AN OBJECT-ORIENTED EXTENSION OF PROLOG

by

MICHAEL KEVIN KONDOR

B. S., Purdue University, 1986

A MASTERS THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Approved by:

William Hankley
Major Professor

CONTENTS

Chapter Page

Table of Figures.....iii

1.0 Introduction..... 1

2.0 Concepts of Object Oriented Programming..... 2
2.1 Object Oriented Programming Approach..... 2
2.2 Object Oriented Language Structures..... 5
2.2.1 Language Features..... 5
2.2.1.1 Objects and Classes..... 5
2.2.1.2 Variables..... 6
2.2.1.3 Inheritance..... 8
2.2.1.4 Encapsulation..... 10
2.2.2 Advantages of Object Oriented Programming Language Structure... 13

3.0 Existing Work in Object Oriented Prolog..... 14
3.1 Concurrent Prolog Model..... 15
3.2 Vulcan--A Preprocessor for Concurrent Prolog..... 17
3.3 Data Abstraction in Prolog/KR..... 18
3.4 Zaniolo's Model..... 19
3.5 SPOOL..... 21

4.0 Object Oriented Extension of Prolog..... 23
4.1 Syntax and Structure of OOP Programs..... 24
4.1.1 Class Definition..... 25
4.1.2 Operations on Classes an Objects.... 28
4.1.3 The Root Class..... 31
4.1.4 OOP Example: Digital Logic Circuit Simulation..... 33
4.2 Underlying Data Structures and Run-Time Predicates..... 37
4.2.1 Underlying Data Structures for Class Definition..... 37
4.2.2 Underlying Data Structures for Instance Creation..... 40
4.2.3 Run-Time Predicates..... 42
4.2.3.1 Variable Access and Assignment Predicates..... 42
4.2.3.2 Message Sending Predicates. 46
4.3 Translation from OOP to C-Prolog..... 51
4.3.1 Scanner..... 51
4.3.2 Definite Clause Grammar Review..... 51

4.3.3 Parser..... 54
4.3.3.1 Parsing Class Definitions.. 54
4.3.3.2 Parsing an Instance Creation Statement..... 57
4.3.3.3 Parsing Top Level Message Transmissions..... 57
4.4 Execution..... 58
4.5 Conclusions and Discussion..... 59

References..... 61

Appendix A Sample Session with OOP..... 66
Appendix B OOP Interface Source..... 71
Appendix C Loader Source..... 72
Appendix D Scanner Source..... 73
Appendix E Parser Source (Definite Clause Grammar). 79
Appendix F Run-Time Support Predicates..... 97
Appendix G C-Prolog Target for Circuit Example.....102
Appendix H Message Trace of Circuit Example.....106
Appendix I Trace of Class Definition: "class and_c".....108
Appendix J Trace of "instance and1 isa and_c".....234
Appendix K Trace of Message Sending Parse: "and2 << in2(1).....247

List of Figures and Tables

Table 2.1	Object Oriented Definitions.....	11
Figure 4.1a	Three Types of OOP Statement.....	25
Figure 4.1b	Class Definition Syntax.....	26
Figure 4.1c	Syntax for Access of Instance and Class Variables.....	27
Figure 4.1d	Statement Syntax.....	30
Figure 4.2	Root Class Variables and Methods.....	32
Figure 4.3a	Circuit Example Diagram.....	34
Figure 4.3b	Circuit Example in OOP.....	35
Figure 4.4	Underlying Data Structures.....	38
Figure 4.5a	Instance Variable Clauses.....	44
Figure 4.5b	Class Variable Clauses.....	45
Figure 4.5c	**send_mesg** Clause.....	49

CHAPTER 1: INTRODUCTION

The purpose of this thesis is to present the design and implementation of OOP, a form of object-oriented Prolog.

Chapter 2 reviews common object-oriented language features, defines them, and, where appropriate, indicates their advantages. Object-oriented programming has evolved into two basic philosophies. These philosophies are discussed in the context of the definitions presented in this chapter.

Chapter 3 compares some of the existing object-oriented Prolog models. The principles behind the design of these models are discussed and a comparison of the models to standard object-oriented language principles is also made.

Chapter 4 presents the syntax for OOP and shows a sample OOP program. Then the data structures and run-time support predicates are explained. Following this, the translation of the OOP source to a target C-Prolog program is discussed using an example.

CHAPTER 2: CONCEPTS OF OBJECT ORIENTED PROGRAMMING

There are several definitions of object oriented programming, and most are partially founded on at least one of two primary philosophies. The first philosophy evolves from the premise that the abstract objects of a program have a one-to-one correspondence to concrete objects [Boo86,Ren82,Sha83, Sta86]. The second philosophy emphasizes programming language features that improve the process of software development [Car84,Cox84,Cox86,Cun86,Fuk86, Fuk87, Gol83,Hew79,Ing78,Ing81,Kah86,Mac85,Rob81,Sny86, Zan84]. These language characteristics include encapsulation, inheritance, and similar attributes.

2.1 Object Oriented Programming Approach

The first philosophy emphasizes the representation of the world situations in an analogous form. Since there is little or no translation involved between these two forms, it is supposed to be easier for inexperienced programmers to develop software. The emphasis is not on the

features of the programming language, but on the use of the programming language. It is the programmer's responsibility to use an object oriented approach.

An example of this approach can be found in [Sta86]. Prolog is used in an object oriented fashion without changing the actual language. Objects are represented using Prolog facts with the following form:

object(<name>,<methods>).

In this form, <name> is the name of the object and <methods> is the list of methods that are represented by Prolog rules. The following is an example from [Sta86]:

```
object(reg_polygon(No_of_sides,Length_of_side),
            [(perimeter(P):- P is
            No_of_sides*Length_of_side),
            description('a regular polygon') ]).
```

From this Prolog fact, we can deduce

- the object is "reg_polygon", and is specified with two parameters

- the object has a method that handles the message requesting the perimeter
- the object has a method that handles the request for a description of the object.

Prolog queries are used to send messages to objects:

```
:- send(<object>,<message>).
```

where send is a rule defined to process messages:

```
send(Object,Message) :-
    object(Object,Methods),
    get_method(Message,Methods, Method),
    call(Method).
get_method(Message,[First|Rest],Method) :-
    fact_or_rule(Message,First,Method).
get_method(Message,[_|Rest],Method) :-
    get_method(Message,Rest,Method).
fact_or_rule(Message,Message,true).
fact_or_rule(Message,(Message :- Body),Body).
```

Using the Prolog query form, and this "send" rule, we can send a message to the "reg_polygon" object requesting the perimeter of a polygon, where the polygon has 4 sides of length 5:

```
:- send(reg_polygon(4,5),perimeter(P)).
```

This instantiates P to the perimeter, 20.

2.2 Object Oriented Language Structures

2.2.1 Language Features

The second philosophy enforces object oriented programming techniques through the use of language structures. In object oriented programming, the object has a direct correspondence to the concrete objects of this world. All worldly objects have state(s); the type of the object will determine the state(s) it has. Therefore, the programmed objects should represent the object's state(s). The programmed objects will also provide encapsulation and inheritance (discussed below), similar to their concrete counterparts; an object's reaction to a particular situation is defined by its own private traits.

2.2.1.1 Object and Classes

The construction of an object involves defining its behavior. This behavior is outlined by

procedures called methods that are private to the object. These methods are used by the object to access its own states. In order to initiate a particular behavior in an object or obtain information about the variables of an object, a message must be sent to that object requesting that action or information. Messages customarily consist of the name of an object, the name of one of its methods, and any parameters required by the method. One or more objects which have the same variables and methods are considered to be members of the same class. If two or more objects can service the same messages, they have the same protocol. An object is described by one particular class and this object is an instance of that class. All objects in a class have the same behavior, but different states. The process of creating a new instance is called instantiation. Thus, a class is the definition of the characteristics belonging to a group of objects, and an instance or object is one member of that group.

2.2.1.2 Variables

The state of a given object is represented by its use of variables; an object's behavior is partially determined by its state when it is invoked. These variables are private to the object; the state of an object can only be changed by its own internal behavior. When choosing the states of an object to be stored in variables, there is some consideration to whether an individual state is a characteristic of single objects or a characteristic of all objects in the class. In order to make the distinction, there are class variables and instance variables. Class variables store information pertinent to all objects in a class. For example, if there exists a class called "dog", the fact that dogs have four legs would be stored in a class variable This variable is shared by all objects derived from this class. Instance variables, however, store information that is characteristic of one object in a class. Referring to the class "dog", if two objects named "spike" and "snoopy" were to represent a bulldog and beagle respectively, that information would be

represented in instance variables, possibly called "breed". Thus, Class variables are those saved states that are shared by all instances of a particular class, and instance variables are those private to a particular object or instance. Both types of variables are defined in the class definition. When an object is instantiated, new instance variables are also instantiated for the exclusive use of that object.

2.2.1.3 Inheritance

Inheritance is a powerful aspect of object oriented programming. It allows "implicit sharing between objects in a class" [Rob81]. This includes the sharing of methods and class variables. If a new class is created by modifying another class, the original class is the superclass and the new class is the subclass. According to [Rob81], these are the modifications that can be made when producing the subclass: "adding instance variables; providing new methods for some of the messages understood by the

superclass; providing methods for new messages (messages not understood by the superclass); adding class variables." In some systems (e.g. LOOPS), there are two types of inheritance: hierarchical and multiple. With hierarchical inheritance, the new class only inherits characteristics from one superclass. Using multiple inheritance, it is possible to create a new class which benefits from the characteristics of more than one superclass. Any conflicts that result from using multiple inheritance are resolved according to each language's conventions. For an example of hierarchical inheritance, assume that there is a class "domestic_animal". Using inheritance, we can create a new class "dog" that has the characteristics of "domestic_animal", as well as some new characteristics. Class "dog" inherits all class variables, instance variables, and methods of "domestic_animal". This set of inherited characteristics is augmented by its own specialized characteristics. Multiple inheritance is handled in a similar fashion.

2.2.1.4 Encapsulation

When using an object oriented programming language, encapsulation is also enforced. Encapsulation forces methods and variables to be private to particular objects or classes. Class variables are private to the class they are declared in, instance variables to the instance they are defined in, and methods to the class they are defined in. One of the objectives of encapsulation is to localize changes made in software maintenance. A change in how a method performs an operation should not affect the rest of the program.

Table 2.1 Object Oriented Definitions

Class

one or more objects which have similar characteristics; a group of objects with the same behavior; a group of objects with the same variables and procedures to manipulate those variables

Class Variable

a variable shared by all objects in one class

Inheritance

the transference or sharing of characteristics (methods and class variables) to a newly created class

Instance

an object; one member of a class

Instance Variable

a variable that is private to an object; an unshared variable

Instantiate

to create an object with an exact state from a given class of objects

Table 2.1 (continued) Object Oriented Definitions

Message

a means of initiating a particular behavior in an object or obtaining information about the variables of an object; a means of communication between objects

Method

an internal procedure of an object; it is defined by the class to which the object is a member

Object

something with saved states and its own procedures to manipulate those states

Protocol

the set of procedures or methods that an object can execute; the set of messages an object can service

Subclass

a class which inherited characteristics from (a) previously defined class(es); these characteristics can be modified.

Superclass

a class from which subclasses were formed

2.2.2 Advantages of Object Oriented Programming Language Structure

The first place advantages appear is in the program development process. Using classes and instantiation to create objects has the benefit of minimizing code duplication because re-entrant code can be used for the methods. This is not only true for objects within the same class, but also for inheritance between superclasses and subclasses. Class definitions also save the programmer from having to re-declare local variables for objects because the instance variables are instantiated with each new object. The scope of all variables is well-defined.

Object oriented language structures also reduce software maintenance. As a result of encapsulation, there are no side effects and all changes are localized to individual methods. Methods are private to the class they are defined in, so other objects do not depend on how the task is done.

CHAPTER 3: EXISTING WORK IN OBJECT ORIENTED PROLOG

This chapter discusses past work in object oriented Prolog. Five implementations of object oriented Prolog are surveyed in this chapter:

Shapiro;Takeuchi. "Object Oriented Programming in Concurrent Prolog"

- uses concurrent Prolog in an object oriented manner.

Kahn; Tribble; Dean; Miller "Objects in Concurrent Logic Programming Languages"

- provide syntactic extensions to the first model in order to make object oriented programming in concurrent Prolog simpler.

Nakashima; Suzuki "Data Abstraction in Prolog/KR"

- Data abstraction and inheritance in Prolog/KR are discussed. Analysis of this paper gives some insight to the life span of objects and the storage of states.

Chapter 3 Existing Work in Object Oriented Prolog

Zaniolo "Object-Oriented Programming in Prolog"

- Prolog is supplemented with object oriented constructs

Fukunaga; Hirose "An Experience with a Prolog-based Object-Oriented Language"

- Prolog is supplemented with object oriented constructs

The last two models supplement Prolog with object oriented constructs and had a significant influence on the design of the model implemented in this thesis. The first of these two models is by Zaniolo; the second is by Fukunaga and Hirose.

3.1 Concurrent Prolog Model

Shapiro and Takeuchi use Concurrent Prolog in an object-oriented style; they do not present any special object-oriented language structures. The paper is based on Concurrent Prolog available at ICOT, which uses active objects analogous to Actors. Actors is similar to the most frequent definition of object-oriented programming; the main difference is in

the terminology. In the Actors model, an object is an actor and "the behavior of each actor is defined by the relationships among the events which are caused by the actor" [Hew79]. The object-oriented style used in Shapiro and Takeuchi's paper is as follows:

(i) Each predicate is an object and each call to a predicate is an instance of that object.

(ii) When an object is called, the first parameter indicates which method should be used.

(iii) Inheritance is imitated by passing messages up the language structure.

Shapiro and Takeuchi rely heavily upon the use of streams. A stream is a list of messages used to maintain the history of states and messages sent. Stream variables (Prolog lists) are used in a recursive manner; each time a method (predicate) is invoked, the stream variable is instantiated to a pair consisting of the message for the predicate invoked and a new stream variable to be used in the

next predicate invocation. The life span of an object is restricted to the life of a predicate in successive recursive calls; once backtracking occurs, an object/predicate is terminated. This is the reason for the use of stream variables.

3.2 Vulcan--A Preprocessor for Concurrent Prolog

A preprocessor for Shapiro's object-oriented use of Concurrent Prolog is presented by Kahn, Tribble, Dean and Miller in [Kah86]. The purpose of the new language, called Vulcan, is to provide a better notation than that provided by Shapiro. Like Shapiro's paper, this one relies heavily upon the use of streams for maintaining a history of states and messages sent to objects. Vulcan processes clauses for methods and classes. A class clause is used for the declaration of the names, and possibly properties, of the state variables of that class. The Vulcan clause

```
class(some_class_name,[List,Of,State,Vars]).
```

is a declaration that all methods of some_class_name "have a message stream, and all the named instance variables." It also causes the Concurrent Prolog clause

```
make(some_class_name,[List,Of,State,Vars],
                              Some_Class_Name) :-

/* This makes a process consuming */
/* the stream Some_Class_Name     */
some_class_name(Some_Class_Name?,List,Of,State,Vars).
```

to be generated. To define a method for some_class_name using Vulcan, the user can use the clause

```
method(some_class_name,some_method_name(State,Vars)) :-
                        method_body.
```

Vulcan provides the programmer with a slightly higher-level notation than unmodified concurrent Prolog.

3.3 Data Abstraction in Prolog/KR

Nakashima and Suzuki, [Nak83], present the use of data abstraction and inheritance in Prolog/KR. Normal Process Objects (NPO) and Coroutine Process

Objects (CPO) are added to standard Prolog to produce an expansion of Prolog called Prolog/KR. Their function is to introduce histories into computation. NPO's are much like normal predicate calls. CPO's allow transfer of control and information. They continue to exist until they fall off the end of their code. Since they can be suspended, they can store state without remaining active; they simply pass control to another object and wait in an inactive state until control is returned to them or the program terminates. Using this feature, it is possible to mimic global storage of state and instance variables: just surround the body of the object's code with a loop which returns the object to the beginning of its code and transfers control to another object. Information is transferred through unification of variables.

3.4 Zaniolo's Model

Another object-oriented syntax structure for Prolog is presented by Zaniolo in [Zan84]. An object declaration has the form:

object with method-list

Message passing is also implemented using the infix operator ":". The syntax for a message clause is

... , object : method, ...

For a message call to succeed, the unification of the object and method must succeed; the proof of the method must also succeed. However, there is not any means to store the state of instance or global variables. The storage of global and local object states is a key feature of object-oriented programming. All states are static and are passed into an object as the initial states. Inheritance is also supported in a clear manner. The clause

sub_object isa object

declares that all the methods of object are inherited by sub_object. Since the isa operator is transitive, hierarchies can be built and multiple inheritance is also possible. If an isa clause is

followed by a non-empty body, the body is used to specify the conditions under which the inheritance relation holds.

3.5 SPOOL

SPOOL, an object oriented programming language based on Prolog, has been implemented by Fukunaga and Hirose [Fuk86,Fuk87]. It is now a program offering of IBM Japan under the name of VM/PROBJ. It supports object-oriented programming by providing the appropriate language constructs. Class definition has the form:

class <class name> has <properties>

The properties include the method definitions and any references to super-classes. Definitions of super-classes are put at the beginning of the properties section:

super-class some_class_name;

The rest of the properties section has the syntax

methods <predicates> end.

Each predicate is a method and is separated by a semicolon. A method is defined as:

<head> :- <body>

where the body is optional. Message passing is allowed with the following syntax:

<receiver> << <message>

Either the message or the receiver may have an unbound variable, thus allowing anonymous message passing. A message can be the head of a Prolog clause. SPOOL tries to unify a message with a head of a method. If a matching head is found, the body is evaluated in the usual succeed/fail manner. As a result of this type of evaluation, it is possible to get a "fail" result instead of an error when a non-existent method is invoked. Prolog allows incremental design of methods and formalization of domain knowledge into declarative data types.

CHAPTER 4: OBJECT ORIENTED EXTENSION OF PROLOG

This chapter is a description of the implementation of Object Oriented Prolog (OOP). OOP is a variation on SPOOL [], using much of the SPOOL syntax. The major difference between OOP and SPOOL is the implementation. OOP is implemented in C-Prolog using Definite Clause Grammars to translate the source OOP program into C-Prolog.

Section 4.1 is a detailed discussion of the OOP language. It includes a view of the syntax and the general structure of an OOP program, as well as information about the semantics of the language. Then an example OOP program is given.

Section 4.2 presents the Prolog data structures used to represent the objects of the OOP program and the run-time predicates that manipulate these data structures.

Section 4.3 shows how OOP source code is translated to C-Prolog. This includes discussion of the scanner, definite clause grammars, and the parser. The parser is a prototype that fully demonstrates the

OOP features used in the circuit example; the parser does not implement all of the features and should not be considered fully functional.

Section 4.4 describes the overall operation of the OOP interpreter: how to invoke the interpreter and what happens once it is invoked.

Section 4.5 concludes the chapter with the status of the implementation, some comments on the size of the translation, and some of the problems encountered during implementation.

4.1 Syntax and Structure of OOP Programs

The syntax of OOP programs is given in Figures 4.1a, 4.1b, 4.1c, 4.1d; they are integrated into this section and appear where they are discussed. OOP programs consist of a list of statements. Each statement is one of the following: class definition, instance creation, top level message transmission.

<class definition>

<instance creation>

<top level message transmission>

Figure 4.1a. Three Types of OOP Statement.

4.1.1 Class Definition

The declaration of classes (Figure 4.1b) includes the name of its superclass, class and instance variable names, and the class methods and their bodies. Currently, only a single superclass is allowed. With minor modification to the translator, multiple inheritance may be supported. Class names must be unique. "class_var" and "instance_var" declarations are optional. Class and instance variables can be accessed only by using methods defined for that class. The methods section may not be empty--if it were, there would be no means of interacting with objects in that class.

```
class <classname> has
  superclass <superclass>;
  [class_var <class_var_list>;]
  [instance_var <instance_var_list>;]
  methods

            [<methods>]+

end <classname>.
```

Figure 4.1b. Class Definition Syntax.

[ ] denotes optional pattern
[ ]+ denotes repeated pattern

Methods are written in Prolog form with the addition of OOP operators for accessing class and instance variables and passing messages. These operations are listed in Figures 4.1c and 4.1d, respectively.

There are two types of operations on instance and class variables: accessing the value of a variable and assignment of a value to a variable (Figure 4.1c). Accessing the value of an instance or class variable is indicated syntactically by using a prefix operator; a ":" indicates that the following variable is an

instance variable and a "::" indicates a class variable. Assignment to an instance or class variable is performed using either of the respective infix operators: "=:", "=::". The latter set of operators only indicate what type of variable is in the left hand side of the given expression.

:X	value of instance variable X
::X	value of class variable X
X =: Y	assign the value of variable Y to instance variable X; Y is neither a class or instance variable
X =:: Y	assign the value of variable Y to class variable X; Y is neither a class or instance variable
X =: :Y	assign the value of instance variable Y to instance variable X
X =: ::Y	assign the value of class variable Y to instance variable X
X =:: :Y	assign the value of instance variable Y to class variable X
X =:: ::Y	assign the value of class variable Y to class variable X

Figure 4.1c. Syntax for Access of Instance and Class Variables.

Methods are distinguished by functor name and arity, as **in1/1**. If, in trying to satisfy a method goal, more than one method exists for a given functor name and arity, OOP will continue trying to satisfy the given goal until either it succeeds or exhausts the list of methods for the object receiving the message. This is consistent with usual Prolog execution.

Inheritance of methods is handled in a manner similar to normal Prolog execution. If a method is declared in a class, a method with the same name and arity may still be inherited from one of its ancestors. The order of execution is determined by choosing the most local definition. Specifically, OOP will first attempt to satisfy the goal using the method as defined in the object's class. If the goal is not satisfied, then method definitions from the object's superclasses are attempted.

4.1.2 Operations on Classes and Objects

The other two types of statement are instance creation and message passing. These features are

shown in Figure 4.1d. Object names must be unique. There are four different <send_op>s in OOP. "<<" provides an operator that acts like message sending in most object oriented programming languages: if failure occurs, an error message is returned. The use of the question mark is a natural extension of the Prolog "fail" characteristic. When failure occurs, no error is reported--the attempt simply fails and backtracking occurs if possible. Usage of these operators is also defined in Figure 4.1d.

INSTANCE CREATION

(1) instance <instance_name> isa <class_name> .

MESSAGE PASSING

(2) <instance_name> <send_op> <method_name_and_args> .

MESSAGE SENDING OPERATORS

<<	results in an error if receiver does not exist or does not understand message
?<	fails w/o any warning if receiver does not exist
<?	fails w/o any warning if invalid message
?<?	fails w/o any warning, regardless of whether receiver exists or message is valid

Figure 4.1d. Statement Syntax.

4.1.3 The Root Class

The predefined root class called root must be the explicit superclass of some declared class. It defines standard variables and methods (Figure 4.2) that may be inherited by all other classes. These variables are accessed by using the methods of **root**, both of which are inherited by successive subclasses. These methods are used primarily for communication between objects and allowing objects to operate on themselves.

```
=========================================================

                    INSTANCE VARIABLES

_Class              /* Contains  the  name of the  class */
                    /* the object belongs to.           */

_Id                 /* Contains the name of the instance */

                         METHODS

    /* Instance "Id" is an object of type "Class" */
                  who_am_I(Class,Id).


      /* Asks "some_object" what its "Class" and */
      /* "Id" (instance name) is                 */
          some_object << who_are_you(Class,Id).


   /* sends the message "do_something" to "inst_1", */
   /* "inst_2", and "inst_3"                        */
  broadcast([inst_1,inst_2,inst_3],do_something(X,Y)).


  /* erases "some_object" and the associated clauses */
                some_object << erase_self.
```

Figure 4.2. Root Class Variables and Methods.

4.1.4 OOP Example: Digital Logic Circuit Simulation

The sample program shown in Figure 4.3b is used to illustrate OOP. The program realizes a simple logic circuit represented in Figure 4.3a. The program uses classes **circuit**, **and_c**, **or_c**, and **out_c**.

The class **circuit** defines characteristics common to all of the gates in the circuit. Its method **in1/1** stores the first input to the gate in an instance variable **Temp**. Method **connect/2** stores information about which input and gate is connected to the output of the current gate.

The class **and_c** defines a generic two-input and-gate. Since it has super_class **circuit**, it inherits all of the variables and methods from **circuit** and anything **circuit** inherited from **root**. The method **in2/1** determines what the output of the circuit is based upon its input value and the value previously stored in the instance variable **Temp** (by **in1/1**). The computed value is sent to the connected input of the next gate in the circuit, as determined by referencing the instance variables **To_inst** and **To_meth**.

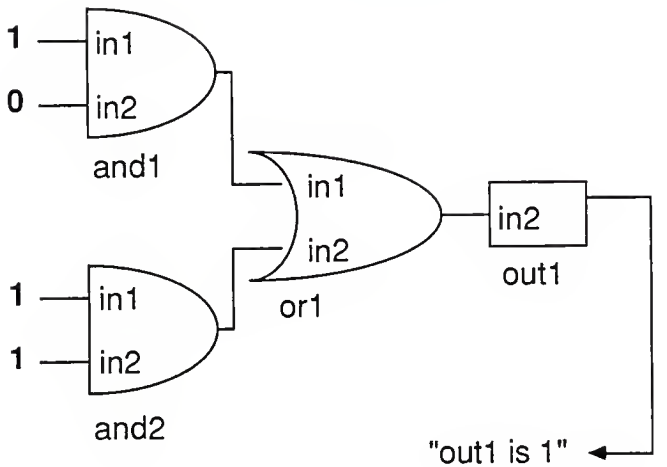

Figure 4.3a Circuit Example Diagram

```
( 1)    class circuit has
( 2)      superclass root;
( 3)      instance_var Temp, To_meth, To_inst;
( 4)      methods
( 5)         in1(X) :- Temp := X.
( 6)         connect(Inst, Meth) :- To_inst := Inst,
                                    To_meth := Meth .
( 7)    end circuit.

( 8)    class and_c has
( 9)      super_class circuit;
(10)      methods
(11)        in2(X) :- Out = (X /\ :Temp),
(12)                  (:To_inst << :To_meth(Out) ).
(13)    end and_c.

(14)    class or_c has
(15)       super_class circuit;
(16)       methods
(17)        in2(X) :- Out = (X \/ :Temp),
(18)                  (:To_inst << :To_meth(Out) ).
(19)    end or_c.

(20)    class out_c has
(21)       super_class circuit;
(22)       methods
(23)         in2(X) :- write(:_Id, " is ", X, "0).
(24)    end out.

(25)    instance and1 isa and_c .
(26)    instance and2 isa and_c.
(27)    instance or1  isa or_c.
(28)    instance out1 isa out_c.
(29)    and1 << connect(or1, in1).
(30)    and2 << connect(or1, in2).
(31)    or1  << connect(out1,in2).

(32)    and1 << in1(1).
(33)    and1 << in2(0).
(34)    and2 << in1(1).
(35)    and2 << in2(1).
```

Figure 4.3b. Circuit Example in OOP.

The class **or_c** is similar to **and_c**. The difference is the use of a logical **OR** instead of a logical **AND**.

Class **out_c** prints the final output of the whole circuit. Note the use of inheritance from the **root** class. In the write statement, the instance variable **_Id** is referenced. It is inherited from **root** through **circuit**.

The main section of the program contains the instantiation of objects and sending of messages to those objects. Lines (25) through (28) create the objects **and1**, **and2**, **or1**, and **out1**, each with characteristics of the appropriate classes. Lines (29) through (31) are messages to each of the logical gates to connect to the appropriate inputs of other gates. Line (29) is a message to object **and1** to connect its output to the **in1** line of the **or1** gate. Lines (32) through (35) supply the logical inputs to the gates.

4.2 Underlying Data Structures and Run-Time Predicates

The target C-Prolog version of an OOP program consists of specific data structures and predicates that reference and manipulate these structures. Figure 4.4 summarizes the underlying data structures discussed in the next two sections. The target for the circuit example of section 4.1.4 can be found in Appendix G.

4.2.1 Underlying Data Structures for Class Definition

When a class in OOP is defined, three types of information must be stored: the superclass relationships of the class; knowledge of the instance and class variables defined for that class; the methods objects in that class may respond to. For each superclass relationship, the following fact is asserted into the Prolog database:

classtree(superclass,subclass).

where **subclass** is the name of the class being defined and **superclass** is the name of a previously defined class. This fact is used to maintain inheritance for the classes.

UNDERLYING DATA STRUCTURES FOR CLASS DEFINITION

```
classtree(superclass,class).
```

defines class-superclass hierarchy

```
c_var_val(class,var_name,val).
```

maintains name, owner, and value of class variable

```
i_var_decl(class,var_name).
```

maintains name and class of instance variable

```
meth_defn(class,meth_name,
    (head(_Inst,orig_parms) :- body)).
```

maintains the class name, method name, and the clause for the method defined

UNDERLYING DATA STRUCTURES FOR INSTANCE CREATION

```
inst(instance_name,class_name)
```

keep track of instance name and the class it belongs to

```
i_var_val(inst,var_name,_UnInstant).
```

maintain the name of the instance, instance variable, and its value

```
c_var_path(inst,var_name,owner).
```

maintain the name of the instance, class variable, and the name of the class it belongs to

```
meth_path(inst,meth_name,def_class).
```

maintain the name of the instance, the name of the method, and where clause for the method can be found

Figure 4.4. Underlying Data Structures.

Since class and instance variables store different types of information, they require different types of data structures. A class variable stores a value shared by all objects in the class. Therefore, when a class variable is declared for a class, a fact is asserted that stores the value for that variable:

c_var_val(class,var_name,value).

class is the name of the class for which the class variable is defined, **var_name** is an atom representing the name of the variable, and **value** is the initial value of the variable. If no initial value is provided in the declaration, an uninstantiated variable is used.

An instance variable, however, stores a value for an individual object in the class. For this reason, only the declaration is asserted:

i_var_decl(class,var_name).

This fact indicates that every object in the class named **class** has an instance variable; the name of this variable is represented by the atom **var_name**.

The third type of information is method definition. For each method predicate in the class definition,

```
meth_defn(class,meth_name,
   (head(_Inst,orig_parms):-body)).
```

is asserted into the Prolog database. This fact contains several pieces of information: **class** is the name of the class in which the method is defined; **meth_name** is the functor of the method; **head** is the same as **meth_name**; **_Inst** is an uninstantiated variable that is necessary for processing messages; **orig_parms** is the original parameter list given in the OOP program; **body** is the executable C-Prolog version of the OOP method's body.

4.2.2 Underlying Data Structures for Instance Creation

When an instance is created, the fact that the object exists must be established:

```
inst(instance_name,class_name).
```

instance_name is the name of the object being instantiated, **class_name** is the name of the class to which the object belongs.

At the moment an object is instantiated, Prolog facts indicating the paths to its class variables and methods, and facts storing the values of

its instance variables, are asserted. The paths are asserted in order to increase execution speed. For each class variable accessible by an object, both local and inherited, the fact

```
c_var_path(inst,var_name,class).
```

is asserted. **inst** is the name of the object that can access the variable, var_name is an atom representing the name of the variable, and **class** is the name of the class that defined the variable. The name of the class is used to find the fact storing the value directly. A similar fact is used to store the path for each of an object's methods:

```
meth_path(inst,meth_name,def_class).
```

inst is the name of the object, **meth_name** the name of the method, **def_class** the name of the class that originally defined the method. The last parameter allows the run-time predicates to obtain the method directly, instead of searching inheritance lattices first.

For each of an object's instance variables, the fact

```
i_var_val(inst,var_name,_UnInstant).
```

stores the value. **inst** and **var_name** have the obvious values. **_UnInstant** is an uninstantiated variable, indicating that the variable does not have a value yet.

4.2.3 Run-Time Predicates

The run-time predicates for OOP are divided into two categories: variable access and assignment predicates; message sending predicates. Other semantic actions are implemented in the parser. All run-time predicates are in Appendix F. Parts of the predicates necessary for discussion in this section are integrated into the text.

4.2.3.1 Variable Access and Assignment Predicates

No special predicates are needed to access normal temporary variables or method parameters; these are handled automatically in Prolog. However, variable access and assignment predicates are needed to manipulate instance and class variable data structures. The operations that may be performed on an instance or class variable are setting the value and retrieving the

value of the variable. Figures 4.5a and 4.5b are the predicates for these operations on instance and class variables, respectively.

```
=========================================================

set_ivar(Inst,Name,Val) :-
     /* if it is a legal variable */
     /* retract the old value, assert the new value */
     retract(i_var_val(Inst,Name,_)),
     assert(i_var_val(Inst,Name,Val)),!.

set_ivar(Inst,Name,Val) :-
     /* else write appropriate message to screen */
     write(Name),
     write(' is not a legal instance variable name'),
     write(' in '),write(Inst),nl,!.


get_ivar(Inst,Name,Val) :-
     /* if it exists, instantiate Val */
     i_var_val(Inst,Name,Val), nonvar(Val),!.

get_ivar(Inst,Name,Val) :-
     /* if the variable is not already instantiated, */
     /* write an appropriate message to the screen */
     i_var_val(Inst,Name,Val), var(Val), !,
     write(Name),write(' is not yet instantiated in '),
     write(Inst),nl,!,fail.

get_ivar(Inst,Name,Val) :-
     /* else write appropriate message to screen */
     write(Name),
     write(' is not an instance variable for '),
     write(Inst),nl,!,fail.
```

Figure 4.5a. Instance Variable Clauses.

```
set_cvar(Inst,Name,Val) :-
     /* if it is a legal variable */
     /* retract the old value, assert the new value */
     c_var_path(Inst,Name,Owner),
     retract(c_var_val(Owner,Name,_)),
     assert(c_var_val(Owner,Name,Val)),!.

set_cvar(Inst,Name,Val) :-
     /* else write appropriate error */
     /* message to screen */
     write(Name),
     write(' is not a legal class variable name'),
     write(' in '),write(Inst),nl,!.


get_cvar(Inst,Name,Val) :-
     /* if it exists, instantiate Val */
     c_var_path(Inst,Name,Owner),
     c_var_val(Owner,Name,Val), nonvar(Val),!.

get_cvar(Inst,Name,Val) :-
     /* if the variable is not already instantiated, */
     /* write an appropriate message to the screen */
     c_var_path(Inst,Name,Owner),
     c_var_val(Owner,Name,Val), var(Val), !,
     write(Name),write(' is not yet instantiated in '),
     write(Inst),nl,!,fail.

get_cvar(Inst,Name,Val) :-
     /* else the variable is not a class */
     /* var for this instance */
     write(Name),
     write(' is not a class variable for '),
     write(Inst),nl,!,fail.
```

Figure 4.5b. Class Variable Clauses.

The predicate **set_ivar(Inst,Name,Val)** ascertains that **Name** is the name of a legal instance variable for **Inst**, retracts the old value, and asserts the new value. If it is not a legal variable, the expected error message is reported.

The predicate **get_ivar(Inst,Name,Val)** is used to retrieve the value of an instance variable. **get_ivar** ascertains that **Name** is a valid variable for **Inst**, verifies that the variable has an instantiated value, and returns the value in **Val**. If the variable is either not set or not valid, the expected error is reported.

The predicates for class variables are **set_cvar/3** and **get_cvar/3**. They operate in a manner analogous to their instance variable counterparts.

4.2.3.2 Message Sending Predicates

The run-time support predicates shown in Figure 4.5c perform operations for message sending. A partial trace of **send_mesg/2** for the circuit example is in Appendix H. For the predicate **send_mesg/2**, **Meth** is the name of the method being invoked, **Inst** is

the instance receiving the message, and **Args** is the list of arguments to the method. Line (2) checks that the method is defined and instantiates **Def_class** to the name of the class that defined the method. Using **Def_class**, line (3) is able to proceed directly to the Prolog fact containing the method and instantiate the method's code to **Method**. Line (4) asserts **Method** at the beginning of the database, effectively stacking method definitions. Lines (5) and (6) break **Method** into components and instantiate the arguments of the method with the list of **Args** and **Inst** of (1). In (5), If is **:-**, **Goal** is the atom corresponding to the principal functor and uninstantiated arguments of **Method**, and **Body** is the body of the method. Line (6) takes **Goal** and instantiates the arguments of the method to **Inst**, the name of the instance being sent the message, and **Args**, the list of arguments given to **send_mesg**. Line (7) calls the predicate that was asserted in (4), using the goal constructed in (6). After the call has completed, (8) retracts the code for the method so that no other instance can access that particular

instance of the method. The remaining predicates in Figure 4.5c are used for failure and error recovery.

```
send_mesg([Meth,Inst|Args],op(_)) :-                    (1)
     meth_path(Inst,Meth,Def_class),                    (2)
     meth_defn(Def_class,Meth,Method),                  (3)
     asserta(Method),                                   (4)
     Method=..([If,Goal,Body]),                         (5)
     Goal=..([Meth,Inst|Args]),                         (6)
     call(Goal),                                        (7)
     retract(Method),!.                                 (8)

 clean_up([Meth,Inst|Args],op(_)) :-                    (9)
     meth_path(Inst,Meth,Def_class),                   (10)
     meth_defn(Def_class,Meth,Method),                 (11)
     retract(Method),!.                                (12)

 send_mesg([Meth,Inst|Args],op(?<)) :-                 (13)
     inst(Inst),                                       (14)
     not(meth_path(Inst,Meth,_)),                      (15)
     write(Meth),                                      (16)
     write(' is not an appropriate message'),          (17)
     write(' for '),                                   (18)
     write(Inst),nl,                                   (19)
     clean_up([Meth,Inst|Args],op(_)).                 (20)

 send_mesg([Meth,Inst|Args],op(<?)) :-                 (21)
     not(inst(Inst)),                                  (22)
     write(Inst),                                      (23)
     write(' is a non-existent object'),               (24)
     nl,                                               (25)
     clean_up([Meth,Inst|Args],op(_)).                 (26)

 send_mesg([Meth,Inst|Args],op(<<)) :-                 (27)
     inst(Inst),                                       (28)
     not(meth_path(Inst,Meth,_)),                      (29)
     write(Meth),                                      (30)
     write(' is not an appropriate message'),          (31)
     write(' for '),                                   (32)
     write(Inst),nl,                                   (33)
     clean_up([Meth,Inst|Args],op(_)).                 (34)
```

Figure 4.5c. send_mesg Clause.

```
send_mesg([Meth,Inst|Args],op(<<)) :-          (35)
    not(inst(Inst)),                           (36)
    write(Inst),                               (37)
    write(' is a non-existent object'),        (38)
    nl,                                        (39)
    clean_up([Meth,Inst|Args],op(_)).          (40)

send_mesg([Meth,Inst|Args],op(?<?)) :-         (41)
    clean_up([Meth,Inst|Args],op(_)).          (42)
```

Figure 4.5c. (continued) send_mesg Clause.

4.3 Translation From OOP to C-Prolog

Translation of an OOP program involves a two step process. The source program is first input to the scanner (Appendix D) which outputs a token list. This token list is then processed by the parser (Appendix E). Before the parser is discussed, there is a brief review of Definite Clause Grammars.

4.3.1 Scanner

The scanner is implemented in C-Prolog and scans the current input stream. The significant part of the scanner is how the tokens are returned. The output variable for **scan/2** is a Prolog list. Each member of the list is in the form

functor(atom)

where **functor** is a tag for the token and **atom** is the actual token. A complete description of the tags is in Appendix D.

4.3.2 Definite Clause Grammar Review

A Definite Clause Grammar (DCG) is an executable

Prolog program that is a natural extension of context-free grammars []. Grammar productions in DCG notation are represented as

nonterminal --> body.

There are two major extensions to context-free grammars that are included in DCG notation: the nonterminal head may have arguments; semantic actions may be included in the body.

The nonterminal may take the same form as a normal Prolog predicate head: **functor(arguments)** or **functor**.

When specifying the body for a production, there are three different types of components: nonterminals, terminals, and semantic actions. Nonterminals in the body also have functor and optional arguments; the nonterminal should be the head of a production. Terminals are enclosed in square brackets. Semantic actions are normal Prolog clauses and are enclosed in curly brackets.

To understand how Prolog handles DCG notation, consider the following DCG production:

```
a(X,Y)-->
    b(X),d(Y),[someterminal],{semantic_action(X)}.
```

When this production is consulted by Prolog, it is

asserted as

```
a(X,Y,L1,L2):-
        b(X,L1,_1),d(Y,_1,_2),
        c(_2,someterminal,L2),
        semantic_action(X).
```

In the head of this predicate, Prolog appends two additional arguments. These arguments handle the token list to be parsed: **L1** is the input token list, **L2** is the output list consisting of the tokens not yet parsed. Each nonterminal in the body also has arguments appended for the same purpose. In this example, b/3 is a call to a nonterminal. The input token list is **L1** and **_1** is the list of tokens not consumed by b/3. The output list of **b/3** (**_1**) is used as the input list for **d/3**.

To process terminals, Prolog uses its own predicate **c/3**. This predicate takes the input token list as its first argument (**_2**), attempts to unify its second argument **someterminal** with the head of its first argument, and returns the tail of the token list in its third argument (**L2**).

Semantic actions are always normal Prolog clauses. No additional arguments are needed because they do not

process the token list; semantic actions are asserted directly and perform actions appropriate to the current context.

4.3.3 Parser

The prototype parser is implemented using the DCG notation described in the previous section. Source code for the parser is in Appendix E. Discussion of the parser is according to the OOP statement category: class definition, instance creation, or top level message transmission. In Appendices I, J, and K there are traces of the parsing for each of these types of statement. These traces are performed on segments of the circuit example.

4.3.3.1 Parsing Class Definitions

The class definition parsing entails getting the class name and its properties. In the semantic actions, the parser verifies that another class by the same name has not already been defined, and reports an error if it is proven otherwise. The semantic actions

also check for a match between the beginning and ending class names for the class definition.

The properties of a class include the declarations of the superclass, class and instance variables, and the methods. Suitable error checking is also provided for superclass and variable declarations (referencing undefined classes, variables already defined for the class, and the like). Error checking for method declarations, however, is left for the C-Prolog interpreter.

Since tokens produced by the scanner are atoms, some problems with variable instantiation for methods arose. The task was to construct the target version of the method from a list of atoms. Consider the following productions:

```
term(X) --> variable(X).
variable(Z) --> [vbl(Z)].
```

Using this form, term returns an uninstantiated variable **X** and Prolog gives the returned variable a temporary anonymous variable name. This name is an underscore followed by an incremented number, such as **_14**. The next time an uninstantiated variable occurs

at the same level, it will be _15, and so forth. This process of assigning anonymous variable names causes problems when a variable name occurs in a method more than once: every time that variable is encountered, Prolog will give it yet another name. If the constructed target method is asserted, it will not work as expected because it has different variables than the source code had. This problem demonstrated some shortcomings of C-Prolog: there is no predicate that facilitates the conversion of atoms to variables; there is no access to anonymous variables.

For lack of a better option, the method is constructed by writing the predicate to a file called **.method** as it is parsed and maintaining a symbol table for anonymous variables. When the complete method has been parsed, **.method** is consulted, thus asserting the method declaration.

Writing to **.method** is separated into two stages. Whenever a **get_ivar/3** or **get_cvar/3** predicate is needed to get a value for an expression, that predicate is written to **.method** immediately. This ensures that the value will be available for the expression using the

instance or class variable. All other parts of the method body are kept in a list of printable elements that is printed to **.method** after the entire method has been parsed.

4.3.3.2 Parsing an Instance Creation Statement

Instance creation has the expected error checking. If an instance with the same name already exists, or the class requested is not defined, the appropriate error is reported. If no error occurs, the data structures discussed in section 4.4.2 are asserted.

4.3.3.3 Parsing Top Level Message Transmissions

The parser handles top level message transmission by calling the **send_mesg/2** predicate in its semantic actions. The production for **top_lev_mes_trans/0** extracts the name of the receiver, the message operator, the message and any arguments. Using these values, the semantic actions call

send_mesg([Message,Receiver|Args],op(Operator))

thus sending the message.

4.4 Execution

A sample session with OOP, using the circuit example, is in Appendix A. The first step to using OOP is invoking the prolog interpreter from the operating system prompt. Once the Prolog prompt appears (?-), consult the file "oop"; this file is the OOP interface and can be found in Appendix B. "oop" begins by printing the OOP banner and consulting the file "load" (Appendix C). The appropriate files for the scanner (Appendix D), parser (Appendix E), and run-time predicates (Appendix F) are consulted by "load".

After the system is loaded, "oop" prompts the user for the name of the file containing the OOP source code. "oop" then scans that file and invokes the parser, passing the token list as the first argument to the parser.

In the parser, structures generated from the declarations store the class hierarchy, variable names, and method names and bodies. Structures generated from the instance creation statements store instance names, and variable and method names for each object. Finally, when a message sending operation is

encountered, a **send_mesg/2** predicate is invoked, with the variables being bound to the name of the instance, the message, and the message sending operator.

4.5 Conclusions and Discussion

The system implemented is a prototype, not a complete system. The system, as implemented, takes approximately 9.7 seconds to load and contains 30.34K of rules facts and productions. The largest portion is the parser, containing 98 productions. Using the circuit example as the only basis, the target C-Prolog version is just about four times the size of its OOP source. The deepest nesting of calls for each type of statement for the circuit example are as follows: class definition, 27 calls; instance creation statement, 12 calls; top level message transmission, 26 calls.

While working on this implementation, some issues became apparent. The first deals with the sequential stacking of methods using asserta and retract. This inhibits possible operations that allow further extension of concurrent objects.

Second, there is the familiar tradeoff between minimizing required memory and maximizing execution speed. As described, each instance creation in the main section causes the assertion of several facts that increase the speed of execution. For example, the fact "meth_path" eliminates the search for the body of the method to be executed and tells "send_mesg" exactly how to get there. The tradeoff is that there is one of these facts asserted for every method that the instance is allowed to invoke. This also occurs with the assertion of c_var_path. The issue causing the most difficulty is the lack of operators in C-Prolog for variable handling (discussed in section 4.3.3.1).

Future directions with this prototype may include adding multiple inheritance, and pseudo instance names for instances to reference themselves and any other classes in it's lineage (ancestors and descendants). Much more testing and extension of the method body and arguments are also possible.

REFERENCES

[Bob85] Bobrow, Daniel G. "If Prolog is the Answer, What is the Question? or What it Takes to Support AI Programming Paradigms", IEEE Transactions on Software Engineering SE-11, 11 (November 1985), 1401-1408.

[Bob86] Bobrow, Daniel G.; Stefik, Mark J. "Perspectives on artificial Intelligence Programming", Science 231, 4741 (28 February 1986), 951-957.

[Boo86] Booch, Grady. "Object-Oriented Development", IEEE Transactions on Software Engineering SE-12, 2 (February 1986), 211-221.

[Car84] Cardelli, Luca. "A Semantics of Multiple Inheritance", International Symposium Sophia-Antipolis, France, June 1984 Proceedings, Lecture Notes in Computer Science 173, 51-67.

[Che87] Chen, Weilin T. "Use of Definite Clause Grammars", Masters Report, Kansas State University, 1987.

[Clo84] Clocksin, W.F.; Mellish, C.S. Programming in Prolog, second edition, Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.

[Cox84] Cox, Brad J. "Message/Object Programming: An Evolutionary Change in Programming Technology", IEEE Software, January 1984, 50-61.

REFERENCES

[Cox86] Cox, Brad J. "Object Oriented Programming", Addison-Wesley, 1986.

[Cun86] Cunningham, Ward; Beck, Kent. "A Diagram for Object-Oriented Programs", Object-Oriented Programming Systems, Languages and Applications, Conference Proceedings (Portland, Oregon, September 29 - October 2, 1986), 361-367.

[Ede87] Edelson, Daniel R. "How Objective Mechanisms Facilitate the Development of Large Software Systems in Three Programming Languages", SIGPLAN Notices 22, 9 (September 1987), 54-62.

[Fuk86] Fukunaga, Koichi; Hirose, Shin-ichi. "An Experience with a Prolog-based Object-Oriented Language", Object-Oriented Programming Systems, Languages and Applications, Conference Proceedings (Portland, Oregon, September 29 - October 2, 1986), 224-231.

[Fuk87] Fukunaga, K.; Hirose, S.; van Dam, E. "VM/PROBJ User's Guide", IBM Japan, 1987. [In Japanese]

[Gen85] Genesereth, Michael R.; Ginsberg, Matthew L. "Logic Programming", Communications of the ACM 28, 9 (September 1985), 933-941.

[Gol83] Goldberg, Adele. "The Influence of an Object-Oriented Language on the Programming Environment", Proceedings of 1983 ACM Computer Science Conference and SIGCSE Symposium (Orlando, Florida, February 15-17, 1983), 35-54.

REFERENCES

[Hew79] Hewitt, Carl. "Control Structure as Patterns of Passing Messages". Artificial Intelligence: An MIT Perspective Volume II: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation. Patrick Henry Winston and Richard Henry Brown, eds. The MIT Press, Cambridge, Massachusetts, 1979, 433-465.

[Ing78] Ingalls, Daniel H. H. "The Smalltalk-76 Programming System Design and Implementation", Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages (Tuscon, Arizona, January 23-25, 1978), 9-15.

[Ing81] Ingalls, Daniel H. H. "Design Principles Behind Smalltalk", Byte August 1981, 286-298.

[Kah86] Kahn, Kenneth; Tribble, Eric Dean; Miller, Mark S.; Bobrow, Daniel G. "Objects in Concurrent Logic Programming Languages", Object-Oriented Programming Systems, Languages and Applications, Conference Proceedings (Portland, Oregon, September 29 - October 2, 1986), 242-257.

[Lee86] Lee, Newton S. "Programming with P-Shell", IEEE Expert, Summer 1986, 50-63.

[Mac85] MacLennan, Bruce J. "A Simple Software Environment Based on Objects and Relations", SIGPLAN Notices 20, 7 (July 1985), 199-207.

[Nak83] Nakashima, Hideyuki; Suzuki, Norihisa. "Data Abstraction in Prolog/KR", New Generation Computing 1 (1983), 49-62.

REFERENCES

[Per80] Pereira, F.; Warren, D. "Definite Clause Grammars for Language Analysis--A Survey of the Formalism and a Comparison with Augmented Transition Networks", Artificial Intelligence, 13, 1980, pp 231-278

[Ren82] Rentsch, Tim. "Object Oriented Programming", SIGPLAN Notices 17, 9 (September 1982), 51-57.

[Rob81] Robson, David. "Object-Oriented Software Systems", Byte August 1981, 74-86.

[Sha83] Shapiro, Ehud; Takeuchi, Akikazu. "Object Oriented Programming in Concurrent Prolog", New Generation Computing 1 (1983), 25-48.

[Sny86] Snyder, Alan. "Encapsulation and Inheritance in Object-Oriented Programming Languages", Object-Oriented Programming Systems, Languages and Applications, Conference Proceedings (Portland, Oregon, September 29-October 2, 1986), 38-45.

[Sta86] Stabler, Edward P., Jr. "Object-Oriented Programming in Prolog", AI Expert, October 1986, 46-57.

[Ste85] Stefik, Mark and Bobrow, Daniel. "Object-Oriented Programming: Themes and Variations", The AI Magazine, 12, pp 40-62.

[Stef86] Stefik, Mark J.; Bobrow, Daniel G.; Kahn, Kenneth M. "Integrating Access-Oriented Programming into a Multiparadigm Environment", IEEE Software, January 1986, 10-18.

REFERENCES

[Ster86] Sterlin, Leon; Shapiro, Ehud. The Art of Prolog, The MIT Press, Cambridge, Massachusetts, 1986.

[Zan84] Zaniolo, Carlo. "Object-Oriented Programming in Prolog", Proceedings, International Symposium on Logic Programming (Atlantic City, IEEE, 1984), 265-270.

APPENDIX A SAMPLE SESSION WITH OOP

```
C-Prolog version 1.4

| ?- [oop].

*****************************************************

                        O O P

        An Object Oriented Extension of Prolog

                (Partial Implementation)

                    by Michael Kondor
*****************************************************

Please wait, loading scanner and parser.

*****************************************************

scanner reconsulted 7104 bytes 2.01667 sec.
yes

parser reconsulted 412 bytes 0.166668 sec.
yes

class_defn reconsulted 1144 bytes 0.266667 sec.
yes
```

```
super_class_decl reconsulted 392 bytes 0.150001 sec.
yes

var_decl reconsulted 2224 bytes 0.716666 sec.
yes

meth_decl reconsulted 2220 bytes 0.683333 sec.
yes

pseudo reconsulted 348 bytes 0.150003 sec.
yes

tlmt reconsulted 616 bytes 0.150002 sec.
yes

operators reconsulted 3008 bytes 0.983334 sec.
yes

inst_create reconsulted 2168 bytes 0.616667 sec.
yes

body reconsulted 5008 bytes 1.65 sec.
yes

preds reconsulted 2576 bytes 0.750002 sec.
yes

support_preds reconsulted 3844 bytes 1.1 sec.
yes

load consulted 31064 bytes 9.76666 sec.
```

APPENDIX A SAMPLE SESSION WITH OOP

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

Please enter the name of the file containing your

OOP program (followed by a period) in one of the

following forms:

     If in the current directory, just the filename

     If in another directory, enter the entire
     path in single quotes
     (e.g. '/usrb/kondor/design/testprog')


What is the filename?   testprog.
**************************************************




out1 is 1
```

APPENDIX A SAMPLE SESSION WITH OOP

```
OOP_Source = testprog

X = [atm(class),atm(circuit),atm(has),atm(super_class),
atm(root),spsym(;),atm(instance_var),vbl(Temp),spsym(,)
,vbl(To_meth),spsym(,),vbl(To_inst),spsym(;),atm(method
s),atm(in1),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Temp
),spsym(=:),vbl(X),spsym(.),atm(connect),spsym(()),vbl(I
nst),spsym(,),vbl(Meth),spsym()),spsym(:-),vbl(To_inst)
,spsym(=:),vbl(Inst),spsym(,),vbl(To_meth),spsym(=:),vb
l(Meth),spsym(.),atm(end),atm(circuit),spsym(.),atm(cla
ss),atm(and_c),atm(has),atm(super_class),atm(circuit),s
psym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),
spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),sp
sym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.),atm(class),atm(or_c),atm(has),atm(super_class),a
tm(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl
(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X)
,spsym(\/),spsym(:),vbl(Temp),spsym()),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)
,spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),
atm(or_c),spsym(.),atm(class),atm(out_c),atm(has),atm(s
uper_class),atm(circuit),spsym(;),atm(methods),atm(in2)
,spsym(()),vbl(X),spsym()),spsym(:-),atm(nl),spsym(,),at
m(nl),spsym(,),atm(write),spsym(()),spsym(:),vbl(_Id),sp
sym()),spsym(,),atm(write),spsym(()),spsym('),qatom( is
),spsym('),spsym()),spsym(,),atm(write),spsym(()),vbl(X)
,spsym()),spsym(,),atm(nl),spsym(,),atm(nl),spsym(,),at
m(nl),spsym(.),atm(end),atm(out_c),spsym(.),atm(instanc
e),atm(andl),atm(isa),atm(and_c),spsym(.),atm(instance)
,atm(and2),atm(isa),atm(and_c),spsym(.),atm(instance),a
tm(or1),atm(isa),atm(or_c),spsym(.),atm(instance),atm(o
ut1),atm(isa),atm(out_c),spsym(.),atm(andl),spsym(<<),a
tm(connect),spsym(()),atm(or1),spsym(,),atm(in1),spsym()
),spsym(.),atm(and2),spsym(<<),atm(connect),spsym(()),at
m(or1),spsym(,),atm(in2),spsym()),spsym(.),atm(or1),sps
ym(<<),atm(connect),spsym(()),atm(out1),spsym(,),atm(in2
),spsym()),spsym(.),atm(andl),spsym(<<),atm(in1),spsym(
(),num(1),spsym()),spsym(.),atm(andl),spsym(<<),atm(in2
),spsym(()),num(0),spsym()),spsym(.),atm(and2),spsym(<<)
,atm(in1),spsym(()),num(1),spsym()),spsym(.),atm(and2),s
psym(<<),atm(in2),spsym(()),num(1),spsym()),spsym(.)]
```

APPENDIX A SAMPLE SESSION WITH OOP

```
Y = []

yes
| oop consulted 34664 bytes 20.3167 sec.

yes

| ?-

[ Prolog execution halted ]
```

APPENDIX B OOP INTERFACE SOURCE

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% File:  oop
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
?- tell(user),nl,nl,tab(10),
   write('****************************************'),
   write('************'),nl,tab(10),
   write('                              O O P'),nl, tab(10),
   write('       An Object Oriented Extension of'),
   write(' Prolog'),nl,nl,nl,tab(10),
   write('               (Partial Implementation)'),nl,
   nl,nl,tab(10),
   write('                    by Michael Kondor'),nl,
   tab(10),
   write('****************************************'),
   write('************'),nl,nl,tab(10),
   write('Please wait, loading scanner and parser.'),
   nl,nl,tab(10),
   write('****************************************'),
   write('************'),nl,
   consult(load),nl,nl,nl,tab(10),
   write('****************************************'),
   write('************'),nl,tab(10),
   write('Please enter the name of the file'),
   write(' containing your   '),nl,nl,tab(10),
   write('OOP program (followed by a period) in one'),
   write(' of the '),nl,nl,tab(10),
   write('following forms: '),nl,nl,tab(15),
   write('If in the current directory, just '),
   write('the filename'),nl,nl,tab(15),
   write('If in another directory, enter the entire '),
   nl,tab(15),write('path in single quotes'),nl,
   tab(15),write('(e.g. '),
   writeq('/usrb/kondor/design/testprog'),
   write(')'),nl,nl,
   tab(10),
   write('What is the filename?  '),

   see(user),read(OOP_Source),seen,tab(10),
   write('****************************************'),
   write('************'),nl,

   told,
   see(OOP_Source), scan(32,X),seen,
   oop(X,Y),seen.
```

APPENDIX C LOADER SOURCE

```
%
% File:  load
%
?- reconsult(scanner).
?- reconsult(parser).
?- reconsult(class_defn).
?- reconsult(super_class_decl).
?- reconsult(var_decl).
?- reconsult(meth_decl).
?- reconsult(pseudo).
?- reconsult(tlmt).
?- reconsult(operators).
?- reconsult(inst_create).
?- reconsult(body).
?- reconsult(preds).
?- reconsult(support_preds).
%
```

APPENDIX D SCANNER SOURCE

```
% "scanner"
% ----------------
% The tokens are returned in the following form:
%  functor(atom).
% The functor describes the token and the argument is
% the token in atomic form.   The types of tokens
% returned are:
%
%   string(X)     chars that appeared between a pair of
%                 " "
%   qatom(X)      chars that appeared between a pair of
%                 ' '
%   vbl(X)        token starting with a capital letter
%                 or underscore
%   atm(X)        token starting with a lowercase letter
%   num(X)        numeric token (does not handle
%                 floating point numbers)
%   spsym(X)      special symbol (token not in above
%                 categories)


scan(0,[string(String),spsym(Quote)|TokenList]) :-
                          get0(C),
                          getstring(C,S),
                          name(String,S),
                          name(Quote,[34]),
                          !,
                          scan(32,TokenList).
scan(1,[qatom(QAtom),spsym(Quote)|TokenList]) :-
                          get0(C),getqatom(C,A),
                          name(QAtom,A),
                          name(Quote,[39]),
                          !,
                          scan(32,TokenList).
scan(26,[]) :- !.     % "^Z" ==> end of input
scan(32,TokenList) :-
             get0(Ch), !,
             scan(Ch,TokenList).
scan(10,TokenList) :-
             get0(Ch), !,
             scan(Ch,TokenList).
scan(Ch,[spsym(Wd)|TokenList]) :-
            getspecialsymbol(Ch,Wd,LastCh), !,
            scan(LastCh,TokenList).
```

APPENDIX D SCANNER SOURCE

```
% get a word or a number
scan(Ch,[Wd|TokenList]) :-
            getword(Ch,Wd1,LastCh),
            (isvar(Wd1,Wd);isatom(Wd1,Wd);
            isnum(Wd1,Wd)),
            !,
            scan(LastCh,TokenList).
scan(Ch,[Wd|TokenList]) :-
            name(Wd,[Ch]), nl,
            write(Wd),
            write('        <== ** invalid token **'),
            nl, nl,
            !, fail.


getword(Ch,[Ch|Wd1],LastCh) :-
                    letter(Ch), !,
                    get0(Ch1),
                    isname(Ch1,Wd1,LastCh).
getword(Ch,[Ch|Wd1],LastCh) :-
                    digit(Ch), !,
                    get0(Ch1),
                    isnumber(Ch1,Wd1,LastCh).

isname(Ch,[Ch|Wd],LastCh) :-
       letter(Ch), !, get0(Ch1), isname(Ch1,Wd,LastCh).
isname(Ch,[Ch|Wd],LastCh) :-
       digit(Ch), !, get0(Ch1), isname(Ch1,Wd,LastCh).
isname(Ch,[],LastCh) :- LastCh=Ch.

isnumber(Ch,[Ch|Wd],LastCh) :-
      digit(Ch), !, get0(Ch1), isnumber(Ch1,Wd,LastCh).
isnumber(Ch,[],LastCh) :- LastCh=Ch.

letter(Ch) :- upperletter(Ch);
              lowerletter(Ch).
lowerletter(Ch) :- Ch>96, Ch<123.     % lowercase letter
upperletter(Ch) :- Ch>64, Ch<91;      % capital
                   Ch = 95.           % underscore
digit(Ch) :- Ch>47, Ch<58.
isvar([H|T],vbl(X)) :- upperletter(H), name(X,[H|T]).
isatom([H|T],atm(X)) :- lowerletter(H), name(X,[H|T]).
isnum([H|T],num(X)) :- digit(H), name(X,[H|T]).
```

APPENDIX D SCANNER SOURCE

```
/* Operaters */
getspecialsymbol(40,'(',C) :- get0(C).
getspecialsymbol(41,')',C) :- get0(C).
getspecialsymbol(91,'[',C) :- get0(C).
getspecialsymbol(93,']',C) :- get0(C).
getspecialsymbol(46,'.',C) :- get0(C).
getspecialsymbol(124,'|',C) :- get0(C).
getspecialsymbol(42,'*',C) :- get0(C).
getspecialsymbol(94,'^',C) :- get0(C).

% token is one of these:    :    :-     ::
%                           (: is 58, - is 45)
getspecialsymbol(58,X,C)    :- get0(C1), !,
                               (C1 \== 45 ->
                               (C1 \== 58 ->
                               (X = ':',C = C1);
                                  (X = '::',get0(C)));
                               (X = (':-'),get0(C))).

% token is one of these:    <    <<     <?
%                           (< is 60, ? is 63)
getspecialsymbol(60,X,C)    :- get0(C1), !,
                               (C1 \== 63 ->
                               (C1 \== 60 ->
                               (X = '<',C = C1);
                                  (X = '<<',get0(C)));
                               (X = ('<?'),get0(C))).

% token is one of these:    >    >=     >>
%                           (> is 62, = is 61)
getspecialsymbol(62,X,C)    :- get0(C1), !,
                               (C1 \== 61 ->
                               (C1 \== 62 ->
                               (X = '>',C = C1);
                                  (X = '>>',get0(C)));
                               (X = ('>='),get0(C))).
```

APPENDIX D SCANNER SOURCE

```
% token is one of these:   =  =:  =::  =:=  =..  ==  =<
%                  (= is 61, : is 58, . is 46, < is 60)
getspecialsymbol(61,X,C)   :- get0(C1), !,
                              (C1 \== 61 ->
                              (C1 \== 58 ->
                              (C1 \== 46 ->
                              (C1 \== 60 ->
                              (X = '=',C = C1);
                                 (X = '=<',get0(C)));
                              (get0(C2),!,(C2==46 ->
                                 (X='=..',get0(C)))));
                              (get0(C2),!,
                                 (C2 \== 58 ->
                                 (C2 \== 61 ->
                                 (X = '=:',C = C2);
                                  (X = '=:=',get0(C)));
                                 (X = '=::'))));
                              (X = '==',get0(C))).

% token is one of these:   ?-  ?<  ?<?
%                         (? is 63, - is 45, < is 60)
getspecialsymbol(63,X,C)   :- get0(C1), !,
                              (C1 \== 45 ->
                              (C1 == 60 ->
                              (get0(C2),!,
                                 (C2 \== 63 ->
                                 (X = '?<',C = C2);
                                (X = '?<?',get0(C)))));
                              (X = (?-),get0(C))).

getspecialsymbol(43,'+',C) :- get0(C).


% token is one of these:   -  -->  ->
%                          (- is 45, > is 62)
getspecialsymbol(45,X,C)   :- get0(C1), !,
                              (C1 \== 62 ->
                              (C1 \== 45 ->
                              (X = '-',C = C1);
                              (get0(C2),!,
                                 (C2 == 62 ->
                                (X = '-->',get0(C)))));
                              (X = '->',get0(C))).
```

APPENDIX D SCANNER SOURCE

```
% token is one of these:    /    //     /\
%                           (/ is 47, \ is 92)
getspecialsymbol(47,X,C)    :- get0(C1), !,
                               (C1 \== 92 ->
                               (C1 \== 47 ->
                               (X = '/',C = C1);
                               (X = '//',get0(C)));
                               (X = ('/\'),get0(C))).


% token is one of these:    \+   \==    \/
%                    (\ is 92, + is 43, / is 47, = is 61)
getspecialsymbol(92,X,C)    :- get0(C1), !,
                               (C1 \== 43 ->
                               (C1 \== 47 ->
                               (C1 == 61 ->
                               (get0(C2),!,
                                  (C2 == 61 ->
                                 (X = '\==',get0(C)))));
                               (X = '\/',get0(C)));
                               (X = (\+),get0(C))).

% token is one of these:    @<  @>  @=<  @>=
%                     (@ is 64, < is 60, > is 62, = is 61)
getspecialsymbol(64,X,C)    :- get0(C1), !,
                               (C1 \== 60 ->
                               (C1 \== 62 ->
                               (C1 == 61 ->
                               (get0(C2),!,
                                   (C2 == 60 ->
                                 (X = '@=<',get0(C)))));
                               (get0(C2),!,
                                   (C2 == 61 ->
                                   (X = '@>=',get0(C));
                                   (X = '@>',C = C2))));
                               (X = '@<',get0(C))).

getspecialsymbol(59,';',C) :- get0(C).
getspecialsymbol(44,',',C) :- get0(C).
getspecialsymbol(34,X,0) :- name(X,[34]).
getspecialsymbol(39,X,1) :- name(X,[39]).
```

APPENDIX D SCANNER SOURCE

```
getstring(34,[]).
getstring(X,[X|T]) :- get0(A),getstring(A,T).

getqatom(39,[]).
getqatom(X,[X|T]) :- get0(A),getqatom(A,T).
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ------------------------------------------------------
%
% File:  parser
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%

oop -->  oop_stmt, oop.
oop -->  [].

% OOP STATEMENT

oop_stmt --> class_defn.
oop_stmt --> inst_create.
oop_stmt --> top_lev_mes_trans.

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ----------------------------------------------------------------
% File:  class_defn
% ----------------------------------------------------------------

class_defn --> [atm(class)],
               classname(_beg_name),
               [atm(has)], properties(_beg_name),
               [atm(_end_name)],
               {match(_beg_name,_end_name)},
               [spsym('.')].

% the terminal [end] was moved down as part of the
%  method definitions.


classname(Name) --> [atm(Name)],
                    {(not(classtree(_,Name))) -> true;
                     (nl,nl,
                     write('Class already defined:  '),
                     write(Name),nl,nl)}.


properties(_class_name) --> supers(_class_name),
                            class_vars(_class_name),
                            inst_vars(_class_name),
                            methods(_class_name).
% ----------------------------------------------------------------
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ---------------------------------------------------------
% File:  super_class_decl
% ---------------------------------------------------------
% SUPERCLASS DECLARATION

supers(_class_name) --> [atm(super_class)],
                        parent(_class_name),
                        [spsym(';')].

parent(_class_name) --> [atm(Parent)],

            {(classtree(_,Parent),
            assert((classtree(Parent,_class_name)))) ->
            true;

            (write('Undefined superclass:  '),
            write(Parent),nl)}.
% ---------------------------------------------------------
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% File:  var_decl
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% VARIABLE DECLARATIONS

class_vars(_class_name) --> [atm(class_var)],
                            cvar_elmt(_class_name),
                            cvarlist(_class_name);
                            {true}.


cvarlist(_class_name) --> [spsym(',')],
                          cvar_elmt(_class_name),
                          cvarlist(_class_name);
                          [spsym(';')].


cvar_elmt(_class_name) --> var_name(Name),
                           rest_elmt(_class_name,Name).

rest_elmt(_class,_var) --> [spsym('=')], init_val(Val),
                {not(c_var_val(_class,_var,_)),
                assert(c_var_val(_class,_var,Val))};

                {not(c_var_val(_class,_var,_)),
                assert(c_var_val(_class,_var,Undef))};

                {c_var_val(_class,_var,_),
                write('Class variable already '),
                write('defined:  '),write('Class = '),
                write(_class_name),
                write(', Variable = '),
                write(Name),nl}.


var_name(Name) --> [vbl(Name)].

init_val(Val) --> [num(Val)]; [atm(Val)].

inst_vars(_class_name) --> [atm(instance_var)],
                           ivar_elmt(_class_name),
                           ivar_list(_class_name),
                           [spsym(';')]; {true}.
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
ivar_list(_class_name) --> [spsym(',')],
                           ivar_elmt(_class_name),
                           ivar_list(_class_name);
                           {true}.

ivar_elmt(_class_name) --> var_name(Name),
          {(not(i_var_decl(_class_name,Name))) ->
          (assert((i_var_decl(_class_name,Name))));

          (write('Instance variable already defined:'),
          write('  Class = '),write(_class_name),
          write(', Variable = '),write(Name),nl)}.
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% File:  meth_decl
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% All methods that are unit clauses must be converted
% to nonunit clauses by using 'true' as the body of the
% clause.  This is necessary for the run-time predicate
% 'send_mesg' to work correctly.
%

methods(_class_name) --> [atm(methods)],
                         methodlist(_class_name).

methodlist(_class_name) --> method_defn(_class_name),
                            rest_mlist(_class_name).

rest_mlist(_class_name) --> [atm(end)];
                            method_defn(_class_name),
                            rest_mlist(_class_name).


method_defn(_class_name) --> {told,tell('.method'),
                              write('meth_defn('),
                              write(_class_name),
                              write(',')},
                              meth_head,
                              meth_tail,
                              {tell(user)}.


meth_head --> [atm(_functor)],
              {write(_functor),
              write(',('),write(_functor),
              write('(_Instance')},
              args(_arglist),
              printargs(_arglist),
              {write(')')}.


printargs([]) --> {true}.
printargs(_arglist) --> {write(','),
                         writestr(_arglist)}.
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
args(_arglist) -->  [spsym('(')],
                    term(Arg1),
                    rest_args(Tail),
                    [spsym(')')],
                    {append(Arg1,Tail,_arglist)}.


args([]) --> {true}.

rest_args(List) --> [spsym(',')], term(H),
                    rest_args(T),
                    {append([','],H,Temp1),
                     append(Temp1,T,List)}.

rest_args([]) --> {true}.


meth_tail --> [spsym((:-))], {write(' :- ')},
              body,
              {consult('.method')};

              [spsym('.')], {write(':-true.'),
              consult('.method')}.
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% File:  body
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
body --> subgoallist, [spsym('.')],
         {write(')).'),nl}.

subgoallist --> subgoal, moresubgoals.

moresubgoals --> [spsym(',')], {write(',')},
                 subgoallist; {true}.

subgoal --> exp(Str), {writestr(Str)}.

exp(S) --> unop(Op), term(T), {X=..[Op,T],fail};
           message(S);
           term(S1),restexp(S1,S);
           [spsym('(')], exp(S1), [spsym(')')],
           {append(['('],S1,Temp),
            append(Temp,[')'],S)}.

restexp(S_in,S_out) --> binop(Op),term(S2),
         {append(S_in,[' ',Op,' '],Temp),
         append(Temp,S2,S_out)};

         spvarasgn(=:),term(T2),
         {name(Quote,[39]),
         append(['set_ivar(_Instance,'],[Quote],Temp1),
         append(Temp1,S_in,Temp2),
         append(Temp2,[Quote],Temp3),
         append(Temp3,[','],Temp4),
         append(Temp4,T2,Temp5),
         append(Temp5,[')'],S_out)};

         spvarasgn(=::),term(T2),
         {name(Quote,[39]),
         append(['set_cvar(_Instance,'],[Quote],Temp1),
         append(Temp1,S_in,Temp2),
         append(Temp2,[Quote],Temp3),
         append(Temp3,[','],Temp4),
         append(Temp4,T2,Temp5),
         append(Temp5,[')'],S_out)}.

restexp(S_in,S_in) --> {true}.
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
term(Str) --> pred(Str); const(Str); variable(Str);
              spvbl(Str);

            [spsym('(')], exp(S1), [spsym(')')],
            {append(['('],S1,Temp),
             append(Temp,[')'],Str)};

            [spsym('(')], term(S1), [spsym(')')],
            {append(['('],S1,Temp),
             append(Temp,[')'],Str)}.

pred(X) --> [atm(_functor)], args(_arglist),
              {(_arglist=[])->(X=[_functor]);
                        (append([_functor],['('],Temp1),
                        append(Temp1,_arglist,Temp2),
                        append(Temp2,[')'],X))}.

const([X]) --> [num(X)];
               [atm(X)].
const(['"',X,'"']) --> [spsym('"')],[string(X)],
                       [spsym('"')].
const([Quote,X,Quote]) --> [spsym(Quote)],
                           [qatom(X)],
                           [spsym(Quote)],
                           {name(Quote,[39])}.

variable([Z]) --> [vbl(Z)].

spvbl([Anon]) --> spvaraccess(:),
             {write('get_ivar(_Instance,')},
             variable([Z]),
             {lookup(vbl(Z),Anon),
             writeq(Z),write(','),
             write(Anon),write('),')};

             spvaraccess(::),
             {write('get_cvar(_Instance,')},
             variable([Z]),
             {lookup(vbl(Z),Anon),
             writeq(Z),write(','),
             write(Anon),write('),')}.
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
message(Str) --> receiver(Receiver), mesgop(Op),
                 mesg(Meth), args(Arglist),
                 {append(['send_mesg(['],Meth,Temp1),
                  append(Temp1,[','],Temp2),
                  append(Temp2,Receiver,Temp3),
                  append(Temp3,[','],Temp4),
                  append(Temp4,Arglist,Temp5),
                  append(Temp5,['],',op(Op),')'],Str)}.

receiver(Receiver) --> [atm(_functor)],
                       {Receiver=[_functor]};
                       variable(Receiver);
                       spvbl(Receiver).

mesg(Meth) --> [atm(_meth)],{Meth=[_meth]};
               variable(Meth);
               spvbl(Meth).
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% File:  operators
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
binop((:-)) --> [spsym((:-))].
binop(-->) --> [spsym(-->)].
binop(->) --> [spsym(->)].
binop(=) --> [spsym(=)].
binop(=..) --> [spsym(=..)].
binop(==) --> [spsym(==)].
binop(\==) --> [spsym(\==)].
binop(@<) --> [spsym(@<)].
binop(@>) --> [spsym(@>)].
binop(@=<) --> [spsym(@=<)].
binop(@>=) --> [spsym(@>=)].
binop(=:=) --> [spsym(=:=)].
binop(=\=) --> [spsym(=\=)].
binop(<) --> [spsym(<)].
binop(>) --> [spsym(>)].
binop(=<) --> [spsym(=<)].
binop(>=) --> [spsym(>=)].
binop(+) --> [spsym(+)].
binop(-) --> [spsym(-)].
binop(/\) --> [spsym(/\)].
binop(\/) --> [spsym(\/)].
binop(*) --> [spsym(*)].
binop(/) --> [spsym(/)].
binop(//) --> [spsym(//)].
binop(>>) --> [spsym(>>)].
binop(^) --> [spsym(^)].
binop(mod) --> [atm(mod)].
binop(is) --> [atm(is)].

unop((:-)) --> [spsym((:-))].
unop((?-)) --> [spsym((?-))].
unop(not) --> [atm(not)].
unop(\+) --> [spsym(\+)].
unop(spy) --> [atm(spy)].
unop(nospy) --> [atm(nospy)].
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
mesgop(<<) --> [spsym(<<)].
mesgop(<?) --> [spsym(<?)].
mesgop(?<) --> [spsym(?<)].
mesgop(?<?) --> [spsym(?<?)].

spvarasgn(=:) --> [spsym(=:)].
spvarasgn(=::) --> [spsym(=::)].

spvaraccess(:) --> [spsym(:)].
spvaraccess(::) --> [spsym(::)].
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% File:  pseudo (stub--not implemented)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
pseudo_inst_name(Name) --> [atm(self)], {Name = self};
                    [atm(super)], {Name = super};
                    [atm(here)], {Name = here};
                    [atm(above)], {Name = above}.
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ------------------------------------------------------
%
% File: inst_create
% ------------------------------------------------------
%
inst_create --> [atm(instance)],
                inst_name(Instance,Err1), [atm(isa)],
                class(_class_name,Err2), [spsym('.')],
                {Err1 = 0,Err2 = 0,
                 assert((inst(Instance,_class_name))),
                 get_c_vars(Instance,_class_name),
                 get_i_vars(Instance,_class_name),
                 get_meth_paths(Instance,_class_name)}.


class(_class_name,Err) --> [atm(_class_name)],
                           {(not(classtree(_,_class_name)),
                            nl,write('Class not defined '),
                            write('in instance '),
                            write('creation: '),
                            write(_class_name),nl,
                            Err = 1);
                            Err = 0}.


inst_name(Instance,Err) --> [atm(Instance)],
                            {(inst(Instance,_),
                            write('Instance already defined:  '),
                            write(Instance),nl,Err=1);Err=0}.


get_c_vars(Instance,_class_name) :-
        c_var_val(_class_name,Var,_),
        not(c_var_path(Instance,Var,_class_name)),
        assert((c_var_path(Instance,Var,_class_name))),
        get_c_vars(Instance,_class_name);

        ((classtree(_super,_class_name),
        not(_class_name=root))->
        get_c_vars(Instance,_super); true).
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
get_i_vars(Instance,_class_name) :-
    i_var_decl(_class_name,Var),
    not(i_var_val(Instance,Var,_)),
    assert((i_var_val(Instance,Var,_UnInstant))),
    get_i_vars(Instance,_class_name);

    (not(i_var_val(Instance,'_Id',_)) ->
    assert((i_var_val(Instance,'_Id',Instance)));
    true),
    (not(i_var_val(Instance,'_Class',_)) ->
    assert((i_var_val(Instance,'_Class',_class_name)));
    true),
    ((classtree(_super,_class_name),
    not(_class_name=root))->
    get_i_vars(Instance,_super); true).


get_meth_paths(Instance,_class_name) :-
        meth_defn(_class_name,Meth,_),
        not(meth_path(Instance,Meth,_class_name)),
        assert((meth_path(Instance,Meth,_class_name))),
        get_meth_paths(Instance,_class_name);

        ((classtree(_super,_class_name),
        not(_class_name=root))->
        get_meth_paths(Instance,_super); true).
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% File:  preds
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
%        PREDICATES USED BY THE ABOVE GRAMMAR
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% ************************
match(X,X) :-  !.
match(X,Y) :-  retract((classtree(_,X))), !,
               cvar_retract(X).
cvar_retract(X) :-  retract((c_var_val(X,_,_))),
                    cvar_retract(X),!,ivar_retract(X).
cvar_retract(X).
ivar_retract(X) :-  retract((i_var_decl(X,_))),
                    ivar_retract(X), !,
                    methdef_retract(X).
ivar_retract(X).
methdef_retract(X) :-  retract((meth_defn(X,_,_))),
                       methdef_retract(X).
methdef_retract(X) :- nl,nl,
          write('Class names at beginning and end'),
          nl,
          write('of class definition did not match.'),
          nl,
          write('Class definition was disregarded').
% ************************
construct(_functor,_arg_list,_body,_meth) :-
              Head =.. [_functor,_Instance|_arg_list],
              _meth =.. [(:-),Head,_body].
% ************************
% makeatom takes a list of atoms and concatenates it
% into      one      atom      the      call      is:
% makeatom(_list_of_atoms,_one_long_atom)

makeatom(List,Atom) :-  expand_list(List,Longlist),
                        name(Atom,Longlist).

expand_list([],[]).
expand_list([H|T],List) :- name(H,H2),
                           expand_list(T,T2),
                           append(H2,T2,List).
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ************************
% compact takes a list of lists and converts it into a
% list of atoms

compact([H|T],L) :- compact(H,L1),compact(T,L2),
                    append(L1,L2,L).
compact([],[]).
compact(X,[X]).
% ************************
append([],L,L).
append([X|L1],L2,[X|L3]) :- append(L1,L2,L3).
% ************************
make_arglist([],[]).
make_arglist([H|[]],[H|[]]).
make_arglist([H|T],[H|[',',|X]]) :- make_arglist(T,X).
% ************************
writestr([]).
writestr([H|T]) :- write(H),writestr(T).
% ************************
lookup(Var_name,Num_var) :- symtab(Var_name,Num_var);

     anon(Num_var), assert((symtab(Var_name,Num_var))).

anon(Num_var) :- anonvar(X),name('_',Underscore),
                 Z is X+1, retract(anonvar(X)),
                 assert(anonvar(Z)),
                 append(Underscore,[Z],N),
                 name(Num_var,N).

anon('_0') :- name(0,[X]),assert((anonvar(X))).

% ************************
strip_args([],[]).
strip_args([Arg|[]],[Arg|[]]).
strip_args([Arg,',',|Tail],[Arg|X]) :-
                                    strip_args(Tail,X).
```

APPENDIX E PARSER SOURCE (DEFINITE CLAUSE GRAMMAR)

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% File:  tlmt
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%

top_lev_mes_trans --> [atm(Receiver)], mesgop(Op),
          mesg(Meth,Arglist),[spsym('.')],
          {strip_args(Arglist,No_commas),
          send_mesg([Meth,Receiver¦No_commas],op(Op))}.

mesg(Meth,Arglist) --> [atm(Meth)], args(Arglist).
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

APPENDIX F RUN-TIME SUPPORT PREDICATES

```
% ------------------------------------------------------
%
% File:  support_preds
% ------------------------------------------------------
%

% ROOT CLASS DEFINITIONS (not fully implemented)
classtree(root,root).
i_var_decl(root,'_Class').
i_var_decl(root,'_Id').
meth_defn(root,who_am_I,
     (who_am_I(_Instance,Class,Id) :-
            i_var_val(_Instance,'_Class',Class),
            i_var_val(_Instance,'_Id',Id))).
meth_defn(root,who_are_you,
     (who_are_you(_Instance,Class,Id) :-
                                    inst(Id,Class))).
meth_defn(root,broadcast,
    (broadcast(_Instance,[_Class|_Class_list],_Message)
     :- true)).
meth_defn(root,erase_self,
     (erase_self(_Instance) :- true)).
```

APPENDIX F RUN-TIME SUPPORT PREDICATES

```
send_mesg([Meth,Inst|Args],op(_)) :-
        meth_path(Inst,Meth,Def_class),
        meth_defn(Def_class,Meth,Method),
        asserta(Method),
        Method=..([If,Goal,Body]),
        Goal=..([Meth,Inst|Args]),
        call(Goal),
        retract(Method),!.

send_mesg([Meth,Inst|Args],op(?<?)) :-
        cleanup([Meth,Inst|Args],op(_)).

clean_up([Meth,Inst|Args],op(_)) :-
        meth_path(Inst,Meth,Def_class),
        meth_defn(Def_class,Meth,Method),
        retract(Method),!.

send_mesg([Meth,Inst|Args],op(?<)) :-
        inst(Inst),
        not(meth_path(Inst,Meth,_)),
        write(Meth),
        write(' is not an appropriate message'),
        write(' for '),
        write(Inst),nl,
        cleanup([Meth,Inst|Args],op(_)).

send_mesg([Meth,Inst|Args],op(<?)) :-
        not(inst(Inst)),
        write(Inst),
        write(' is a non-existent object'),
        nl,
        cleanup([Meth,Inst|Args],op(_)).


send_mesg([Meth,Inst|Args],op(<<)) :-
        inst(Inst),
        not(meth_path(Inst,Meth,_)),
        write(Meth),
        write(' is not an appropriate message'),
        write(' for '),
        write(Inst),nl,
        cleanup([Meth,Inst|Args],op(_)).
```

APPENDIX F RUN-TIME SUPPORT PREDICATES

```
send_mesg([Meth,Inst|Args],op(<<)) :-
        not(inst(Inst)),
        write(Inst),
        write(' is a non-existent object'),
        nl,
        cleanup([Meth,Inst|Args],op(_)).
```

APPENDIX F RUN-TIME SUPPORT PREDICATES

```
/* ============  instance var operations  =========  */

set_ivar(Inst,Name,Val) :-
        /* if it is a legal variable retract the old */
        /* value, assert the new value */
        retract(i_var_val(Inst,Name,_)),
        assert(i_var_val(Inst,Name,Val)),!.

set_ivar(Inst,Name,Val) :-
     /* write appropriate message to the screen */
     write(Name),
     write(' is not a legal instance variable name'),
     write(' in '),write(Inst),nl,!.


get_ivar(Inst,Name,Val) :-
        /* if it exists, instantiate Val */
        i_var_val(Inst,Name,Val), nonvar(Val),!.

get_ivar(Inst,Name,Val) :-
     /* if the variable is not already instantiated, */
     /* write an appropriate message to the screen */
     i_var_val(Inst,Name,Val), var(Val), !,
     write(Name),write(' is not yet instantiated in '),
     write(Inst),nl,!,fail.

get_ivar(Inst,Name,Val) :-
     /* write appropriate message to the screen */
     write(Name),
     write(' is not an instance variable for '),
     write(Inst),nl,!,fail.
```

APPENDIX F RUN-TIME SUPPORT PREDICATES

```
/* ==============  class var operations  ==========  */


set_cvar(Inst,Name,Val) :-
        /* if it is a legal variable retract the old */
        /* value, assert the new value */
        c_var_path(Inst,Name,Owner),
        retract(c_var_val(Owner,Name,_)),
        assert(c_var_val(Owner,Name,Val)),!.

set_cvar(Inst,Name,Val) :-
        /* write appropriate error message to screen */
        write(Name),
        write(' is not a legal class variable name'),
        write(' in '),write(Inst),nl,!.


get_cvar(Inst,Name,Val) :-
        /* if it exists, instantiate Val */
        c_var_path(Inst,Name,Owner),
        c_var_val(Owner,Name,Val), nonvar(Val),!.

get_cvar(Inst,Name,Val) :-
        /* if variable is not already instantiated, */
        /* write appropriate message to the screen */
        c_var_path(Inst,Name,Owner),
        c_var_val(Owner,Name,Val), var(Val), !,
        write(Name),
        write(' is not yet instantiated in '),
        write(Inst),nl,!,fail.

get_cvar(Inst,Name,Val) :-
    /* variable is not a class var for this instance */
    write(Name),write(' is not a class variable for '),
    write(Inst),nl,!,fail.
```

APPENDIX G C-PROLOG TARGET FOR CIRCUIT EXAMPLE

Circuit Example in C-Prolog: Declarations.

```
classtree(root,circuit).
i_var_decl(circuit,'Temp').
i_var_decl(circuit,'To_meth').
i_var_decl(circuit,'To_inst').
meth_defn(circuit,in1,(in1(_Instance,X):-
     set_ivar(_Instance,'Temp',X))).
meth_defn(circuit,connect,
     (connect(_Instance,Inst,Meth):-
          set_ivar(_Instance,'To_inst',Inst),
          set_ivar(_Instance,'To_meth',Meth))).


classtree(circuit,and_c).
meth_defn(and_c,in2,(in2(_Instance,X) :-
     get_ivar(_Instance,'Temp',_1),
     Out is (X /\ _1),
     get_ivar(_Instance,'To_inst',_2),
     get_ivar(_Instance,'To_meth',_3),
     send_mesg([_3,_2,Out],op(<<)))).


classtree(circuit,or_c).
meth_defn(or_c,in2,(in2(_Instance,X) :-
     get_ivar(_Instance,'Temp',_1),
     Out is (X \/ _1),
     get_ivar(_Instance,'To_inst',_2),
     get_ivar(_Instance,'To_meth',_3),
     send_mesg([_3,_2,Out],op(<<)))).


classtree(circuit,out_c).
meth_defn(out_c,in2,(in2(_Instance,X) :-
     get_ivar(_Instance,'_Id',_1),
     write(_1),
     write(' is '),
     write(X),nl)).
```

APPENDIX G C-PROLOG TARGET FOR CIRCUIT EXAMPLE

Circuit Example in C-Prolog: Object Instantiation.

```
inst(and1,and_c).
i_var_val(and1,'Temp',_UnInstant).
i_var_val(and1,'To_meth',_UnInstant).
i_var_val(and1,'To_inst',_UnInstant).
i_var_val(and1,'_Class',and_c).
i_var_val(and1,'_Id',and1).
meth_path(and1,in2,and_c).
meth_path(and1,in1,circuit).
meth_path(and1,connect,circuit).
meth_path(and1,who_am_I,root).
meth_path(and1,who_are_you,root).
meth_path(and1,broadcast,root).
meth_path(and1,erase_self,root).

inst(and2,and_c).
i_var_val(and2,'Temp',_UnInstant).
i_var_val(and2,'To_meth',_UnInstant).
i_var_val(and2,'To_inst',_UnInstant).
i_var_val(and2,'_Class',and_c).
i_var_val(and2,'_Id',and2).
meth_path(and2,in2,and_c).
meth_path(and2,in1,circuit).
meth_path(and2,connect,circuit).
meth_path(and2,who_am_I,root).
meth_path(and2,who_are_you,root).
meth_path(and2,broadcast,root).
meth_path(and2,erase_self,root).

inst(or1,or_c).
i_var_val(or1,'Temp',_UnInstant).
i_var_val(or1,'To_meth',_UnInstant).
i_var_val(or1,'To_inst',_UnInstant).
i_var_val(or1,'_Class',or_c).
i_var_val(or1,'_Id',or1).
meth_path(or1,in2,or_c).
meth_path(or1,in1,circuit).
meth_path(or1,connect,circuit).
meth_path(or1,who_am_I,root).
meth_path(or1,who_are_you,root).
meth_path(or1,broadcast,root).
meth_path(or1,erase_self,root).
```

APPENDIX G C-PROLOG TARGET FOR CIRCUIT EXAMPLE

Circuit Example in C-Prolog: Object Instantiation.

(Continued)

```
inst(out1,out_c).
i_var_val(out1,'Temp',_UnInstant).
i_var_val(out1,'To_meth',_UnInstant).
i_var_val(out1,'To_inst',_UnInstant).
i_var_val(out1,'_Class',out_c).
i_var_val(out1,'_Id',out1).
meth_path(out1,in2,out_c).
meth_path(out1,in1,circuit).

meth_path(out1,connect,circuit).
meth_path(out1,who_am_I,root).
meth_path(out1,who_are_you,root).
meth_path(out1,broadcast,root).
meth_path(out1,erase_self,root).
```

APPENDIX G C-PROLOG TARGET FOR CIRCUIT EXAMPLE

Circuit Example in C-Prolog: Message Sending.

```
?- send_mesg([connect,and1,or1,in1],op(<<)).
?- send_mesg([connect,and2,or1,in2],op(<<)).
?- send_mesg([connect,or1,out1,in2],op(<<)).

?- send_mesg([in1,and1,1],op(<<)).
?- send_mesg([in2,and1,0],op(<<)).
?- send_mesg([in1,and2,1],op(<<)).
?- send_mesg([in2,and2,1],op(<<)).
```

APPENDIX H MESSAGE TRACE OF CIRCUIT EXAMPLE

```
=========================================================
*  (1) 1 Call: send_mesg([connect,and1,or1,in1],op(<<))
   (2) 2 Call: meth_path(and1,connect,_66122)
   (2) 2 Exit: meth_path(and1,connect,circuit)
   (3) 2 Call: meth_defn(circuit,connect,_66123)
   (3) 2 Exit: meth_defn(circuit,connect,
          (connect(_48,_49,_50):-
            set_ivar(_48,To_inst,_49),
            set_ivar(_48,To_meth,_50)))
   (4) 2 Call: asserta((connect(_48,_49,_50):-
          set_ivar(_48,To_inst,_49),
          set_ivar(_48,To_meth,_50)))
   (4) 2 Exit: asserta((connect(_48,_49,_50):-
          set_ivar(_48,To_inst,_49),
          set_ivar(_48,To_meth,_50)))
   (5) 2 Call: (connect(_48,_49,_50):-
          set_ivar(_48,To_inst,_49),
          set_ivar(_48,To_meth,_50))=..[_43,_44,_45]
   (5) 2 Exit: (connect(_48,_49,_50):-
          set_ivar(_48,To_inst,_49),
          set_ivar(_48,To_meth,_50))=..
          [:-,connect(_48,_49,_50),
           (set_ivar(_48,To_inst,_49),
            set_ivar(_48,To_meth,_50))]
   (6) 2 Call: connect(_48,_49,_50)=..
          [connect,and1,or1,in1]
   (6) 2 Exit: connect(and1,or1,in1)=..
          [connect,and1,or1,in1]
   (7) 2 Call: connect(and1,or1,in1)
   (8) 3 Call: set_ivar(and1,To_inst,or1)
   (9) 4 Call: retract(i_var_val(and1,To_inst,_73))
   (9) 4 Exit: retract(i_var_val(and1,To_inst,_73))
  (10) 4 Call: assert(i_var_val(and1,To_inst,or1))
  (10) 4 Exit: assert(i_var_val(and1,To_inst,or1))
   (8) 3 Exit: set_ivar(and1,To_inst,or1)
  (11) 3 Call: set_ivar(and1,To_meth,in1)
  (12) 4 Call: retract(i_var_val(and1,To_meth,_93))
  (12) 4 Exit: retract(i_var_val(and1,To_meth,_93))
  (13) 4 Call: assert(i_var_val(and1,To_meth,in1))
  (13) 4 Exit: assert(i_var_val(and1,To_meth,in1))
  (11) 3 Exit: set_ivar(and1,To_meth,in1)
   (7) 2 Exit: connect(and1,or1,in1)
```

APPENDIX H MESSAGE TRACE OF CIRCUIT EXAMPLE

```
   (14) 2 Call: retract((connect(and1,or1,in1):-
            set_ivar(and1,To_inst,or1),
            set_ivar(and1,To_meth,in1)))
   (14) 2 Exit: retract((connect(and1,or1,in1):-
            set_ivar(and1,To_inst,or1),
            set_ivar(and1,To_meth,in1)))
*  (1) 1 Exit: send_mesg([connect,and1,or1,in1],op(<<))

============================================================
```

Assume that all assertions in the main section have been completed and tracing begins with the first **send_mesg** query in Appendix G. Trace level numbers are given in parentheses.

The method **connect** is invoked in object **and1** (1). The method is found to exist (2). The method code is instantiated (3) from the class **circuit**, asserted (4), broken into its component goal (**connect**) and code parts (5). Before level (6), **Goal** is **connect(_44,_45,_46)**, **Meth** is **connect**, **Inst** is **and1**, and **Args** is **[in1,or1]**. Then the variables in **Goal** are instantiated so that **Goal** is **connect(and1,or1,in1)** (6), and **send_mesg** calls **(connect(and1,or1,in1))** (7). The subgoals of **connect** are invoked (8-13). Finally, the method code is retracted (14) and the **send_mesg** is complete (1).

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3629) 2 Call: oop([atm(class),atm(and_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_157)

(3630) 3 Call: oop_stmt([atm(class),atm(and_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_515810)

(3633) 4 Call: class_defn([atm(class),atm(and_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_515810)

(3634) 5 Call: c([atm(class),atm(and_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(class),_515850)

(3634) 5 Exit: c([atm(class),atm(and_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),sps

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
ym(.),atm(end),atm(and_c),spsym(.)],atm(class),[atm(and
_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm
(methods),atm(in2),spsym((),vbl(X),spsym()),spsym(:-),v
bl(Out),atm(is),spsym((),vbl(X),spsym(/\),spsym(:),vbl(
Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),
spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym
()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])


   (3635) 5 Call: classname(_515851,[atm(and_c),atm(has
),atm(super_class),atm(circuit),spsym(;),atm(methods),a
tm(in2),spsym((),vbl(X),spsym()),spsym(:-),vbl(Out),atm
(is),spsym((),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym
()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],_515852)


   (3636) 6 Call: c([atm(and_c),atm(has),atm(super_clas
s),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()
,vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vb
l(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),sps
ym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_m
eth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(e
nd),atm(and_c),spsym(.)],atm(_5650),_515852)


   (3636) 6 Exit: c([atm(and_c),atm(has),atm(super_clas
s),atm(circuit),spsym(;),atm(methods),atm(in2),spsym(()
,vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vb
l(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),sps
ym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_m
eth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(e
nd),atm(and_c),spsym(.)],atm(and_c),[atm(has),atm(super
_class),atm(circuit),spsym(;),atm(methods),atm(in2),sps
ym((),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)])


   (3638) 6 Call: classtree(_5651,and_c)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3638) 6 Fail: classtree(_5651,and_c)

(3635) 5 Exit: classname(and_c,[atm(and_c),atm(has),
atm(super_class),atm(circuit),spsym(;),atm(methods),atm
(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(i
s),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),
spsym(.),atm(end),atm(and_c),spsym(.)],[atm(has),atm(su
per_class),atm(circuit),spsym(;),atm(methods),atm(in2),
spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(

(3639) 5 Call: c([atm(has),atm(super_class),atm(circ
uit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),sps
ym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(
/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym
(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(
(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and
_c),spsym(.)],atm(has),_515853)

(3639) 5 Exit: c([atm(has),atm(super_class),atm(circ
uit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),sps
ym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(
/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym
(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(
(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and
_c),spsym(.)],atm(has),[atm(super_class),atm(circuit),s
psym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),
spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),sp
sym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)])

(3640) 5 Call: properties(and_c,[atm(super_class),at
m(circuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],_515854)


   (3641) 6 Call: supers(and_c,[atm(super_class),atm(ci
rcuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),s
psym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_515937)


   (3642) 7 Call: c([atm(super_class),atm(circuit),spsy
m(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),sps
ym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym
(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],atm(super_class),_515951)


   (3642) 7 Exit: c([atm(super_class),atm(circuit),spsy
m(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym()),sps
ym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym
(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],atm(super_class),[atm(circuit),spsym(;),atm(method
s),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out)
,atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),s
psym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<
<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sps
ym()),spsym(.),atm(end),atm(and_c),spsym(.)])


   (3643) 7 Call: parent(and_c,[atm(circuit),spsym(;),a
tm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-)
,vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vb
l(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst
),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sps
```

```
ym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_
515952)


   (3644) 8 Call: c([atm(circuit),spsym(;),atm(methods)
,atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),a
tm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),sps
ym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<)
,spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5667),_
515952)


   (3644) 8 Exit: c([atm(circuit),spsym(;),atm(methods)
,atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),a
tm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),sps
ym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<)
,spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(circuit)
,[spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym(
)),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)])


   (3643) 7 Exit: parent(and_c,[atm(circuit),spsym(;),a
tm(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-)
,vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vb
l(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst
),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sps
ym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[
spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym())
,spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),s
psym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),s
psym(.)])


   (3646) 7 Call: c([spsym(;),atm(methods),atm(in2),sps
ym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
```

),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym(())),spsym(())),spsym(.),
atm(end),atm(and_c),spsym(.)],spsym(;),_515937)

(3646) 7 Exit: c([spsym(;),atm(methods),atm(in2),sps
ym(()),vbl(X),spsym(())),spsym(:-),vbl(Out),atm(is),spsym(
()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym(())),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym(())),spsym(())),spsym(.),
atm(end),atm(and_c),spsym(.)],spsym(;),[atm(methods),at
m(in2),spsym(()),vbl(X),spsym(())),spsym(:-),vbl(Out),atm(
is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym(
)),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sp
sym(:),vbl(To_meth),spsym(()),vbl(Out),spsym(())),spsym(())
,spsym(.),atm(end),atm(and_c),spsym(.)])

(3641) 6 Exit: supers(and_c,[atm(super_class),atm(ci
rcuit),spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),s
psym(())),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym(())),spsym(())),spsym(.),atm(end),atm(a
nd_c),spsym(.)],[atm(methods),atm(in2),spsym(()),vbl(X),
spsym(())),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),sps
ym(/\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym(()),sp
sym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),sps
ym(()),vbl(Out),spsym(())),spsym(())),spsym(.),atm(end),atm(
and_c),spsym(.)])

(3647) 6 Call: class_vars(and_c,[atm(methods),atm(in
2),spsym(()),vbl(X),spsym(())),spsym(:-),vbl(Out),atm(is),
spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym(())),s
psym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(
:),vbl(To_meth),spsym(()),vbl(Out),spsym(())),spsym(())),sps
ym(.),atm(end),atm(and_c),spsym(.)],_515938)

(3648) 7 Call: c([atm(methods),atm(in2),spsym(()),vbl
(X),spsym(())),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X)
,spsym(/\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)

,spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),
atm(and_c),spsym(.)],atm(class_var),_5693)

(3648) 7 Fail: c([atm(methods),atm(in2),spsym(()),vbl
(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X)
,spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)
,spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),
atm(and_c),spsym(.)],atm(class_var),_5693)

(3649) 7 Call: [atm(methods),atm(in2),spsym(()),vbl(X
),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),s
psym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),
spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),at
m(and_c),spsym(.)]=_5692

(3649) 7 Exit: [atm(methods),atm(in2),spsym(()),vbl(X
),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),s
psym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),
spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),at
m(and_c),spsym(.)]=[atm(methods),atm(in2),spsym(()),vbl(
X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)]

(3650) 7 Call: true

(3650) 7 Exit: true

(3647) 6 Exit: class_vars(and_c,[atm(methods),atm(in
2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),
spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),s
psym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(
:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),sps

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
ym(.),atm(end),atm(and_c),spsym(.)],[atm(methods),atm(i
n2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is)
,spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),
spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym
(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),sp
sym(.),atm(end),atm(and_c),spsym(.)])


   (3651) 6 Call: inst_vars(and_c,[atm(methods),atm(in2
),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),s
psym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),sp
sym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:
),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsy
m(.),atm(end),atm(and_c),spsym(.)],_515939)


   (3652) 7 Call: c([atm(methods),atm(in2),spsym(()),vbl
(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X)
,spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)
,spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),
atm(and_c),spsym(.)],atm(instance_var),_5707)


   (3652) 7 Fail: c([atm(methods),atm(in2),spsym(()),vbl
(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X)
,spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)
,spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),
atm(and_c),spsym(.)],atm(instance_var),_5707)


   (3653) 7 Call: [atm(methods),atm(in2),spsym(()),vbl(X
),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),s
psym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),
spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),at
m(and_c),spsym(.)]=_5706


   (3653) 7 Exit: [atm(methods),atm(in2),spsym(()),vbl(X
),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),s
psym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),
spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),s
```

```
psym(( ),vbl(Out),spsym( )),spsym( )),spsym(.),atm(end),at
m(and_c),spsym(.)]=[atm(methods),atm(in2),spsym(( ),vbl(
X),spsym( )),spsym(:-),vbl(Out),atm(is),spsym(( ),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym( )),spsym(,),spsym(( )
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(( ),vbl(Out),spsym( )),spsym( )),spsym(.),atm(end),a
tm(and_c),spsym(.)]

   (3654) 7 Call: true

   (3654) 7 Exit: true

   (3651) 6 Exit: inst_vars(and_c,[atm(methods),atm(in2
),spsym(( ),vbl(X),spsym( )),spsym(:-),vbl(Out),atm(is),s
psym(( ),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym( )),sp
sym(,),spsym(( ),spsym(:),vbl(To_inst),spsym(<<),spsym(:
),vbl(To_meth),spsym(( ),vbl(Out),spsym( )),spsym( )),spsy
m(.),atm(end),atm(and_c),spsym(.)],[atm(methods),atm(in
2),spsym(( ),vbl(X),spsym( )),spsym(:-),vbl(Out),atm(is),
spsym(( ),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym( )),s
psym(,),spsym(( ),spsym(:),vbl(To_inst),spsym(<<),spsym(
:),vbl(To_meth),spsym(( ),vbl(Out),spsym( )),spsym( )),sps
ym(.),atm(end),atm(and_c),spsym(.)])

   (3655) 6 Call: methods(and_c,[atm(methods),atm(in2),
spsym(( ),vbl(X),spsym( )),spsym(:-),vbl(Out),atm(is),sps
ym(( ),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym( )),spsy
m(,),spsym(( ),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(( ),vbl(Out),spsym( )),spsym( )),spsym(

   (3656) 7 Call: c([atm(methods),atm(in2),spsym(( ),vbl
(X),spsym( )),spsym(:-),vbl(Out),atm(is),spsym(( ),vbl(X)
,spsym(/\),spsym(:),vbl(Temp),spsym( )),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)
,spsym(( ),vbl(Out),spsym( )),spsym( )),spsym(.),atm(end),
atm(and_c),spsym(.)],atm(methods),_516100)

   (3656) 7 Exit: c([atm(methods),atm(in2),spsym(( ),vbl
```

```
(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vbl(X)
,spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym((
),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth)
,spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),
atm(and_c),spsym(.)],atm(methods),[atm(in2),spsym((),vb
l(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vbl(X
),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(
(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth
),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end)
,atm(and_c),spsym(.)])


   (3657) 7 Call: methodlist(and_c,[atm(in2),spsym((),v
bl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vbl(
X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym
((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end
),atm(and_c),spsym(.)],_515854)


   (3658) 8 Call: method_defn(and_c,[atm(in2),spsym((),
vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vbl
(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsy
m((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_me
th),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],_516120)


   (3659) 9 Call: told


   (3659) 9 Exit: told


   (3660) 9 Call: tell(.method)


   (3660) 9 Exit: tell(.method)


meth_defn(and_c,

   (3664) 9 Call: meth_head([atm(in2),s
psym((),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsy
```

```
m(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.
),atm(end),atm(and_c),spsym(.)],_516132)


   (3665) 10 Call: c([atm(in2),spsym(()),vbl(X),spsym())
,spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),s
psym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],atm(_5721),_516183)


   (3665) 10 Exit: c([atm(in2),spsym(()),vbl(X),spsym())
,spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),s
psym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],atm(in2),[spsym(()),vbl(X),spsym())),spsym(:-),v
bl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(
Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),
spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym
())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)])


in2,(in2(_Instance

   (3670) 10 Call: args(_516184,[spsy
m(()),vbl(X),spsym())),spsym(:-),vbl(Out),atm(is),spsym((
),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,)
,spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(
To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),a
tm(end),atm(and_c),spsym(.)],_516185)


   (3671) 11 Call: c([spsym(()),vbl(X),spsym())),spsym(:-
),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),v
bl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_ins
t),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sp
sym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],
spsym(()),_516237)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3671) 11 Exit: c([spsym(()),vbl(X),spsym()),spsym(:-
),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),v
bl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_ins
t),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sp
sym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],
spsym(()),[vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),sp
sym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),sps
ym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:)
,vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym
(.),atm(end),atm(and_c),spsym(.)])


(3672) 11 Call: term(_516238,[vbl(X),spsym()),spsym(
:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:)
,vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_i
nst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),
spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)
],_516239)


(3673) 12 Call: pred(_5728,[vbl(X),spsym()),spsym(:-
),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),v
bl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_ins
t),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sp
sym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],
_5730)


(3674) 13 Call: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5742
),_516277)


(3674) 13 Fail: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5742
),_516277)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3673) 12 Back to: pred(_5728,[vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5730)

(3673) 12 Fail: pred(_5728,[vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5730)

(3675) 12 Call: const(_5728,[vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5730)

(3676) 13 Call: c([vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],num(_5745),_5730)

(3676) 13 Fail: c([vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],num(_5745),_5730)

(3677) 13 Call: c([vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(),vbl(Out),spsym()),sp

```
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5745
),_5730)


   (3677) 13 Fail: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5745
),_5730)


   (3678) 13 Call: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym("),
_516286)


   (3678) 13 Fail: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym("),
_516286)


   (3675) 12 Back to: const(_5728,[vbl(X),spsym()),spsy
m(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(
:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3679) 13 Call: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(_57
45),_516286)


   (3679) 13 Fail: c([vbl(X),spsym()),spsym(:-),vbl(Out
```

```
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(_57
45),_516286)


   (3675) 12 Back to: const(_5728,[vbl(X),spsym()),spsy
m(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(
:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3675) 12 Fail: const(_5728,[vbl(X),spsym()),spsym(:
-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),
vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_in
st),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),s
psym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)]
,_5730)


   (3680) 12 Call: variable(_5728,[vbl(X),spsym()),spsy
m(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(
:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3681) 13 Call: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_5749
),_5730)


   (3681) 13 Exit: c([vbl(X),spsym()),spsym(:-),vbl(Out
),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),
spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(
<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(X),[s
psym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),sps
```

```
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(a
nd_c),spsym(.)])


   (3680) 12 Exit: variable([X],[vbl(X),spsym()),spsym(
:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:)
,vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_i
nst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),
spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)
],[spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)])


   (3672) 11 Exit: term([X],[vbl(X),spsym()),spsym(:-),
vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl
(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst)
,spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsy
m()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[s
psym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(a
nd_c),spsym(.)])


   (3682) 11 Call: rest_args(_516240,[spsym()),spsym(:-
),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),v
bl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_ins
t),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sp
sym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],
_516241)


   (3683) 12 Call: c([spsym()),spsym(:-),vbl(Out),atm(i
s),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),
spsym(.),atm(end),atm(and_c),spsym(.)],spsym(,),_516314
)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3683) 12 Fail: c([spsym()),spsym(:-),vbl(Out),atm(i
s),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),
spsym(.),atm(end),atm(and_c),spsym(.)],spsym(,),_516314
)

(3682) 11 Back to: rest_args(_516240,[spsym()),spsym
(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],_516241)

(3684) 12 Call: true

(3684) 12 Exit: true

(3682) 11 Exit: rest_args([],[spsym()),spsym(:-),vbl
(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Te
mp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],[spsy
m())),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/
\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((
),vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_
c),spsym(.)])

(3685) 11 Call: c([spsym()),spsym(:-),vbl(Out),atm(i
s),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),
spsym(.),atm(end),atm(and_c),spsym(.)],spsym())),_516185
)

(3685) 11 Exit: c([spsym()),spsym(:-),vbl(Out),atm(i

```
s),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),
spsym(.),atm(end),atm(and_c),spsym(.)],spsym()),[spsym(
:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:)
,vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_i
nst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),
spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)
])


   (3686) 11 Call: append([X],[],_516184)


   (3687) 12 Call: append([],[],_5760)


   (3687) 12 Exit: append([],[],[])


   (3686) 11 Exit: append([X],[],[X])


   (3670) 10 Exit: args([X],[spsym(()),vbl(X),spsym()),s
psym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],[spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(a
nd_c),spsym(.)])


   (3688) 10 Call: printargs([X],[spsym(:-),vbl(Out),at
m(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],_516132)


,   (3690) 11 Call: writestr([X])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

X (3692) 12 Call: writestr([])

(3692) 12 Exit: writestr([])

(3690) 11 Exit: writestr([X])

(3688) 10 Exit: printargs([X],[spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

) (3664) 9 Exit: meth_head([atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3694) 9 Call: meth_tail([spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_516120)

(3695) 10 Call: c([spsym(:-),vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

atm(end),atm(and_c),spsym(.)],spsym(:-),_5767)

(3695) 10 Exit: c([spsym(:-),vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],spsym(:-),[vbl(Out),atm(i
s),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),
spsym(.),atm(end),atm(and_c),spsym(.)])

:- (3697) 10 Call: body([vbl(Out),atm(is),spsym(()),
vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),s
psym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To
_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm
(end),atm(and_c),spsym(.)],_5766)

(3698) 11 Call: subgoallist([vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],_516436)

(3699) 12 Call: subgoal([vbl(Out),atm(is),spsym(()),v
bl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),sp
sym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_
meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(
end),atm(and_c),spsym(.)],_516447)

(3700) 13 Call: exp(_516458,[vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],_516447)

(3701) 14 Call: unop(_5800,[vbl(Out),atm(is),spsym((
),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,)

,spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(
To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),a
tm(end),atm(and_c),spsym(.)],_5801)

(3702) 15 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(:-),_5801)

(3702) 15 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(:-),_5801)

(3703) 15 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(?-),_5801)

(3703) 15 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(?-),_5801)

(3704) 15 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(not),_5801)

(3704) 15 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a

```
tm(and_c),spsym(.)],atm(not),_5801)


   (3705) 15 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(\+),_5801)


   (3705) 15 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(\+),_5801)


   (3706) 15 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(spy),_5801)


   (3706) 15 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(spy),_5801)


   (3707) 15 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(nospy),_5801)


   (3707) 15 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(nospy),_5801)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3701) 14 Fail: unop(_5800,[vbl(Out),atm(is),spsym(( ),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,) ,spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),a tm(end),atm(and_c),spsym(.)],_5801)

(3708) 14 Call: message(_5797,[vbl(Out),atm(is),spsy m(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym (,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5799)

(3709) 15 Call: receiver(_516496,[vbl(Out),atm(is),s psym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),sp sym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsy m(.),atm(end),atm(and_c),spsym(.)],_516497)

(3710) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X), spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(() ,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth), spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a tm(and_c),spsym(.)],atm(_5821),_5820)

(3710) 16 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X), spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(() ,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth), spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a tm(and_c),spsym(.)],atm(_5821),_5820)

(3711) 16 Call: variable(_5818,[vbl(Out),atm(is),sps ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:), vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(

(3712) 17 Call: c([vbl(Out),atm(is),spsym(()),vbl(X), spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],vbl(_5830),_5820)


   (3712) 17 Exit: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],vbl(Out),[atm(is),spsym(()),vbl(X),s
psym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),
spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),at
m(and_c),spsym(.)])


   (3711) 16 Exit: variable([Out],[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(
X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym
(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end
),atm(and_c),spsym(.)])


   (3709) 15 Exit: receiver([Out],[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(
X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym
(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end
),atm(and_c),spsym(.)])


   (3713) 15 Call: mesgop(_5817,[atm(is),spsym(()),vbl(X
),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(
(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth
),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end)
,atm(and_c),spsym(.)],_516498)


   (3714) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
```

```
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(<<),_516498)


   (3714) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(<<),_516498)


   (3715) 16 Call: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(<?),_516498)


   (3715) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(<?),_516498)


   (3716) 16 Call: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(?<),_516498)


   (3716) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(?<),_516498)


   (3717) 16 Call: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(?<?),_516498)


   (3717) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(?<?),_516498)


   (3713) 15 Fail: mesgop(_5817,[atm(is),spsym((),vbl(X
),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(
(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth
),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end)
,atm(and_c),spsym(.)],_516498)


   (3712) 17 Back to: c([vbl(Out),atm(is),spsym((),vbl(
X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym
((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end
),atm(and_c),spsym(.)],vbl(_5830),_5820)


   (3712) 17 Fail: c([vbl(Out),atm(is),spsym((),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],vbl(_5830),_5820)


   (3711) 16 Fail: variable(_5818,[vbl(Out),atm(is),sps
ym((),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy
m(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(


   (3718) 16 Call: spvbl(_5818,[vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],_5820)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3719) 17 Call: spvaraccess(:,[vbl(Out),atm(is),spsy
m(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],_5833)

(3720) 18 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(:),_5833)

(3720) 18 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(:),_5833)

(3719) 17 Fail: spvaraccess(:,[vbl(Out),atm(is),spsy
m(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],_5833)

(3721) 17 Call: spvaraccess(::,[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(

(3722) 18 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(::),_5835)

(3722) 18 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(::),_5835)


   (3721) 17 Fail: spvaraccess(::,[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(


   (3718) 16 Fail: spvbl(_5818,[vbl(Out),atm(is),spsym(
()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),
atm(end),atm(and_c),spsym(.)],_5820)


   (3709) 15 Fail: receiver(_516496,[vbl(Out),atm(is),s
psym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),sp
sym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:
),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsy
m(.),atm(end),atm(and_c),spsym(.)],_516497)


   (3708) 14 Back to: message(_5797,[vbl(Out),atm(is),s
psym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),sp
sym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:
),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsy
m(.),atm(end),atm(and_c),spsym(.)],_5799)


   (3708) 14 Fail: message(_5797,[vbl(Out),atm(is),spsy
m(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.
),atm(end),atm(and_c),spsym(.)],_5799)


   (3723) 14 Call: term(_5804,[vbl(Out),atm(is),spsym((
),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,)
,spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(
To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),a
tm(end),atm(and_c),spsym(.)],_5805)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3724) 15 Call: pred(_5804,[vbl(Out),atm(is),spsym((
),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,)
,spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(
To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),a
tm(end),atm(and_c),spsym(.)],_5805)

(3725) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(_5842),_516521)

(3725) 16 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(_5842),_516521)

(3724) 15 Back to: pred(_5804,[vbl(Out),atm(is),spsy
m(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],_5805)

(3724) 15 Fail: pred(_5804,[vbl(Out),atm(is),spsym((
),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,)
,spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(
To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),a
tm(end),atm(and_c),spsym(.)],_5805)

(3726) 15 Call: const(_5804,[vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],_5805)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3727) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],num(_5845),_5805)

(3727) 16 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],num(_5845),_5805)

(3728) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(_5845),_5805)

(3728) 16 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],atm(_5845),_5805)

(3729) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym('),_516530)

(3729) 16 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym('),_516530)

(3726) 15 Back to: const(_5804,[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsy

```
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(


   (3730) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(_5845),_516530)


   (3730) 16 Fail: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],spsym(_5845),_516530)


   (3726) 15 Back to: const(_5804,[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(


   (3726) 15 Fail: const(_5804,[vbl(Out),atm(is),spsym(
()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],_5805)


   (3731) 15 Call: variable(_5804,[vbl(Out),atm(is),sps
ym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy
m(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(


   (3732) 16 Call: c([vbl(Out),atm(is),spsym(()),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],vbl(_5849),_5805)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3732) 16 Exit: c([vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(Out),[atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3731) 15 Exit: variable([Out],[vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3723) 14 Exit: term([Out],[vbl(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3733) 14 Call: restexp([Out],_5797,[atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5799)

(3734) 15 Call: binop(_5858,[atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5859)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3735) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(:-),_5859)


   (3735) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(:-),_5859)


   (3736) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(-->),_5859)


   (3736) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(-->),_5859)


   (3737) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(->),_5859)


   (3737) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(->),_5859)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3738) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=),_5859)


   (3738) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=),_5859)


   (3739) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=..),_5859)


   (3739) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=..),_5859)


   (3740) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(==),_5859)


   (3740) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(==),_5859)


   (3741) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
```

```
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(\==),_5859)


   (3741) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(\==),_5859)


   (3742) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@<),_5859)


   (3742) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@<),_5859)


   (3743) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@>),_5859)


   (3743) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@>),_5859)


   (3744) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
```

,spsym(.)],spsym(@=<),_5859)

(3744) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@=<),_5859)

(3745) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@>=),_5859)

(3745) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(@>=),_5859)

(3746) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=:=),_5859)

(3746) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=:=),_5859)

(3747) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=\=),_5859)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3747) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=\=),_5859)


   (3748) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(<),_5859)


   (3748) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(<),_5859)


   (3749) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(>),_5859)


   (3749) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(>),_5859)


   (3750) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=<),_5859)


   (3750) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
```

```
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(=<),_5859)


   (3751) 16 Call: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(>=),_5859)


   (3751) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(>=),_5859)


   (3752) 16 Call: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(+),_5859)


   (3752) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(+),_5859)


   (3753) 16 Call: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(-),_5859)


   (3753) 16 Fail: c([atm(is),spsym((),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
```

```
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(-),_5859)


   (3754) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(/\),_5859)


   (3754) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(/\),_5859)


   (3755) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(\/),_5859)


   (3755) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(\/),_5859)


   (3756) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(*),_5859)


   (3756) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(*),_5859)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3757) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(/),_5859)

   (3757) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(/),_5859)

   (3758) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(//),_5859)

   (3758) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(//),_5859)

   (3759) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(>>),_5859)

   (3759) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(>>),_5859)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3760) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(^),_5859)


   (3760) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],spsym(^),_5859)


   (3761) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],atm(mod),_5859)


   (3761) 16 Fail: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],atm(mod),_5859)


   (3762) 16 Call: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],atm(is),_5859)


   (3762) 16 Exit: c([atm(is),spsym(()),vbl(X),spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],atm(is),[spsym(()),vbl(X),spsym(/\),spsym(:),
vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_in
st),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),s
psym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)]
)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3734) 15 Exit: binop(is,[atm(is),spsym(()),vbl(X),sp
sym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),s
psym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),sp
sym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm
(and_c),spsym(.)],[spsym(()),vbl(X),spsym(/\),spsym(:),v
bl(Temp),spsym())),spsym(,),spsym((),spsym(:),vbl(To_ins
t),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sp
sym())),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])


(3763) 15 Call: term(_5860,[spsym(()),vbl(X),spsym(/\
),spsym(:),vbl(Temp),spsym())),spsym(,),spsym((),spsym(:
),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()
,vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c
),spsym(.)],_5799)


(3764) 16 Call: pred(_5860,[spsym(()),vbl(X),spsym(/\
),spsym(:),vbl(Temp),spsym())),spsym(,),spsym((),spsym(:
),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()
,vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c
),spsym(.)],_5799)


(3765) 17 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym((),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],atm(_5905),_516609)


(3765) 17 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym((),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],atm(_5905),_516609)


(3764) 16 Back to: pred(_5860,[spsym(()),vbl(X),spsym
(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym((),spsy
m(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym
((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(an
```

```
d_c),spsym(.)],_5799)


   (3764) 16 Fail: pred(_5860,[spsym(()),vbl(X),spsym(/\
),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:
),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()
,vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c
),spsym(.)],_5799)


   (3766) 16 Call: const(_5860,[spsym(()),vbl(X),spsym(/
\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((
),vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_
c),spsym(.)],_5799)


   (3767) 17 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],num(_5908),_5799)


   (3767) 17 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],num(_5908),_5799)


   (3768) 17 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],atm(_5908),_5799)


   (3768) 17 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],atm(_5908),_5799)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3769) 17 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym("),_516618)


   (3769) 17 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym("),_516618)


   (3766) 16 Back to: const(_5860,[spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym())),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_5799)


   (3770) 17 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(_5908),_516618)


   (3770) 17 Exit: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(()),[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym(
)),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sp
sym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())
,spsym(.),atm(end),atm(and_c),spsym(.)])


   (3771) 17 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],qatom(
_5909),_516619)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3771) 17 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(|),spsym(,),spsym(|),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(|),vbl(Out),spsym(|)
,spsym(|),spsym(.),atm(end),atm(and_c),spsym(.)],qatom(
_5909),_516619)

(3770) 17 Back to: c([spsym(|),vbl(X),spsym(/\),spsy
m(:),vbl(Temp),spsym(|),spsym(,),spsym(|),spsym(:),vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(|),vbl(O
ut),spsym(|),spsym(|),spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(_5908),_516618)

(3770) 17 Fail: c([spsym(|),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym(|),spsym(,),spsym(|),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(|),vbl(Out)
,spsym(|),spsym(|),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(_5908),_516618)

(3766) 16 Back to: const(_5860,[spsym(|),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym(|),spsym(,),spsym(|),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(|),vbl(Out),spsym(|),spsym(|),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_5799)

(3766) 16 Fail: const(_5860,[spsym(|),vbl(X),spsym(/
\),spsym(:),vbl(Temp),spsym(|),spsym(,),spsym(|),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(|
),vbl(Out),spsym(|),spsym(|),spsym(.),atm(end),atm(and_
c),spsym(.)],_5799)

(3772) 16 Call: variable(_5860,[spsym(|),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym(|),spsym(,),spsym(|),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(|),vbl(Out),spsym(|),spsym(|),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_5799)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3773) 17 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym(())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym(())),spsym(())),spsym(.),atm(end),atm(and_c),spsym(.
)],vbl(_5912),_5799)


   (3773) 17 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym(())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym(())),spsym(())),spsym(.),atm(end),atm(and_c),spsym(.
)],vbl(_5912),_5799)


   (3772) 16 Fail: variable(_5860,[spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym(())),spsym(())),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_5799)


   (3774) 16 Call: spvbl(_5860,[spsym(()),vbl(X),spsym(/
\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym(()),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((
),vbl(Out),spsym(())),spsym(())),spsym(.),atm(end),atm(and_
c),spsym(.)],_5799)


   (3775) 17 Call: spvaraccess(:,[spsym(()),vbl(X),spsym
(/\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym(()),spsy
m(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym
(()),vbl(Out),spsym(())),spsym(())),spsym(.),atm(end),atm(an
d_c),spsym(.)],_5919)


   (3776) 18 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym(())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym(())),spsym(())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(:),_5919)


   (3776) 18 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym(())),spsym(,),spsym(()),spsym(:),vbl(To_
```

```
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(:),_5919)


   (3775) 17 Fail: spvaraccess(:,[spsym(()),vbl(X),spsym
(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsy
m(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym
(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(an
d_c),spsym(.)],_5919)


   (3777) 17 Call: spvaraccess(::,[spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_5921)


   (3778) 18 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(::),_5921)


   (3778) 18 Fail: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(::),_5921)


   (3777) 17 Fail: spvaraccess(::,[spsym(()),vbl(X),spsy
m(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),sps
ym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsy
m(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(a
nd_c),spsym(.)],_5921)


   (3774) 16 Fail: spvbl(_5860,[spsym(()),vbl(X),spsym(/
\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((
),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_
```

```
c),spsym(.)],_5799)


   (3779) 16 Call: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(()),_5894)


   (3779) 16 Exit: c([spsym(()),vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],spsym(()),[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym(
)),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sp
sym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())
,spsym(.),atm(end),atm(and_c),spsym(.)])


   (3780) 16 Call: exp(_5895,[vbl(X),spsym(/\),spsym(:)
,vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_i
nst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),
spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)
],_5896)


   (3781) 17 Call: unop(_5941,[vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.
)],_5942)


   (3782) 18 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
:-),_5942)


   (3782) 18 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
```

```
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
:-),_5942)


   (3783) 18 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
?-),_5942)


   (3783) 18 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
?-),_5942)


   (3784) 18 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(no
t),_5942)


   (3784) 18 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(no
t),_5942)


   (3785) 18 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
\+),_5942)


   (3785) 18 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
\+),_5942)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3786) 18 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(|)),spsym(,),spsym(|(),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(|(),vbl(Out),spsym(|))
,spsym(|)),spsym(.),atm(end),atm(and_c),spsym(.)],atm(sp
y),_5942)

(3786) 18 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(|)),spsym(,),spsym(|(),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(|(),vbl(Out),spsym(|))
,spsym(|)),spsym(.),atm(end),atm(and_c),spsym(.)],atm(sp
y),_5942)

(3787) 18 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(|)),spsym(,),spsym(|(),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(|(),vbl(Out),spsym(|))
,spsym(|)),spsym(.),atm(end),atm(and_c),spsym(.)],atm(no
spy),_5942)

(3787) 18 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(|)),spsym(,),spsym(|(),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(|(),vbl(Out),spsym(|))
,spsym(|)),spsym(.),atm(end),atm(and_c),spsym(.)],atm(no
spy),_5942)

(3781) 17 Fail: unop(_5941,[vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym(|)),spsym(,),spsym(|(),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(|(),vbl(Out)
,spsym(|)),spsym(|)),spsym(.),atm(end),atm(and_c),spsym(.
)],_5942)

(3788) 17 Call: message(_5895,[vbl(X),spsym(/\),spsy
m(:),vbl(Temp),spsym(|)),spsym(,),spsym(|(),spsym(:),vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(|(),vbl(O
ut),spsym(|)),spsym(|)),spsym(.),atm(end),atm(and_c),spsy
m(.)],_5896)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3789) 18 Call: receiver(_516680,[vbl(X),spsym(/\),s
psym(:),vbl(Temp),spsym(())),spsym(,),spsym((),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vb
l(Out),spsym(())),spsym(())),spsym(.),atm(end),atm(and_c),s
psym(.)],_516681)


   (3790) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(())),spsym(,),spsym((),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym())
,spsym(())),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5
962),_5961)


   (3790) 19 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(())),spsym(,),spsym((),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym())
,spsym(())),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5
962),_5961)


   (3791) 19 Call: variable(_5959,[vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym(())),spsym(,),spsym((),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(
Out),spsym(())),spsym(())),spsym(.),atm(end),atm(and_c),sps
ym(.)],_5961)


   (3792) 20 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(())),spsym(,),spsym((),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym())
,spsym(())),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_5
971),_5961)


   (3792) 20 Exit: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym(())),spsym(,),spsym((),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym())
,spsym(())),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(X)
,[spsym(/\),spsym(:),vbl(Temp),spsym(())),spsym(,),spsym(
(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth
),spsym((),vbl(Out),spsym(())),spsym(())),spsym(.),atm(end)
,atm(and_c),spsym(.)])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3791) 19 Exit: variable([X],[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3789) 18 Exit: receiver([X],[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3793) 18 Call: mesgop(_5958,[spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_516682)

(3794) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(<<),_516682)

(3794) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(<<),_516682)

(3795) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),

```
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(<?),_51
6682)


   (3795) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(<?),_51
6682)


   (3796) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(?<),_51
6682)


   (3796) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(?<),_51
6682)


   (3797) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(?<?),_5
16682)


   (3797) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(?<?),_5
16682)


   (3793) 18 Fail: mesgop(_5958,[spsym(/\),spsym(:),vbl
(Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst)
,spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsy
m())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],_5
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

16682)

(3792) 20 Back to: c([vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_5971),_5961)

(3792) 20 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_5971),_5961)

(3791) 19 Fail: variable(_5959,[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5961)

(3798) 19 Call: spvbl(_5959,[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(

(3799) 20 Call: spvaraccess(:,[vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5974)

(3800) 21 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:),_5974)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3800) 21 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
:),_5974)

(3799) 20 Fail: spvaraccess(:,[vbl(X),spsym(/\),spsy
m(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(O
ut),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsy
m(.)],_5974)

(3801) 20 Call: spvaraccess(::,[vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_5976)

(3802) 21 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
::),_5976)

(3802) 21 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
::),_5976)

(3801) 20 Fail: spvaraccess(::,[vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_5976)

(3798) 19 Fail: spvbl(_5959,[vbl(X),spsym(/\),spsym(
:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3789) 18 Fail: receiver(_516680,[vbl(X),spsym(/\),s
psym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),s
psym(.)],_516681)


   (3788) 17 Back to: message(_5895,[vbl(X),spsym(/\),s
psym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),s
psym(.)],_5896)


   (3788) 17 Fail: message(_5895,[vbl(X),spsym(/\),spsy
m(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(O
ut),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsy
m(.)],_5896)


   (3803) 17 Call: term(_5945,[vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],_5946)


   (3804) 18 Call: pred(_5945,[vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],_5946)


   (3805) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5
983),_516705)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3805) 19 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5
983),_516705)

   (3804) 18 Back to: pred(_5945,[vbl(X),spsym(/\),spsy
m(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(O
ut),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsy
m(.)],_5946)

   (3804) 18 Fail: pred(_5945,[vbl(X),spsym(/\),spsym(:
),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],_5946)

   (3806) 18 Call: const(_5945,[vbl(X),spsym(/\),spsym(
:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(

   (3807) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],num(_5
986),_5946)

   (3807) 19 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],num(_5
986),_5946)

   (3808) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
```

```
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5
986),_5946)


   (3808) 19 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_5
986),_5946)


   (3809) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
"),_516714)


   (3809) 19 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
"),_516714)


   (3806) 18 Back to: const(_5945,[vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_5946)


   (3810) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
_5986),_516714)


   (3810) 19 Fail: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
```

```
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(
_5986),_516714)


   (3806) 18 Back to: const(_5945,[vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_5946)


   (3806) 18 Fail: const(_5945,[vbl(X),spsym(/\),spsym(
:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3811) 18 Call: variable(_5945,[vbl(X),spsym(/\),sps
ym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_5946)


   (3812) 19 Call: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_5
990),_5946)


   (3812) 19 Exit: c([vbl(X),spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(X)
,[spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(
(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth
),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end)
,atm(and_c),spsym(.)])


   (3811) 18 Exit: variable([X],[vbl(X),spsym(/\),spsym
(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Ou
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],[spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),sp
sym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_
meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(
end),atm(and_c),spsym(.)])


   (3803) 17 Exit: term([X],[vbl(X),spsym(/\),spsym(:),
vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_in
st),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),s
psym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)]
,[spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(
(),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth
),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end)
,atm(and_c),spsym(.)])


   (3813) 17 Call: restexp([X],_5895,[spsym(/\),spsym(:
),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)],_5896)


   (3814) 18 Call: binop(_5999,[spsym(/\),spsym(:),vbl(
Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),
spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym
()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_60
00)


   (3815) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:-),_60
00)


   (3815) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:-),_60
00)
```

```
   (3816) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(-->),_6
000)


   (3816) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(-->),_6
000)


   (3817) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(->),_60
00)


   (3817) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(->),_60
00)


   (3818) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=),_600
0)


   (3818) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=),_600
0)


   (3819) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
```

```
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=..),_6
000)


   (3819) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=..),_6
000)


   (3820) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(==),_60
00)


   (3820) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(==),_60
00)


   (3821) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(\==),_6
000)


   (3821) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(\==),_6
000)


   (3822) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
```

)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@<),_60
00)

(3822) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@<),_60
00)

(3823) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@>),_60
00)

(3823) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@>),_60
00)

(3824) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@=<),_6
000)

(3824) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@=<),_6
000)

(3825) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@>=),_6
000)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3825) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(@>=),_6
000)


   (3826) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=:=),_6
000)


   (3826) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=:=),_6
000)


   (3827) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=\=),_6
000)


   (3827) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=\=),_6
000)


   (3828) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(<),_600
0)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3828) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(<),_600
0)

(3829) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(>),_600
0)

(3829) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(>),_600
0)

(3830) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=<),_60
00)

(3830) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(=<),_60
00)

(3831) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(>=),_60
00)

(3831) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(>=),_60
00)


   (3832) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(+),_600
0)


   (3832) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(+),_600
0)


   (3833) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(-),_600
0)


   (3833) 19 Fail: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(-),_600
0)


   (3834) 19 Call: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(/\),_60
00)


   (3834) 19 Exit: c([spsym(/\),spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(/\),[sp
```

```
sym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)])


   (3814) 18 Exit: binop(/\,[spsym(/\),spsym(:),vbl(Tem
p),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym
(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)])


   (3835) 18 Call: term(_6001,[spsym(:),vbl(Temp),spsym
()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)


   (3836) 19 Call: pred(_6001,[spsym(:),vbl(Temp),spsym
()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)


   (3837) 20 Call: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],atm(_6048),_516793)


   (3837) 20 Fail: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],atm(_6048),_516793)


   (3836) 19 Back to: pred(_6001,[spsym(:),vbl(Temp),sp
sym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<
),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsy
m()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3836) 19 Fail: pred(_6001,[spsym(:),vbl(Temp),spsym
()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3838) 19 Call: const(_6001,[spsym(:),vbl(Temp),spsy
m())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3839) 20 Call: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],num(_6051),_5896)

(3839) 20 Fail: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],num(_6051),_5896)

(3840) 20 Call: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],atm(_6051),_5896)

(3840) 20 Fail: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],atm(_6051),_5896)

(3841) 20 Call: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym())),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],spsym("),_516802)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3841) 20 Fail: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],spsym("),_516802)


   (3838) 19 Back to: const(_6001,[spsym(:),vbl(Temp),s
psym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<
<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),sps
ym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)


   (3842) 20 Call: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],spsym(_6051),_516802)


   (3842) 20 Exit: c([spsym(:),vbl(Temp),spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],spsym(:),[vbl(Temp),sps
ym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<)
,spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)])


   (3843) 20 Call: c([vbl(Temp),spsym()),spsym(,),spsym
(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end
),atm(and_c),spsym(.)],qatom(_6052),_516803)


   (3843) 20 Fail: c([vbl(Temp),spsym()),spsym(,),spsym
(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end
),atm(and_c),spsym(.)],qatom(_6052),_516803)


   (3842) 20 Back to: c([spsym(:),vbl(Temp),spsym()),sp
sym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:
),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsy
m(.),atm(end),atm(and_c),spsym(.)],spsym(_6051),_516802
)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3842) 20 Fail: c([spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(_6051),_516802)

(3838) 19 Back to: const(_6001,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3838) 19 Fail: const(_6001,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3844) 19 Call: variable(_6001,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3845) 20 Call: c([spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_6055),_5896)

(3845) 20 Fail: c([spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_6055),_5896)

(3844) 19 Fail: variable(_6001,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3846) 19 Call: spvbl(_6001,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3847) 20 Call: spvaraccess(:,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6062)

(3848) 21 Call: c([spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:),_6062)

(3848) 21 Exit: c([spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:),[vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3847) 20 Exit: spvaraccess(:,[spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

get_ivar(_Instance, (3850) 20 Call: variable([_6063],[vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_5896)

(3851) 21 Call: c([vbl(Temp),spsym()),spsym(,),spsym

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_6063),_5896)

(3851) 21 Exit: c([vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(Temp),[spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3850) 20 Exit: variable([Temp],[vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3852) 20 Call: lookup(vbl(Temp),_6059)

(3853) 21 Call: symtab(vbl(Temp),_6059)

(3853) 21 Fail: symtab(vbl(Temp),_6059)

(3854) 21 Call: anon(_6059)

(3855) 22 Call: anonvar(_6121)

(3855) 22 Fail: anonvar(_6121)

(3854) 21 Back to: anon(_6059)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3856) 22 Call: name(0,[_6121])

   (3856) 22 Exit: name(0,[48])

   (3857) 22 Call: assert(anonvar(48))

   (3857) 22 Exit: assert(anonvar(48))

   (3854) 21 Exit: anon(_0)

   (3858) 21 Call: assert(symtab(vbl(Temp),_0))

   (3858) 21 Exit: assert(symtab(vbl(Temp),_0))

   (3852) 20 Exit: lookup(vbl(Temp),_0)

'Temp',_0),   (3846) 19 Exit: spvbl([_0],[spsym(:),vbl(
Temp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),
spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym
()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[sp
sym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<
),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsy
m()),spsym(.),atm(end),atm(and_c),spsym(.)])

   (3835) 18 Exit: term([_0],[spsym(:),vbl(Temp),spsym(
)),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sp
sym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym())
,spsym(.),atm(end),atm(and_c),spsym(.)],[spsym()),spsym
(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)])

   (3863) 18 Call: append([X],[ ,/\, ],_6002)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3864) 19 Call: append([],[ ,/\, ],_6160)

   (3864) 19 Exit: append([],[ ,/\, ],[ ,/\, ])

   (3863) 18 Exit: append([X],[ ,/\, ],[X, ,/\, ])

   (3865) 18 Call: append([X, ,/\, ],[_0],_5895)

   (3866) 19 Call: append([ ,/\, ],[_0],_6167)

   (3867) 20 Call: append([/\, ],[_0],_6172)

   (3868) 21 Call: append([ ],[_0],_6179)

   (3869) 22 Call: append([],[_0],_6186)

   (3869) 22 Exit: append([],[_0],[_0])

   (3868) 21 Exit: append([ ],[_0],[ ,_0])

   (3867) 20 Exit: append([/\, ],[_0],[/\, ,_0])

   (3866) 19 Exit: append([ ,/\, ],[_0],[ ,/\, ,_0])

   (3865) 18 Exit: append([X, ,/\, ],[_0],[X, ,/\, ,_0]
)

   (3813) 17 Exit: restexp([X],[X, ,/\, ,_0],[spsym(/\)
,spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()),spsym(:)
,vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),
```

```
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],[spsym()),spsym(,),spsym((),spsym(:),vbl(To_
inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out)
,spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.
)])


   (3780) 16 Exit: exp([X, ,/\, ,_0],[vbl(X),spsym(/\),
spsym(:),vbl(Temp),spsym()),spsym(,),spsym((),spsym(:),
vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),v
bl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),
spsym(.)],[spsym()),spsym(,),spsym((),spsym(:),vbl(To_i
nst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),
spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)
])


   (3870) 16 Call: c([spsym()),spsym(,),spsym((),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((
),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_
c),spsym(.)],spsym()),_5799)


   (3870) 16 Exit: c([spsym()),spsym(,),spsym((),spsym(
:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((
),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_
c),spsym(.)],spsym()),[spsym(,),spsym((),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)])


   (3871) 16 Call: append([(],[X, ,/\, ,_0],_5897)


   (3872) 17 Call: append([],[X, ,/\, ,_0],_6209)


   (3872) 17 Exit: append([],[X, ,/\, ,_0],[X, ,/\, ,_0
])


   (3871) 16 Exit: append([(],[X, ,/\, ,_0],[(,X, ,/\,
,_0])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3873) 16 Call: append([(,X, ,/\, ,_0],[)],_5860)

   (3874) 17 Call: append([X, ,/\, ,_0],[)],_6214)

   (3875) 18 Call: append([ ,/\, ,_0],[)],_6221)

   (3876) 19 Call: append([/\, ,_0],[)],_6226)

   (3877) 20 Call: append([ ,_0],[)],_6231)

   (3878) 21 Call: append([_0],[)],_6236)

   (3879) 22 Call: append([],[)],_6241)

   (3879) 22 Exit: append([],[)],[)])

   (3878) 21 Exit: append([_0],[)],[_0,)])

   (3877) 20 Exit: append([ ,_0],[)],[ ,_0,)])

   (3876) 19 Exit: append([/\, ,_0],[)],[/\, ,_0,)])

   (3875) 18 Exit: append([ ,/\, ,_0],[)],[ ,/\, ,_0,)]
)

   (3874) 17 Exit: append([X, ,/\, ,_0],[)],[X, ,/\, ,_
0,)])

   (3873) 16 Exit: append([(,X, ,/\, ,_0],[)],[(,X, ,/\
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
, ,_0,)])

   (3763) 15 Exit: term([(,X, ,/\, ,_0,)],[spsym((),vbl
(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsy
m((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_me
th),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],[spsym(,),spsym((),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)])

   (3880) 15 Call: append([Out],[ ,is, ],_5861)

   (3881) 16 Call: append([],[ ,is, ],_6253)

   (3881) 16 Exit: append([],[ ,is, ],[ ,is, ])

   (3880) 15 Exit: append([Out],[ ,is, ],[Out, ,is, ])

   (3882) 15 Call: append([Out, ,is, ],[(,X, ,/\, ,_0,)
],_5797)

   (3883) 16 Call: append([ ,is, ],[(,X, ,/\, ,_0,)],_6
260)

   (3884) 17 Call: append([is, ],[(,X, ,/\, ,_0,)],_626
5)

   (3885) 18 Call: append([ ],[(,X, ,/\, ,_0,)],_6272)

   (3886) 19 Call: append([],[(,X, ,/\, ,_0,)],_6279)

   (3886) 19 Exit: append([],[(,X, ,/\, ,_0,)],[(,X, ,/
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
\, ,_0,)])


   (3885) 18 Exit: append([ ],[(,X, ,/\, ,_0,)],[ ,(,X,
 ,/\, ,_0,)])


   (3884) 17 Exit: append([is, ],[(,X, ,/\, ,_0,)],[is,
 ,(,X, ,/\, ,_0,)])


   (3883) 16 Exit: append([ ,is, ],[(,X, ,/\, ,_0,)],[
,is, ,(,X, ,/\, ,_0,)])


   (3882) 15 Exit: append([Out, ,is, ],[(,X, ,/\, ,_0,)
],[Out, ,is, ,(,X, ,/\, ,_0,)])


   (3733) 14 Exit: restexp([Out],[Out, ,is, ,(,X, ,/\,
,_0,)],[atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(
Temp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),
spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym
())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],[sp
sym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:
),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsy
m(.),atm(end),atm(and_c),spsym(.)])


   (3700) 13 Exit: exp([Out, ,is, ,(,X, ,/\, ,_0,)],[vb
l(Out),atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(T
emp),spsym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),s
psym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym(
)),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],[sps
ym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:)
,vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym
(.),atm(end),atm(and_c),spsym(.)])


   (3887) 13 Call: writestr([Out, ,is, ,(,X, ,/\, ,_0,)
])


Out   (3889) 14 Call: writestr([ ,is, ,(,X, ,/\, ,_0,)]
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
)
    (3891) 15 Call: writestr([is, ,(,X, ,/\, ,_0,)])
is   (3893) 16 Call: writestr([ ,(,X, ,/\, ,_0,)])
    (3895) 17 Call: writestr([(,X, ,/\, ,_0,)])
(   (3897) 18 Call: writestr([X, ,/\, ,_0,)])
X   (3899) 19 Call: writestr([ ,/\, ,_0,)])
    (3901) 20 Call: writestr([/\, ,_0,)])
/\   (3903) 21 Call: writestr([ ,_0,)])
    (3905) 22 Call: writestr([_0,)])
_0   (3907) 23 Call: writestr([)])
)   (3909) 24 Call: writestr([])
   (3909) 24 Exit: writestr([])
   (3907) 23 Exit: writestr([)])
   (3905) 22 Exit: writestr([_0,)])
   (3903) 21 Exit: writestr([ ,_0,)])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3901) 20 Exit: writestr([/\, ,_0,)])

   (3899) 19 Exit: writestr([ ,/\, ,_0,)])

   (3897) 18 Exit: writestr([X, ,/\, ,_0,)])

   (3895) 17 Exit: writestr([(,X, ,/\, ,_0,)])

   (3893) 16 Exit: writestr([ ,(,X, ,/\, ,_0,)])

   (3891) 15 Exit: writestr([is, ,(,X, ,/\, ,_0,)])

   (3889) 14 Exit: writestr([ ,is, ,(,X, ,/\, ,_0,)])

   (3887) 13 Exit: writestr([Out, ,is, ,(,X, ,/\, ,_0,)
])

   (3699) 12 Exit: subgoal([vbl(Out),atm(is),spsym(()),v
bl(X),spsym(/\),spsym(:),vbl(Temp),spsym())),spsym(,),sp
sym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_
meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(
end),atm(and_c),spsym(.)],[spsym(,),spsym(()),spsym(:),v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)])

   (3910) 12 Call: moresubgoals([spsym(,),spsym(()),spsy
m(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym
(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(an
d_c),spsym(.)],_516436)

   (3911) 13 Call: c([spsym(,),spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(

(3911) 13 Exit: c([spsym(,),spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
)),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(
spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)])

, (3913) 13 Call: subgoallist([spsym(()),spsym(:),vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(O
ut),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsy
m(.)],_6305)

(3914) 14 Call: subgoal([spsym(()),spsym(:),vbl(To_in
st),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),s
psym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)]
,_517388)

(3915) 15 Call: exp(_517399,[spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],_517388)

(3916) 16 Call: unop(_6332,[spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
)),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(

(3917) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(:-),_6333)

(3917) 17 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(:-),_6333)

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3918) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(?-),_6333)


   (3918) 17 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(?-),_6333)


   (3919) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(n
ot),_6333)


   (3919) 17 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(n
ot),_6333)


   (3920) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(\+),_6333)


   (3920) 17 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(\+),_6333)


   (3921) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(s
py),_6333)


   (3921) 17 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
```

```
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(s
py),_6333)


   (3922) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(n
ospy),_6333)


   (3922) 17 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(n
ospy),_6333)


   (3916) 16 Fail: unop(_6332,[spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3923) 16 Call: message(_6329,[spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6331)


   (3924) 17 Call: receiver(_517437,[spsym(()),spsym(:),
vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),v
bl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),
spsym(.)],_517438)


   (3925) 18 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_
6353),_6352)


   (3925) 18 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_
6353),_6352)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3926) 18 Call: variable(_6350,[spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6352)


   (3927) 19 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_
6362),_6352)


   (3927) 19 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_
6362),_6352)


   (3926) 18 Fail: variable(_6350,[spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6352)


   (3928) 18 Call: spvbl(_6350,[spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym
(.)],_6352)


   (3929) 19 Call: spvaraccess(:,[spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6365)


   (3930) 20 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(:),_6365)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3930) 20 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(:),_6365)

(3929) 19 Fail: spvaraccess(:,[spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6365)

(3931) 19 Call: spvaraccess(::,[spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6367)

(3932) 20 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(::),_6367)

(3932) 20 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(::),_6367)

(3931) 19 Fail: spvaraccess(::,[spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6367)

(3928) 18 Fail: spvbl(_6350,[spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],_6352)

(3924) 17 Fail: receiver(_517437,[spsym(()),spsym(:),
vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),v

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
bl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),
spsym(.)],_517438)


   (3923) 16 Back to: message(_6329,[spsym((),spsym(:),
vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),v
bl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),
spsym(.)],_6331)


   (3923) 16 Fail: message(_6329,[spsym((),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6331)


   (3933) 16 Call: term(_6336,[spsym((),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3934) 17 Call: pred(_6336,[spsym((),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out
),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(


   (3935) 18 Call: c([spsym((),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_
6374),_517462)


   (3935) 18 Fail: c([spsym((),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_
6374),_517462)


   (3934) 17 Back to: pred(_6336,[spsym((),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(
Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6337)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3934) 17 Fail: pred(_6336,[spsym(()),spsym(:),vbl(To
_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out
)),spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(


   (3936) 17 Call: const(_6336,[spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t)),spsym())),spsym())),spsym(.),atm(end),atm(and_c),spsym
(.)],_6337)


   (3937) 18 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],num(_
6377),_6337)


   (3937) 18 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],num(_
6377),_6337)


   (3938) 18 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_
6377),_6337)


   (3938) 18 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out)),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_
6377),_6337)


   (3939) 18 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
("),_517471)


   (3939) 18 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(''),_517471)

(3936) 17 Back to: const(_6336,[spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6337)

(3940) 18 Call: c([spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(_6377),_517471)

(3940) 18 Exit: c([spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(()),[spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3941) 18 Call: c([spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],qatom(_6378),_517472)

(3941) 18 Fail: c([spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],qatom(_6378),_517472)

(3940) 18 Back to: c([spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(_6377),_517471)

(3940) 18 Fail: c([spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(_6377),_517471)

   (3936) 17 Back to: const(_6336,[spsym((),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6337)

   (3936) 17 Fail: const(_6336,[spsym((),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],_6337)

   (3942) 17 Call: variable(_6336,[spsym((),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6337)

   (3943) 18 Call: c([spsym((),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_
6381),_6337)

   (3943) 18 Fail: c([spsym((),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_
6381),_6337)

   (3942) 17 Fail: variable(_6336,[spsym((),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6337)

   (3944) 17 Call: spvbl(_6336,[spsym((),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym((),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],_6337)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3945) 18 Call: spvaraccess(:,[spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6388)


   (3946) 19 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(:),_6388)


   (3946) 19 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(:),_6388)


   (3945) 18 Fail: spvaraccess(:,[spsym(()),spsym(:),vbl
(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(
Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),sps
ym(.)],_6388)


   (3947) 18 Call: spvaraccess(::,[spsym(()),spsym(:),vb
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6390)


   (3948) 19 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(::),_6390)


   (3948) 19 Fail: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(::),_6390)


   (3947) 18 Fail: spvaraccess(::,[spsym(()),spsym(:),vb
```

```
l(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)],_6390)


   (3944) 17 Fail: spvbl(_6336,[spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],_6337)


   (3949) 17 Call: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(()),_6363)


   (3949) 17 Exit: c([spsym(()),spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],spsym
(()),[spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_me
th),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)])


   (3950) 17 Call: exp(_6364,[spsym(:),vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6365)


   (3951) 18 Call: unop(_6410,[spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6411
)


   (3952) 19 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:-),_641
1)


   (3952) 19 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
```

```
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:-),_641
1)


   (3953) 19 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(?-),_641
1)


   (3953) 19 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(?-),_641
1)


   (3954) 19 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(not),_6411
)


   (3954) 19 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(not),_6411
)


   (3955) 19 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(\+),_641
1)


   (3955) 19 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(\+),_641
1)


   (3956) 19 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(spy),_6411
)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3956) 19 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(spy),_6411
)


   (3957) 19 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(nospy),_64
11)


   (3957) 19 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(nospy),_64
11)


   (3951) 18 Fail: unop(_6410,[spsym(:),vbl(To_inst),sp
sym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()
),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6411
)


   (3958) 18 Call: message(_6364,[spsym(:),vbl(To_inst)
,spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsy
m())),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6
365)


   (3959) 19 Call: receiver(_517533,[spsym(:),vbl(To_in
st),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),s
psym())),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)]
,_517534)


   (3960) 20 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_6431),_64
30)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3960) 20 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],atm(_6431),_64
30)

(3961) 20 Call: variable(_6428,[spsym(:),vbl(To_inst
),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sps
ym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_
6430)

(3962) 21 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_6440),_64
30)

(3962) 21 Fail: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],vbl(_6440),_64
30)

(3961) 20 Fail: variable(_6428,[spsym(:),vbl(To_inst
),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),sps
ym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_
6430)

(3963) 20 Call: spvbl(_6428,[spsym(:),vbl(To_inst),s
psym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym(
)),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_643
0)

(3964) 21 Call: spvaraccess(:,[spsym(:),vbl(To_inst)
,spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsy
m()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6
443)

(3965) 22 Call: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()

```
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:),_6443
)


   (3965) 22 Exit: c([spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(:),[vbl(
To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(O
ut),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsy
m(.)])


   (3964) 21 Exit: spvaraccess(:,[spsym(:),vbl(To_inst)
,spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsy
m()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[v
bl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),s
psym(.)])


get_ivar(_Instance,   (3967) 21 Call: variable([_6444],
[vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),
vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c)
,spsym(.)],_6430)


   (3968) 22 Call: c([vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],vbl(_6444),_6430)


   (3968) 22 Exit: c([vbl(To_inst),spsym(<<),spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],vbl(To_inst),[spsym(<<)
,spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)])


   (3967) 21 Exit: variable([To_inst],[vbl(To_inst),sps
ym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym
(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),s
psym()),spsym(.),atm(end),atm(and_c),spsym(.)])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(3969) 21 Call: lookup(vbl(To_inst),_6440)

(3970) 22 Call: symtab(vbl(To_inst),_6440)

(3970) 22 Fail: symtab(vbl(To_inst),_6440)

(3971) 22 Call: anon(_6440)

(3972) 23 Call: anonvar(_6502)

(3972) 23 Exit: anonvar(48)

(3973) 23 Call: name(_,_517657)

(3973) 23 Exit: name(_,[95])

(3974) 23 Call: _6503 is 48+1

(3974) 23 Exit: 49 is 48+1

(3975) 23 Call: retract(anonvar(48))

(3975) 23 Exit: retract(anonvar(48))

(3976) 23 Call: assert(anonvar(49))

(3976) 23 Exit: assert(anonvar(49))

(3977) 23 Call: append([95],[49],_517658)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3978) 24 Call: append([],[49],_6524)

(3978) 24 Exit: append([],[49],[49])

(3977) 23 Exit: append([95],[49],[95,49])

(3979) 23 Call: name(_6440,[95,49])

(3979) 23 Exit: name(_1,[95,49])

(3971) 22 Exit: anon(_1)

(3980) 22 Call: assert(symtab(vbl(To_inst),_1))

(3980) 22 Exit: assert(symtab(vbl(To_inst),_1))

(3969) 21 Exit: lookup(vbl(To_inst),_1)

'To_inst',_1), (3963) 20 Exit: spvbl([_1],[spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3959) 19 Exit: receiver([_1],[spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3985) 19 Call: mesgop(_6427,[spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),
atm(end),atm(and_c),spsym(.)],_517535)


   (3986) 20 Call: c([spsym(<<),spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),at
m(and_c),spsym(.)],spsym(<<),_517535)


   (3986) 20 Exit: c([spsym(<<),spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),at
m(and_c),spsym(.)],spsym(<<),[spsym(:),vbl(To_meth),sps
ym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(
and_c),spsym(.)])


   (3985) 19 Exit: mesgop(<<,[spsym(<<),spsym(:),vbl(To
_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm
(end),atm(and_c),spsym(.)],[spsym(:),vbl(To_meth),spsym
(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),atm(an
d_c),spsym(.)])


   (3987) 19 Call: mesg(_517536,[spsym(:),vbl(To_meth),
spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),a
tm(and_c),spsym(.)],_517537)


   (3988) 20 Call: c([spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],atm(_6559),_6558)


   (3988) 20 Fail: c([spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],atm(_6559),_6558)


   (3989) 20 Call: variable(_6556,[spsym(:),vbl(To_meth
),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end)
,atm(and_c),spsym(.)],_6558)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3990) 21 Call: c([spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],vbl(_6568),_6558)


   (3990) 21 Fail: c([spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],vbl(_6568),_6558)


   (3989) 20 Fail: variable(_6556,[spsym(:),vbl(To_meth
),spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end)
,atm(and_c),spsym(.)],_6558)


   (3991) 20 Call: spvbl(_6556,[spsym(:),vbl(To_meth),s
psym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),at
m(and_c),spsym(.)],_6558)


   (3992) 21 Call: spvaraccess(:,[spsym(:),vbl(To_meth)
,spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),
atm(and_c),spsym(.)],_6571)


   (3993) 22 Call: c([spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],spsym(:),_6571)


   (3993) 22 Exit: c([spsym(:),vbl(To_meth),spsym(()),vb
l(Out),spsym())),spsym())),spsym(.),atm(end),atm(and_c),s
psym(.)],spsym(:),[vbl(To_meth),spsym(()),vbl(Out),spsym
())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)])


   (3992) 21 Exit: spvaraccess(:,[spsym(:),vbl(To_meth)
,spsym(()),vbl(Out),spsym())),spsym())),spsym(.),atm(end),
atm(and_c),spsym(.)],[vbl(To_meth),spsym(()),vbl(Out),sp
sym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)])


get_ivar(_Instance,   (3995) 21 Call: variable([_6572],
[vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym())),spsym
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(.),atm(end),atm(and_c),spsym(.)],_6558)

   (3996) 22 Call: c([vbl(To_meth),spsym(()),vbl(Out),sp
sym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],
vbl(_6572),_6558)

   (3996) 22 Exit: c([vbl(To_meth),spsym(()),vbl(Out),sp
sym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)],
vbl(To_meth),[spsym(()),vbl(Out),spsym())),spsym())),spsym
(.),atm(end),atm(and_c),spsym(.)])

   (3995) 21 Exit: variable([To_meth],[vbl(To_meth),sps
ym(()),vbl(Out),spsym())),spsym())],spsym(.),atm(end),atm(
and_c),spsym(.)],[spsym(()),vbl(Out),spsym())),spsym())),s
psym(.),atm(end),atm(and_c),spsym(.)])

   (3997) 21 Call: lookup(vbl(To_meth),_6568)

   (3998) 22 Call: symtab(vbl(To_meth),_6568)

   (3998) 22 Fail: symtab(vbl(To_meth),_6568)

   (3999) 22 Call: anon(_6568)

   (4000) 23 Call: anonvar(_6630)

   (4000) 23 Exit: anonvar(49)

   (4001) 23 Call: name(_,_517943)

   (4001) 23 Exit: name(_,[95])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(4002) 23 Call: _6631 is 49+1

(4002) 23 Exit: 50 is 49+1

(4003) 23 Call: retract(anonvar(49))

(4003) 23 Exit: retract(anonvar(49))

(4004) 23 Call: assert(anonvar(50))

(4004) 23 Exit: assert(anonvar(50))

(4005) 23 Call: append([95],[50],_517944)

(4006) 24 Call: append([],[50],_6652)

(4006) 24 Exit: append([],[50],[50])

(4005) 23 Exit: append([95],[50],[95,50])

(4007) 23 Call: name(_6568,[95,50])

(4007) 23 Exit: name(_2,[95,50])

(3999) 22 Exit: anon(_2)

(4008) 22 Call: assert(symtab(vbl(To_meth),_2))

(4008) 22 Exit: assert(symtab(vbl(To_meth),_2))
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

(3997) 21 Exit: lookup(vbl(To_meth),_2)

'To_meth',_2), (3991) 20 Exit: spvbl([_2],[spsym(:),v
bl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.
),atm(end),atm(and_c),spsym(.)],[spsym(()),vbl(Out),spsy
m()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

(3987) 19 Exit: mesg([_2],[spsym(:),vbl(To_meth),sps
ym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(
and_c),spsym(.)],[spsym(()),vbl(Out),spsym()),spsym()),s
psym(.),atm(end),atm(and_c),spsym(.)])

(4013) 19 Call: args(_517538,[spsym(()),vbl(Out),spsy
m()),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6
365)

(4014) 20 Call: c([spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(()),_518
109)

(4014) 20 Exit: c([spsym(()),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],spsym(()),[vbl
(Out),spsym()),spsym()),spsym(.),atm(end),atm(and_c),sp
sym(.)])

(4015) 20 Call: term(_518110,[vbl(Out),spsym()),spsy
m()),spsym(.),atm(end),atm(and_c),spsym(.)],_518111)

(4016) 21 Call: pred(_6684,[vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

(4017) 22 Call: c([vbl(Out),spsym()),spsym()),spsym(

(4017) 22 Fail: c([vbl(Out),spsym()),spsym()),spsym(

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4016) 21 Back to: pred(_6684,[vbl(Out),spsym()),sps
ym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

   (4016) 21 Fail: pred(_6684,[vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

   (4018) 21 Call: const(_6684,[vbl(Out),spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

   (4019) 22 Call: c([vbl(Out),spsym()),spsym()),spsym(

   (4019) 22 Fail: c([vbl(Out),spsym()),spsym()),spsym(

   (4020) 22 Call: c([vbl(Out),spsym()),spsym()),spsym(

   (4020) 22 Fail: c([vbl(Out),spsym()),spsym()),spsym(

   (4021) 22 Call: c([vbl(Out),spsym()),spsym()),spsym(

   (4021) 22 Fail: c([vbl(Out),spsym()),spsym()),spsym(

   (4018) 21 Back to: const(_6684,[vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

   (4022) 22 Call: c([vbl(Out),spsym()),spsym()),spsym(

   (4022) 22 Fail: c([vbl(Out),spsym()),spsym()),spsym(

   (4018) 21 Back to: const(_6684,[vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4018) 21 Fail: const(_6684,[vbl(Out),spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

   (4023) 21 Call: variable(_6684,[vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],_6686)

   (4024) 22 Call: c([vbl(Out),spsym()),spsym()),spsym(

   (4024) 22 Exit: c([vbl(Out),spsym()),spsym()),spsym(
ym()),spsym(.),atm(end),atm(and_c),spsym(.)])

   (4023) 21 Exit: variable([Out],[vbl(Out),spsym()),sp
sym()),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym())
,spsym()),spsym(.),atm(end),atm(and_c),spsym(.)])

   (4015) 20 Exit: term([Out],[vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym()),sps
ym()),spsym(.),atm(end),atm(and_c),spsym(.)])

   (4025) 20 Call: rest_args(_518112,[spsym()),spsym())
,spsym(.),atm(end),atm(and_c),spsym(.)],_518113)

   (4026) 21 Call: c([spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],spsym(,),_518186)

   (4026) 21 Fail: c([spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],spsym(,),_518186)

   (4025) 20 Back to: rest_args(_518112,[spsym()),spsym
()),spsym(.),atm(end),atm(and_c),spsym(.)],_518113)

   (4027) 21 Call: true
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4027) 21 Exit: true

   (4025) 20 Exit: rest_args([],[spsym()),spsym()),spsy
m(.),atm(end),atm(and_c),spsym(.)],[spsym()),spsym()),s
psym(.),atm(end),atm(and_c),spsym(.)])

   (4028) 20 Call: c([spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],spsym()),_6365)

   (4028) 20 Exit: c([spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],spsym()),[spsym()),spsym(.),atm
(end),atm(and_c),spsym(.)])

   (4029) 20 Call: append([Out],[],_517538)

   (4030) 21 Call: append([],[],_6716)

   (4030) 21 Exit: append([],[],[])

   (4029) 20 Exit: append([Out],[],[Out])

   (4013) 19 Exit: args([Out],[spsym((),vbl(Out),spsym(
)),spsym()),spsym(.),atm(end),atm(and_c),spsym(.)],[sps
ym()),spsym(.),atm(end),atm(and_c),spsym(.)])

   (4031) 19 Call: append([send_mesg([],[_2],_517539)

   (4032) 20 Call: append([],[_2],_6721)

   (4032) 20 Exit: append([],[_2],[_2])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4031) 19 Exit: append([send_mesg([],[_2],[send_mesg
([,_2])

   (4033) 19 Call: append([send_mesg([,_2],[,],_517540)

   (4034) 20 Call: append([_2],[,],_6726)

   (4035) 21 Call: append([],[,],_6733)

   (4035) 21 Exit: append([],[,],[,])

   (4034) 20 Exit: append([_2],[,],[_2,,])

   (4033) 19 Exit: append([send_mesg([,_2],[,],[send_me
sg([,_2,,])

   (4036) 19 Call: append([send_mesg([,_2,,],[_1],_5175
41)

   (4037) 20 Call: append([_2,,],[_1],_6738)

   (4038) 21 Call: append([,],[_1],_6743)

   (4039) 22 Call: append([],[_1],_6748)

   (4039) 22 Exit: append([],[_1],[_1])

   (4038) 21 Exit: append([,],[_1],[,,_1])

   (4037) 20 Exit: append([_2,,],[_1],[_2,,,_1])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4036) 19 Exit: append([send_mesg([,_2,,],[_1],[send
_mesg([,_2,,,_1])

   (4040) 19 Call: append([send_mesg([,_2,,,_1],[,],_51
7542)

   (4041) 20 Call: append([_2,,,_1],[,],_6753)

   (4042) 21 Call: append([,,_1],[,],_6760)

   (4043) 22 Call: append([_1],[,],_6765)

   (4044) 23 Call: append([],[,],_6770)

   (4044) 23 Exit: append([],[,],[,])

   (4043) 22 Exit: append([_1],[,],[_1,,])

   (4042) 21 Exit: append([,,_1],[,],[,,_1,,])

   (4041) 20 Exit: append([_2,,,_1],[,],[_2,,,_1,,])

   (4040) 19 Exit: append([send_mesg([,_2,,,_1],[,],[se
nd_mesg([,_2,,,_1,,])

   (4045) 19 Call: append([send_mesg([,_2,,,_1,,],[Out]
,_517543)

   (4046) 20 Call: append([_2,,,_1,,],[Out],_6775)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(4047) 21 Call: append([,,_1,,],[Out],_6780)

(4048) 22 Call: append([_1,,],[Out],_6785)

(4049) 23 Call: append([,],[Out],_6790)

(4050) 24 Call: append([],[Out],_6795)

(4050) 24 Exit: append([],[Out],[Out])

(4049) 23 Exit: append([,],[Out],[,,Out])

(4048) 22 Exit: append([_1,,],[Out],[_1,,,Out])

(4047) 21 Exit: append([,,_1,,],[Out],[,,_1,,,Out])

(4046) 20 Exit: append([_2,,,_1,,],[Out],[_2,,,_1,,,
Out])

(4045) 19 Exit: append([send_mesg([,_2,,,_1,,],[Out]
,[send_mesg([,_2,,,_1,,,Out])

(4051) 19 Call: append([send_mesg([,_2,,,_1,,,Out],[
],,op(<<),)],_6364)

(4052) 20 Call: append([_2,,,_1,,,Out],[],,op(<<),)]
,_6800)

(4053) 21 Call: append([,,_1,,,Out],[],,op(<<),)],_6
807)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4054) 22 Call: append([_1,,,Out],[],,op(<<),)],_681
2)

   (4055) 23 Call: append([,,Out],[],,op(<<),)],_6817)

   (4056) 24 Call: append([Out],[],,op(<<),)],_6822)

   (4057) 25 Call: append([],[],,op(<<),)],_6827)

   (4057) 25 Exit: append([],[],,op(<<),)],[],,op(<<),)
])

   (4056) 24 Exit: append([Out],[],,op(<<),)],[Out,],,o
p(<<),)])

   (4055) 23 Exit: append([,,Out],[],,op(<<),)],[,,Out,
],,op(<<),)])

   (4054) 22 Exit: append([_1,,,Out],[],,op(<<),)],[_1,
,,Out,],,op(<<),)])

   (4053) 21 Exit: append([,,_1,,,Out],[],,op(<<),)],[,
,_1,,,Out,],,op(<<),)])

   (4052) 20 Exit: append([_2,,,_1,,,Out],[],,op(<<),)]
,[_2,,,_1,,,Out,],,op(<<),)])

   (4051) 19 Exit: append([send_mesg([,_2,,,_1,,,Out],[
],,op(<<),)],[send_mesg([,_2,,,_1,,,Out,],,op(<<),)])

   (3958) 18 Exit: message([send_mesg([,_2,,,_1,,,Out,]
,,op(<<),)],[spsym(:),vbl(To_inst),spsym(<<),spsym(:),v
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
b1(To_meth),spsym(()),vb1(Out),spsym())),spsym())),spsym(.
),atm(end),atm(and_c),spsym(.)],[spsym())),spsym(.),atm(
end),atm(and_c),spsym(.)])


   (3950) 17 Exit: exp([send_mesg([,_2,,,_1,,,Out,],,op
(<<),)],[spsym(:),vb1(To_inst),spsym(<<),spsym(:),vb1(T
o_meth),spsym(()),vb1(Out),spsym())),spsym())),spsym(.),at
m(end),atm(and_c),spsym(.)],[spsym())),spsym(.),atm(end)
,atm(and_c),spsym(.)])


   (4058) 17 Call: c([spsym())),spsym(.),atm(end),atm(an
d_c),spsym(.)],spsym())),_6337)


   (4058) 17 Exit: c([spsym())),spsym(.),atm(end),atm(an
d_c),spsym(.)],spsym())),[spsym(.),atm(end),atm(and_c),s
psym(.)])


   (4059) 17 Call: append([(],[send_mesg([,_2,,,_1,,,Ou
t,],,op(<<),)],_6366)


   (4060) 18 Call: append([],[send_mesg([,_2,,,_1,,,Out
,],,op(<<),)],_6850)


   (4060) 18 Exit: append([],[send_mesg([,_2,,,_1,,,Out
,],,op(<<),)],[send_mesg([,_2,,,_1,,,Out,],,op(<<),)])


   (4059) 17 Exit: append([(],[send_mesg([,_2,,,_1,,,Ou
t,],,op(<<),)],[(,send_mesg([,_2,,,_1,,,Out,],,op(<<),)
])


   (4061) 17 Call: append([(,send_mesg([,_2,,,_1,,,Out,
],,op(<<),)],[)],_6336)


   (4062) 18 Call: append([send_mesg([,_2,,,_1,,,Out,],
,op(<<),)],[)],_6855)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4063) 19 Call: append([_2,,,_1,,,Out,],,op(<<),)],[
)],_6862)

   (4064) 20 Call: append([,,_1,,,Out,],,op(<<),)],[)],
_6867)

   (4065) 21 Call: append([_1,,,Out,],,op(<<),)],[)],_6
872)

   (4066) 22 Call: append([,,Out,],,op(<<),)],[)],_6877
)

   (4067) 23 Call: append([Out,],,op(<<),)],[)],_6882)

   (4068) 24 Call: append([],,op(<<),)],[)],_6887)

   (4069) 25 Call: append([op(<<),)],[)],_6892)

   (4070) 26 Call: append([)],[)],_6899)

   (4071) 27 Call: append([],[)],_6908)

   (4071) 27 Exit: append([],[)],[)])

   (4070) 26 Exit: append([)],[)],[),)])

   (4069) 25 Exit: append([op(<<),)],[)],[op(<<),),)])

   (4068) 24 Exit: append([],,op(<<),)],[)],[],,op(<<),
),)])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4067) 23 Exit: append([Out,],,op(<<),)],[])],[Out,],
,op(<<),),)])

   (4066) 22 Exit: append([,,Out,],,op(<<),)],[])],[,,Ou
t,],,op(<<),),)])

   (4065) 21 Exit: append([_1,,,Out,],,op(<<),)],[])],[_
1,,,Out,],,op(<<),),)])

   (4064) 20 Exit: append([,,_1,,,Out,],,op(<<),)],[])],
[,,_1,,,Out,],,op(<<),),)])

   (4063) 19 Exit: append([_2,,,_1,,,Out,],,op(<<),)],[
)],[_2,,,_1,,,Out,],,op(<<),),)])

   (4062) 18 Exit: append([send_mesg([,_2,,,_1,,,Out,],
,op(<<),)],[])],[send_mesg([,_2,,,_1,,,Out,],,op(<<),),)
])

   (4061) 17 Exit: append([(,send_mesg([,_2,,,_1,,,Out,
],,op(<<),)],[])],[(,send_mesg([,_2,,,_1,,,Out,],,op(<<)
,),)])

   (3933) 16 Exit: term([(,send_mesg([,_2,,,_1,,,Out,],
,op(<<),),)],[spsym((),spsym(:),vbl(To_inst),spsym(<<),
spsym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym(
)),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(.),atm
(end),atm(and_c),spsym(.)])

   (4072) 16 Call: restexp([(,send_mesg([,_2,,,_1,,,Out
,],,op(<<),),)],_6329,[spsym(.),atm(end),atm(and_c),sps
ym(.)],_6331)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4073) 17 Call: binop(_6915,[spsym(.),atm(end),atm(a
nd_c),spsym(.)],_6916)

   (4074) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(:-),_6916)

   (4074) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(:-),_6916)

   (4075) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(-->),_6916)

   (4075) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(-->),_6916)

   (4076) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(->),_6916)

   (4076) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(->),_6916)

   (4077) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=),_6916)

   (4077) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=),_6916)

   (4078) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=..),_6916)

   (4078) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=..),_6916)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4079) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(==),_6916)

   (4079) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(==),_6916)

   (4080) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(\==),_6916)

   (4080) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(\==),_6916)

   (4081) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@<),_6916)

   (4081) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@<),_6916)

   (4082) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@>),_6916)

   (4082) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@>),_6916)

   (4083) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@=<),_6916)

   (4083) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@=<),_6916)

   (4084) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@>=),_6916)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4084) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(@>=),_6916)

   (4085) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=:=),_6916)

   (4085) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=:=),_6916)

   (4086) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=\=),_6916)

   (4086) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=\=),_6916)

   (4087) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(<),_6916)

   (4087) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(<),_6916)

   (4088) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(>),_6916)

   (4088) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(>),_6916)

   (4089) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=<),_6916)

   (4089) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=<),_6916)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4090) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(>=),_6916)

   (4090) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(>=),_6916)

   (4091) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(+),_6916)

   (4091) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(+),_6916)

   (4092) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(-),_6916)

   (4092) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(-),_6916)

   (4093) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(/\),_6916)

   (4093) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(/\),_6916)

   (4094) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(\/),_6916)

   (4094) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(\/),_6916)

   (4095) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(*),_6916)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4095) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(*),_6916)

   (4096) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(/),_6916)

   (4096) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(/),_6916)

   (4097) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(//),_6916)

   (4097) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(//),_6916)

   (4098) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(>>),_6916)

   (4098) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(>>),_6916)

   (4099) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(^),_6916)

   (4099) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(^),_6916)

   (4100) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],atm(mod),_6916)

   (4100) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],atm(mod),_6916)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4101) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],atm(is),_6916)

   (4101) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],atm(is),_6916)

   (4073) 17 Fail: binop(_6915,[spsym(.),atm(end),atm(a
nd_c),spsym(.)],_6916)

   (4102) 17 Call: spvarasgn(=:,[spsym(.),atm(end),atm(
and_c),spsym(.)],_6919)

   (4103) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=:),_6919)

   (4103) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=:),_6919)

   (4102) 17 Fail: spvarasgn(=:,[spsym(.),atm(end),atm(
and_c),spsym(.)],_6919)

   (4104) 17 Call: spvarasgn(=::,[spsym(.),atm(end),atm
(and_c),spsym(.)],_6927)

   (4105) 18 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=::),_6927)

   (4105) 18 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(=::),_6927)

   (4104) 17 Fail: spvarasgn(=::,[spsym(.),atm(end),atm
(and_c),spsym(.)],_6927)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4106) 17 Call: true


   (4106) 17 Exit: true


   (4072) 16 Exit: restexp([(,send_mesg([,_2,,,_1,,,Out
,],,op(<<),),)],[(,send_mesg([,_2,,,_1,,,Out,],,op(<<),
),)],[spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(.),
atm(end),atm(and_c),spsym(.)])


   (3915) 15 Exit: exp([(,send_mesg([,_2,,,_1,,,Out,],,
op(<<),),)],[spsym((),spsym(:),vbl(To_inst),spsym(<<),s
psym(:),vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()
),spsym(.),atm(end),atm(and_c),spsym(.)],[spsym(.),atm(
end),atm(and_c),spsym(.)])


   (4107) 15 Call: writestr([(,send_mesg([,_2,,,_1,,,Ou
t,],,op(<<),),)])


(    (4109) 16 Call: writestr([send_mesg([,_2,,,_1,,,Out
,],,op(<<),),)])


send_mesg([    (4111) 17 Call: writestr([_2,,,_1,,,Out,]
,,op(<<),),)])


_2    (4113) 18 Call: writestr([,,_1,,,Out,],,op(<<),),)
])


,    (4115) 19 Call: writestr([_1,,,Out,],,op(<<),),)])


_1    (4117) 20 Call: writestr([,,Out,],,op(<<),),)])


,    (4119) 21 Call: writestr([Out,],,op(<<),),)])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
Out   (4121) 22 Call: writestr([],,op(<<),),)])

],   (4123) 23 Call: writestr([op(<<),),)])

op(<<)   (4125) 24 Call: writestr([),)])

)   (4127) 25 Call: writestr([)])

)   (4129) 26 Call: writestr([])

   (4129) 26 Exit: writestr([])

   (4127) 25 Exit: writestr([)])

   (4125) 24 Exit: writestr([),)])

   (4123) 23 Exit: writestr([op(<<),),)])

   (4121) 22 Exit: writestr([],,op(<<),),)])

   (4119) 21 Exit: writestr([Out,],,op(<<),),)])

   (4117) 20 Exit: writestr([,,Out,],,op(<<),),)])

   (4115) 19 Exit: writestr([_1,,,Out,],,op(<<),),)])

   (4113) 18 Exit: writestr([,,_1,,,Out,],,op(<<),),)])

   (4111) 17 Exit: writestr([_2,,,_1,,,Out,],,op(<<),),
)])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (4109) 16 Exit: writestr([send_mesg([,_2,,,_1,,,Out,
],,op(<<),),)])

   (4107) 15 Exit: writestr([(,send_mesg([,_2,,,_1,,,Ou
t,],,op(<<),),)])

   (3914) 14 Exit: subgoal([spsym(()),spsym(:),vbl(To_in
st),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),s
psym())),spsym())),spsym(.),atm(end),atm(and_c),spsym(.)]
,[spsym(.),atm(end),atm(and_c),spsym(.)])

   (4130) 14 Call: moresubgoals([spsym(.),atm(end),atm(
and_c),spsym(.)],_6305)

   (4131) 15 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(,),_6935)

   (4131) 15 Fail: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(,),_6935)

   (4132) 15 Call: [spsym(.),atm(end),atm(and_c),spsym(

   (4132) 15 Exit: [spsym(.),atm(end),atm(and_c),spsym(

   (4133) 15 Call: true

   (4133) 15 Exit: true

   (4130) 14 Exit: moresubgoals([spsym(.),atm(end),atm(
and_c),spsym(.)],[spsym(.),atm(end),atm(and_c),spsym(.)
])
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
   (3913) 13 Exit: subgoallist([spsym(()),spsym(:),vbl(T
o_inst),spsym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Ou
t),spsym()),spsym()),spsym(.),atm(end),atm(and_c),spsym
(.)],[spsym(.),atm(end),atm(and_c),spsym(.)])


   (3910) 12 Exit: moresubgoals([spsym(,),spsym(()),spsy
m(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),spsym
(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end),atm(an
d_c),spsym(.)],[spsym(.),atm(end),atm(and_c),spsym(.)])


   (3698) 11 Exit: subgoallist([vbl(Out),atm(is),spsym(
(),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,
),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl
(To_meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),
atm(end),atm(and_c),spsym(.)],[spsym(.),atm(end),atm(an
d_c),spsym(.)])


   (4134) 11 Call: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(.),_5766)


   (4134) 11 Exit: c([spsym(.),atm(end),atm(and_c),spsy
m(.)],spsym(.),[atm(end),atm(and_c),spsym(.)])


)).


   (3697) 10 Exit: body([vbl(Out),atm(is),spsym(()),vbl(
X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym
(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(end
),atm(and_c),spsym(.)],[atm(end),atm(and_c),spsym(.)])


   (4137) 10 Call: consult(.method)
```

```
(4137) 10 Exit: consult(.method)

(3694) 9 Exit: meth_tail([spsym(:-),vbl(Out),atm(is)
,spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),
spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym
(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),sp
sym(.),atm(end),atm(and_c),spsym(.)],[atm(end),atm(and_
c),spsym(.)])

(4138) 9 Call: tell(user)

(4138) 9 Exit: tell(user)

(3658) 8 Exit: method_defn(and_c,[atm(in2),spsym(()),
vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl
(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsy
m(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_me
th),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(en
d),atm(and_c),spsym(.)],[atm(end),atm(and_c),spsym(.)])

(4139) 8 Call: rest_mlist(and_c,[atm(end),atm(and_c)
,spsym(.)],_515854)

(4140) 9 Call: c([atm(end),atm(and_c),spsym(.)],atm(
end),_6954)

(4140) 9 Exit: c([atm(end),atm(and_c),spsym(.)],atm(
end),[atm(and_c),spsym(.)])

(4139) 8 Exit: rest_mlist(and_c,[atm(end),atm(and_c)
,spsym(.)],[atm(and_c),spsym(.)])

(3657) 7 Exit: methodlist(and_c,[atm(in2),spsym(()),v
bl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),vbl(
X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_met
h),spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end
),atm(and_c),spsym(.)],[atm(and_c),spsym(.)])


   (3655) 6 Exit: methods(and_c,[atm(methods),atm(in2),
spsym((),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),sps
ym((),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsy
m(,),spsym((),spsym(:),vbl(To_inst),spsym(<<),spsym(:),
vbl(To_meth),spsym((),vbl(Out),spsym()),spsym()),spsym(


   (3640) 5 Exit: properties(and_c,[atm(super_class),at
m(circuit),spsym(;),atm(methods),atm(in2),spsym((),vbl(
X),spsym()),spsym(:-),vbl(Out),atm(is),spsym((),vbl(X),
spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),spsym(()
,spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_meth),
spsym((),vbl(Out),spsym()),spsym()),spsym(.),atm(end),a
tm(and_c),spsym(.)],[atm(and_c),spsym(.)])


   (4141) 5 Call: c([atm(and_c),spsym(.)],atm(_5647),_5
15855)


   (4141) 5 Exit: c([atm(and_c),spsym(.)],atm(and_c),[s
psym(.)])


   (4142) 5 Call: match(and_c,and_c)


   (4142) 5 Exit: match(and_c,and_c)


   (4143) 5 Call: c([spsym(.)],spsym(.),_515810)


   (4143) 5 Exit: c([spsym(.)],spsym(.),[])


   (3633) 4 Exit: class_defn([atm(class),atm(and_c),atm
(has),atm(super_class),atm(circuit),spsym(;),atm(method
s),atm(in2),spsym((),vbl(X),spsym()),spsym(:-),vbl(Out)
```

```
,atm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),s
psym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<
<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),sps
ym())),spsym(.),atm(end),atm(and_c),spsym(.)],[])


   (3630) 3 Exit: oop_stmt([atm(class),atm(and_c),atm(h
as),atm(super_class),atm(circuit),spsym(;),atm(methods)
,atm(in2),spsym(()),vbl(X),spsym())),spsym(:-),vbl(Out),a
tm(is),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),sps
ym())),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<)
,spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym())),spsym
())),spsym(.),atm(end),atm(and_c),spsym(.)],[])


   (4144) 3 Call: oop([],_157)


   (4145) 4 Call: oop_stmt([],_518892)


No leashing.


   (4148) 5 Call: class_defn([],_518892)


   (4149) 6 Call: c([],atm(class),_518932)


   (4149) 6 Fail: c([],atm(class),_518932)


   (4148) 5 Back to: class_defn([],_518892)


   (4148) 5 Fail: class_defn([],_518892)


   (4145) 4 Back to: oop_stmt([],_518892)


   (4150) 5 Call: inst_create([],_518892)
```

APPENDIX I TRACE OF CLASS DEFINITION: "class and_c"

```
(4151) 6 Call: c([],atm(instance),_518913)

(4151) 6 Fail: c([],atm(instance),_518913)

(4150) 5 Back to: inst_create([],_518892)

(4150) 5 Fail: inst_create([],_518892)

(4152) 5 Call: top_lev_mes_trans([],_518892)

(4153) 6 Call: c([],atm(_6968),_518913)

(4153) 6 Fail: c([],atm(_6968),_518913)

(4152) 5 Back to: top_lev_mes_trans([],_518892)

(4152) 5 Fail: top_lev_mes_trans([],_518892)

(4145) 4 Fail: oop_stmt([],_518892)

(4144) 3 Back to: oop([],_157)

(4144) 3 Exit: oop([],[])

(3629) 2 Exit: oop([atm(class),atm(and_c),atm(has),a
tm(super_class),atm(circuit),spsym(;),atm(methods),atm(
in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is
),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())
,spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsy
m(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),s
psym(.),atm(end),atm(and_c),spsym(.)],[])
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
   (8241) 7 Call: inst_create([atm(instance),atm(and1),
atm(isa),atm(and_c),spsym(.)],_520763)

   (8242) 8 Call: c([atm(instance),atm(and1),atm(isa),a
tm(and_c),spsym(.)],atm(instance),_520795)

   (8242) 8 Exit: c([atm(instance),atm(and1),atm(isa),a
tm(and_c),spsym(.)],atm(instance),[atm(and1),atm(isa),a
tm(and_c),spsym(.)])

   (8243) 8 Call: inst_name(_12482,_520796,[atm(and1),a
tm(isa),atm(and_c),spsym(.)],_520797)

   (8244) 9 Call: c([atm(and1),atm(isa),atm(and_c),spsy
m(.)],atm(_12482),_520797)

   (8244) 9 Exit: c([atm(and1),atm(isa),atm(and_c),spsy
m(.)],atm(and1),[atm(isa),atm(and_c),spsym(.)])

   (8245) 9 Call: inst(and1,_12488)

   (8245) 9 Fail: inst(and1,_12488)

   (8246) 9 Call: _12487=0

   (8246) 9 Exit: 0=0

   (8243) 8 Exit: inst_name(and1,0,[atm(and1),atm(isa),
atm(and_c),spsym(.)],[atm(isa),atm(and_c),spsym(.)])

   (8247) 8 Call: c([atm(isa),atm(and_c),spsym(.)],atm(
isa),_520798)

   (8247) 8 Exit: c([atm(isa),atm(and_c),spsym(.)],atm(
isa),[atm(and_c),spsym(.)])

   (8248) 8 Call: class(_12483,_520799,[atm(and_c),spsy
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
m(.)],_520800)

   (8249) 9 Call: c([atm(and_c),spsym(.)],atm(_12483),_
520800)

   (8249) 9 Exit: c([atm(and_c),spsym(.)],atm(and_c),[s
psym(.)])

   (8250) 9 Call: not classtree(_12499,and_c)

   (8251) 9 Call: classtree(_12499,and_c)

   (8251) 9 Exit: classtree(circuit,and_c)

   (8250) 9 Back to: not classtree(_12499,and_c)

   (8250) 9 Fail: not classtree(_12499,and_c)

   (8252) 9 Call: _12498=0

   (8252) 9 Exit: 0=0

   (8248) 8 Exit: class(and_c,0,[atm(and_c),spsym(.)],[
spsym(.)])

   (8253) 8 Call: c([spsym(.)],spsym(.),_520763)

   (8253) 8 Exit: c([spsym(.)],spsym(.),[])

   (8254) 8 Call: 0=0

   (8254) 8 Exit: 0=0

   (8255) 8 Call: 0=0
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8255) 8 Exit: 0=0

(8256) 8 Call: assert(inst(and1,and_c))

(8256) 8 Exit: assert(inst(and1,and_c))

(8257) 8 Call: get_c_vars(and1,and_c)

(8258) 9 Call: c_var_val(and_c,_12514,_12515)

(8258) 9 Fail: c_var_val(and_c,_12514,_12515)

(8260) 9 Call: and_c=root

(8260) 9 Fail: and_c=root

(8261) 0 Call: circuit=root

(8261) 0 Fail: circuit=root

(8262) 0 Call: root=root

(8262) 0 Exit: root=root

(8257) 8 Exit: get_c_vars(and1,and_c)

(8263) 8 Call: get_i_vars(and1,and_c)

(8264) 9 Call: i_var_decl(and_c,_12577)

(8264) 9 Fail: i_var_decl(and_c,_12577)

(8266) 9 Call: i_var_val(and1,_Id,_12580)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8266) 9 Fail: i_var_val(and1,_Id,_12580)

(8268) 9 Call: i_var_val(and1,_Class,_12581)

(8268) 9 Fail: i_var_val(and1,_Class,_12581)

(8270) 9 Call: and_c=root

(8270) 9 Fail: and_c=root

(8271) 0 Call: i_var_val(and1,Temp,_12639)

(8271) 0 Fail: i_var_val(and1,Temp,_12639)

(8272) 0 Call: i_var_val(and1,Temp,_12676)

(8272) 0 Exit: i_var_val(and1,Temp,_12676)

(8273) 0 Call: i_var_val(and1,To_meth,_12676)

(8273) 0 Fail: i_var_val(and1,To_meth,_12676)

(8274) 0 Call: i_var_val(and1,Temp,_12713)

(8274) 0 Exit: i_var_val(and1,Temp,_12713)

(8275) 0 Call: i_var_val(and1,To_meth,_12713)

(8275) 0 Exit: i_var_val(and1,To_meth,_12713)

(8276) 0 Call: i_var_val(and1,To_inst,_12713)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8276) 0 Fail: i_var_val(and1,To_inst,_12713)

(8277) 0 Call: i_var_val(and1,Temp,_12750)

(8277) 0 Exit: i_var_val(and1,Temp,_12750)

(8278) 0 Call: i_var_val(and1,To_meth,_12750)

(8278) 0 Exit: i_var_val(and1,To_meth,_12750)

(8279) 0 Call: i_var_val(and1,To_inst,_12750)

(8279) 0 Exit: i_var_val(and1,To_inst,_12750)

(8280) 0 Call: i_var_val(and1,_Id,_12752)

(8280) 0 Exit: i_var_val(and1,_Id,and1)

(8281) 0 Call: i_var_val(and1,_Class,_12753)

(8281) 0 Exit: i_var_val(and1,_Class,and_c)

(8282) 0 Call: circuit=root

(8282) 0 Fail: circuit=root

(8283) 0 Call: i_var_val(and1,_Class,_12803)

(8283) 0 Exit: i_var_val(and1,_Class,and_c)

(8284) 0 Call: i_var_val(and1,_Id,_12803)

(8284) 0 Exit: i_var_val(and1,_Id,and1)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8285) 0 Call: i_var_val(and1,_Id,_12805)

(8285) 0 Exit: i_var_val(and1,_Id,and1)

(8286) 0 Call: i_var_val(and1,_Class,_12806)

(8286) 0 Exit: i_var_val(and1,_Class,and_c)

(8287) 0 Call: root=root

(8287) 0 Exit: root=root

(8263) 8 Exit: get_i_vars(and1,and_c)

(8288) 8 Call: get_meth_paths(and1,and_c)

(8289) 9 Call: meth_defn(and_c,_12846,_12847)

(8289) 9 Exit: meth_defn(and_c,in2,(in2(_12860,_1286
1):-get_ivar(_12860,Temp,_12862),_12863 is _12861/\_128
62,get_ivar(_12860,To_inst,_12864),get_ivar(_12860,To_m
eth,_12865),send_mesg([_12865,_12864,_12863],op(<<))))

(8290) 9 Call: not meth_path(and1,in2,and_c)

(8291) 9 Call: meth_path(and1,in2,and_c)

(8291) 9 Fail: meth_path(and1,in2,and_c)

(8290) 9 Exit: not meth_path(and1,in2,and_c)

(8292) 9 Call: assert(meth_path(and1,in2,and_c))

(8292) 9 Exit: assert(meth_path(and1,in2,and_c))
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8293) 9 Call: get_meth_paths(and1,and_c)

(8294) 10 Call: meth_defn(and_c,_12888,_12889)

(8294) 10 Exit: meth_defn(and_c,in2,(in2(_12902,_129
03):-get_ivar(_12902,Temp,_12904),_12905 is _12903/\_12
904,get_ivar(_12902,To_inst,_12906),get_ivar(_12902,To_
meth,_12907),send_mesg([_12907,_12906,_12905],op(<<))))

(8295) 10 Call: not meth_path(and1,in2,and_c)

(8296) 10 Call: meth_path(and1,in2,and_c)

(8296) 10 Exit: meth_path(and1,in2,and_c)

(8295) 10 Back to: not meth_path(and1,in2,and_c)

(8295) 10 Fail: not meth_path(and1,in2,and_c)

(8294) 10 Back to: meth_defn(and_c,_12888,_12889)

(8294) 10 Fail: meth_defn(and_c,_12888,_12889)

(8298) 10 Call: and_c=root

(8298) 10 Fail: and_c=root

(8299) 0 Call: meth_path(and1,in1,circuit)

(8299) 0 Fail: meth_path(and1,in1,circuit)

(8300) 0 Call: meth_path(and1,in1,circuit)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8300) 0 Exit: meth_path(and1,in1,circuit)

(8301) 0 Call: meth_path(and1,connect,circuit)

(8301) 0 Fail: meth_path(and1,connect,circuit)

(8302) 0 Call: meth_path(and1,in1,circuit)

(8302) 0 Exit: meth_path(and1,in1,circuit)

(8303) 0 Call: meth_path(and1,connect,circuit)

(8303) 0 Exit: meth_path(and1,connect,circuit)

(8304) 0 Call: circuit=root

(8304) 0 Fail: circuit=root

(8305) 0 Call: meth_path(and1,who_am_I,root)

(8305) 0 Fail: meth_path(and1,who_am_I,root)

(8306) 0 Call: meth_path(and1,who_am_I,root)

(8306) 0 Exit: meth_path(and1,who_am_I,root)

(8307) 0 Call: meth_path(and1,who_are_you,root)

(8307) 0 Fail: meth_path(and1,who_are_you,root)

(8308) 0 Call: meth_path(and1,who_am_I,root)

(8308) 0 Exit: meth_path(and1,who_am_I,root)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8309) 0 Call: meth_path(and1,who_are_you,root)
(8309) 0 Exit: meth_path(and1,who_are_you,root)
(8310) 0 Call: meth_path(and1,broadcast,root)
(8310) 0 Fail: meth_path(and1,broadcast,root)
(8311) 0 Call: meth_path(and1,who_am_I,root)
(8311) 0 Exit: meth_path(and1,who_am_I,root)
(8312) 0 Call: meth_path(and1,who_are_you,root)
(8312) 0 Exit: meth_path(and1,who_are_you,root)
(8313) 0 Call: meth_path(and1,broadcast,root)
(8313) 0 Exit: meth_path(and1,broadcast,root)
(8314) 0 Call: meth_path(and1,erase_self,root)
(8314) 0 Fail: meth_path(and1,erase_self,root)
(8315) 0 Call: meth_path(and1,who_am_I,root)
(8315) 0 Exit: meth_path(and1,who_am_I,root)
(8316) 0 Call: meth_path(and1,who_are_you,root)
(8316) 0 Exit: meth_path(and1,who_are_you,root)
(8317) 0 Call: meth_path(and1,broadcast,root)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(8317) 0 Exit: meth_path(and1,broadcast,root)

(8318) 0 Call: meth_path(and1,erase_self,root)

(8318) 0 Exit: meth_path(and1,erase_self,root)

(8319) 0 Call: root=root

(8319) 0 Exit: root=root

(8293) 9 Exit: get_meth_paths(and1,and_c)

(8288) 8 Exit: get_meth_paths(and1,and_c)

(8241) 7 Exit: inst_create([atm(instance),atm(and1),
atm(isa),atm(and_c),spsym(.)],[])

(8236) 6 Exit: oop_stmt([atm(instance),atm(and1),atm
(isa),atm(and_c),spsym(.)],[])

(8320) 6 Call: oop([],_157)

(8321) 7 Call: oop_stmt([],_522060)

(8322) 8 Call: class_defn([],_522060)

(8323) 9 Call: c([],atm(class),_522081)

(8323) 9 Fail: c([],atm(class),_522081)

(8322) 8 Back to: class_defn([],_522060)

(8322) 8 Fail: class_defn([],_522060)
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
No leashing.

   (8326) 8 Call: inst_create([],_522060)

   (8327) 9 Call: c([],atm(instance),_522100)

   (8327) 9 Fail: c([],atm(instance),_522100)

   (8326) 8 Back to: inst_create([],_522060)

   (8326) 8 Fail: inst_create([],_522060)

   (8321) 7 Back to: oop_stmt([],_522060)

   (8328) 8 Call: top_lev_mes_trans([],_522060)

   (8329) 9 Call: c([],atm(_13181),_522081)

   (8329) 9 Fail: c([],atm(_13181),_522081)

   (8328) 8 Back to: top_lev_mes_trans([],_522060)

   (8328) 8 Fail: top_lev_mes_trans([],_522060)

   (8321) 7 Fail: oop_stmt([],_522060)

   (8320) 6 Back to: oop([],_157)

   (8320) 6 Exit: oop([],[])

   (8235) 5 Exit: oop([atm(instance),atm(and1),atm(isa)
,atm(and_c),spsym(.)],[])
```

APPENDIX J TRACE OF "instance andl isa and_c."

```
(7472) 4 Exit: oop([atm(class),atm(out_c),atm(has),a
tm(super_class),atm(circuit),spsym(;),atm(methods),atm(
in2),spsym(()),vbl(X),spsym()),spsym(:-),atm(nl),spsym(,
),atm(nl),spsym(,),atm(write),spsym(()),spsym(:),vbl(_Id
),spsym()),spsym(,),atm(write),spsym(()),spsym('),qatom(
 is ),spsym('),spsym()),spsym(,),atm(write),spsym(()),vb
l(X),spsym()),spsym(,),atm(nl),spsym(,),atm(nl),spsym(,
),atm(nl),spsym(.),atm(end),atm(out_c),spsym(.),atm(ins
tance),atm(andl),atm(isa),atm(and_c),spsym(.)],[])


(6983) 3 Exit: oop([atm(class),atm(or_c),atm(has),at
m(super_class),atm(circuit),spsym(;),atm(methods),atm(i
n2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is)
,spsym(()),vbl(X),spsym(\/),spsym(:),vbl(Temp),spsym()),
spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsym
(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),sp
sym(.),atm(end),atm(or_c),spsym(.),atm(class),atm(out_c
),atm(has),atm(super_class),atm(circuit),spsym(;),atm(m
ethods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),atm
(nl),spsym(,),atm(nl),spsym(,),atm(write),spsym(()),spsy
m(:),vbl(_Id),spsym()),spsym(,),atm(write),spsym(()),sps
ym('),qatom( is ),spsym('),spsym()),spsym(,),atm(write)
,spsym(()),vbl(X),spsym()),spsym(,),atm(nl),spsym(,),atm
(nl),spsym(,),atm(nl),spsym(.),atm(end),atm(out_c),spsy
m(.),atm(instance),atm(andl),atm(isa),atm(and_c),spsym(


(6470) 2 Exit: oop([atm(class),atm(and_c),atm(has),a
tm(super_class),atm(circuit),spsym(;),atm(methods),atm(
in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is
),spsym(()),vbl(X),spsym(/\),spsym(:),vbl(Temp),spsym())
,spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),spsy
m(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),s
psym(.),atm(end),atm(and_c),spsym(.),atm(class),atm(or_
c),atm(has),atm(super_class),atm(circuit),spsym(;),atm(
methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),vb
l(Out),atm(is),spsym(()),vbl(X),spsym(\/),spsym(:),vbl(T
emp),spsym()),spsym(,),spsym(()),spsym(:),vbl(To_inst),s
psym(<<),spsym(:),vbl(To_meth),spsym(()),vbl(Out),spsym(
)),spsym()),spsym(.),atm(end),atm(or_c),spsym(.),atm(cl
ass),atm(out_c),atm(has),atm(super_class),atm(circuit),
spsym(;),atm(methods),atm(in2),spsym(()),vbl(X),spsym())
,spsym(:-),atm(nl),spsym(,),atm(nl),spsym(,),atm(write)
,spsym(()),spsym(:),vbl(_Id),spsym()),spsym(,),atm(write
),spsym(()),spsym('),qatom( is ),spsym('),spsym()),spsym
(,),atm(write),spsym(()),vbl(X),spsym()),spsym(,),atm(nl
),spsym(,),atm(nl),spsym(,),atm(nl),spsym(.),atm(end),a
tm(out_c),spsym(.),atm(instance),atm(andl),atm(isa),atm
```

APPENDIX J TRACE OF "instance and1 isa and_c."

```
(and_c),spsym(.)],[])

   (5932) 1 Exit: oop([atm(class),atm(circuit),atm(has)
,atm(super_class),atm(root),spsym(;),atm(instance_var),
vbl(Temp),spsym(,),vbl(To_meth),spsym(,),vbl(To_inst),s
psym(;),atm(methods),atm(in1),spsym(()),vbl(X),spsym()),
spsym(:-),vbl(Temp),spsym(=:),vbl(X),spsym(.),atm(conne
ct),spsym(()),vbl(Inst),spsym(,),vbl(Meth),spsym())),spsy
m(:-),vbl(To_inst),spsym(=:),vbl(Inst),spsym(,),vbl(To_
meth),spsym(=:),vbl(Meth),spsym(.),atm(end),atm(circuit
),spsym(.),atm(class),atm(and_c),atm(has),atm(super_cla
ss),atm(circuit),spsym(;),atm(methods),atm(in2),spsym((
),vbl(X),spsym()),spsym(:-),vbl(Out),atm(is),spsym(()),v
bl(X),spsym(/\),spsym(:),vbl(Temp),spsym()),spsym(,),sp
sym(()),spsym(:),vbl(To_inst),spsym(<<),spsym(:),vbl(To_
meth),spsym(()),vbl(Out),spsym()),spsym()),spsym(.),atm(
end),atm(and_c),spsym(.),atm(class),atm(or_c),atm(has),
atm(super_class),atm(circuit),spsym(;),atm(methods),atm
(in2),spsym(()),vbl(X),spsym()),spsym(:-),vbl(Out),atm(i
s),spsym(()),vbl(X),spsym(\/),spsym(:),vbl(Temp),spsym()
),spsym(,),spsym(()),spsym(:),vbl(To_inst),spsym(<<),sps
ym(:),vbl(To_meth),spsym(()),vbl(Out),spsym()),spsym()),
spsym(.),atm(end),atm(or_c),spsym(.),atm(class),atm(out
_c),atm(has),atm(super_class),atm(circuit),spsym(;),atm
(methods),atm(in2),spsym(()),vbl(X),spsym()),spsym(:-),a
tm(nl),spsym(,),atm(nl),spsym(,),atm(write),spsym(()),sp
sym(:),vbl(_Id),spsym()),spsym(,),atm(write),spsym(()),s
psym('),qatom( is ),spsym('),spsym()),spsym(,),atm(writ
e),spsym(()),vbl(X),spsym()),spsym(,),atm(nl),spsym(,),a
tm(nl),spsym(,),atm(nl),spsym(.),atm(end),atm(out_c),sp
sym(.),atm(instance),atm(and1),atm(isa),atm(and_c),spsy
m(.)],[])
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
(11294) 16 Call: oop_stmt([atm(and2),spsym(<<),atm(i
n2),spsym(()),num(1),spsym()),spsym(.)],_524655)

(11295) 17 Call: class_defn([atm(and2),spsym(<<),atm
(in2),spsym(()),num(1),spsym()),spsym(.)],_524655)

(11296) 18 Call: c([atm(and2),spsym(<<),atm(in2),sps
ym(()),num(1),spsym()),spsym(.)],atm(class),_524676)

(11296) 18 Fail: c([atm(and2),spsym(<<),atm(in2),sps
ym(()),num(1),spsym()),spsym(.)],atm(class),_524676)

(11295) 17 Back to: class_defn([atm(and2),spsym(<<),
atm(in2),spsym(()),num(1),spsym()),spsym(.)],_524655)

(11295) 17 Fail: class_defn([atm(and2),spsym(<<),atm
(in2),spsym(()),num(1),spsym()),spsym(.)],_524655)

(11297) 17 Call: inst_create([atm(and2),spsym(<<),at
m(in2),spsym(()),num(1),spsym()),spsym(.)],_524655)

(11298) 18 Call: c([atm(and2),spsym(<<),atm(in2),sps
ym(()),num(1),spsym()),spsym(.)],atm(instance),_524676)

(11298) 18 Fail: c([atm(and2),spsym(<<),atm(in2),sps
ym(()),num(1),spsym()),spsym(.)],atm(instance),_524676)

(11297) 17 Back to: inst_create([atm(and2),spsym(<<)
,atm(in2),spsym(()),num(1),spsym()),spsym(.)],_524655)

(11297) 17 Fail: inst_create([atm(and2),spsym(<<),at
m(in2),spsym(()),num(1),spsym()),spsym(.)],_524655)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
No leashing.

   (11301) 17 Call: top_lev_mes_trans([atm(and2),spsym(
<<),atm(in2),spsym(()),num(1),spsym()),spsym(.)],_524655
)

   (11302) 18 Call: c([atm(and2),spsym(<<),atm(in2),sps
ym(()),num(1),spsym()),spsym(.)],atm(_18637),_524695)

   (11302) 18 Exit: c([atm(and2),spsym(<<),atm(in2),sps
ym(()),num(1),spsym()),spsym(.)],atm(and2),[spsym(<<),at
m(in2),spsym(()),num(1),spsym()),spsym(.)])

   (11303) 18 Call: mesgop(_18638,[spsym(<<),atm(in2),s
psym(()),num(1),spsym()),spsym(.)],_524696)

   (11304) 19 Call: c([spsym(<<),atm(in2),spsym(()),num(
1),spsym()),spsym(.)],spsym(<<),_524696)

   (11304) 19 Exit: c([spsym(<<),atm(in2),spsym(()),num(
1),spsym()),spsym(.)],spsym(<<),[atm(in2),spsym(()),num(
1),spsym()),spsym(.)])

   (11303) 18 Exit: mesgop(<<,[spsym(<<),atm(in2),spsym
(()),num(1),spsym()),spsym(.)],[atm(in2),spsym(()),num(1)
,spsym()),spsym(.)])

   (11305) 18 Call: mesg(_18639,_524697,[atm(in2),spsym
(()),num(1),spsym()),spsym(.)],_524698)

   (11306) 19 Call: c([atm(in2),spsym(()),num(1),spsym()
),spsym(.)],atm(_18639),_524736)

   (11306) 19 Exit: c([atm(in2),spsym(()),num(1),spsym()
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
),spsym(.)],atm(in2),[spsym(()),num(1),spsym()),spsym(.)
])


   (11307) 19 Call: args(_524697,[spsym(()),num(1),spsym
()),spsym(.)],_524698)


   (11308) 20 Call: c([spsym(()),num(1),spsym()),spsym(.
)],spsym(()),_524756)


   (11308) 20 Exit: c([spsym(()),num(1),spsym()),spsym(.
)],spsym(()),[num(1),spsym()),spsym(.)])


   (11309) 20 Call: term(_524757,[num(1),spsym()),spsym
(.)],_524758)


   (11310) 21 Call: pred(_18654,[num(1),spsym()),spsym(


   (11311) 22 Call: c([num(1),spsym()),spsym(.)],atm(_1
8668),_524796)


   (11311) 22 Fail: c([num(1),spsym()),spsym(.)],atm(_1
8668),_524796)


   (11310) 21 Back to: pred(_18654,[num(1),spsym()),sps
ym(.)],_18656)


   (11310) 21 Fail: pred(_18654,[num(1),spsym()),spsym(


   (11312) 21 Call: const(_18654,[num(1),spsym()),spsym
(.)],_18656)


   (11313) 22 Call: c([num(1),spsym()),spsym(.)],num(_1
8671),_18656)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
   (11313) 22 Exit: c([num(1),spsym()),spsym(.)],num(1)
,[spsym()),spsym(.)])

   (11312) 21 Exit: const([1],[num(1),spsym()),spsym(.)
],[spsym()),spsym(.)])

   (11309) 20 Exit: term([1],[num(1),spsym()),spsym(.)]
,[spsym()),spsym(.)])

   (11314) 20 Call: rest_args(_524759,[spsym()),spsym(.
)],_524760)

   (11315) 21 Call: c([spsym()),spsym(.)],spsym(,),_524
831)

   (11315) 21 Fail: c([spsym()),spsym(.)],spsym(,),_524
831)

   (11314) 20 Back to: rest_args(_524759,[spsym()),spsy
m(.)],_524760)

   (11316) 21 Call: true

   (11316) 21 Exit: true

   (11314) 20 Exit: rest_args([],[spsym()),spsym(.)],[s
psym()),spsym(.)])

   (11317) 20 Call: c([spsym()),spsym(.)],spsym()),_524
698)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
   (11317) 20 Exit: c([spsym()),spsym(.)],spsym()),[sps
ym(.)])

   (11318) 20 Call: append([1],[],_524697)

   (11319) 21 Call: append([],[],_18688)

   (11319) 21 Exit: append([],[],[])

   (11318) 20 Exit: append([1],[],[1])

   (11307) 19 Exit: args([1],[spsym((),num(1),spsym()),
spsym(.)],[spsym(.)])

   (11305) 18 Exit: mesg(in2,[1],[atm(in2),spsym((),num
(1),spsym()),spsym(.)],[spsym(.)])

   (11320) 18 Call: c([spsym(.)],spsym(.),_524655)

   (11320) 18 Exit: c([spsym(.)],spsym(.),[])

   (11321) 18 Call: strip_args([1],_18640)

   (11321) 18 Exit: strip_args([1],[1])

   (11322) 18 Call: send_mesg([in2,and2,1],op(<<))

   (11323) 19 Call: meth_path(and2,in2,_524885)

   (11323) 19 Exit: meth_path(and2,in2,and_c)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
(11324) 19 Call: meth_defn(and_c,in2,_524886)

(11324) 19 Exit: meth_defn(and_c,in2,(in2(_18705,_18
706):-get_ivar(_18705,Temp,_18707),_18708 is _18706/\_1
8707,get_ivar(_18705,To_inst,_18709),get_ivar(_18705,To
_meth,_18710),send_mesg([_18710,_18709,_18708],op(<<)))
)

(11325) 19 Call: asserta((in2(_18705,_18706):-get_iv
ar(_18705,Temp,_18707),_18708 is _18706/\_18707,get_iva
r(_18705,To_inst,_18709),get_ivar(_18705,To_meth,_18710
),send_mesg([_18710,_18709,_18708],op(<<))))

(11325) 19 Exit: asserta((in2(_18705,_18706):-get_iv
ar(_18705,Temp,_18707),_18708 is _18706/\_18707,get_iva
r(_18705,To_inst,_18709),get_ivar(_18705,To_meth,_18710
),send_mesg([_18710,_18709,_18708],op(<<))))

(11326) 19 Call: (in2(_18705,_18706):-get_ivar(_1870
5,Temp,_18707),_18708 is _18706/\_18707,get_ivar(_18705
,To_inst,_18709),get_ivar(_18705,To_meth,_18710),send_m
esg([_18710,_18709,_18708],op(<<)))=..[_18702,_18703,_1
8704]

(11326) 19 Exit: (in2(_18705,_18706):-get_ivar(_1870
5,Temp,_18707),_18708 is _18706/\_18707,get_ivar(_18705
,To_inst,_18709),get_ivar(_18705,To_meth,_18710),send_m
esg([_18710,_18709,_18708],op(<<)))=..[:-,in2(_18705,_1
8706),(get_ivar(_18705,Temp,_18707),_18708 is _18706/\_
18707,get_ivar(_18705,To_inst,_18709),get_ivar(_18705,T
o_meth,_18710),send_mesg([_18710,_18709,_18708],op(<<))
)]

(11327) 19 Call: in2(_18705,_18706)=..[in2,and2,1]

(11327) 19 Exit: in2(and2,1)=..[in2,and2,1]
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
(11328) 19 Call: in2(and2,1)

(11329) 20 Call: get_ivar(and2,Temp,_18730)

(11330) 21 Call: i_var_val(and2,Temp,_18730)

(11330) 21 Exit: i_var_val(and2,Temp,1)

(11331) 21 Call: nonvar(1)

(11331) 21 Exit: nonvar(1)

(11329) 20 Exit: get_ivar(and2,Temp,1)

(11332) 20 Call: _18731 is 1/\1

(11332) 20 Exit: 1 is 1/\1

(11333) 20 Call: get_ivar(and2,To_inst,_18732)

(11334) 21 Call: i_var_val(and2,To_inst,_18732)

(11334) 21 Exit: i_var_val(and2,To_inst,or1)

(11335) 21 Call: nonvar(or1)

(11335) 21 Exit: nonvar(or1)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
   (11333) 20 Exit: get_ivar(and2,To_inst,or1)

   (11336) 20 Call: get_ivar(and2,To_meth,_18733)

   (11337) 21 Call: i_var_val(and2,To_meth,_18733)

   (11337) 21 Exit: i_var_val(and2,To_meth,in2)

   (11338) 21 Call: nonvar(in2)

   (11338) 21 Exit: nonvar(in2)

   (11336) 20 Exit: get_ivar(and2,To_meth,in2)

   (11339) 20 Call: send_mesg([in2,or1,1],op(<<))

   (11340) 21 Call: meth_path(or1,in2,_524993)

   (11340) 21 Exit: meth_path(or1,in2,or_c)

   (11341) 21 Call: meth_defn(or_c,in2,_524994)

   (11341) 21 Exit: meth_defn(or_c,in2,(in2(_18745,_187
46):-get_ivar(_18745,Temp,_18747),_18748 is _18746\/_18
747,get_ivar(_18745,To_inst,_18749),get_ivar(_18745,To_
meth,_18750),send_mesg([_18750,_18749,_18748],op(<<))))

   (11342) 21 Call: asserta((in2(_18745,_18746):-get_iv
ar(_18745,Temp,_18747),_18748 is _18746\/_18747,get_iva
r(_18745,To_inst,_18749),get_ivar(_18745,To_meth,_18750
),send_mesg([_18750,_18749,_18748],op(<<))))
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

(11342) 21 Exit: asserta((in2(_18745,_18746):-get_ivar(_18745,Temp,_18747),_18748 is _18746\/_18747,get_ivar(_18745,To_inst,_18749),get_ivar(_18745,To_meth,_18750),send_mesg([_18750,_18749,_18748],op(<<))))

(11343) 21 Call: (in2(_18745,_18746):-get_ivar(_18745,Temp,_18747),_18748 is _18746\/_18747,get_ivar(_18745,To_inst,_18749),get_ivar(_18745,To_meth,_18750),send_mesg([_18750,_18749,_18748],op(<<)))=..[_18740,_18741,_18742]

(11343) 21 Exit: (in2(_18745,_18746):-get_ivar(_18745,Temp,_18747),_18748 is _18746\/_18747,get_ivar(_18745,To_inst,_18749),get_ivar(_18745,To_meth,_18750),send_mesg([_18750,_18749,_18748],op(<<)))=..[:-,in2(_18745,_18746),(get_ivar(_18745,Temp,_18747),_18748 is _18746\/_18747,get_ivar(_18745,To_inst,_18749),get_ivar(_18745,To_meth,_18750),send_mesg([_18750,_18749,_18748],op(<<)))]

(11344) 21 Call: in2(_18745,_18746)=..[in2,or1,1]

(11344) 21 Exit: in2(or1,1)=..[in2,or1,1]

(11345) 21 Call: in2(or1,1)

(11346) 22 Call: get_ivar(or1,Temp,_18770)

(11347) 23 Call: i_var_val(or1,Temp,_18770)

(11347) 23 Exit: i_var_val(or1,Temp,0)

(11348) 23 Call: nonvar(0)

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
(11348) 23 Exit: nonvar(0)

(11346) 22 Exit: get_ivar(or1,Temp,0)

(11349) 22 Call: _18771 is 1\/0

(11349) 22 Exit: 1 is 1\/0

(11350) 22 Call: get_ivar(or1,To_inst,_18772)

(11351) 23 Call: i_var_val(or1,To_inst,_18772)

(11351) 23 Exit: i_var_val(or1,To_inst,out1)

(11352) 23 Call: nonvar(out1)

(11352) 23 Exit: nonvar(out1)

(11350) 22 Exit: get_ivar(or1,To_inst,out1)

(11353) 22 Call: get_ivar(or1,To_meth,_18773)

(11354) 23 Call: i_var_val(or1,To_meth,_18773)

(11354) 23 Exit: i_var_val(or1,To_meth,in2)

(11355) 23 Call: nonvar(in2)

(11355) 23 Exit: nonvar(in2)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
(11353) 22 Exit: get_ivar(or1,To_meth,in2)

(11356) 22 Call: send_mesg([in2,out1,1],op(<<))

(11357) 23 Call: meth_path(out1,in2,_525101)

(11357) 23 Exit: meth_path(out1,in2,out_c)

(11358) 23 Call: meth_defn(out_c,in2,_525102)

(11358) 23 Exit: meth_defn(out_c,in2,(in2(_18785,_18
786):-nl,nl,get_ivar(_18785,_Id,_18787),write(_18787),w
rite( is ),write(_18786),nl,nl,nl))

(11359) 23 Call: asserta((in2(_18785,_18786):-nl,nl,
get_ivar(_18785,_Id,_18787),write(_18787),write( is ),w
rite(_18786),nl,nl,nl))

(11359) 23 Exit: asserta((in2(_18785,_18786):-nl,nl,
get_ivar(_18785,_Id,_18787),write(_18787),write( is ),w
rite(_18786),nl,nl,nl))

(11360) 23 Call: (in2(_18785,_18786):-nl,nl,get_ivar
(_18785,_Id,_18787),write(_18787),write( is ),write(_18
786),nl,nl,nl)=..[_18780,_18781,_18782]

(11360) 23 Exit: (in2(_18785,_18786):-nl,nl,get_ivar
(_18785,_Id,_18787),write(_18787),write( is ),write(_18
786),nl,nl,nl)=..[:-,in2(_18785,_18786),(nl,nl,get_ivar
(_18785,_Id,_18787),write(_18787),write( is ),write(_18
786),nl,nl,nl)]
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
   (11361) 23 Call: in2(_18785,_18786)=..[in2,out1,1]

   (11361) 23 Exit: in2(out1,1)=..[in2,out1,1]

   (11362) 23 Call: in2(out1,1)




   (11365) 24 Call: get_ivar(out1,_Id,_525161)

   (11366) 25 Call: i_var_val(out1,_Id,_525161)

   (11366) 25 Exit: i_var_val(out1,_Id,out1)

   (11367) 25 Call: nonvar(out1)

   (11367) 25 Exit: nonvar(out1)

   (11365) 24 Exit: get_ivar(out1,_Id,out1)

out1 is 1




   (11362) 23 Exit: in2(out1,1)
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
   (11374) 23 Call: retract((in2(out1,1):-n1,n1,get_iva
r(out1,_Id,_18787),write(_18787),write( is ),write(1),n
1,n1,n1))

   (11374) 23 Exit: retract((in2(out1,1):-n1,n1,get_iva
r(out1,_Id,_18787),write(_18787),write( is ),write(1),n
1,n1,n1))

   (11356) 22 Exit: send_mesg([in2,out1,1],op(<<))

   (11345) 21 Exit: in2(or1,1)

   (11375) 21 Call: retract((in2(or1,1):-get_ivar(or1,T
emp,_18747),_18748 is 1\/_18747,get_ivar(or1,To_inst,_1
8749),get_ivar(or1,To_meth,_18750),send_mesg([_18750,_1
8749,_18748],op(<<))))

   (11375) 21 Exit: retract((in2(or1,1):-get_ivar(or1,T
emp,_18747),_18748 is 1\/_18747,get_ivar(or1,To_inst,_1
8749),get_ivar(or1,To_meth,_18750),send_mesg([_18750,_1
8749,_18748],op(<<))))

   (11339) 20 Exit: send_mesg([in2,or1,1],op(<<))

   (11328) 19 Exit: in2(and2,1)

   (11376) 19 Call: retract((in2(and2,1):-get_ivar(and2
,Temp,_18707),_18708 is 1/\_18707,get_ivar(and2,To_inst
,_18709),get_ivar(and2,To_meth,_18710),send_mesg([_1871
0,_18709,_18708],op(<<))))

   (11376) 19 Exit: retract((in2(and2,1):-get_ivar(and2
,Temp,_18707),_18708 is 1/\_18707,get_ivar(and2,To_inst
,_18709),get_ivar(and2,To_meth,_18710),send_mesg([_1871
```

APPENDIX K TRACE OF MESSAGE SENDING PARSE:
"and2 << in2(1)."

```
0,_18709,_18708],op(<<))))

   (11322) 18 Exit: send_mesg([in2,and2,1],op(<<))

   (11301) 17 Exit: top_lev_mes_trans([atm(and2),spsym(
<<),atm(in2),spsym(()),num(1),spsym()),spsym(.)],[])

   (11294) 16 Exit: oop_stmt([atm(and2),spsym(<<),atm(i
n2),spsym(()),num(1),spsym()),spsym(.)],[])
```

AN OBJECT-ORIENTED EXTENSION OF PROLOG

by

MICHAEL KEVIN KONDOR

B. S., Purdue University, 1986

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

ABSTRACT

This thesis defines a prototype of object-oriented Prolog called OOP. The design is based largely on the early language SPOOL by IBM Japan. OOP incorporates simple inheritance, class and instance variables, and method definition using Prolog clauses. OOP is implemented using Definite Clause Grammars to translate the OOP source to a target C-Prolog program. The translation to C-Prolog and execution of the resulting program is illustrated using a small logic circuit simulation example.