), is executed. This is analogous to the otherwise clause in a Pascal case statement. It is optional. If omitted, the selector must match one of the choice's labels. Otherwise, there is an error.

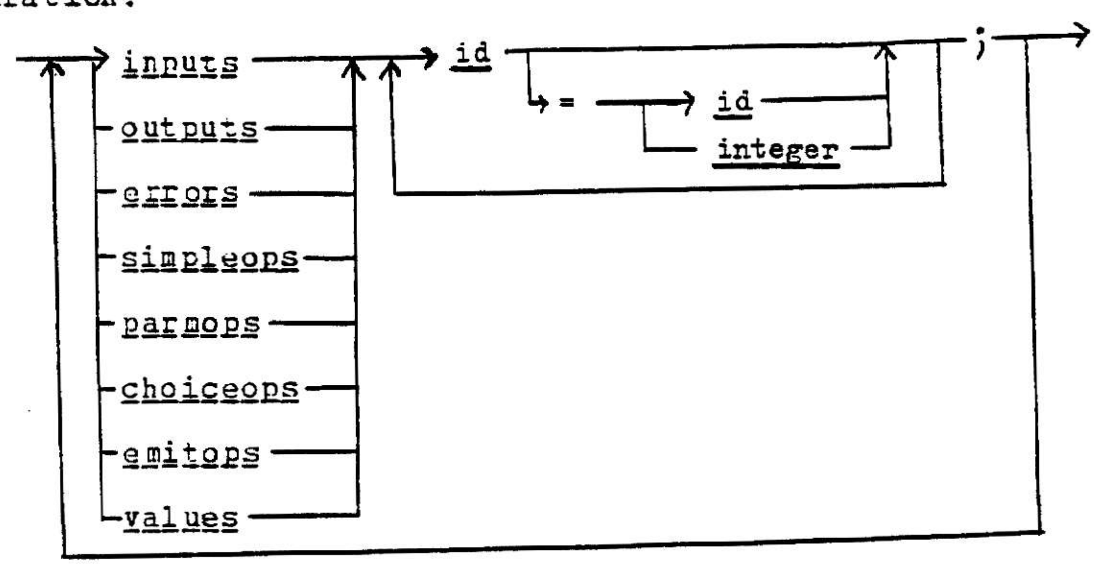
The choice action controls the decision process. The input token (in case of input choice) or the value returned (in case of a rule or a semantic choice) is only used as a determiner for the selection of a particular set of actions. It does not modify or manipulate any data. Our dialect has no rule choices.

5.3 Implementation of S/SL

S/SL is implemented by a translator which translates S/SL to an intermediate language. That language can be called SL-code. This is a machine-language-like instruction set encoded into a sequence of numbers. The result is a Pascal array of integers. The translated S/SL program is then interpreted by SL-code interpreter. That interpreter scans through the array by using a large case statement. Each instruction and semantic operation of SL-code has a counter part label in the case statement.

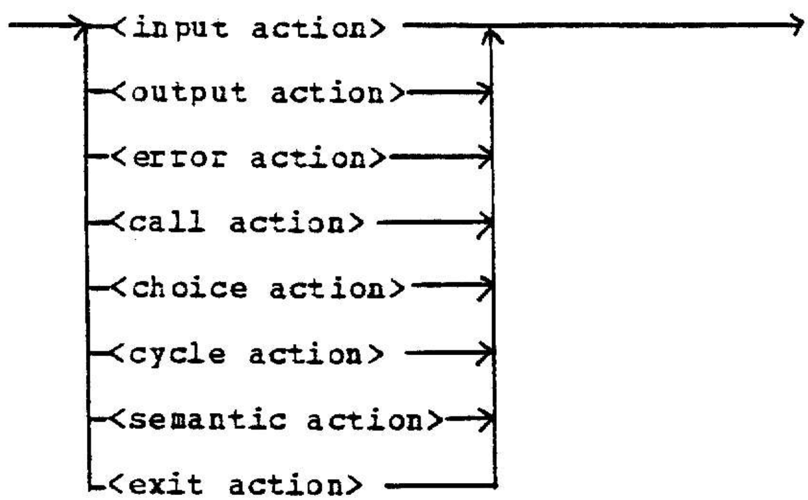
5.4 S/SL Syntax

declaration:



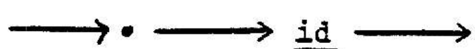
rule:

action:

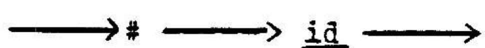


input action:

output action:

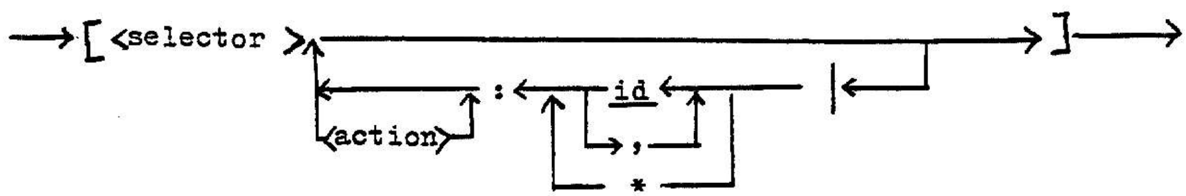


error action:

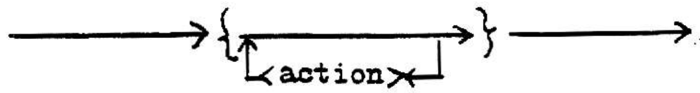


call action:

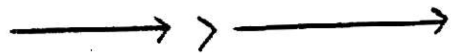
choice action:



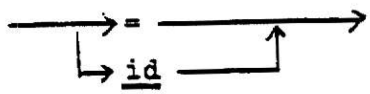
cycle action:



exit action:



selector:



Chapter 6

DESIGN OF INTERPASS

Interpass is a new pass and at the same time a major pass of the S-Compiler. Interpass includes a main controlling program coded in S/SL and a set of semantic routines coded in Pascal. This chapter discusses the design of Interpass. First, we give the detailed description of data structures and semantic operations used in Interpass. Then the general summary of how we built the S/SL program is given.

6.1 Data Structures and Semantic Operations of Interpass

6.1.1 S-stack and Associated Semantic Operations

This is a global stack which corresponds to the compile time stack defined in the S-code. It is a stack of pointers pointing to the descriptors.[a] This stack is implemented as a Pascal array of pointers.

As the compilation process in interpass progresses, the descriptors are continuously created and destroyed on the S-stack. Therefore push and pop operations are needed for this creation and destruction.

---

[a] The actions which the S-compiler should take upon recognition of a specific program element, are defined in terms of descriptors. These descriptors contain information about the objects which exist during the compilation process.

The stack semantics for assignment and addressing in the S-program requires that certain conditions on the descriptores held in the stack are valid. For example, the correct interpretation of the "assign" instruction and valid code generation is possible only if the second topmost descriptor on the stack is of ref mode.[9] At runtime the value held in the topmost descriptor is to be transferred to the location designated by the second topmost descriptor. The addressing instructions modify or replace the top descriptor on the stack. This means a new result descriptor must occupy the top of the stack at the end of translation. It is also necessary to set the mode, type or output pointer fields of the descriptor. Push and set operations are used to accomplish this. The emission of the right tokens in the output stream is possible only by examining the information. Verification operations and choice operations on the S-stack descriptors are needed for this.

Here is a list of some operations associated with S-stack.

--push descriptors

--pop the S-stack

--Set the mode field of the top of S-stack

--Set the output pointer field of the top of S-stack

--Return S-code type field of top of S-stack

---

[9] The mode or "access rule" in S-code is grouped in two modes, namely VAL and REF. VAL access rules describe values. REF access rules describe references to areas which have an associated address of some sort.

--Verify if top of S-stack is ref mode and so on.

6.1.2 Tag Stack, Tag Table and Associated Semantic Operations

The tag stack is a stack of tag values associated with the descriptors. Tags are S-code equivalent of identifiers. It is implemented as an array of integers within a predefined subrange. This stack is used to push something onto S-stack.

The tag table is an array of records subscripted by tag values. A tag record stores two kinds of information about a tag. One field of the record stores the tag state. A tag may be undefined, referenced or defined. The other field points to the descriptor associated with the given tag. This table is used to locate the descriptor when the tag naming it appears in the input. The table entries are modified at compile time when the tags are declared or undeclared.

Whenver a tag is read from the input stream, it is held temporarily in the tag stack. Since tags may be specified prior to their definition, it is necessary to verify the state of the tag whenever the tag is referred to. The tag value stored in the tag stack is used to create a new descriptor relating to that tag. The pointer to that descriptor will be stored in the tag table at the tag value.

As the arguments to a token are scanned further, the appropriate fields of the descriptor referring to the tag are set up. The setting operations may use the constant values being held in the count stack temporarily.

While interpreting the type conversion instructions it is necessary to examine the tag held in the tag stack in order to determine whether the conversion will be valid or not. Return or choice operations are needed to accomplish this.

Here is a list of some semantic oprations associated with the tag stack and with the tag table:

--read a tag from input, and push it onto the tag stack

--set the state of the tag

--select the descriptor

--verify if top of the tag stack is a record type, and so on.

6.1.3 <u>Count Stack and Associated Semantic Operations</u>

Count stack is a stack of integers. During the compilation of an S-progam, the count stack is used to temporarily hold the constant values which may appear in various program elements. In particular, the count stack can store:

--Size of an S-code type

--Number of a repetition value

--Number of a fixed repetition value

--Number of an indefinite repetition value

--Displacement

--Lower and upper bound of a range

--Byte, number and converted string values, and so on.

These values are placed on the count stack through push operations. It is sometimes necessary to increment or decrement the constant value on the count stack. Compile time addition and multiplication of the top two constants on the count stack is also needed in some translations. Therefore, simple semantic routines for these operations are introduced.

The count stack always holds the values temporarily. Once the constant has been used, it is discarded through a pop operation.

Here is a list of semantic operations for the count stack.

--push byte, number, etc.

--pop

--add top and second top of count stack and replace it by their sum

--add one(increment) to top of count stack

--subtract one(decrement) from top of count stack

--return the value of top of count stack and so on.

6.1.4 <u>Index Table Stack and Associated Semantic Operations</u>

This is a stack of tables of S-code jump indexes. Each entry contains a state(undefined, referenced, defined) and a Pass 6 label. Ordinarily, only the top table of the stack is visible. A maximum stack depth of two tables is sufficient.

The backward and forward jump instructions access the jump labels through indexing. When a jump instruction is translated, an index entry relating to the destination of the jump is changed. In particular, the state of the index will be changed depending on the kind of jump. The index will be undefined if the jump is forward jump. A set of Pass 6 labels is associated with the destination indices. It is necessary to allocate a new pass 6 label every time the destination is defined or referenced. Further, once a jump has occurred the corresponding index should be made available for reuse. But the Pass 6 labels can never be reused. A new index table is pushed when the compile time control enters a new scope level.

The following semantic operations operate on the index table stack:

--push a new index table on the index table stack with all entries undefined

--allocate a new pass 6 label and store it in the pass 6 label field of the table entry

--pop the index table stack

--set the state of the index, and so on.

6.1.5 Reachable Stack and Associated Semantic Operations

This is a stack of boolean values. It is used to indicate whether the current point in S-code is reachable, i.e., whether the runtime control can reach the current point. Initially it has only top entry with a value false. The top of the stack is turned true by the tokens associated with the backward destinations, forward destinations, switch destinations and the labels endif and else. The top is turned false by the tokens associated with backward jumps, forward jumps, unconditional gotos and switches. The stack is pushed when statically nested procedures are entered and popped when they are exited.

The following semantic operations are required to push, pop, set or return a boolean value to indicate the reachability of the S-code:

--push true or false onto the reachable stack

--pop the reachable stack

--set the top of reachable stack to true or false

--return the value of the top of reachable stack.

6.2 General Coding Methodology

Here is an example of how S/SL program is constructed. "add" is a simple arithmetic instruction which, at runtime, adds the top descriptor and the second topmost descriptor. The operations involved are:

1. If the top descriptor is of ref mode then change the top of the S-stack to hold the contents of the area referred to by this descriptor. And emit code "tpushind".

2. If the second topmost descriptor is of ref mode then change the second top of the S-stack is to hold the contents of the area referred to by this descriptor. And emit code "tpush2ndind".

3. Verify that the top and the second topmost descriptor type tags are equal.

4. Emit a pass 6 "tadd" token in the output token stream.

5. Push the S-code type field of the S-stack top descriptor onto the tag stack.

6. Pop the top and the second top descriptors from the S-stack.

7. Depending on the data type, emit the value, i.e., a Pass 6 type argument to "add".

S/SL invokes the following semantic routines for interpreting the "add" instruction.

```
@ForceTOSValue
@ForceSOSValue
oVerifySTosSosMDataDotTypesMatch
oEmitToken(tadd)
oPushTagWithSMDataDotType
oPopS
oPopS
[ oChooseTag
      | INT:
          oEmitValue(P6word)
      | REAL:
          oEmitValue(P6shortreal)
      | LREAL:
          oEmitValue(P6real)
      | *:
        oAbort(eArithmeticType)
]
```

@PushResultDescriptor

The S/SL rules @ForceTOSValue, @ForceSOSValue and @PushResultDescriptor invoke in turn the following semantic routines:

```
ForceTOSValue:
        [ oChooseSMDataDotMode
              | REFMode:
                   [ oChooseSMDataDotType
                         | INT, BOOL, CHAR, REAL, LREAL,
                           SIZE, OADDR, AADDR, GADDR,
                           PADDR, RADDR:
                             oSetSMDataDotMode(VALMode)
                             oEmitToken(tpushind)
                             oPushTagWithSMDataDotType
                             oEmitTagMTypeDotP6Type
                             oPopTag
                         | *:
                            oSetSMDataDotMode(VALADDRMode)
                   ]
              | *:
        ]

ForceSOSValue:
        [ oChooseSSosMDataDotMode
              | REFMode:
                  [ oChooseSSosMDataDotType
                        | INT, BOOL, CHAR, REAL, LREAL,
```

```
                              SIZE, OADDR, AADDR, GADDR, PADDR,
                              RADDR:
                                oSetSSosMDataMode(VALMode)
                                oEmitTokenAfterSSos(tpush2ndind)
                                oPushTagWithSSosMDataDotType
                                oEmitTagMTypeDotP6TypeBefore
                                 SMDataDotOutputpointer
                                oPopTag
                            | *:
                               oSetSSosMDataDotMode(VALADDRMode)
                        ]
                  | *:
              [

PushResultDescriptor:
              oPushSNewResultDescriptor
              oSetSMDataDotTypeToTag
              oPopTag
              oSetSMDataDotOutputPointer
              oTurnonSMDataDotHasP6Counterpart
```

Chapter 7

INTERESTING PROBLEMS

This chapter discusses some of the interesting issues which we met during the design of Interpass. The solutions adopted in each case are also described.

7.1 Unnesting of Procedure Bodies

7.1.1 Introducing the Problem

The **segment** instruction in S-code allows the existence of out of sequence code in the S-program.

segment-instruction

::= **bseg** <program-element>*[10] **eseg**

The intention of this instruction is that the enclosed elements must be located elsewhere and cannot be generally scanned in strict sequential order. In other words, the following piece of S-code:

sequence-1 **bseg** sequence-2 **eseg** sequence-3

will generate target code for sequence-1 and sequence-3 in direct control sequence while sequence-2 will be located somewhere else.

Besides the "bseg" and "eseg" tokens may be nested thus allowing for nested procedures. Pass 6 of the Pascal/32 compiler has no provision for handling such nested procedures. A direct handling of the non-sequential code can result in a waste of memory space.

---

[10] The asterisk(*) has the following meaning: "program-element" may occur zero or more times at this point.

7.1.2 Solution

The solution adopted was to use files for the treatment of code in the non sequential order. This was implemented as follows.

1. A main file or an output file is written physically in the order the tokens are produced. The main file is divided into fragments. It is assumed that within a fragment the physical and the logical ordering of the tokens is the same. But on the whole, the main file contains tokens which are not in order at all, because the fragments of this file have no inherent ordering, i.e.. they can be scattered within the main file.

2. There is another file called map file which allows Pass 6 to read the fragments of the output file in the correct logical order. The map file contains a list of pointers to the fragments. The physical ordering of the pointers is kept the same as the logical ordering of fragments to which they point. Pass 6 will read the map file sequentailly.

3. A fragment stack is defined. Each entry of this stack is the root of a list of pointers pointing to the fragments. The list itself is a double linked list of fragment pointer nodes. Each node stores a pair of main file pointers. The pair of main file pointers are a begin-fragment pointer and an end-fragment pointer. The begin-fragment pointer points to the first token of the fragment and the end-fragment pointer points to the last of the fragment.

7.2 Value Forcing

7.2.1 Introducing the Problem

The descriptors on the S-stack can be of ref mode. A value forcing operation on such a descriptor requires the insertion of a pass 6 "pushind" token at a place in the token stream just after the instruction token which generated this descriptor. The instruction token may be residing at a position other than the end of the token stream and the value forcing operation can be performed on a descriptor which is below the top of the S-stack. This requires inserting the "pushind" token at a spot other than at the end of the output token stream.

7.2.2 The Solution

The solution adopted is to insert the token in the middle of the stream by splitting a fragment. The insertion mechanism is implemented as follows:

1. The descriptor on the S-stack contains a field called output pointer which stores the pointer to the spot in the output stream.
2. The fragment containing this token is split into two fragments. This is done by searching for that fragment through a scan of the fragment pointers in the list attached to the top entry of the fragment.
3. The found fragment is split into two and a new fragment is linked between the two pieces. The inserted fragment contains the pointers to the tokens to be inserted. The token itself is written physically at the end of the main file.

In this way, the logical ordering of the tokens is arraged without disturbing the physical ordering of the emitted tokens in the output stream.

Chapter 8

CONCLUSION

As a participant of the Perkin Elmer Simula System project, we had an wonderful opportunity to know the characteristics of a group project. Besides, as a result of working for Interpass design, we got an opportunity to study the components of S-Port, the low level intermediate language S-code, and the fascinating language S/SL.

We also observed that the selection of Syntax/Semantic Language as the main control language was made well because the structure of S/SL suits perfectly the control structure of S-code.

Concerning the present status of the project, we can say that the work of Interpass which includes the coding of S/SL and the coding of semantic routines in Pascal is about 70 % complete. However no testing has done on the code developed so far except running through S/SL assembler.

The design of Interpass so far does not include the treatment of initializing attributes of a record type in S-code. Future work can be done so that Interpass design will include some additions to the related data structure and semantic operations, particularly in the record descriptors. Interpass design will also be modified to include the code for the utility library routines. The routines will involve the reading of S-code tokens from Interpass input stream, writing the tokens to Interpass output stream, allocating and freeing the descriptors during the compilation process and converting the text strings and input literals from S-code into internal form. One more thing what we can suggest is that for the efficiency and the convenience, making a S/SL pretty printer can be done as a part of the project.

Finally we can say that we have contributed to introducing a running SIMULA program in K.S.U. We can't run it yet, and it is just a commencement, but as the oriental saying indicates, "the commencement means it's already done the rest."

Appendix I

REFERENCES

REFERENCES

Dahl, O.J., and Nygaard, K. 1966 September. "SIMULA--an ALGOL-Based Simulation Language", Comm. ACM 9, vol.9, No.9, pp.671-678.

Dahl, O.J., and Nygaard, K. 1978 August. "The Development of the SIMULA Language", ACM SIGPLAN Notices, vol.13, No.8, pp.245-272.

Holt, R.C., Cody, J.R. and Wortman, D.B. 1980 March. "Introduction to S/SL: Syntax/Semantic Language", University of Toronto.

Ichbiah, J.D., and Morse, S.P. 1969 December. "General Concept of the Simula 67 Programming Language", DR.SA.69. 132 ND. Compagnie Internationale pour l'Informatique, pp.65-93.

Jensen, P., Krogdahl, S., Myhre, O., Robertson, P.S., and Syrrist, G. 1982 October. "S-Port: Definition of S-Code v3.0", NCC:Norwegian Computing Center.

Millard, Geoffrey E., Myhre, O., and Syrrist, G. 1982 June. "S-Port: The Environment Interface", Norsk Regnesentral/Norwegian Computing Center.

"Internal Documentation For The PASCAL/32 Compiler", 1980. Department of Computer Science, Kansas State University.

Appendix II

S/SL PROGRAM LISTING

```
***** Tue Mar 13 16:56:45 1984
***** KANSAS STATE UNIVERSITY  *****
***** COMPUTER SCIENCE  DEPT   *****
***** MINI-COMPUTER LABORATORY *****

       *
** *   *       *   **
* * *  *          *  *
* * *  *       *  *  *
*   *  * *     *  *  *
*   *   *      *   ****
            *  *
             **
```

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

P-474

```
% Interpass Input Tokens

Inputs:

access
accessv
add
alt
and
anone
asscall
assign
asspar
assrep
attr
bdest
bjump
bjumpif
body
bseq
eseq
call
call_tos
c_aaddr
c_char
c_dot
c_gaddr
c_int
c_ireal
c_oaddr
c_paddr
c_raddr
c_real
c_record
c_size
compare
const
constspec
convert
deco
delete
deref
dinitarea
dist
div
dsize
dup
else
empty
endif
endmacro
endmodule
endprofile
endprogram
endrecord
```

```
 56 endroutine                          *
 57 endskip                             *
 58 eqv                                 *
 59 eval                                *
 60 existing                            *
 61 exit                                *
 62 export                              *
 63 external                            *
 64 false                               *
 65 fetch                               *
 66 fdest                               *
 67 fixrep                              *
 68 fjump                               *
 69 fjumpif                             *
 70 getobj                              *
 71 global                              *
 72 gnone                               *
 73 goto                                *
 74 if                                  *
 75 imp                                 *
 76 inco                                *
 77 index                               *
 78 indexv                              *
 79 info                                *
 80 initarea                            *
 81 insert                              *
 82 interface                           *
 83 internal                            *
 84 import                              *
 85 known                               *
 86 label                               *
 87 labelspec                           *
 88 line                                *
 89 local                               *
 90 locate                              *
 91 macro                               *
 92 main                                *
 93 mark                                *
 94 mcall                               *
 95 module                              *
 96 mpar                                *
 97 mult                                *
 98 neg                                 *
 99 nobody                              *
100 nosize                              *
101 not                                 *
102 nowhere                             *
103 onone                               *
104 or                                  *
105 pop                                 *
106 popall                              *
107 precall                             *
108 prefix                              *
109 profile                             *
110 program                             *
111 push                                *
```

```
pushc
pushlen
pushv
Qlt
Qle
Qeq
Qge
Qgt
Qne
range
record
refer
rem
remote
remotev
rep
repcall
restore
routine
routinespec
rupdate
save
sdest
select
selectv
setobj
setswitch
skipif
sub
switch
system
tag
text
t_geto
t_inito
t_seto
true
update
xor
zeroarea
visible         % These are not used in S-code
endvisible      % but are stored in visible
taglist         % and taglist files for
endtaglist      % identification.
;

% Simple Semantic Operations

SimpleOps:

oAddcount
oAddTagListEntryFromTagAndCount
oAdjustMinimumTagToTag
oAppendTagSosMProfileDotImportListWithTagMAttribute
oBeginSegment
oCheckTagMRecordInitialization
```

```
168 oCloseInputFileAndPop
169 oCloseOutputCopyFile
170 oComputeRangeSizeAndAlignmentUnCount
171 oCreateBodyDescriptorAtTag
172 oCreateLabelDescriptorAtTag
173 oCreateProfileDescriptorAtTag
174 oCreateSwitchDescriptorAtTagWithCountLabels
175 oDecrementCountByOne
176 oEmitStringToOutputCopyFile
177 oEndSegment
178 oFixThis
179 oIncrementCountByOne
180 oMakeTaglistTableEmpty
181 oMarkAndCopySSection
182 oMarkSSection
183 oMergeSSections
184 oMultiplyCount
185 oOpenOutputCopyFileWithStringVisible
186 oOpenOutputCopyFileWithStringTaglist
187 oPopCount
188 oPopIndexes
189 oPopMarkedSSectionMEmpty
190 oPopP6Label
191 oPopReachable
192 oPopS
193 oPopString
194 oPopTag
195 oPopUpdateAndRestoreTag
196 oPushAndOpenInputFileWithStringVisible
197 oPushAndOpenInputFileWithStringTaglist
198 oPushCountWithDuplicate
199 oPushCountWithInputByte
200 oPushCountWithInputIntegerLiteral
201 oPushCountWithInputNumber
202 oPushCountWithSMDataDotDisplacement
203 oPushCountWithSMDataDotFixrep
204 oPushCountWithSSosMDataDotFixrep
205 oPushCountWithTagMDataDotDisplacement
206 oPushCountWithTagMSwitchDotFirstP6Label
207 oPushCountWithTagMSwitchDotLastP6Label
208 oPushCountWithTagMTypeDotHasRepField
209 oPushCountWithTagMTypeDotIndefAlternateLength
210 oPushCountWithTagMTypeDotLength
211 oPushDownIndexes
212 oPushP6LabelWithNewLabel
213 oPushReachableWithFalse
214 oPushReachableWithTrue
215 oPushSNewResultDescriptor
216 oPushStringWithInputString
217 oPushSWithDuplicate
218 oPushSWithTagMDataDescriptor
219 oPushTagAndStringWithInput
220 oPushTagWithInputTag
221 oPushTagWithSMDataDotType
222 oPushTagWithSSosMDataDotType
223 oPushTagWithTagMTypeDotIndefRepTag
```

```
oPushUpdateWithTagAndUndefine                                                     *
oSetIndexSubCountDotP6LabelToNewLabel                                             *
oSetReachableToFalse                                                              *
oSetReachableToTrue                                                               *
oSetSDotHasP6CounterPart                                                          *
oSetSMDataDotMode                                                                 *
oSetSMDataDotOutputPointer                                                        *
oSetSMDataDotTypeToTag                                                            *
oSetTagMBodyDotP6plabToNewP6plab                                                  *
oSetTagMDataDotDisplacementToCount                                                *
oSetTagMProfileDotIdToInputString                                                 *
oSetTagMProfileDotNatureToInputString                                             *
oSetTagSosMDataDotFixrepToCount                                                   *
oSetTagSosMDataDotLowerBoundToCount                                               *
oSetTagSosMDataDotUpperBoundToCount                                               *
oSetTagSosMDataDotRepToCount                                                      *
oSetTagSosMProfileDotBodyTagToTag                                                 *
oSkipInputString                                                                  *
oStoreValuePackedAddressConstantTag                                               *
oStoreValueWithInputLongRealLiteralAtCount                                        *
oStoreValueWithInputLongStringAndPushCountLength                                  *
oStoreValueWithInputRealLiteralAtCount                                            *
oSwapMarkedSSections                                                              *
oSwapP6Labels                                                                     *
oSwapString                                                                       *
oTranslateTagUsingTagListTable                                                    *
oTurnOnSMDataDotHasP6Counterpart                                                  *
oTurnOnOutputCopying                                                              *
oTurnOffOutputCopying                                                             *
oTurnOnTagSosMDataDotHasRange                                                     *
oTurnOnTagSosMDataDotHasFixrep                                                    *
oVerifyAllIndexesUndefined                                                        *
oVerifySTosSosMDataDotTypesMatch                                                  *
oVerifyTagDotKindIsType                                                           *
;


% Semantic Operations with Parameters

ParmOps:

oAbort                                  % (errorCode)                             *
oAlignAreaToCount                       % (AreaName)                              *
oCreateDataDescriptorAtTag              % (DataDescrKind)                         *
oEmitToken                              % (P6Token)                               *
oEmitTokenAfterSSos                     % (P6Token)                               *
oEmitTokenAfterS3os                     % (P6Token)                               *
oEmitTokenToOutputCopyFile              % (TokenValue)                            *
oEmitValue                              % (Integer)                               *
oEmitValueAfterSSos                     % (Integer)                               *
oEmitValueAfterS3os                     % (Integer)                               *
oIncrAreaDisplacementByCount            % (AreaName)                              *
oPushCount                              % (Value)                                 *
oPushTagWithType                        % (ScodeType)                             *
oSetIndexSubCountDotState               % (IndexState)                            *
oSetSMDataDotMode                       % (DescriptorMode)                        *
```

```
oSetSMDataDotType                  % (ScodeType)
oSetSSosMDataDotMode               % (DescriptorMode)
oSetTagDotState                    % (TagState)
oSetTagMProfileDotProfileKind      % (ProfileKind)
oSetTagMProfileDotNature           % (ProfileNature)
oVerifyIndexSubCountDotState       % (IndexState)
oVerifySDotDescriptorKind          % (DataDescrKind)
oVerifySMDataDotMode               % (DescriptorMode)
oVerifySMDataDotType               % (ScodeType)
oVerifySSosMDataDotMode            % (DescriptorMode)
oVerifySSosMDataDotType            % (ScodeType)
oVerifyStringEqualsInputString     % (ErrorCode)
oVerifyTagDotState                 % (TagState)
;


% Choice Operations

ChoiceOps:

oChooseCountComparison
oChooseCountSosValue
oChooseCountValue
oChooseReachable
oChooseSDotDescriptorKind
oChooseSMDataDotHasP6CounterPart
oChooseSMDataDotMode
oChooseSMDataDotType
oChooseSMDataDotHasFixrep
oChooseSMDataDotHasRange
oChooseSSosMDataDotMode
oChooseSSosMDataDotType
oChooseSEmpty
oChooseTagDotState
oChooseTagMTypeDotHasRepField
oChooseTagMTypeStructured
;


% Emit Semantic Operations

EmitOps:

oEmitCount
oEmitIndexSubCountDotP6Label
oEmitP6Label
oEmitSMDataDotP6Level
oEmitSMDataDotP6Mode
oEmitSMDataDotDisplacement
oEmitSSosMDataDotP6Mode
oEmitStringToOutputCopy
oEmitTagMBodyDotP6Plab
oEmitTagMDataDotDisplacement
oEmitTagMLabelDotP6Stlab
oEmitTagMProfileDotId8Chars
oEmitTagMTypeDotLength
```

```
oEmitTagMTypeDotP6Type                                                     •
oEmitTagMTypeDotP6TypeBeforeSMDataDotOutputPointer                         •
oEmitValueAreaCountBytes                                                   •
;


% VALUES OF S/SL

Values:

% Scode Type:

BOOL                                                                       •
CHAR                                                                       •
INT                                                                        •
REAL                                                                       •
LREAL                                                                      •
SIZE                                                                       •
UADDR                                                                      •
AADDR                                                                      •
GADDR                                                                      •
PADDR                                                                      •
RADDR                                                                      •
SINT          % special -- not a type, but a kind of range                 •

% AreaName:

GlobalArea                                                                 •
ConstantArea                                                               •
LocalArea                                                                  •


% DescriptorMode:

REFMode                                                                    •
VALMode                                                                    •
VALADDRMode                                                                •

% IndexState:

UndefinedIndexState                                                        •
DefinedIndexState                                                          •
ReferencedIndexState                                                       •


% TagState:

UndefinedTagState                                                          •
SpecifiedTagState                                                          •
DefinedTagState                                                            •

% DataDescrKind:

GlobalDescrKind                                                            •
LocalDescrKind                                                             •
ConstantDescrKind                                                          •
```

```
AttributeDescrKind                                                          *
BodyDescrKind                                                               *
DeclaredTypeDescrKind                                                       *
BuiltinTypeDescrKind                                                        *
SwitchDescrKind                                                             *
LabelDescrKind                                                              *
ImportDescrKind                                                             *
ExportDescrKind                                                             *
ExitDescrKind                                                               *
ResultDescrKind                                                             *
ProfileDescrKind                                                            *

% ProfileKind: Tells what kind of routine a profile defines

knownProfileKind                                                            *
systemProfileKind                                                           *
externalProfileKind                                                         *
interfaceProfileKind                                                        *
ordinaryProfileKind                                                         *

% ProfileNature:

SIMULANature                                                                *
CNature                                                                     *
Pas32Nature                                                                 *

% TokenValue: These are redefined as values so ParmOps
%              can emit them to special places.

otVisible      = visible                                                    *   152
otEndVisible   = endvisible                                                 *   153
otTaglist      = taglist                                                    *   154
otEndTaglist   = endtaglist                                                 *   155
otModule       = module                                                     *    95

% BooleanValue:

P6True = 1          % constant values emitted to Pass 6                     *
P6False = 0                                                                 *

InterpassTrue = 1   % returned by interpass boolean valued ChoiceOps        *
InterpassFalse = 0                                                          *

yes = 1             % also used for boolean valued ChoiceOps                *
no = 0                                                                      *

% P6Relation: Relational operators

P6equal                                                                     *
P6greaterthan                                                               *
P6greaterthanequal                                                          *
P6lessthan                                                                  *
P6lessthanequal                                                             *
P6notequal                                                                  *

% P6Mode: P6plab and P6stlab are passed as actuals to ParmOps
```

```
%          to tell which kind of P6  label is wanted.

P6globalMode
P6constMode
P6localMode
P6plabMode
P6stlabMode

% P6Type:

P6word
P6shortreal
P6real
P6wordpair

% Pass 6 conversion kinds:

stdconv       % int to shortreal
stdlongconv   % int to (long)real
stdlengthen   % short real to (long)real
stdround      % some kind of real to int

% Misc.

P6ReturnAddressDisplacement = 4
P6FirstFormalDisplacement = 8

% Result values for oChooseCountComparison

CompGreater
CompEqual
CompLess



% ErrorCode: These are declared as values so they can
%            be emitted by ParmOps oAbort.  At the
%            moment all errors are fatal.

eArithmeticType
eConvertType
eIllegalRelation
eProfileDescriptor
eStackNotEmpty
eType
eUndefinedTag
eTagKind
eFixrepsDiffer
eDuplicateDefinition
eDeferredMacros

% The following are non-fatal errors

erModuleNamesMismatch
erCheckCodeMismatch
```

```
% P6Token: These are the output tokens produced by Interpass
%          as Pass 6 Input Tokens.



tpushconst
tpushind
tpushaddr
tpushlabel
textid
tfield
trange
tassign
tcopy
tnot
tand
tor
tneg
tadd
tsub
tmul
tdiv
tmod
tcompare
tcompstruct
tdeflabel
tjump
tfalsejump
tcaseexpr
tcasejump
tpop
tlconst
tconvrt
tpushlabel
tlongjump
tdefstlab


% Pass 6 New Input Tokens

taccess
tdup
tupdate
tupdatecopy
trupdate
trupdatecopy
tpop2nd
tpush2ndind
tsetobj
tgetobj
tsimpleindex
;

_!_

% PROGRAMS
```

```
SProgram:                                                        *

    =program                                                   110
    oPushStringWithInputString                                 216
    @ProgramBody                                                 *
    =endprogram                                                 54
    oPopString                                                 193
;


ProgramBody:                                                     *

[ =
    | main:                                                     92
        @ProgramElements                                         *
    | macro:                                                    91
        oAbort(eDeferredMacros)                                265     497
        =module                                                 95
        oSkipInputString                                       241
        oSkipInputString                                       241
        @MacroDefinitions                                        *
        =endmodule                                              52
    | global:                                                   71
        @GlobalModule                                            *
        =endmodule                                              52
    | *:
        { [ =
            | module:                                           95
                @ModuleHeader                                    *
                =body                                           19
                @ProgramElements                                 *
                =endmodule                                      52
            | *: >
          ]
        }
    ]
;

ModuleHeader:                                                    *

    oPushStringWithInputString   % Module id                   216
    oPushStringWithInputString   % Check code                  216
    [ =
        | existing:                                             60
            oSwapString                                        248
            oPushAndOpenInputFileWithStringVisible             196
            =visible                                           152
            oVerifyStringEqualsInputString(eModuleNamesMismatch) 291   501
            oSwapString                                        248
            oVerifyStringEqualsInputString(eCheckCodeMismatch) 291     502
            @TranslateVisibleDeclarations                        *
            =endvisible                                        153
            oCloseInputFileAndPop                              168
        | *:
            oSwapString                                        248
```

```
                oOpenOutputCopyFileWithStringVisible                     185
                oTurnOnOutputCopying                                     251
                oEmitTokenToOutputCopyFile(otModule)                     271   425
                oEmitStringToOutputCopyFile                              176
                oSwapString                                              248
                oEmitStringToOutputCopyFile                              176
                @TranslateVisibleDeclarations                              *
                oEmitTokenToOutputCopyFile(otEndVisible)                 271   422
                oCloseOutputCopyFile                                     169
                oSwapString                                              248
                oOpenOutputCopyFileWithStringTaglist                     186
                oEmitTokenToOutputCopyFile(otTaglist)                    271   423
                oEmitStringToOutputCopyFile                              176
                oSwapString                                              248
                oEmitStringToOutputCopyFile                              176
                { [ =
                    | tag:                                               143
                        oPushTagAndStringWithInput                       219
                        oPopTag                                          194
                        oPushCountWithInputNumber                        201
                        oPopCount                                        187
                    | *: >
                ] }
                oEmitTokenToOutputCopyFile(otEndTaglist)                 271   424
                oCloseOuputCopyFile                                        *
        ]
    ;

    TranslateVisibleDeclarations:                                          *

    [ =
        | constspec:        @TranslateConstSpec                           38     *
        | record:           @TranslateRecordDescriptor                   122     *
        | labelspec:        @TranslateLabelSpec                           87     *
        | profile:          @TranslateProfile                            109     *
        | info:             @TranslateInfo                                79     *
        | setswitch:        @TranslateSetSwitch                          138     *
        | insert:                                                         81
            oTurnOffOutputCopying                                        252
            @TranslateInsert                                               *
            oTurnOnOutputCopying                                         251
        ]
    ;


    TranslateInsert:                                                       *

        oPushStringWithInputString                                       216
        oPushStringWithInputString                                       216
        oPushTagAndStringWithInput                                       219
        oPushTagAndStringWithInput                                       219
        oSwapString                                                      248
        oPushAndOpenInputFileWithStringTaglist                           197
        =taglist                                                         154
        oVerifyStringEqualsInputString(erModuleNamesMismatch)            291   501
        oSwapString                                                      248
```

```
oVerifyStringEqualsInputString(erCheckCodeMismatch)            291    502
oMakeTaglistTableEmpty                                         180
{ [ =
    | tag:                                                     143
        oPushTagAndStringWithInput                             219
        oPushCountWithInputNumber                              201
        oAddTaglistEntryFromTagAndCount                          *
        oPopCount                                              187
        oPopTag                                                194
        oPopString                                             193
    | *: >
] }
=endtaglist                                                    155
oSwapString                                                    248
oCloseInputFileAndPop                                          168
oPushAndOpenInputFileWithStringVisible                         196
=visible                                                       152
oVerifyStringEqualsInputString(erModuleNamesMismatch)          291    501
oSwapString                                                    248
oVerifyStringEqualsInputString(erCheckCodeMismatch)            291    502
[ =
   | constspec:                                                 38
       oPushTagAndStringWithInput                              219
       oTranslateTagUsingTaglistTable                            *
       @TranslateConstSpec                                       *
       oPopString                                              193
   | record:                                                   122
       oPushTagAndStringWithInput                              219
       oTranslateTagUsingTaglistTable                            *
       @TranslateRecordDescriptor                                *
   | labelspec:                                                 87
       oPushTagAndStringWithInput                              219
       oTranslateTagUsingTaglistTable                            *
       @TranslateLabelSpec                                       *
       oPopString                                              193
   | profile:                                                  109
       oPushTagAndStringWithInput                              219
       oTranslateTagUsingTaglistTable                            *
       @TranslateProfile                                         *
       oPopString                                              193
   | routinespec:                                              131
       oPushTagAndStringWithInput                              219
       oTranslateTagUsingTaglistTable                            *
       @TranslateRoutineSpec                                     *
       oPopString                                              193
   | info:                                                      79
       @TranslateInfo                                            *
   | line:                                                      88
       @TranslateLine                                            *
   | setswitch:                                                138
       @TranslateSetSwitch                                       *
   | insert:                                                    81
       oPushStringWithInputString                              216
       oPopString                                              193
       oPushStringWithInputString                              216
       oPopString                                              193
```

```
    ]
    =endvisible                                               153
    oCloseInputFileAndPop                                     168
    oMakeTaglistTableEmpty                                    180
    oPopString                                                193
    oPopString                                                193
    oPopTag                                                   194
    oPopTag                                                   194
;


ProgramElements:                                                *
{ [ =
      | constspec:                                             38
           oPushTagWithInputTag                               220
           @TranslateConstSpec                                  *
      | const:                                                 37
           @TranslateConstantDefinition                         *
      | record:                                               122
           oPushTagWithInputTag                               220
           @TranslateRecordDescriptor                           *
      | push:                                                 111
           @TranslatePush                                       *
      | pushv:                                                114
           @TranslatePush                                       *
           @FoceTOSValue                                        *
      | pushc:                                                112
           @TranslateNonRecordConstValue                        *
      | dup:                                                   47
           oPushSWithDuplicate                                217
           @ForceTOSValue                                       *
           oEmitToken(tdup)                                   268   545
      | pop:                                                  105
           [ oChooseSDotDescriptorKind                        304
                 | profileDescrKind                             *
                      oAbort(eProfileDescriptor)              265   490
                 | *:
                      oPopS                                   192
           ]
      | popall:                                               106
           oPushCountWithInputByte                            199
           { [ oChooseCountValue                              302
                  | 0:
                      [ oChooseSEmpty                         312
                            | yes:                            435
                                 oPopCount                    187
                                 >
                            | *:
                                 oAbort(eStackNotEmpty)       265   491
                                 >
                      ]
                  | *:
                      oPopS                                   192
                      oDecrementCountByOne                    175
           ] }
      | assign:                                                12
```

```
            @ForceTOSValue                                                  *
            oVerifySSosMDataDotMode(REFMode)                               289   370
            oVerifySTosSosMDataDotTypesMatch                               256
            oPushTagWithSMDataDotType                                      221
            [ oChooseSMDataDotMode                                         306
                | VALMode:                                                 371
                    oEmitToken(tassign)                                    268   516
                    oEmitTagMTypeDotP6Type                                 336
                | VALADDRMode:                                             372
                    oEmitToken(tcopy)                                      268   517
                    oEmitTagMTypeDotLength                                 335
            ]
            oPopTag                                                        194
            oPopS                                                          192
            oPopS                                                          192
        | update:                                                          149
            @ForceTOSValue                                                  *
            oVerifySTosSosMDataDotTypesMatch                               256
            oVerifySSosMDataDotMode(REFMode)                               289   370
            oPushTagWithSMDataDotType                                       *
            oEmitTokenAfterSSos(tdup)                                      269   545
            [ oChooseSMDataDotMode                                         306
                | VALMode:                                                 371
                    oEmitToken(tupdate)                                    268   546
                    oEmitTagMTypeDotP6Type                                 336
                | VALADDRMode:                                              *
                    oEmitToken(tupdatecopy)                                268   547
                    oEmitTagMTypeDotLength                                 335
            ]
            oPopTag                                                        194
            oPopS                                                          192
            @ForceTOSValue                                                  *
        | rupdate:                                                         132
            @ForceTOSValue                                                  *
            oVerifySTosSosMDataDotTypesMatch                               256
            oVerifySMDataDotMode(REFMode)                                  287   370
            oPushTagWithSSosMDataDotType                                   222
            oEmitTokenAfterSSos(tdup)                                      269   545
            [ oChooseSSosMDataDotMode                                      310
                | VALMode:                                                 371
                    oEmitToken(trupdate)                                   268   548
                    oEmitTagMTypeDotP6Type                                 336
                | VALADDRMode:                                             372
                    oEmitToken(trupdatecopy)                               268   549
                    oEmitTagMTypeDotLength                                 335
            ]
            oPopTag                                                        194
            oPopS                                                          192
        | fetch:                                                            65
            @ForceTOSValue                                                  *
        | refer:                                                           123
            oVerifySMDataDotType(GADDR)                                    288   356
            @ForceTOSValue                                                  *
            oEmitToken(tadd)                                               268   522
            oEmitTagMTypeDotP6Type                                         336
            oPopS                                                          192
```

```
            oPushSNewResultDescriptor                        215
            oSetSMDataDotMode(REFMode)                       279    370
            oPushTagWithInputTag                             220
            oSetSMDataDotTypeToTag                           231
            oPopTag                                          194
            oTurnOnSMDataDotHasP6Counterpart                 250
            oSetSMDataDotOutputPointer                       230
        | deref:                                             42
            oVerifySMDataDotMode(REFMode)                    287    370
            oEmitToken(tpushconst)                           268    509
            oEmitValue(0)                                    272
            oPopS                                            192
            oPushTagWithType(GADDR)                          277    356
            @PushResultDescriptor                            *
        | select:                                            135
            @TranslateSelect                                 *
        | selectv:                                           136
            @TranslateSelect                                 *
            @ForceTOSValue                                   *
        | remote:                                            125
            @TranslateRemote                                 *
        | remotev:                                           126
            @TranslateRemote                                 *
            @ForceTOSValue                                   *
        | index:                                             77
            @TranslateIndex                                  *
        | indexv:                                            78
            @TranslateIndex                                  *
            @ForceTOSValue                                   *
        | inco:                                              76
            oVerifySMDataDotType(SIZE)                       288    353
            @ForceTOSValue                                   *
            oVerifySSosMDataDotType(DADDR)                   290    354
            @ForceSOSValue                                   *
            oEmitToken(tadd)                                 268    522
            oEmitValue(P6word)                               272    458
            oPopS                                            192
            oPopS                                            192
            oPushTagWithType(DADDR)                          277    354
            @PushResultDescriptor                            *
        | deco:                                              40
            oVerifySMDataDotType(SIZE)                       288    353
            @ForceTOSValue                                   *
            oVerifySSosMDataDotType(DADDR)                   290    354
            @ForceSOSValue                                   *
            oEmitToken(tsub)                                 268    523
            oPopS                                            192
            oPopS                                            192
            oPushTagWithType(DADDR)                          277    354
            @PushResultDescriptor                            *
        | dist:                                              44
            oVerifySMDataDotType(DADDR)                      288    354
            @ForceTOSValue                                   *
            oVerifySSosMDataDotType(DADDR)                   290    354
            @ForceSOSValue                                   *
            oEmitToken(tsub)                                 268    523
```

```
                oPopS                                                   192
                oPopS                                                   192
                oPushTagWithType(SIZE)                                  277    353
                @PushResultDescriptor                                     #
        | dsize:                                                         46
                oVerifySMDataDotType(INT)                               288    350
                @ForceSOSValue                                            #
                oEmitToken(tpushconst)                                  268    509
                oPushCountWithTagMTypeDotHasRepField                    208
                oEmitTagMTypeDotLength                                  335
                oEmitToken(tmul)                                        268    524
                oEmitValue(P6word)                                      272    458
                oEmitTagMTypeDotP6Type                                  336
                oEmitToken(tpushconst)                                  268    509
                oPushCountWithTag                                         #
                oEmitToken(tadd)                                        268    522
                oEmitValue(P6word)                                      272    458
                oPopS                                                   192
                oPushTagWithType(SIZE)                                  277    353
                @PushResultDesCriptor                                     #
        | locate:                                                        90
                oVerifySMDAtaDotTYpe(AADDR)                               #    355
                @ForceTOSValue                                            #
                @ForceSOSValue                                            #
                [ oChooseSSosMDataDotType                               311
                        | DADDR:                                        354

                        | GADDR:                                        356
                                oEmitToken(tadd)                        268    522
                                oEmitValue(P6word)                      272    458
                        | *:
                                oAbort(eType)                           265    492
                ]
                oPopS                                                   192
                oPopS                                                   192
                oPushTagWithType(GADDR)                                 277    356
                @PushResultDescriptor                                     #
        | setobj:                                                       137
                oVerifySMDataDotType(INT)                               288    350
                @ForceTOSValue                                            #
                oVerifySSosMDataDotType(DADDR)                          290    354
                @ForceSOSValue                                            #
                oEmitToken(tsetobj)                                     268    552
                oPopS                                                   192
                oPopS                                                   192
        | getobj:                                                        70
                oVerifySMDataDotType(INT)                               288    350
                @ForceTOSValue                                            #
                oEmitToken(tgetobj)                                     268    553
                oPopS                                                   192
                oPushTagWithType(DADDR)                                 277    354
                @PushResultDescriptor                                     #
        | access:                                                         5
                @TranslateAccess                                          #
        | accessv:                                                        6
                @TranslateAccess                                          #
```

```
                @ForceTOSVaue                                           *
        | add:                                                          7
                @ForceTOSValue                                          *
                @ForceSOSValue                                          *
                oVerifySTosSosMDataDotTypesMatch                      256
                oEmitToken(tadd)                                      268     522
                oPushTagWithSMDataDotType                             221
                oPopS                                                 192
                oPopS                                                 192
                [ oChooseTag                                            *
                        | INT:                                        350
                                oEmitValue(P6word)                    272     458
                        | REAL:                                       351
                                oEmitValue(P6shortreal)               272     459
                        | LREAL:                                      352
                                oEmitValue(P6real)                    272     460
                        | *:
                                oAbort(eArithmeticType)               265     487
                ]
                @PushResultDescriptor                                   *
        | sub:                                                        140
                @ForceTOSValue                                          *
                @ForceSOSValue                                          *
                oVerifySTosSosMDataDotTypesMatch                      256
                oEmitToken(tsub)                                      268     523
                oPushTagWithSMDataDotType                             221
                oPopS                                                 192
                oPopS                                                 192
                [ oChooseTag                                            *
                        | INT:                                        350
                                oEmitValue(P6word)                    272     458
                        | REAL:                                       351
                                oEmitValue(P6shortreal)               272     459
                        | LREAL:                                      352
                                oEmitValue(P6real)                    272     460
                        | *:
                                oAbort(eArithmeticType)               265     487
                ]
                @PushResultDescriptor                                   *

        | mult:                                                        97
                @ForceTOSValue                                          *
                @ForceSOSValue                                          *
                oVerifySTosSosMDataDotTypesMatch                      256
                oEmitToken(tmul)                                      268     524
                oPushTagWithSMDataDotType                             221
                oPopS                                                 192
                oPopS                                                 192
                [ oChooseTag                                            *
                        | INT:                                        350
                                oEmitValue(P6word)                    272     458
                        | REAL:                                       351
                                oEmitValue(P6shortreal)               272     459
                        | LREAL:                                      352
                                oEmitValue(P6real)                    272     460
                        | *:
```

```
                        oAbort(eArithmeticType)               265    487
                ]
                @PushResultDesriptor                            *
        | div:                                                 45
                @ForceTOSValue                                  *
                @ForceSOSValue                                  *
                oVerifySTosSosMDataDotTypesMatch              256
                oEmitToken(tdiv)                              268    525
                oPushTagWithSMDataDotType                     221
                oPopS                                         192
                oPopS                                         192
                [ oChooseTag                                    *
                        | INT:                                350
                            oEmitValue(P6word)                272    458
                        | REAL:                               351
                            oEmitValue(P6shortreal)           272    459
                        | LREAL:                              352
                            oEmitValue(P6real)                272    460
                        | *:
                            oAbort(eArithmeticType)           265    487
                ]
                @PushResultDescriptor                           *
        | rem:                                                124
                oVerifySMDataDotType(INT)                     288    350
                @ForceTOSValue                                  *
                oVerifySSosMDataDotType(INT)                  290    350
                @ForceSOSValue                                  *
                oEmitToken(tmod)                              268    526
                oPopS                                         192
                oPopS                                         192
                oPushTagWithType(BOOL)                        277    348
                @PushResultDescriptor                           *
        | neg:                                                 98
                @ForceTOSValue                                  *
                oEmitToken(tneg)                              268    521
                oPushTagWithSMDataDotType                     221
                oPopS                                         192
                [ oChooseTag                                    *
                        | INT:                                350
                            oEmitValue(P6word)                272    458
                        | REAL:                               351
                            oEmitValue(P6shortreal)           272    459
                        | LREAL:                              352
                            oEmitValue(P6real)                272    460
                        | *:
                            oAbort(eArithmeticType)           265    487
                ]
                @PushResultDescriptor                           *
        | and:                                                  9
                oVerifySMDataDotType(BOOL)                    288    348
                @ForceTOSValue                                  *
                oVerifySSosMDataDotType(BOOL)                 290    348
                @ForceSOSValue                                  *
                oEmitToken(tand)                              268    519
                oPopS                                         192
                oPopS                                         192
```

```
                oPushTagWithType(BOOL)                                      277    348
                @PushResultDescriptor                                         *
        | or:                                                               104
                oVerifySMDataDotType(BOOL)                                  288    348
                @ForceTOSValue                                                *
                @ForceSOSValue                                                *
                oVerifySTosSosMDataDotTypesMatch                            256
                oEmitToken(tor)                                             268    520
                oPopS                                                       192
                oPopS                                                       192
                oPushTagWithType(BOOL)                                      277    348
                @PushResultDescriptor                                         *
        | xor:                                                              150
                oVerifySMDataDotType(BOOL)                                  288    348
                @ForceTOSValue                                                *
                @ForceSOSValue                                                *
                oVerifySTosSosMDataDotTypesMatch                            256
                oEmitToken(tcompare)                                        268    527
                oEmitValue(P6notequal)                                      272    445
                oPopS                                                       192
                oPopS                                                       192
                oPushTagWithType(BOOL)                                      277    348
                @PushResultDescriptor                                         *
        | imp:                                                               75
                oVerifySMDataDotType(BOOL)                                  288    348
                @ForceTOSValue                                                *
                @ForceSOSValue                                                *
                oVerifySTosSosMDataDotTypesMatch                            256
                oEmitToken(tnot)                                            268    518
                oEmitToken(tand)                                            268    519
                oEmitToken(tnot)                                            268    518
        | eqv:                                                               58
                oVerifySMDataDotType(BOOL)                                  288    348
                @ForceTOSVaue                                                 *
                @ForceSOSValue                                                *
                oVerifySTosSosMDataDotTypesMatch                            256
                oEmitToken(tcompare)                                        268    527
                oEmitValue(P6equal)                                         272    440
                oPopS                                                       192
                oPopS                                                       192
                oPushSNewResultDescriptor                                   215
                oSetSMDataDotType(BOOL)                                     280    348
                oSetSMDataDotOutputPointer                                  230
        | not:                                                              101
                oVerifySMDataDotType(BOOL)                                  288    348
                @ForceTOSValue                                                *
                oEmitToken(tnot)                                            268    518
                oPushTagWithType(BOOL)                                      277    348
                @PushResultDescriptor                                         *
        | compare:                                                           36
                @ForceTOSValue                                                *
                @ForceSOSValue                                                *
                @CheckRelation                                                *
                oPushTagWithSMDataDotType                                   221
                [ oChooseTagMTypeStructured                                 315
                        | yes:                                              435
```

```
                        oEmitToken(tcompstruct)                                        268   528
                        oEmitCount                                                     323
                        oEmitTagMTypeDotLength                                         335
                    | *:
                        oEmitToken(tcompare)                                           268   527
                        oEmitCount                                                     323
                        oEmitTagMTypeDotP6Type                                         336
                ]
                oPopS                                                                  192
                oPopS                                                                  192
                oPushTagWithType(BOOL)                                                 277   348
                oPushResultDescriptor                                                  *
            | convert:                                                                 39
                oPushTagWithInputTag                                                   220
                [ oChooseSMDataDotType                                                 307
                    | CHAR:                                                            349
                        [ oChooseTag                                                   *
                            | CHAR, INT:                                               349   350

                            | *:
                                oAbort(eConvertType)                                   265   488
                        ]
                    | BOOL:                                                            348
                        [ oChooseTag                                                   *
                            | BOOL:                                                    348

                            | *:
                                oAbort(eConvertType)                                   265   488
                        ]
                    | INT:                                                             350
                        [ oChooseTag                                                   *
                            | INT:                                                     350

                            | CHAR:                                                    349
                                oEmitToken(trange)                                     268   515
                                oEmitValue(0)                                          272
                                oEmitValue(255)                                        272
                            | REAL:                                                    351
                                oEmitToken(tconvrt)                                    268   536
                                oEmitValue(stdconv)                                    272   465
                                oEmitValue(P6shortreal)                                272   459
                                oEmitValue(4)                                          272
                                oEmitValue(4)                                          272
                            | LREAL:                                                   352
                                oEmitToken(tconvrt)                                    268   536
                                oEmitValue(stdlongconv)                                272   466
                                oEmitValue(8)                                          272
                                oEmitValue(8)                                          272
                            | *:
                                oAbort(eConvertType)                                   265   488
                        ]
                    | REAL:                                                            351
                        [ oChooseTag                                                   *
                            | REAL:                                                    351

                            | INT:                                                     350
```

```
                        oEmitToken(tconvrt)                 268   536
                        oEmitValue(stdround)                272   468
                        oEmitValue(4)                       272
                        oEmitValue(4)                       272
                    | LREAL:                                352
                        oEmitToken(tconvrt)                 268   536
                        oEmitValue(stdlengthen)             272   467
                        oEmitValue(8)                       272
                        oEmitValue(8)                       272
                    | *:
                        oAbort(eConvertType)                265   488
                ]
        | LREAL:                                            352
            [ oChooseTag                                      *
                    | LREAL:                                352

                    | INT:                                  350
                        oEmitToken(tconvrt)                 268   536
                        oEmitValue(stdround)                272   468
                        oEmitValue(P6word)                  272   458
                        oEmitValue(4)                       272
                        oEmitValue(4)                       272
                    | REAL:                                 351
                        oEmitToken(tconvrt)                 268   536
                        oEmitValue(stdshorten)              272     *
                        oEmitValue(P6shortreal)             272   459
                        oEmitValue(4)                       272
                        oEmitValue(4)                       272
                    | *:
                        oAbort(eConvertType)                265   488
                ]
        | SIZE:                                             353
            [ oChooseTag                                      *
                    | SIZE:                                 353

                    | *:
                        oAbort(eConvertType)                265   488
                ]
        | AADDR:                                            355
            [ oChooseTag                                      *
                    | AADDR:                                355

                    | *:
                        oAbort(eConvertType)                265   488
                ]
        | DADDR:                                            354
            [ oChooseTag                                      *
                    | DADDR:                                354

                    | GADDR:                                356
                        [ oChooseSMDataDotMode              306
                                | REFMode:                  370
                                    oEmitToken(tpushind)    268   510
                                    oEmitValue(P6word)      272   458
                                | *:
                        ]
```

```
                                oEmitToken(tpushconst)              268   509
                                oEmitValue(0)                       272
                            | *:
                                oAbort(eConvertType)                265   488
                        ]
                | GADDR:                                            356
                    [ oChooseTag                                      *
                        | GADDR:                                    356

                        | AADDR:                                    355
                            [ oChooseSMDataDotMode                  306
                                | REFMode:                          370
                                    oEmitToken(tpushind)            268   510
                                    oEmitValue(P6word)              272   458
                                | *:
                            ]
                            oEmitToken(tpop2nd)                     268   550
                        | DADDR:                                    354
                            [ oChooseSMDataDotMode                  306
                                | REFMode:                          370
                                    oEmitToken(tpushind)            268   510
                                    oEmitValue(P6wordpair)          272   461
                                | *:
                            ]
                            oEmitToken(tpop)                        268   534
                        | *:
                            oAbort(eConvertType)                    265   488
                    ]
                | PADDR:                                            357
                    [ oChooseTag                                      *
                        | PADDR:                                    357


                        | *:
                            oAbort(eConvertType)                    265   488
                    ]
                | RADDR:                                            358
                    [ oChooseTag                                      *
                        | RADDR:                                    358

                        | *:
                            oAbort(eConvertType)                    265   488
                    ]
                | *:
            ]
            oPopS                                                   192
            @PushResultDescriptor                                     *
        | labelspec:                                                 87
            oPushTagWithInputTag                                    220
            @TranslateLabelSpec                                       *
        | label:                                                     86
            [ oChooseSEmpty                                         312
                | yes:                                              435

                | *:
                    oAbort(eStackNotEmpty)                          265   491
```

```
            ]
            oPushTagWithInputTag                                          220
            [ oChooseTagDotState                                          313
                | UndefinedTagState:                                      383
                    oCreateLabelDescriptorAtTag                           172
                    oSetTagDotState(SpecifiedTagState)                    282   384
                | SpecifiedTagState:                                      384

                | *:
                    oAbort(eDuplicateDefinition)                          265   496
            ]
            oEmitToken(tdefstlab)                                         268   539
            oEmitTagMLabelDotP6Stlab                                      333
            oSetTagDotState(DefinedTagState)                              282   385
        | goto:                                                           73
            oVerifySMDataDotType(PADDR)                                   288   357
            @ForceTOSValue                                                *
            oPopS                                                         192
            [ oChooseSEmpty                                               312
                | yes:                                                    435
                    oEmitToken(tlongJump)                                 268   538
                    oEmitTagMlabelDotP6Stlab                              *
                    oEmitSMDataDotP6Level                                 326
                | *:
                    oAbort(eStackNotEmpty)                                265   491
            ]
        | switch:                                                         141
            [ oChooseSEmpty                                               312
                | yes:                                                    435

                | *:
                    oAbort(eStackNotEmpty)                                265   491
            ]
            oPushTagWithInputTag                                          220
            oVerifyTagDotState(UndefinedTagState)                         292   383
            oPushCountWithInputNumber                                     201
            oVerifySMDataDotType(INT)                                     288   350
            @ForceTOSValue                                                *
            oEmitToken(tcaseexpr)                                         268   532
            oPopS                                                         192
            oCreateSwitchDescriptorAtTagWithCountLabels                   174
            oEmitToken(tcasejump)                                         268   533
            oEmitValue(0)                                                 272
            oDecrementCountByOne                                          175
            oEmitCount                                                    323
            oEmitValue(0)                                                 272
            oEmitValue(0)                                                 272
            oPushCountWithTagMSwitchDotFirstP6Label                       206
            { oPushCountWithTagMSwitchDotlastP6Label                      *
              [ oChooseCountComparison                                    300
                  | CompGreater: >                                        477
                  | *:
              ]
              oPopCount                                                   187
              oEmitCount                                                  323
              oIncrementCountByOne                                        179
```

```
                }
                oPopCount                                                   187
                oPopCount                                                   187
                oSetTagDotState(DefinedTagState)                            282    385
        | sdest:                                                            134
                oPushTagWithInputTag                                        220
                oVerifyTagDotState(DefinedTagState)                         292    385
                oPushCountWithInputNumber                                   201
                oPushCountWithTagMSwitchDotFirstP6Label                     206
                [ oChooseSEmpty                                             312
                        | yes:                                              435

                        | *:
                                oAbort(eStackNotEmpty)                      265    491
                ]
                oEmitToken(tdeflabel)                                       268    529
                oAddCount                                                     *
                oEmitCount                                                  323
                oPopCount                                                   187
        | fJumpIf:                                                           69
                oPushDownIndexes                                            211
                @ForceTOSValue                                                *
                @ForceSOSValue                                                *
                @CheckRelation                                                *
                oPushCountWithInputByte                                     199
                oSetIndexSubCountDotState(ReferencedIndexState)             278    378
                oSetIndexSubCountDotP6LabelToNewLabel                       225
                oPushTagWithSMDataDotType                                   221
                [ oChooseTagMTypeStructured                                 315
                        | yes:                                              435
                                oEmitToken(tcompstruct)                     268    528
                                oEmitCount                                  323
                                oEmitTagMTypeDotLength                      335
                        | *:
                                oEmitToken(tcompare)                        268    527
                                oEmitCount                                  323
                                oEmitTagMTypeDotP6Type                      336
                ]
                oPopCount                                                   187
                oPopS                                                       192
                oPopS                                                       192
                oPopTag                                                     194
                oEmitToken(tnot)                                            268    518
                oEmitToken(tfalsejump)                                      268    531
                oEmitIndexSubCountDotP6Label                                324
                oSetIndexSubCountDotState(ReferencedIndexState)             278    378
                oPopCount                                                   187
        | fJump:                                                             68
                oPushDownIndexes                                            211
                [ oChooseSEmpty                                             312
                        | yes:                                              435

                        | *:
                                oAbort(eStackNotEmpty)                      265    491
                ]
                oPushCountWithInputByte                                     199
```

```
                oVerifyIndexSubCountDotState(UndefinedIndexState)          285   376
                oSetIndexSubCountDotP6LabelToNewLabel                      225
                oEmitToken(tjump)                                          268   530
                oEmitIndexSubCountDotP6Label                               324
                oSetIndexSubCountDotState(ReferencedIndexState)            278   378
                oPopCount                                                  187
        | ldest:                                                            66
                oPushDownIndexes                                           211
                [ oChooseSEmpty                                            312
                        | yes:                                             435

                        | *:
                            oAbort(eStackNotEmpty)                         265   491
                ]
                oPushCountWithInputByte                                    199
                oVerifyIndexSubCountDotState(ReferencedIndexState)         285   378
                oEmitToken(tdeflabel)                                      268   529
                oEmitIndexSubCountDotP6Label                               324
                oSetIndexSubCountDotState(UndefinedIndexState)             278   376
                oVerifyAllIndexesUndefined                                 255
                oPopCount                                                  187
        | bjumpif:                                                          18
                oPushDownIndexes                                           211
                @ForceTUSValue                                               *
                @ForceSUSValue                                               *
                @CheckRelation                                               *
                oPushCountWithInputByte                                    199
                oVerifyIndexSubCountDotState(DefinedIndexState)            285   377
                oPushTagWithSMDataDotType                                  221
                oSwapCount                                                   *
                [ oChooseTagMTypeStructured                                315
                        | yes:                                             435
                            oEmitToken(tcompstruct)                        268   528
                            oEmitCount                                     323
                            oEmitTagMTypeDotLength                         335
                        | *:
                            oEmitToken(tcompare)                           268   527
                            oEmitCount                                     323
                            oEmitTagMTypeDotP6Type                         336
                ]
                oPopCount                                                  187
                oPopS                                                      192
                oPopS                                                      192
                oPopTag                                                    194
                oEmitToken(tnot)                                           268   518
                oEmitToken(tfalsejump)                                     268   531
                oEmitIndexSubCountDotP6Label                               324
                oPopCount                                                  187
                oSetIndexSubCountDotState(UndefinedIndexState)             278   376
        | bjump:                                                            17
                oPushDownIndexes                                           211
                oVerifyIndexSubCountDotState(DefinedIndexState)            285   377
                oSetIndexSubCountDotState(UndefinedIndexState)             278   376
                oVerifyAllIndexesUndefined                                 255
                oEmitToken(tjump)                                          268   530
                oEmitIndexSubCountDotP6Label                               324
```

```
    | bdest:                                                        16
        oPushDownIndexes                                            211
        [ oChooseSEmpty                                             312
            | yes:                                                  435

            | *:
                oAbort(eStackNotEmpty)                              265    491
        ]
        oPushCountWithInputByte                                     197
        oVerifyIndexSubCountDotState(UndefinedIndexState)           285    376
        oSetIndexSubCountDotP6LabelToNewLabel                       225
        oEmitToken(tdeflabel)                                       268    529
        oEmitIndexSubCountDotP6Label                                324
        oPopCount                                                   187
        oSetIndexSubCountDotState(DefinedIndexState)                278    377
    | skipif:                                                       139
        @ForceTOSValue                                              *
        @ForceSOSValue                                              *
        @CheckRelation                                              *
        oPushTagWithSMDataDotType                                   221
        oPopS                                                       192
        oPopS                                                       192
        [ oChooseTagMTypeStructured                                 315
            | yes:                                                  435
                oEmitToken(tcompstruct)                             268    528
                oEmitCount                                          323
                oEmitTagMTypeDotLength                              335
            | *:
                oEmitToken(tcompare)                                268    527
                oEmitCount                                          323
                oEmitTagMTypeDotP6Type                              336
        ]
        oPushP6LabelWithNewLabel                                    212
        oEmitToken(tnot)                                            268    518
        oEmitToken(tfalsejump)                                      268    531
        oEmitP6Label                                                325
        oMarkAndCopySSection                                        181
        @ProgramElements                                            739
        =endskip                                                     57
        [ oChooseReachable                                          303
            | InterpassTrue:                                        432

            | *:
                oSetReachableToTrue                                 227
        ]
        [ oChooseSEmpty                                             312
            | yes:                                                  435

            | *:
                oAbort(eStackNotEmpty)                              265    491
        ]
        oEmitToken(tdeflabel)                                       268    529
        oEmitP6Label                                                325
        oPopP6Label                                                 190
        oPopMarkedSSectionMEmpty                                    189
    | if:                                                            74
```

```
            @ForceTOSValue                                          *
            @ForceSOSValue                                          *
            @CheckRelation                                          *
            oPushTagWithSMDataDotType                             221
            [ oChooseTagMTypeStructured                           315
                | yes:                                            435
                    oEmitToken(tcompstruct)                       268   528
                    oEmitCount                                    323
                    oEmitTagMTypeDotLength                        335
                | *:
                    oEmitToken(tcompare)                          268   527
                    oEmitCount                                    323
                    oEmitTagMTypeDotP6Type                        336
            ]
            oPopCount                                             187
            oPopS                                                 192
            oPopS                                                 192
            oPushP6LabelWithNewLabel                              212
            oEmitToken(tfalsejump)                                268   531
            oEmitP6Label                                          325
            oMarkAndCopySSection                                  181
            @ProgramElements                                      739
            [ =
              | else:                                              48
                  @ForceTOSValue                                    *
                  oPushP6LablewithNewLabel                          *
                  oEmitToken(tjump)                               268   530
                  oEmitP6Label                                    325
                  oSwapMarkedSSections                            246
                  oSwapP6Labels                                   247
                  oEmitToken(tdeflabel)                           268   529
                  oPopP6Label                                     190
                  @ProgramElements                                739
              | *:

            ]
            =endif                                                 50
            @ForceTOSValue                                          *
            oMergeSSections                                       183
            oEmitToken(tdeflabel)                                 268   529
            oEmitP6Label                                          325
            oPopP6Label                                           190
        | bseg:                                                    20
            oMarkSSection                                         182
            oBeginSegment                                         166
            oPushReachableWithTrue                                214
            @ProgramElements                                      739
            =eseg                                                  21
            oPopReachablee                                          *
            oEndSegment                                           177
            oPopMarkedSSectionMEmpty                              189
        | profile:                                                109
            oPushTagWithInputTag                                  220
            @TranslateProfile                                       *
        | *: >
      ] ]
```

```
;



TranslateProfile:                                                          *

    oCreateProfileDescriptorAtTag                                          173
    oSetTagDotState(DefinedTagState)                                       282     385
    [ =
        | known:                                                           85
            oSetTagMProfileDotProfileKind(knownProfileKind)                283     406
            oPushTagWithInputTag                                           220
            oCreateBodyDescriptorAtTag                                     171
            oSetTagDotState(SpecifiedTagState)                             282     384
            oSetTagMBodyDotP6ptabToNewP6ptab                               *
            oSetTagSosMProfileDotBodyTagToTag                              240
            oPopTag                                                        194
            oSetTagMProfileDotIdToInputString                              234
            oSetTagMProfileDotNature(SIMULANature)                         284     414
            oPopTag                                                        194
        | system:                                                          142
            oSetTagMProfileDotProfileKind(systemProfileKind)               283     407
            oPushTagWithInputTag                                           220
            oCreateBodyDescriptorAtTag                                     171
            oSetTagDotState(DefinedTagState)                               282     385
            oSetTagSosMProfileDotBodyTagToTag                              240
            oEmitToken(textId)                                             268     513
            oSetTagMBodyDotP6ptabToNewP6ptab                               232
            oEmitTagMBodyDotP6ptab                                         *
            oPopTag                                                        194
            oSetTagMProfileDotIdToInputString                              234
            oEmitTagMProfileDotId8Chars                                    334
            oSetTagMProfileDotNature(CNature)                              284     415
            oSetTagMProfileDotIdToInputString                              234
        | external:                                                        63
            oSetTagMProfileDotProfileKind(externalProfileKind)             283     408
            oPushTagWithInputTag                                           220
            oCreateBodyDescriptorAtTag                                     171
            oSetTagDotState(DefinedTagState)                               282     385
            oSetTagSosMProfileDotBodyTagToTag                              240
            oEmitToken(textId)                                             268     513
            oSetTagMBodyDotP6ptabToNewP6ptab                               232
            oEmitTagMBodyDotP6ptab                                         *
            oPopTag                                                        194
            oSetTagMProfileDotIdToInputString                              234
            oSetTagMProfileDotNatureToInputString                          235
            oEmitTagMProfileDotId8Chars                                    334
            oPopTag                                                        194
        | interface:                                                       82
            oSetTagMProfileDotProfileKind(interfaceProfileKind)            283     409
            oSetTagMProfileDotIdToInputString                              234
        | *:
            oSetTagMProfileDotProfileKkind(ordinaryProfileKind)            *       410
            oSetTagMProfileDotNature(SIMULANature)                         284     414
    ]
    oPushCount(P6FirstFormalDisplacement)                                  276     473
```

```
    oSetAreaToCount(LocalArea)                                          *     365
    [ [ =
          | import:                                                    84
              oPushTagWithInputTag                                     220
              oPushUpdateWithTagAndUndefine                            224
              oVerifyTagDotState(UndefinedTagState)                    292   383
              oCreateDataDescriptorAtTag(ImportDescrKind)              267   398
              oAppendTagSosMProfileDotImportListWithTagMAttribute      165
              @QuantityDescriptor                                       *
              @AllocateLocal                                            *
              oPopUpdateAndRestoreTag                                  195
              oPopTag                                                  194
          | *: >
    ] ]
    [ =
          | export:                                                    62
              oPushTagWithInputTag                                     220
              oPushUpdateWithTagAndUndefine                            224
              oVerifyTagDotState(UndefinedTagState)                    292   383
              oCreateDataDescriptorAtTag(ExportDescrKind)              267   399
              oSetTagSosMProfileDotExportToTag                          *
              @QuantityDescriptor                                       *
              oPopUpdateAndRestoreTag                                  195
              oPopTag                                                  194
          | exit:                                                      61
              oPushTagWithInputTag                                     220
              oPushUpdateWWithTagAndUndefine                            *
              oCreateDataDescriptorAtTag(ExitDescrKind)                267   400
              oSetTagMDataDotTagToTag                                   *
              oSetTagSosMProfileDotExitToTag                            *
              oPushCount(P6ReturnAddressDisplacement)                  276   472
              oPopCount                                                187
              oPopUpdateAndRestoreTag                                  195
              oPopTag                                                  194
          | *:
    ]
    *endprofile                                                         53
;


AllocateLocal:                                                          *

    oAlignAreaToCount(LocalArea)                                       266   365
    oPopCount                                                          187
    oSetTagMDataDotDisplacementToCount                                 233
    oIncrAreaDisplacementByCount(LocalArea)                            275   365
    oPopCount                                                          187
;

TranslateConstSpec:                                                     *

    oVerifyTagDotState(UndefinedTagState)                              292   383
    oCreateDataDescriptorAtTag(ConstantDescrKind)                      267   391
    oSetTagDotState(SpecifiedTagState)                                 282   384
    @QuantityDescriptor                                                 *
    @AllocateConstant                                                   *
```

```
    oPopTag                                                   194
;

AllocateConstant:                                              *

    oAlignAreaToCount(ConstantArea)                           266    364
    oPopCount                                                 187
    oSetTagMDataDotDisplacementToCount                        233
    oIncrAreaDisplacementByCount(ConstantArea)                275    364
    oPopCount                                                 187
;

QuantityDescriptor:                                            *

% pre: a data descriptor on top of S-stack
% post: alignment and size on count Tos and Sos respectively

    oPushTagWithInputTag                                      220
    [ oChooseTag                                               *
         | CHAR:                                              349
              oPushCount(1)                                   276
              oPushCountWithDuplicate                         198
         | BOOL, SINT:                                        348    359
              oPushCount(2)                                   276
              oPushCountWithDuplicate                         198
         | REAL, UADDR, AADDR, SIZE, PADDR, RADDR:            351    354    35
              oPushCount(4)                                   276
              oPushCountWithDuplicate                         198
         | GADDR:                                             356
              oPushCount(8)                                   276
              oPushCount(4)                                   276
         | LREAL:                                             352
              oPushCount(8)                                   276
              oPushCountWithDuplicate                         198
         | INT:                                               350
              [ *
                  | range:                                    121
                      oTurnOnTagSosMDataDotHasRange           253
                      oPushCountWithInputNumber               201
                      oSetTagSosMDataDotLowerBoundToCount     237
                      oPushCountWithInputNumber               201
                      oSetTagSosMDataDotUpperBoundToCount     238
                      oComputeRangeSizeAndAlignmentOnCount    170
                  | *:
                      oPushCount(4)                           275
                      oPushCountWithDuplicate                 198
              ]
         | *:
              oVerifyTagDotKindIsType                         257
              [ oChooseTagMTypeDotHasIndefRepField             *
                   | InterpassTrue:                           432
                        =fixrep                               67
                        oPushCountWithInputNumber             201
                        oTurnOnTagSosMDataDotHasFixrep        254
                        oSetTagSosMDataDotFixrepToCount       236
                        oPushTagWithTagMTypeDotIndefRepTag    223
```

```
                                oPushCountWithTagMTypeDotLength                        210
                                oPopTag                                                194
                                oMultiplyCount                                           *
                                oPushCountWithTagMTypeDotIndefAlternateLength          209
                                oAddCount                                                *
                            | *:
                                oPushContWithTagMTypeDotLength                           *
                        ]
        ]
        oPushCountWithTagMTypeDotAlignment                                               *
        [ =
            | rep:                                                                     127
                oSwapCount                                                               *
                oPushCountWithInputNumber                                              201
                oSetTagSosMDataDotRepToCount                                           239
                oMultiplyCount                                                         184
                oSwapCount                                                               *
            | *:
                oPushCount(1)                                                          276
                oSetTagSosMDataDotRepToCount                                           239
                oPopCount                                                              187
        ]
        oSetTagSosMDataDotTypeToTag                                                      *
        oSwapCount                                                                       *
        oSetTagSosMDataDotLengthToCount                                                  *
        oSwapCount                                                                       *
        oPopTag                                                                        194
    ;


    PushResultDescriptor:                                                                *

        oPushSNewResultDescriptor                                                      215
        oSetSMDataDotTypeToTag                                                         231
        oPopTag                                                                        194
        oSetSMDataDotOutputPointer                                                     230
        oTurnOnSMDataDotHasP6Counterpart                                               250
    ;

    TranslatePush:                                                                       *

        oPushTagWithInputTag                                                           220
        [ oChooseTagDotState                                                           313
            | UndefinedTagState:                                                       383
                    oAbort(eUndefinedTag)                                              265    493
            | *:
        ]
        oPushSWithTagMDataDescriptor                                                   218
        oEmitToken(tpushaddr)                                                          268    511
        oEmitSMDataDotP6Mode                                                           327
        oEmitTagMDataDotDisplacement                                                   332
        oEmitSMDataDotP6Level                                                          326
        oSetSMDataDotMode(REFMode)                                                     279    370
        oSetSMDataDotOutputPointer                                                     230
        oPopTag                                                                        194
    ;
```

```

TranslateSelect:                                                        *

    oPushTagWithInputTag                                              220
    oVerifySDotMode(REFMode)                                            *     370
    oPops                                                               *
    oPushSNewResultDescriptor                                         215
    oPushSWithTagMDataDescriptor                                      218
    oPushTagWithSMDataDotType                                         221
    oPopS                                                             192
    oPopTag                                                           194
    oSetSMDataDotTypeToTag                                            231
    oPopTag                                                           194
    oEmitToken(tfield)                                                268     514
    oEmitTagMDataDotDisplacement                                      332
;

TranslateRemote:                                                        *

    oPushTagWithInputTag                                              220
    oVerifySMDataDotType(UADDR)                                       288     354
    @ForceTOSValue                                                      *
    oPopS                                                             192
    oPushSWithTagMDataDescriptor                                      218
    oEmitToken(tfield)                                                268     514
    oEmitTagMDataDotDsplacement                                         *
    oSetSMDataDotMode(REFMode)                                        279     370
    oSetSDotHasP6Counterpart                                            *
    oPopTag                                                           194
;

TranslateIndex:                                                         *

    oVerifySMDataDotType(INT)                                           *     350
    @ForceTOSValue                                                      *
    oVerifySSosDotMode(REFMode)                                         *     370
    oPushTagWithSSosMDataDotType                                      222
    oPopS                                                             192
    oPushCountWithTagMTypeDotLength                                     *
    oEmitToken(tsimpleindex)                                          268     554
    oEmitCount                                                        323
    oPopCount                                                         187
    oPopTag                                                           194
;


TranslateAccess:                                                        *

    oPushCountWithInputByte                                           199
    oPushTagWithInputTag                                              220
    oPushSWithTagMDataDescriptor                                      218
    oVerifySDotDescriptorKind(AttributeDescrKind)                     286     392
    oEmitToken(taccess)                                               268     544
    oEmitCount                                                        323
    oEmitTagMDataDotDisplcement                                         *
    oPopCount                                                         187
```

```
    @PushResultDescriptor                                              1766
    oSetSMDataDotMode(REFMode)                                         279   370
;


TranslateLabelSpec:                                                      *

    oVerifyTagDotState(UndefinedTagState)                              292   383
    oCreateLabelDescriptorAtTag                                        172
    oSetTagDotState(SpecifiedTagState)                                 282   384
;

ForceTOSValue:                                                           *

    [ oChooseSMDataDotMode                                             306
          | REFMode:                                                   370
              [ oChooseSMDataDotType                                   307
                    | INT, BOOL, CHAR, REAL, LREAL, SIZE, DADDR, AADDR,   350   348   349   351   352   35
                      GADDR, PADDR, RADDR:                             356   357   358
                          oSetSMDataDotMode(VALMode)                   279   371
                          oEmitToken(tpushInd)                         268   510
                          oPushTagWithSMDataDotType                    221
                          oEmitTagMTypeDotP6Type                       336
                          oPopTag                                      194
                    | *:
                          oSetSMDataDotMode(VALADDRMode)               279   372
              ]
          | *:
    ]
;

ForceSOSValue:                                                           *

    [ oChooseSSosMDataDotMode                                          310
          | REFMode:                                                   370
              [ oChooseSSosMDataDotType                                311
                    | INT, BOOL, CHAR, REAL, LREAL, SIZE, DADDR, AADDR,   350   348   349   351   352   35
                      GADDR, PADDR, RADDR:                             356   357   358
                          oSetSSosMDataDotMode(VALMode)                281   371
                          oEmitTokenAfterSSos(tpush2ndInd)             269   551
                          oPushTagWithSSosMDataDotType                 222
                          oEmitTagMTypeDotP6TypeBeforeSMDataDotOutputPointer   337
                          oPopTag                                      194
                    | *:
                          oSetSSosMDataDotMode(VALADDRMode)            281   372
              ]
          | *:
    ]
;


TranslateNonRecordConstValue:                                            *

    [ =
        | true:                                                        148
            oEmitToken(tpushconst)                                     268   509
```

```
            oEmitValue(True)                                          272     *
            oPushTagWithType(BOOL)                                    277   348
        | false:                                                       64
            oEmitToken(tpushconst)                                    268   509
            oEmitValue(False)                                         272     *
            oPushTagWithType(BOOL)                                    277   348
        | c_char:                                                      25
            oPushCountWithInputByte                                   199
            oEmitToken(tpushconst)                                    268   509
            oEmitCount                                                323
            oPopCount                                                 187
            oPushTagWithType(CHAR)                                    277   349
        | c_int:                                                       28
            oPushCountWithInputIntegerLiteral                         200
            oEmitToken(tpushconst)                                    268   509
            oEmitCount                                                323
            oPushTagWithType(INT)                                     277   350
        | c_real:                                                      33
            oStoreValueWithInputRealLiteralAtCount                    245
            oFixThis                                                  178
            oPushTagWithType(REAL)                                    277   351
        | c_lreal:                                                     29
            oStoreValueWithInputLongRealLiteralAtCount                243
            oFixThis                                                  178
            oPushTagWithType(LREAL)                                   277   352
        | text:                                                       144
            oStoreValueWithInputLongStringAndPushCountLength          244
            oPushTagWithType(CHAR)                                    277   349
        | c_size:                                                      35
            oPushTagWithInputTag                                      220
            oPushCountWithTagMTypeDotLength                           210
            oEmitToken(tpushconst)                                    268   509
            oEmitCount                                                323
            oPopCount                                                 187
            oPopTag                                                   194
            oPushTagWithType(SIZE)                                    277   353
        | nosize:                                                     100
            oEmitToken(tpushconst)                                    268   509
            oEmitValue(0)                                             272
            oPushTagWithType(SIZE)                                    277   353
        | anone:                                                       10
            oEmitToken(tpushconst)                                    268   509
            oEmitValue(0)                                             272
            oPushTagWithType(AADDR)                                   277   355
        | c_oaddr:                                                     30
            oPushTagWithInputTag                                      220
            oPushSWithTagMDataDescriptor                              218
            [ oChooseSDotDescriptorKind                               304
                | GlobalDescrKind, ConstantDescrKind:                 389   391

                | *:
                    oAbortIeTagKind)                                  265   494
            ]
            oEmitToken(tpushaddr)                                     268   511
            oEmitSMDataDotPbMode                                      327
            oEmitSMDataDotDisplacement                                328
```

```
            oEmitSMDataDotP6Level                                        326
            oPop$                                                        192
            oPopTag                                                      194
            oPushTagWithType(OADDR)                                      277    354
        | onone:                                                         103
            oEmitToken(tpushconst)                                       268    509
            oEmitValue(0)                                                272
            oPushTagWithType(OADDR)                                      277    354
        | gnone:                                                          72
            oEmitToken(tpushconst)                                       268    509
            oEmitValue(0)                                                272
            oPushTagWithType(GADDR)                                      277    356
        | c_paddr:                                                        31
            oPushTagWithInputTag                                         220
            oEmitToken(tpushlabel)                                       268    537
            oEmitValue(P6stlabMode)                                      272    454
            oEmitTagMLabelDotP6stlab                                       *
            oPopTag                                                      194
            oPushTagWithType(PADDR)                                      277    357
        | nowhere:                                                       102
            oEmitToken(tpushconst)                                       268    509
            oEmitValue(0)                                                272
            oPushTagWithType(PADDR)                                      277    357
        | c_raddr:                                                        32
            oPushTagWithInputTag                                         220
            oEmitToken(tpushlabel)                                       268    537
            oEmitValue(P6stlabMode)                                      272    454
            oEmitTagMLabelDotP6stlab                                       *
            oPopTag                                                      194
            oPushTagWithType(RADDR)                                      277    358
        | nobody:                                                         99
            oEmitToken(tpushconst)                                       268    509
            oEmitValue(0)                                                272
            oPushTagWithType(RADDR)                                      277    358
        | *:
            { [ =
                | c_dot:                                                  26
                    oPushTagWithInputTag                                 220
                    oPushCountWithTagMDataDotDisplacement                205
                | *:
                    >
                    oAddCount                                              *
            ] }
            [ =
                | c_aaddr:                                                24
                    oPushTagWithInputTag                                 220
                    oPushTagWithType(AADDR)                              277    355
                | c_gaddr:                                                27
                    oPushTaggWithInputTag                                  *
                    oPushTagWithType(GADDR)                              277    356
            ]
        ]
;

AddAttributeDisplacementToCount:                                           *

```

```
    oPushTagWithInputTag                                                  223
    oPushSWithTagMDataDescriptor                                          218
    oVerifySDotDescriptorKind(AttributeDescrKind)                         286    392
    oAddCount                                                             *
    oPopS                                                                 192
    oPopTag                                                               194
;


CheckRelation:                                                            *

    oVerifySTosSosMDataDotTypesMatch                                      256
    oPushTagWithSMDataDotType                                             221
    [ oChooseTagMTypeDotHasIndefRepField                                  *
          | yes:                                                          435
              oPushCountWithSMDataDotFixrep                               203
              oPushCountWithSSosMDataDotFixrep                            204
              [ oChooseCountComparison                                    300
                    | CompEqual:                                          478

                    | *:
                        oAbort(eFixrepsDiffer)                            265    495
              ]
              oPopCount                                                   187
              oPopCount                                                   187
          | *:
    ]
    oPopTag                                                               194
    [ -
       | Qlt:                                                             115
             [ oChooseSMDataDotType                                       307
                   | CHAR, INT, REAL, LREAL, SIZE, DADDR:                 349    350    351    352    353    354
                       oPushCountWithRelationValue(P6lessthan)            *      443
                   | *:
                       oAbort(eIllegalRelation)                           265    489
             ]
       | Qle:                                                             116
             [ oChooseSMDataDotType                                       307
                   | CHAR, INT, REAL, LREAL, SIZE, DADDR:                 349    350    351    352    353    354
                       oPushCountWithRelationValue(P6lessthanequal)       *      444
                   | *:
                       oAbort(eIllegalRelation)                           265    489
             ]
       | Qeq:                                                             117
             oPushCountWithRelationValue(P6equal)                         *      440
       | Qge:                                                             118
             [ oChooseSMDataDotType                                       307
                   | CHAR, INT, REAL, LREAL, SIZE, DADDR:                 349    350    351    352    353    354
                       oPushCountWithRelationValue(P6greaterthanequal)    *      442
                   | *:
                       oAbort(eIllegalRelation)                           265    489
             ]
       | Qgt:                                                             119
             [ oChooseSMDataDotType                                       307
                   | CHAR, INT, REAL, LREAL, SIZE, DADDR:                 349    350    351    352    353    354
                       oPushCountWithRelationValue(P6greaterthan)         *      441
```

```
                        | *:
                            oAbort(eIllegalRelation)                      265   489
                    ]
               | One:                                                     120
                    oPushCountWithRelationValue(P6notequal)                 *   445
       ]
   ;
   _!_                                                                      *     *
```

```
cross reference   * is def   = is asg

-A-
 AADDR                        355*   917  1214  1216  1241  1705  1865  1884  1947  2006
 access                         5*   948
 accessv                        6*   950
 add                            7*   953
 AddAttributeDisplaceme-     2014*
   ntToCount
 AllocateConstant            1679  1683*
 AllocateLocal               1633  1664*
 alt                            8*
 and                            9*  1056
 anone                         10*  1944
 asscall                       11*
 assign                        12*   783
 asspar                        13*
 assrep                        14*
 attr                          15*
 AttributeDescrKind           392*  1843  2018

-B-
 bdest                         16*  1456
 bjump                         17*  1449
 bjumpif                       18*  1421
 body                          19*   590
 BodyDescrKind                393*
 BOOL                         348*  1038  1057  1059  1064  1067  1074  1077  1085  1088  1096  1105  1108  1111  1130
                             1144  1702  1865  1884  1905  1909
 bseg                          20*  1554
 BuiltinTypeDescrKind         395*

-C-
 call                          22*
 call_tos                      23*
 CHAR                         349*  1135  1137  1153  1699  1865  1884  1915  1931  2047  2054  2063  2070
 CheckRelation               1116  1367  1425  1474  1514  2025*
 CNature                      415*  1600
 compare                       36*  1113
 CompEqual                    478*  2034
 CompGreater                  477*  1338
 CompLess                     479*
 const                         37*   744
 ConstantArea                 364*  1685  1688
 ConstantDescrKind            391*  1676  1952
 constspec                     38*   647   693   741
 convert                       39*  1132
 c_aaddr                       24*  2004
 c_char                        25*  1910
 c_dot                         26*  1996
 c_gaddr                       27*  2007
 c_int                         28*  1916
 c_ireal                       29*  1925
 c_oaddr                       30*  1948
 c_paddr                       31*  1972
```

```
 c_raddr                    32*  1983
 c_real                     33*  1921
 c_record                   34*
 c_size                     35*  1932

-D-
 DeclaredTypeDescrKind     394*
 deco                       40*   880
 DefinedIndexState         377*  1427   1451   1470
 DefinedTagState           385*  1301   1347   1350   1575   1592   1606
 delete                     41*
 deref                      42*   847
 dinitarea                  43*
 dist                       44*   890
 div                        45*  1011
 dsize                      46*   900
 dup                        47*   756

-E-
 eArithmeticType           487*   969    988   1008   1027   1053
 eConvertType              488*  1140   1147   1169   1186   1205   1212   1219   1235   1258   1266   1273
 eDeferredMacros           497*   577
 eDuplicateDefinition      496*  1297
 eFixrepsDiffer            495*  2037
 eIllegalRelation          489*  2050   2057   2066   2073
 else                       48*  1535
 EmitOps                   321*
 empty                      49*
 endif                      50*  1548
 endmacro                   51*
 endmodule                  52*   582    585    592
 endprofile                 53*  1660
 endprogram                 54*   566
 endrecord                  55*
 endroutine                 56*
 endskip                    57*  1494
 endtaglist                155*   424    684
 endvisible                153*   422    612    729
 eProfileDescriptor        490*   763
 eqv                        58*  1095
 erCheckCodeMismatch       502*   610    672    691
 erModuleNamesMismatch     501*   608    670    689
 eseg                       21*  1559
 eStackNotEmpty            491*   776   1287   1312   1319   1357   1397   1412   1462   1505
 eTagKind                  494*  1955
 eType                     492*   927
 eUndefinedTag             493*  1780
 eval                       59*
 existing                   60*   604
 exit                       61*  1648
 ExitDescrKind             400*  1651
 export                     62*  1639
 ExportDescrKind           399*  1643
 external                   63*  1602
 externalProfileKind       408*  1603
```

```
-F-
 False                     1908
 false                       64* 1906
 fdest                       66* 1406
 fetch                       65*  832
 fixrep                      67* 1731
 fjump                       68* 1391
 fjumpif                     69* 1363
 FoceTOSValue              753
 ForceSOSValue             873   884   894   902   919   937   955   974   994  1013  1034  1060  1069  1079  1090
                          1115  1366  1424  1473  1513  1879*
 ForceTOSValue             758   784   800   815   817   833   858   863   868   871   882   892   918   935   943
                           973   993  1012  1032  1041  1058  1068  1078  1089  1109  1114  1304  1325  1365  1423
                          1512  1536  1549  1813  1826  1860*
 ForceTOsValue             836
 ForceTOSVaue              952  1097

-G-
 GADDR                     356*  835   852   923   931  1225  1237  1239  1708  1866  1885  1971  2009
 getobj                     70*  941
 global                     71*  583
 GlobalArea                363*
 GlobalDescrKind           389* 1952
 GlobalModule              584
 gnone                      72* 1968
 goto                       73* 1302

-I-
 if                         74* 1511
 imp                        75* 1087
 import                     84* 1626
 ImportDescrKind           398* 1630
 inco                       76*  869
 index                      77*  864
 indexv                     78*  866
 info                       79*  651   717
 initarea                   80*
 insert                     81*  653   723
 INT                       350*  901   934   942   962   981  1001  1020  1031  1033  1046  1137  1149  1151  1175
                          1324  1714  1825  1865  1884  1920  2047  2054  2063  2070
 interface                  82* 1616
 interfaceProfileKind      409* 1617
 internal                   83*
 interpassFalse            433*
 interpassTrue             432* 1496  1730

-K-
 known                      85* 1577
 knownProfileKind          406* 1578

-L-
 label                      86* 1282
 LabelDescrKind            397*
 labelspec                  87*  649   702  1279
 line                       88*  719
 local                      89*
```

```
 LocalArea               365* 1624  1666  1669
 LocalDescrKind          390*
 locate                   90*  916
 LREAL                   352*  966   985  1005  1024  1050  1163  1180  1188  1190  1711  1865  1884  1928  2047  2054
                        2063  2070

-M-
 macro                    91*  576
 MacroDefinitions        581
 main                     92*  574
 mark                     93*
 mcall                    94*
 module                   95*  425   578   588
 ModuleHeader            589   599*
 mpar                     96*
 mult                     97*  992

-N-
 neg                      98* 1040
 no                      436*
 nobody                   99* 1990
 nosize                  100* 1940
 not                     101* 1107
 nowhere                 102* 1979

-O-
 oAbort                  265*  577   763   776   927   969   988  1008  1027  1053  1140  1147  1169  1186  1205  1212
                        1219  1235  1258  1266  1273  1287  1297  1312  1319  1357  1397  1412  1462  1505  1780  1955
                        2037  2050  2057  2066  2073
 oAddCount              1360  1740  2001  2019
 oAddcount               162*
 OADDR                   354*  872   878   883   888   891   893   921   936   946  1221  1223  1249  1705  1812  1865
                        1884  1963  1967  2047  2054  2063  2070
 oAddTagListEntryFromTa- 163*
   gAndCount
 oAddTaglistEntryFromTa- 678
   gAndCount
 oAdjustMinimumTagToTag  164*
 oAlignAreaToCount       266* 1666  1685
 oAppendTagSosMProfileD- 165* 1631
   otImportListWithTagMAttribute
 oBeginSegment           166* 1556
 oCheckTagMRecordInitia- 167*
   lization
 oChooseCountComparison  300* 1337  2033
 oChooseCountSosValue    301*
 oChooseCountValue       302*  769
 oChooseReachable        303* 1495
 oChooseSDotDescriptorK- 304*  761  1951
   ind
 oChooseSEmpty           312*  771  1283  1306  1315  1353  1393  1408  1458  1501
 oChooseSMDataDotHasFix- 308*
   rep
 oChooseSMDataDotHasP6C- 305*
   ounterPart
 oChooseSMDataDotHasRan- 309*
```

```
  ge
oChooseSMDataDotMode       306*  788   805  1226  1242  1250  1862
oChooseSMDataDotType       307* 1134  1864  2046  2053  2062  2069
oChooseSSosMDataDotMod-    310*  822  1881
  e
oChooseSSosMDataDotTyp-    311*  920  1883
  e
oChooseTag                 961   980  1000  1019  1045  1136  1143  1150  1172  1189  1208  1215  1222  1238  1261  1269
                           1698
oChooseTagDotState         313* 1290  1778
oChooseTagMTypeDotHasI-    1729  2029
  ndefRepField
oChooseTagMTypeDotHasR-    314*
  epField
oChooseTagMTypeStructu-    315* 1118  1372  1430  1478  1516
  red
oCloseInputFileAndPop      168*  613   686   730
oCloseOuputCopyFile        640
oCloseOutputCopyFile       169*  624
oComputeRangeSizeAndAl-    170* 1722
  ignmentOnCount
oCreateBodyDescriptorA-    171* 1580  1591  1605
  tTag
oCreateDataDescriptorA-    267* 1630  1643  1651  1676
  tTag
oCreateLabelDescriptor-    172* 1292  1856
  AtTag
oCreateProfileDescript-    173* 1574
  orAtTag
oCreateSwitchDescripto-    174* 1328
  rAtTagWithCountLabels
oDecrementCountByOne       175*  781  1331
oEmitCount                 323* 1121  1125  1332  1342  1361  1375  1379  1433  1437  1481  1485  1519  1523  1832  1845
                           1913  1919  1936
oEmitIndexSubCountDotP-    324* 1388  1403  1417  1446  1455  1468
  6Label
oEmitP6Label               325* 1491  1508  1531  1539  1552
oEmitSMDataDotDisplace-    328* 1959
  ment
oEmitSMDataDotP6Level      326* 1310  1787  1960
oEmitSMDataDotP6Mode       327* 1785  1958
oEmitSSosMDataDotP6Mod-    329*
  e
oEmitStringToOutputCop-    330*
  Y
oEmitStringToOutputCop-    176*  619   621   628   630
  yFile
oEmitTagMBodyDotP6Plab     331*
oEmitTagMBodyDotP6plab     1596  1610
oEmitTagMDataDotDispla-    332* 1786  1806
  cement
oEmitTagMDataDotDisplc-    1846
  ement
oEmitTagMDataDotDsplac-    1817
  ement
oEmitTagMLabelDotP6Stl-    333* 1300
```

```
  ab
oEmitTagMLabelDotP6stl- 1976  1987
  ab
oEmitTagMlabelDotP6Stl- 1309
  ab
oEmitTagMProfileDotidd-  334* 1599  1614
  Chars
oEmitTagMTypeDotLength   335*  794   811   828   905  1122  1376  1434  1482  1520
oEmitTagMTypeDotP6Type   336*  791   808   825   838   908  1126  1380  1438  1486  1524  1870
oEmitTagMTypeDotP6Type-  337* 1889
  BeforeSMDataDotOutputPointer
oEmitToken               268*  759   790   793   807   810   824   827   837   849   874   885   895   903   906   909
                         911   924   938   944   957   976   996  1015  1035  1042  1061  1071  1081  1092  1093  1094
                        1100  1110  1120  1124  1154  1158  1164  1176  1181  1193  1199  1228  1232  1244  1248  1252
                        1256  1299  1308  1326  1329  1359  1374  1378  1386  1387  1402  1416  1432  1436  1444  1445
                        1454  1467  1480  1484  1489  1490  1507  1518  1522  1530  1538  1542  1551  1594  1608  1784
                        1805  1816  1831  1844  1868  1903  1907  1912  1918  1935  1941  1945  1957  1965  1969  1974
                        1980  1985  1991
oEmitTokenAfterS3os      270*
oEmitTokenAfterSSos      269*  804   821  1887
oEmitTokenToOutputCopy-  271*  618   623   627   639
  File
oEmitValue               272*  850   875   907   912   925   963   965   967   982   984   986  1002  1004  1006  1021
                        1023  1025  1047  1049  1051  1082  1101  1155  1156  1159  1160  1161  1162  1165  1166  1167
                        1177  1178  1179  1182  1183  1184  1194  1195  1196  1197  1200  1201  1202  1203  1229  1233
                        1245  1253  1330  1333  1334  1904  1908  1942  1946  1966  1970  1975  1981  1986  1992
oEmitValueAfterS3os      274*
oEmitValueAfterSSos      273*
oEmitValueAreaCountByt-  338*
  es
oEndSegment              177* 1561
oFixThis                 178* 1923  1927
oIncrAreaDisplacementB-  275* 1669  1698
  yCount
oIncrementCountByOne     179* 1343
oMakeTaglistTableEmpty   180*  673   731
oMarkAndCopySSection     181* 1492  1532
oMarkSSection            182* 1555
oMergeSSections          183* 1550
oMultiplyCount           184* 1751
oMulyiplyCount          1738
onone                    103* 1964
oOpenOutputCopyFilewit-  186*  626
  hStringTaglist
oOpenOutputCopyFilewit-  185*  616
  hStringVisible
oPopCount                187*  636   679   773  1341  1345  1346  1362  1382  1390  1405  1420  1440  1447  1469  1526
                        1655  1667  1670  1686  1689  1756  1833  1847  1914  1937  2039  2040
oPopIndexes              188*
oPopMarkedSSectionMEmp-  189* 1510  1562
  ty
oPopP6Label              190* 1509  1543  1553
oPopReachable            191*
oPopReachablee          1560
oPopS                    192*  765   780   797   798   814   831   839   851   876   877   886   887   896   897   913
                         929   930   939   940   945   959   960   978   979   998   999  1017  1018  1036  1037  1044
```

```
                            1062  1063  1072  1073  1083  1084  1102  1103  1128  1129  1277  1305  1327  1383  1384  1441
                            1442  1476  1477  1527  1528  1801  1814  1829  1961  2020
oPops                       1797
oPopString                   193*  567   681   697   706   711   716   725   727   732   733
oPopTag                      194*  634   680   734   735   796   813   830   844  1385  1443  1584  1587  1597  1611  1615
                            1635  1647  1657  1680  1737  1762  1770  1790  1802  1804  1820  1834  1871  1890  1938  1962
                            1977  1988  2021  2043
oPopUpdateAndRestoreTa-      195* 1634  1646  1656
  g
oPushAndOpenInputFileW-      197*  668
  ithStringTaglist
oPushAndOpenInputFileW-      196*  606   687
  ithStringVisible
oPushContWithTagMTypeD-     1742  1830
  otLength
oPushCount                   276* 1623  1654  1700  1703  1706  1709  1710  1712  1724  1754
oPushCountWithDuplicat-      198* 1701  1704  1707  1713  1725
  e
oPushCountWithInputByt-      199*  768  1368  1399  1414  1426  1464  1840  1911
  e
oPushCountWithInputInt-      200* 1917
  egerLiteral
oPushCountWithInputNum-      201*  635   677  1323  1351  1718  1720  1732  1749
  ber
oPushCountWithRelation-     2048  2055  2060  2064  2071  2076
  Value
oPushCountWithSMDataDo-      202*
  tDisplacement
oPushCountWithSMDataDo-      203* 2031
  tFixrep
oPushCountWithSSosMDat-      204* 2032
  aDotFixrep
oPushCountWithTag            910
oPushCountWithTagMData-      205* 1998
  DotDisplacement
oPushCountWithTagMSwit-      206* 1335  1352
  chDotFirstP6Label
oPushCountWithTagMSwit-      207*
  chDotLastP6Label
oPushCountWithTagMSwit-     1336
  chDotlastP6Label
oPushCountWithTagMType-     1745
  DotAlignment
oPushCountWithTagMType-      208*  904
  DotHasRepField
oPushCountWithTagMType-      209* 1739
  DotIndefAlternateLength
oPushCountWithTagMType-      210* 1736  1934
  DotLength
oPushDownIndexes             211* 1364  1392  1407  1422  1450  1457
oPushP6LabelWithNewLab-      212* 1488  1529
  el
oPushP6LableWithNewLab-     1537
  el
oPushReachableWithFals-      213*
  e
```

```
>PushReachableWithTrue    214* 1557
>PushSNewResultDescrip-   215*  840  1104  1768  1798
  tor
>PushStringWithInputSt-   216*  564   601   602   663   664   724   726
  ring
>PushSWithDuplicate       217*  757
>PushSWithTagMDataDesc-   218* 1783  1799  1815  1842  1950  2017
  riptor
>PushTagAndStringWithI-   219*  633   665   666   676   694   699   703   708   713
  nput
>PushTaggWithInputTag    2008
>PushTagWithInputTag      220*  742   747   842  1133  1280  1289  1321  1349  1564  1579  1590  1604  1627  1640  1649
                         1697  1777  1795  1811  1841  1933  1949  1973  1984  1997  2005  2016
>PushTagWithSMDataDotT-   803
  Ype
>PushTagWithSMDataDotT-   221*  787   958   977   997  1016  1043  1117  1371  1428  1475  1515  1800  1869  2028
  ype
>PushTagWithSSosMDataD-   222*  820  1828  1888
  otType
>PushTagWithTagMTypeDo-   223* 1735
  tIndefRepTag
>PushTagWithType          277*  852   878   888   898   914   931   946  1038  1064  1074  1085  1111  1130  1905  1909
                         1915  1920  1924  1928  1931  1939  1943  1947  1963  1967  1971  1978  1982  1989  1993  2006
                         2009
>PushUpdateWithTagAndU-   224* 1628  1641
  ndefine
>PushUpdateWWithTagAnd-  1650
  Jndefine
>r                        104* 1066
>rdinaryProfileKind       410* 1620
>SetAreaToCount          1624
>SetIndexSubCountDotP6-   225* 1370  1401  1466
  LabelToNewLabel
>SetIndexSubCountDotSt-   278* 1369  1389  1404  1418  1448  1452  1470
  ate
>SetReachableToFalse      226*
>SetReachableToTrue       227* 1499
>SetSDotHasP6CounterPa-   228*
  rt
>SetSDotHasP6Counterpa-  1819
  rt
>SetSMDataDotMode         229*  279*  841  1788  1818  1849  1867  1873
>SetSMDataDotOutputPoi-   230*  846  1106  1771  1789
  nter
>SetSMDataDotType         280* 1105
>SetSMDataDotTypeToTag    231*  843  1769  1803
>SetSSosMDataDotMode      281* 1886  1892
>SetTagDotState           282* 1293  1301  1347  1575  1581  1592  1606  1677  1857
>SetTagMBodyDotP6labTo-  1582
  NewP6plab
>SetTagMBodyDotP6plabT-   232* 1595  1609
  oNewP6plab
>SetTagMDataDotDisplac-   233* 1668  1687
  ementToCount
>SetTagMDataDotTagToTa-  1652
  g
```

```
oSetTagMProfileDotIdTo-   234* 1585  1598  1601  1612  1618
  InputString
oSetTagMProfileDotNatu-   284* 1586  1600  1621
  re
oSetTagMProfileDotNatu-   235* 1613
  reToInputString
oSetTagMProfileDotProf-   283* 1578  1589  1603  1617
  ileKind
oSetTagMProfileDotProf- 1620
  ileKkind
oSetTagSosMDataDotFixr-   236* 1734
  epToCount
oSetTagSosMDataDotLeng- 1760
  thToCount
oSetTagSosMDataDotLowe-   237* 1719
  rBoundToCount
oSetTagSosMDataDotRepT-   239* 1750  1755
  oCount
oSetTagSosMDataDotType- 1758
  ToTag
oSetTagSosMDataDotUppe-   238* 1721
  rBoundToCount
oSetTagSosMProfileDotB-   240* 1583  1593  1607
  odyTagToTag
oSetTagSosMProfileDotE- 1653
  xitToTag
oSetTagSosMProfileDotE- 1644
  xportToTag
oSkipInputString          241*  579   580
oStoreValuePackedAddre-   242*
  ssConstantTag
oStoreValueWithInputLo-   243* 1926
  ngRealLiteralAtCount
oStoreValueWithInputLo-   244* 1930
  ngStringAndPushCountLength
oStoreValueWithInputRe-   245* 1922
  alLiteralAtCount
oSwapCount              1429  1748  1752  1759  1761
oSwapMarkedSSections      246* 1540
oSwapP6Labels             247* 1541
oSwapString               248*  605   609   615   620   625   629   667   671   685   690
otEndTaglist              424*  639
otEndVisible              422*  623
otModule                  425*  618
oTranslateTagUsingTagL-   249*
  istTable
oTranslateTagUsingTagl-   695   700   704   709   714
  istTable
otTaglist                 423*  627
oTurnOffOutputCopying     252*  654
oTurnOnOutputCopying      251*  617   656
oTurnOnSMDataDotHasP6C-   250*  845  1772
  ounterpart
oTurnOnTagSosMDataDotH-   254* 1733
  asFixrep
oTurnOnTagSosMDataDotH-   253* 1717
```

```
   asRange
 otVisible               421*
 oVerifyAllIndexesUndef-  255* 1419  1453
   ined
 oVerifyIndexSubCountDo-  285* 1400  1415  1427  1451  1465
   tState
 oVerifySDotDescriptorK-  286* 1843  2018
   ind
 oVerifySDotMode         1796
 oVerifySMDataDotMode     287*  819   848
 oVerifySMDAtaDotTYpe     917
 oVerifySMDataDotTYpe    1825
 oVerifySMDataDotType     288*  835   870   881   891   901   934   942  1031  1057  1067  1077  1088  1096  1108  1303
                         1324  1812
 oVerifySSosDotMode      1827
 oVerifySSosMDataDotMod-  289*  785   802
   e
 oVerifySSosMDataDotTyp-  290*  872   883   893   936  1033  1059
   e
 oVerifySTosSosMDataDot-  256*  786   801   818   956   975   995  1014  1070  1080  1091  1099  2027
   TypesMatch
 oVerifyStringEqualsInp-  291*  608   610   670   672   689   691
   utString
 oVerifyTagDotKindIsTyp-  257* 1728
   e
 oVerifyTagDotState       292* 1322  1350  1629  1642  1675  1855

-P-
 P6equal                  440* 1101  2060
 P6False                  430*
 P6FirstFormalDisplacem-  473* 1623
   ent
 P6globalMode             450*
 P6greaterthan            441* 2071
 P6greaterthanequal       442* 2064
 P6constMode              451*
 P6lessthan               443* 2048
 P6lessthanequal          444* 2055
 P6localMode              452*
 P6notequal               445* 1082  2076
 P6plabMode               453*
 P6real                   460*  967   986  1006  1025  1051
 P6ReturnAddressDisplac-  472* 1654
   ement
 P6shortreal              459*  965   984  1004  1023  1049  1160  1201
 P6stlabMode              454* 1975  1986
 P6True                   429*
 P6word                   458*  875   907   912   925   963   982  1002  1021  1047  1195  1229  1245
 P6wordpair               461* 1251
 PADDR                    357* 1260  1262  1303  1705  1866  1885  1978  1982
 Pas32Nature              416*
 pop                      105*  760
 popall                   106*  767
 precall                  107*
 prefix                   108*
 profile                  109*  650   707  1563
```

```
ProfileDescrKind          402*
profileDescrKind          762
program                   110*  563
ProgramBody               565   571*
ProgramElements           575   591   739* 1493  1533  1544  1558
push                      111*  749
pushc                     112*  754
pushlen                   113*
PushResultDescriiptor     899
PushResultDesCriptor      915
PushResultDescriptor      853   879   889   932   947   971   990  1029  1039  1055  1065  1075  1086  1112  1131  1278
                         1766* 1848
PushResultDesriptor      1010
pushv                     114*  751

Q-
Qeq                       117* 2059
Qge                       118* 2061
Qgt                       119* 2068
Qle                       116* 2052
Qlt                       115* 2045
Qne                       120* 2075
QuantityDescriptor       1632  1645  1678  1692*

R-
RADOR                     358* 1268  1270  1705  1866  1885  1989  1993
range                     121* 1716
REAL                      351*  964   983  1003  1022  1048  1157  1171  1173  1198  1705  1865  1884  1924  2047  2054
                         2063  2070
record                    122*  648   698   746
refer                     123*  834
ReferencedIndexState      378* 1369  1389  1404  1415
REFMode                   370*  785   802   819   841   848  1227  1243  1251  1788  1796  1818  1827  1849  1863  1882
rem                       124* 1030
remote                    125*  859
remotev                   126*  861
rep                       127* 1747
repcall                   128*
restore                   129*
ResultDescrKind           401*
routine                   130*
routinespec               131*  712
rupdate                   132*  816

S-
save                      133*
sdest                     134* 1348
select                    135*  854
selectv                   136*  856
setobj                    137*  933
setswitch                 138*  652   721
SIMULANature              414* 1586  1621
SINT                      359* 1702
SIZE                      353*  870   881   898   914  1207  1209  1705  1865  1884  1939  1943  2047  2054  2063  2070
skipif                    139* 1471
SpecifiedTagState         384* 1293  1294  1581  1677  1857
```

```
Program                 561*
tdconv                  465* 1159
tdlengthen              467* 1182
tdlongconv              466* 1165
tdround                 468* 1177  1194
tdshorten              1200
ub                      140*  972
witch                   141* 1314
witchDescrKind          396*
ystem                   142* 1588
ystemProfileKind        407* 1589

-
access                  544* 1844
add                     522*  837   874   911   924   957
ag                      143*  632   675
aglist                  154*  423   669
and                     519* 1061  1093
assign                  516*  790
caseexpr                532* 1326
casejump                533* 1329
compare                 527* 1081  1100  1124  1378  1436  1484  1522
compstruct              528* 1120  1374  1432  1480  1518
convrt                  536* 1158  1164  1176  1181  1193  1199
copy                    517*  793
deflabel                529* 1359  1416  1467  1507  1542  1551
defstlab                539* 1299
div                     525* 1015
dup                     545*  759   804   821
ext                     144* 1929
extid                   513* 1594  1608
falsejump               531* 1387  1445  1490  1530
field                   514* 1805  1816
getobj                  553*  944
jump                    530* 1402  1454  1538
lconst                  535*
longjump                538* 1308
mod                     526* 1035
mul                     524*  906   996
neg                     521* 1042
not                     518* 1092  1094  1110  1386  1444  1489
or                      520* 1071
pop                     534* 1256
pop2nd                  550* 1248
push2ndind              551* 1887
pushaddr                511* 1784  1957
pushconst               509*  849   903   909  1232  1903  1907  1912  1918  1935  1941  1945  1965  1969  1980  1991
pushind                 510* 1228  1244  1252  1868
pushlabel               512*  537* 1974  1985
range                   515* 1154
ranslateAccess          949   951  1838*
ranslateConstantDefin-  745
 ition
ranslateConstSpec       647   696   743  1673*
ranslateIndex           865   867  1823*
ranslateInfo            651   718
```

```
 TranslateInsert           655   661*
 TranslateLabelSpec        649   705  1281  1853*
 TranslateLine             720
 TranslateNonRecordCons-   755  1899*
   tValue
 TranslateProfile          650   710  1565  1572*
 TranslatePush             750   752  1775*
 TranslateRecordDescrip-   648   701   748
   tor
 TranslateRemote           860   862  1809*
 TranslateRoutineSpec      715
 TranslateSelect           855   857  1793*
 TranslateSetSwitch        652   722
 TranslateVisibleDeclar-   611   622   644*
   ations
 True                     1904
 true                      148* 1902
 trupdate                  548*  824
 trupdatecopy              549*  827
 tsetobj                   552*  938
 tsimpleindex              554* 1831
 tsub                      523*  885   895   976
 tupdate                   546*  807
 tupdatecopy               547*  810
 t_geto                    145*
 t_inito                   146*
 t_seto                    147*

-U-
 UndefinedIndexState       376* 1400  1418  1448  1452  1465
 UndefinedTagState         383* 1291  1322  1629  1642  1675  1779  1855
 update                    149*  799

-V-
 VALADDRMode               372*  792   826  1873  1892
 VALADDRRMode              809
 VALMode                   371*  789   806   823  1867  1886
 visible                   152*  421   607   688

-X-
 xor                       150* 1076

-Y-
 yes                       435*  772  1119  1284  1307  1316  1354  1373  1394  1409  1431  1459  1479  1502  1517  2030

-Z-
 zeroarea                  151*

-_-
 _                         557*  557* 2079* 2079*

end exref  522 identifiers  2059 total references
3314 collisions.
```

DESIGN AND IMPLEMENTATION OF S-COMPILER INTERPASS

by

Mija Chung Kim

B.A., Seoul National University, 1975
M.A., Seoul National University, 1978

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

ABSTRACT

The purposes of the report were: 1) to introduce and analyse the Perkin Elmer SIMULA system; 2) to describe the design of Interpass which is a main part of the compiler to be used in the Perkin Elmer SIMULA system.

The Perkin Elmer SIMULA system is a complete SIMULA system on 8/32 Perkin Elmer machines. It is to be run under the Unix operating system. It consists of a portable front-end compiler, a portable run-time support system and an S-Compiler. The portable front-end compiler and the portable run-time support system are provided by S-Port which was brought from Norway. Also there is an Environment Interface to facilitate the access to the Unix operating system.

The interpass is a brand new pass, and it plays the major role for the S-Compiler along with the modified passes of the Pascal/32 compiler. The Pascal/32 compiler is an extensive version of Hartmann compiler and it is available here at K.S.U. The Interpass is written in Syntax/Semantic Language(S/SL) and this primitive language S/SL calls semantic routines coded in Pascal.

The design of Interpass and the coding of S/SL which we intensively worked for are almost 70 % completed. The future work can be done about the design of Interpass by adding more data structures and semantic operations which we didn't do, especially for record descriptors and utility routines.