SAL and XML (XSAL)

Sam Owre[1], David Dill[2], and N. Shankar[1]

Draft Report
March, 2001

This report was developed and is maintained by SRI International. SRI's part of the SAL project is funded by DARPA/AFRL contract number F33615-00-C-3043.

[1] Computer Science Laboratory, SRI International, Menlo Park, California
[2] Stanford University, Stanford, California

Abstract

The Symbolic Analysis Laboratory (SAL) is a framework for combining different tools for abstraction, program analysis, theorem proving, and model checking toward the calculation of properties (symbolic analysis) of transition systems. The language is described in the draft technical report at `http://www.csl.sri.com/projects/cara/reports/language-report.ps.gz`. In order to be as independent as possible of the choice of operating system or implemetation language for tools, XML was chosen as the language for describing the abstract syntax.

Chapter 1

Introduction

This is a draft report on XSAL: the XML translation of SAL specifications. It is expected that tools written for SAL will communicate using XSAL, which is much easier to generate and parse than the SAL language itself. This report describes the correlation between the SAL and XSAL languages. It is very much a work in progress—the tables are quite out-of-date and incomplete; the SAL DTD and the examples are much closer to finished.

Chapter 2 describes the correlation between SAL and XSAL using tables indexed by the nonterminals of the SAL language. Chapter 3 describes the SAL DTD, and Chapter 4 gives an example.

Chapter 2

SAL and XSAL

The following tables provide a correlation between the concrete syntax from the SAL language report and the abstract syntax defined by the SAL Document Type Definition (DTD). In some sense, the concrete syntax is simply the prettyprinted version of the DTD.

Here are some general principles:

- Attributes are used to keep information that is purely syntactical
- Restrict use of optionals to the ends of entities, e.g., in *ConstantDeclarations* there is an optional *VarDecls*, in the DTD the VARDECLS entity is not optional, but it may be empty.
- As much as possible, retain the names of the nonterminals in the concrete syntax
- The DTD is positional, generally the order of subentities in the DTD should reflect the order in the concrete syntax.

For example, there is no difference in semantics between the constant declarations

```
a, b: INTEGER;
c: INTEGER;
```

and

```
a: INTEGER;
b, c: INTEGER;
```

These are represented in the SAL DTD as:

```
<CONSTANTDECLARATION CHAIN="YES">
   <IDENTIFIER>a</IDENTIFIER>
   <VARDECLS></VARDECLS>
   <TYPENAME>INTEGER</TYPENAME>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION CHAIN="NO">
   <IDENTIFIER>b</IDENTIFIER>
   <VARDECLS></VARDECLS>
   <TYPENAME>INTEGER</TYPENAME>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION CHAIN="NO">
   <IDENTIFIER>c</IDENTIFIER>
   <VARDECLS></VARDECLS>
   <TYPENAME>INTEGER</TYPENAME>
</CONSTANTDECLARATION>
```

The second example would be the same, except that the first CHAIN would be "NO", and the second would be "YES".

Another example is infix applications, which don't directly exist in the DTD. Instead, there is an INFIX attribute that is set to "YES" if it is an infix operation. Thus A AND B is treated as AND(A, B) and translates to

```
<APPLICATION INFIX="YES">
  <NAMEEXPR>AND</NAMEEXPR>
  <TUPLELITERAL>
    <NAMEEXPR>A</NAMEEXPR>
    <NAMEEXPR>B</NAMEEXPR>
  </TUPLELITERAL>
</APPLICATION>
```

Contexts

Nonterminal	**SAL**	**XML**
Context	*Identifier* [{*Parameters*}] : CONTEXT = *ContextBody*	<CONTEXT> *Identifier Parameters ContextBody* </CONTEXT>
Parameters	[*TypeDecls*] TYPE ; $\{VarDecls\}^*_{,}$	<PARAMETERS> *TypeDecls VarDecls* </PARAMETERS>
ContextBody	BEGIN *Declarations* END	<CONTEXTBODY> $\{Declarations\}^+$ </CONTEXTBODY>
Declarations	$\{Declaration\ ;\}^+$	$\{Declaration\}^+$
Declaration	*TypeDeclaration* |*ConstantDeclaration* |*AssertionDeclaration* |*ModuleDeclaration* |*ContextDeclaration*	*TypeDeclaration* |*ConstantDeclaration* |*AssertionDeclaration* |*ModuleDeclaration* |*ContextDeclaration*
TypeDeclaration	*Identifier* : TYPE [= *TypeDef*]	<TYPEDECLARATION> *Identifier* [*TypeDef*] </TYPEDECLARATION>
TypeDef	*ScalarType* |*DataType* |*Type*	*ScalarType* |*DataType* |*Type*
ScalarType	{$\{Identifier\}^+_{,}$ }	<SCALARTYPE> $\{Identifier\}^+$ </SCALARTYPE>
DataType	DATATYPE *Constructors* END	<DATATYPE> *Constructors* </DATATYPE>
Constructors	$\{Constructor\}^+_{,}$	$\{Constructor\}^+$
Constructor	*Identifier* [(*Accessors*)]	<CONSTRUCTOR> *Identifier Accessors* </CONSTRUCTOR>
Accessors	$\{Accessor\}^+_{,}$	$\{Accessor\}^+$
Accessor	*Identifier* : *Type*	<ACCESSOR> *Identifier Type* </ACCESSOR>

Nonterminal	**SAL**	**XML**
ConstantDeclaration	*Identifier* [(*VarDecls*)] : *Type* [= *Expression*]	<CONSTANTDECLARATION> *Identifier* *Type* $\{\mathit{Expression}\}^{?}$ </CONSTANTDECLARATION>
AssertionDeclaration	*Identifier* : *AssertionForm* = *AssertionExpression*	<ASSERTIONDECLARATION> *Identifier* *AssertionForm* *AssertionExpression* </ASSERTIONDECLARATION>
ModuleDeclaration	*Identifier* [[*VarDecls*]] : MODULE = *Module*	<MODULEDECLARATION> *Identifier* $\{\mathit{VarDecl}\}^{*}$ *Module* </MODULEDECLARATION>
ContextDeclaration	*Identifier* : CONTEXT = *ContextName*	<CONTEXTDECLARATION> *Identifier* *ContextName* </CONTEXTDECLARATION>
ContextName	*Identifier* {*ActualParameters* }	<CONTEXTNAME> *Identifier* *ActualParameters* </CONTEXTNAME>

Types

Nonterminal	**SAL**	**XML**
Type	*BasicType* |*Name* |*Subrange* |*ArrayType* |*TupleType* |*FunctionType* |*RecordType*	*BasicType* |*Name* |*Subrange* |*ArrayType* |*TupleType* |*FunctionType* |*RecordType*
BasicType	BOOLEAN |REAL |INTEGER |NZINTEGER |NATURAL |NZREAL	<TYPENAME>BOOLEAN</TYPENAME> ...
Subrange	[*Bound* .. *Bound*]	<SUBRANGE> *Bound Bound* </SUBRANGE>
Bound	*Unbounded* |*Expression*	*Unbounded* |*Expression*
Unbounded	_	<NAMEEXPR>_</NAMEEXPR>
ArrayType	ARRAY *IndexType* OF *Type*	<ARRAYTYPE> *IndexType ElementType* </ARRAYTYPE>
IndexType	INTEGER |*Subrange* |*ScalarTypeName*	<TYPENAME>INTEGER</TYPENAME> |*Subrange* |*ScalarTypeName*
ScalarTypeName	*Name*	*Name*
TupleType	[$\{\mathit{Type}\}^{+}_{,}$]	<TUPLETYPE> $\{\mathit{Type}\}^{+}$ </TUPLETYPE>
FunctionType	[*Type* -> *Type*]	<FUNCTIONTYPE> *Type Type* </FUNCTIONTYPE>
RecordType	[# $\{\mathit{FieldDecl}\}^{+}_{,}$ #]	<RECORDTYPE> $\{\mathit{FieldDecl}\}^{+}$ </RECORDTYPE>
FieldDecl	*Identifier* : *Type*	<FIELDDECL> *Identifier Type* </FIELDDECL>

Expressions

Nonterminal	**SAL**	**XML**
Expression	*NameExpr* |*NextVariable* |*Numeral* |*Application* |*InfixApplication* |*ArraySelection* |*RecordSelection* |*TupleSelection* |*UpdateExpression* |*LambdaAbstraction* |*QuantifiedExpression* |*LetExpression* |*SetExpression* |*ArrayLiteral* |*RecordLiteral* |*TupleLiteral* |*Conditional* |*ParenExpression*	*NameExpr* |*NextVariable* |*Numeral* |*Application* |*InfixApplication* |*ArraySelection* |*RecordSelection* |*TupleSelection* |*UpdateExpression* |*LambdaAbstraction* |*QuantifiedExpression* |*LetExpression* |*SetExpression* |*ArrayLiteral* |*RecordLiteral* |*TupleLiteral* |*Conditional* |*ParenExpression*
NextVariable	*Identifier* '	<NEXTOPERATOR> <NAMEEXPR> *Identifier* </NAMEEXPR> </NEXTOPERATOR>
Application	*Expression Argument*	<APPLICATION> *Expression Argument* </APPLICATION>
Argument	($\{Expression\}^{+}_{,}$)	<TUPLELITERAL> $\{Expression\}^{*}$ </TUPLELITERAL>

Nonterminal	**SAL**	**XML**

InfixApplication	*Expression Identifier Expression*	`<APPLICATION INFIX="YES">`
		`<NAMEEXPR>`
		Identifier
		`</NAMEEXPR>`
		`<TUPLELITERAL>`
		Expression Expression
		`</TUPLELITERAL>`
		`</APPLICATION>`

ArraySelection	*Expression*`[`*Expression*`]`	`<ARRAYSELECTION>`
		Expression Expression
		`</ARRAYSELECTION>`

RecordSelection	*Expression*`.`*Identifier*	`<RECORDSELECTION>`
		Expression Identifier
		`</RECORDSELECTION>`

TupleSelection	*Expression*`.`*Numeral*	`<TUPLESELECTION>`
		Expression Numeral
		`</TUPLESELECTION>`

UpdateExpression	*Expression* `WITH`
	UpdatePosition `:=` *Expression*
		`<UPDATEEXPRESSION>`
		Expression
		UpdateSelection
		Expression
		`</UPDATEEXPRESSION>`

UpdatePosition	{ *Argument* | `[`*Expression*`]` | `.`*Identifier* | `.`*Numeral* $\}^{+}$	*See Below*
UpdateSelection	*See Below*	*Application* | *ArraySelection* | *RecordSelection* | *TupleSelection*

UpdateExpressions are represented as selections in the DTD. Thus

```
F WITH (x,y).a := 3
```

becomes

```
<UPDATEEXPRESSION>
  <NAMEEXPR>F<NAMEEXPR>
  <RECORDSELECTION>
    <APPLICATION>
      <NAMEEXPR>F</NAMEEXPR>
      <TUPLELITERAL>
        <NAMEEXPR>x</NAMEEXPR>
        <NAMEEXPR>y</NAMEEXPR>
      </TUPLELITERAL>
    </APPLICATION>
    a
  </RECORDSELECTION>
</UPDATEEXPRESSION>
```

Expressions (continued)

Nonterminal	SAL	XML
LambdaAbstraction	LAMBDA (*VarDecls*) : *Expression*	<LAMBDAABSTRACTION> *VarDecls Expression* </LAMBDAABSTRACTION>
QuantifiedExpression	*Quantifier* (*VarDecls*) : *Expression*	<QUANTIFIEDEXPRESSION> *VarDecls Expression* </QUANTIFIEDEXPRESSION>
LetExpression	LET *LetDeclarations* IN *Expression*	<LETEXPRESSION> *LetDeclarations Expression* </LETEXPRESSION>
LetDeclarations	$\{\mathit{LetDeclaration}\}^{+}_{,}$	$\{\mathit{LetDeclaration}\}^{+}$
LetDeclaration	*Identifier* : *Type* = *Expression*	<LETDECLARATION> *Identifier Type Expression* </LETDECLARATION>
SetExpression	*SetListExpression* |*SetPredExpression*	*SetListExpression* |*SetPredExpression*
SetPredExpression	{*Identifier* : *Type* | *Expression* }	<SETPREDEXPRESSION> *Identifier Type Expression* </SETPREDEXPRESSION>
SetListExpression	{ $\{\mathit{Expression}\}^{+}_{,}$ }	<SETLISTEXPRESSION> $\{\mathit{Expression}\}^{+}$ </SETLISTEXPRESSION>
ArrayLiteral	[[*IndexVarDecl*] *Expression*]	<ARRAYLITERAL> *IndexVarDecl Expression* </ARRAYLITERAL>
RecordLiteral	(# $\{\mathit{RecordEntry}\}^{+}_{,}$ #)	<RECORDLITERAL> $\{\mathit{RecordEntry}\}^{+}$ </RECORDLITERAL>
RecordEntry	*Identifier* := *Expression*	<RECORDENTRY> *Identifier Expression* </RECORDENTRY>

Nonterminal	**SAL**	**XML**
TupleLiteral	($\{Expression\}^{+}_{,}$)	<TUPLELITERAL> $\{Expression\}^{+}$ </TUPLELITERAL>

Conditional	IF *Expression*
	THEN *Expression*
		<CONDITIONAL>
		Expression
		Expression
		Expression
		</CONDITIONAL>

	{ ELSIF *Expression* THEN *Expression*}*	<CONDITIONAL ELSIF="YES"> *Expression* *Expression* *Expression* </CONDITIONAL>
	ELSE *Expression* ENDIF	
ParenExpression	(*Expression*)	*See Below*

ParenExpression Every *Expression* element has a PARENS attribute that reflects the number of parentheses. Thus `((x + 1))` corresponds to

```
<APPLICATION INFIX="YES" PARENS="2">
  <NAMEEXPR>+</NAMEEXPR>
  <TUPLELITERAL>
    <NAMEEXPR>x</NAMEEXPR>
    <NUMERAL>1</NUMERAL>
  </TUPLELITERAL>
</APPLICATION>
```

Modules

Nonterminal	SAL	XML
Module	*BaseModule* |*SynchronousComposition* |*AsynchronousComposition* |*MultiSynchronous* |*MultiAsynchronous* |*Hiding* |*NewOutput* |*Renaming* |*ModuleName* |*ParenModule*	*BaseModule* |*SynchronousComposition* |*AsynchronousComposition* |*MultiSynchronous* |*MultiAsynchronous* |*Hiding* |*NewOutput* |*Renaming* |*ModuleName* |*ParenModule*

BaseModule	`BEGIN` *BaseDeclarations* `END`	`<BASEMODULE>`
		BaseDeclarations
		`</BASEMODULE>`

BaseDeclarations	$\{BaseDeclaration\}^*$	$\{BaseDeclaration\}^*$
BaseDeclaration	*InputDecl* |*OutputDecl* |*GlobalDecl* |*LocalDecl* |*DefDecl* |*InitDecl* |*TransDecl*	*InputDecl* |*OutputDecl* |*GlobalDecl* |*LocalDecl* |*DefDecl* |*InitDecl* |*TransDecl*

Nonterminal	**SAL**	**XML**
InputDecl	INPUT *VarDecls*	<INPUTDECL> *VarDecls* </INPUTDECL>
OutputDecl	OUTPUT *VarDecls*	<OUTPUTDECL> *VarDecls* </OUTPUTDECL>
GlobalDecl	GLOBAL *VarDecls*	<GLOBALDECL> *VarDecls* </GLOBALDECL>
LocalDecl	LOCAL *VarDecls*	<LOCALDECL> *VarDecls* </LOCALDECL>
DefDecl	DEFINITION *Definitions*	<DEFDECL> *Definitions* </DEFDECL>
InitDecl	INITIALIZATION $\{DefinitionOrCommand\}^{+}_{;}$	<INITDECL> $\{DefinitionOrCommand\}^{+}$ </INITDECL>
TransDecl	TRANSITION $\{DefinitionOrCommand\}^{+}_{;}$	<TRANSDECL> $\{DefinitionOrCommand\}^{+}$ </TRANSDECL>

Nonterminal	**SAL**	**XML**
DefinitionOrCommand	*Definition* |`[` *SomeCommands* `]`	*Definition* |*SomeCommands*
Definition	*SimpleDefinition* |*ForallDefinition*	*SimpleDefinition* |*ForallDefinition*

SimpleDefinition	*Lhs RhsDefinition*	`<SIMPLEDEFINITION>`
		Lhs RhsDefinition
		`</SIMPLEDEFINITION>`

Lhs	*Identifier* [`'`] $\{\mathit{Access}\}^*$	*NextVariable* |*ArraySelection* |*RecordSelection* |*TupleSelection*
Access	*ArrayAccess* |*RecordAccess* |*TupleAccess*	
ArrayAccess	`[` *Expression* `]`	
RecordAccess	`.` *Identifier*	
TupleAccess	`.` *Numeral*	
RhsDefinition	*RhsExpression* |*RhsSelection*	*RhsExpression* |*RhsSelection*

RhsExpression	`=` *Expression*	`<RHSEXPRESSION>`
		Expression
		`</RHSEXPRESSION>`

RhsSelection	`IN` *Expression*	`<RHSSELECTION>`
		Expression
		`</RHSSELECTION>`

ForallDefinition	`(` `FORALL` `(` *VarDecls* `)`
	`:` *Definitions* `)`
		`<FORALLDEFINITION>`
		VarDecls Definitions
		`</FORALLDEFINITION>`

Definitions	$\{\mathit{Definition}\}^{+}_{;}$	$\{\mathit{Definition}\}^{+}$
SomeCommands	$\{\mathit{SomeCommand}\}^{+}_{\texttt{[]}}$	$\{\mathit{SomeCommand}\}^{+}$

Lhs An *Lhs* of the form `x'[3].a` corresponds to

```
<RECORDSELECTION>
  <ARRAYSELECTION>
    <NEXTOPERATOR><NAMEEXPR>x</NAMEEXPR></NEXTOPERATOR>
    <NUMERAL>3</NUMERAL>
  </ARRAYSELECTION>
  a
</RECORDSELECTION>
```

Modules (continued)

Nonterminal	**SAL**	**XML**
SomeCommand	*NamedCommand* |*MultiCommand*	*NamedCommand* |*MultiCommand*
NamedCommand	[*Identifier* :] *GuardedCommand*	<NAMEDCOMMAND> <LABEL> *Identifier* </LABEL> *GuardedCommand* </NAMEDCOMMAND>
GuardedCommand	*Guard* --> *Assignments*	*Guard Assignments*
Guard	*Expression*	<GUARD> *Expression* </GUARD>
Assignments	$\{SimpleDefinition\}^{+}_{;}$	<ASSIGNMENTS> $\{SimpleDefinition\}^{+}$ </ASSIGNMENTS>
MultiCommand	([] (*VarDecls*) : *SomeCommand*)	<MULTICOMMAND> *VarDecls SomeCommand* </MULTICOMMAND>
SynchronousComposition	*Module* || *Module*	<SYNCHRONOUSCOMPOSITION> *Module Module* </SYNCHRONOUSCOMPOSITION>
AsynchronousComposition	*Module* [] *Module*	<ASYNCHRONOUSCOMPOSITION> *Module Module* </ASYNCHRONOUSCOMPOSITION>
MultiSynchronous	(|| (*Identifier* : *Subrange*) : *Module*)	<MULTISYNCHRONOUS> *Identifier Subrange Module* </MULTISYNCHRONOUS>
MultiAsynchronous	([] (*Identifier* : *Subrange*) : *Module*)	<MULTIASYNCHRONOUS> *Identifier Subrange Module* </MULTIASYNCHRONOUS>

Nonterminal	**SAL**	**XML**
Hiding	LOCAL $\{\mathit{Identifier}\}^{+}_{,}$ IN *Module*	<HIDING> $\{\mathit{Identifier}\}^{+}$ *Module* </HIDING>
NewOutput	OUTPUT *VarDecls* IN *Module*	<NEWOUTPUT> *VarDecls Module* </NEWOUTPUT>
Renaming	[WITH *NewVarDecls*] RENAME *Renames* IN *Module*	<RENAMING> *NewVarDecls Renames Module* </RENAMING>
NewVarDecls	$\{\mathit{InputDecl} \mid \mathit{OutputDecl} \mid \mathit{GlobalDecl}\}^{+}_{;}$	<NEWVARDECLS> $\{\mathit{InputDecl} \mid \mathit{OutputDecl} \mid \mathit{GlobalDecl}\}^{+}$ </NEWVARDECLS>
Renames	$\{\mathit{Lhs}$ TO $\mathit{Lhs}\}^{+}_{,}$	<RENAMES> *Lhs Lhs* </RENAMES>
ModuleName	*Name* [[$\{\mathit{Expression}\}^{+}_{,}$]]	<MODULENAME> *Name* $\{\mathit{Expression}\}^{+}$ </MODULENAME>
ParenModule	(*Module*)	*See Below*

Chapter 3

The SAL DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- The DTD for the Symbolic Analysis Laboratory (SAL) Language -->

<!ENTITY % TYPE
  "( TYPENAME
   | QUALIFIEDTYPENAME
   | SUBRANGE
   | ARRAYTYPE
   | TUPLETYPE
   | FUNCTIONTYPE
   | RECORDTYPE )">

<!ENTITY % TYPEDEF "( %TYPE; | SCALARTYPE | DATATYPE )">
<!ENTITY % SETEXPRESSION "(SETLISTEXPRESSION | SETPREDEXPRESSION)">

<!ENTITY % EXPRESSION
  "( NAMEEXPR
   | QUALIFIEDNAMEEXPR
   | NEXTOPERATOR
   | NUMERAL
   | APPLICATION
   | ARRAYSELECTION
   | RECORDSELECTION
   | TUPLESELECTION
   | UPDATEEXPRESSION
   | LAMBDAABSTRACTION
   | QUANTIFIEDEXPRESSION
   | LETEXPRESSION
   | %SETEXPRESSION;
   | ARRAYLITERAL
   | RECORDLITERAL
   | TUPLELITERAL
   | CONDITIONAL )">

<!ENTITY % MODULE
 "( BASEMODULE
  | SYNCHRONOUSCOMPOSITION
  | ASYNCHRONOUSCOMPOSITION
  | MULTISYNCHRONOUS
  | MULTIASYNCHRONOUS
  | HIDING
  | NEWOUTPUT
  | RENAMING
  | MODULEINSTANCE
  | OBSERVEMODULE)">

<!ENTITY % OBSERVERMODULE
 "( BASEOBSERVERMODULE
  | SYNCHRONOUSOBSERVERCOMPOSITION
  | ASYNCHRONOUSOBSERVERCOMPOSITION
  | MULTISYNCHRONOUSOBSERVER
  | MULTIASYNCHRONOUSOBSERVER
  | OBSERVERHIDING
  | OBSERVERNEWOUTPUT
  | OBSERVERRENAMING
  | MODULEINSTANCE)">

<!ENTITY % LHS "(%EXPRESSION;)">

<!ENTITY % DEFINITION "( SIMPLEDEFINITION | FORALLDEFINITION )">

<!ENTITY % SOMECOMMAND "(GUARDEDCOMMAND|LABELEDCOMMAND|MULTICOMMAND)">

<!ENTITY % DEFINITIONORCOMMAND "(%DEFINITION;|SOMECOMMANDS)">

<!ENTITY % NAME "#PCDATA">
<!ENTITY % QUALIFIEDNAME "(IDENTIFIER, CONTEXTNAME)">

<!ENTITY % INDEXTYPE "(TYPENAME|SUBRANGE)">
<!ENTITY % BOUND "(UNBOUNDED|%EXPRESSION;)">
<!ENTITY % UPDATEPOSITION "(TUPLELITERAL|ARRAYPOSITION|IDENTIFIER|NUMERAL)">

<!ENTITY % MODULEASSERTION "(MODULEMODELS|MODULEIMPLEMENTS|MODULEREFINES)">
<!ENTITY % ASSERTIONEXPRESSION
  "(ASSERTIONPROPOSITION|QUANTIFIEDASSERTION|%MODULEASSERTION;)">

<!ELEMENT CONTEXT (IDENTIFIER, PARAMETERS, CONTEXTBODY)>

<!ELEMENT PARAMETERS (TYPEDECLS?, VARDECLS?)>
<!ELEMENT TYPEDECLS (TYPEDECL*)>
<!ELEMENT VARDECLS (VARDECL*)>
<!ELEMENT CONTEXTBODY ( CONSTANTDECLARATION
                      | TYPEDECLARATION
                      | ASSERTIONDECLARATION
                      | CONTEXTDECLARATION
                      | MODULEDECLARATION
                      | OBSERVERMODULEDECLARATION)+>
<!ELEMENT CONSTANTDECLARATION (IDENTIFIER,VARDECLS,%TYPE;,(%EXPRESSION;)?)>
```

```
<!ELEMENT TYPEDECLARATION (IDENTIFIER,(%TYPEDEF;)?)>

<!ELEMENT CONTEXTDECLARATION (IDENTIFIER, CONTEXTNAME)>
<!ELEMENT CONTEXTNAME (IDENTIFIER, ACTUALPARAMETERS?)>
<!ELEMENT ACTUALPARAMETERS (ACTUALTYPES, ACTUALEXPRS)>
<!ELEMENT ACTUALTYPES ((%TYPE;)*)>
<!ELEMENT ACTUALEXPRS ((%EXPRESSION;)*)>

<!ELEMENT MODULEDECLARATION (IDENTIFIER, VARDECLS, %MODULE;)>

<!ELEMENT BASEMODULE ( INPUTDECL
                     | OUTPUTDECL
                     | GLOBALDECL
                     | LOCALDECL
                     | DEFDECL
                     | INITDECL
                     | TRANSDECL )+>
<!ELEMENT INPUTDECL (VARDECL+)>
<!ELEMENT OUTPUTDECL (VARDECL+)>
<!ELEMENT GLOBALDECL (VARDECL+)>
<!ELEMENT LOCALDECL (VARDECL+)>
<!ELEMENT OBSERVEDDECL (VARDECL+)>

<!ELEMENT DEFDECL ((%DEFINITION;)+)>
<!ELEMENT INITDECL ((%DEFINITIONORCOMMAND;)+)>
<!ELEMENT TRANSDECL ((%DEFINITIONORCOMMAND;)+)>

<!ELEMENT SIMPLEDEFINITION (%LHS;, (RHSEXPRESSION|RHSSELECTION))>
<!ELEMENT ARRAYACCESS (%EXPRESSION;)>
<!ELEMENT RECORDACCESS (IDENTIFIER)>
<!ELEMENT TUPLEACCESS (NUMERAL)>
<!ELEMENT RHSEXPRESSION (%EXPRESSION;)>
<!ELEMENT RHSSELECTION (%EXPRESSION;)>
<!ELEMENT FORALLDEFINITION (VARDECL+, (%DEFINITION;)+)>
```

```
<!ELEMENT LABELEDCOMMAND (LABEL, GUARDEDCOMMAND)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT GUARDEDCOMMAND (GUARD, ASSIGNMENTS)>
<!ELEMENT GUARD (%EXPRESSION;)>
<!ELEMENT ASSIGNMENTS (SIMPLEDEFINITION+)>
<!ELEMENT SOMECOMMANDS ((%SOMECOMMAND;)+)>
<!ELEMENT MULTICOMMAND (VARDECL+, %SOMECOMMAND;)>

<!ELEMENT SYNCHRONOUSCOMPOSITION (%MODULE;, %MODULE;)>
<!ELEMENT ASYNCHRONOUSCOMPOSITION (%MODULE;, %MODULE;)>
<!ELEMENT MULTISYNCHRONOUS (INDEXVARDECL, %MODULE;)>
<!ELEMENT MULTIASYNCHRONOUS (INDEXVARDECL, %MODULE;)>
<!ELEMENT HIDING (IDENTIFIER+, %MODULE;)>
<!ELEMENT NEWOUTPUT (VARDECL+, %MODULE;)>
<!ELEMENT RENAMING (NEWVARDECLS?, RENAME+, %MODULE;)>
<!ELEMENT NEWVARDECLS ((INPUTDECL|OUTPUTDECL|GLOBALDECL)+)>
<!ELEMENT RENAME (%LHS;, %LHS;)>
<!ELEMENT MODULEINSTANCE ((MODULENAME|QUALIFIEDMODULENAME), MODULEACTUALS)>
<!ELEMENT MODULENAME (%NAME;)>
<!ELEMENT QUALIFIEDMODULENAME (%QUALIFIEDNAME;)>
<!ELEMENT MODULEACTUALS ((%EXPRESSION;)*)>
<!ELEMENT OBSERVEMODULE (%MODULE;, %OBSERVERMODULE;)>

<!ELEMENT SCALARTYPE (SCALARELEMENT+)>
<!ELEMENT SCALARELEMENT (#PCDATA)>
<!ELEMENT DATATYPE (CONSTRUCTOR+)>
<!ELEMENT CONSTRUCTOR (IDENTIFIER, ACCESSOR*)>
<!ELEMENT ACCESSOR (IDENTIFIER, %TYPE;)>
<!ELEMENT BASICTYPE (BOOLEAN|REAL|INTEGER|NATURAL|NZREAL)>
<!ELEMENT BOOLEAN EMPTY>
<!ELEMENT REAL EMPTY>
<!ELEMENT INTEGER EMPTY>
<!ELEMENT NATURAL EMPTY>
```

```
<!ELEMENT NZREAL EMPTY>
<!ELEMENT TYPENAME (%NAME;)>
<!ELEMENT QUALIFIEDTYPENAME (%QUALIFIEDNAME;)>
<!ELEMENT SUBRANGE (%BOUND;, %BOUND;)>
<!ELEMENT UNBOUNDED EMPTY>
<!ELEMENT ARRAYTYPE (%INDEXTYPE;, %TYPE;)>
<!ELEMENT TUPLETYPE ((%TYPE;)+)>
<!ELEMENT RECORDTYPE (FIELDDECLARATION+)>
<!ELEMENT FIELDDECLARATION (IDENTIFIER, %TYPE;)>
<!ELEMENT FUNCTIONTYPE (%TYPE;, %TYPE;)>

<!ELEMENT NEXTOPERATOR (NAMEEXPR)>
<!ELEMENT NAMEEXPR (%NAME;)>
<!ELEMENT QUALIFIEDNAMEEXPR (%QUALIFIEDNAME;)>

<!-- APPLICATIONS have 2 parts: the operator and the argument, which -->
<!--   is a TUPLEEXPRESSION for functions of arity > 1 -->
<!-- "A AND B" is an <APPLICATION INFIX="YES">, with operator AND -->
<!ELEMENT APPLICATION (%EXPRESSION;, %EXPRESSION;)>
<!ATTLIST APPLICATION INFIX (YES|NO) "NO">

<!ELEMENT ARRAYSELECTION (%EXPRESSION;, %EXPRESSION;)>
<!ELEMENT RECORDSELECTION (%EXPRESSION;, IDENTIFIER)>
<!ELEMENT TUPLESELECTION (%EXPRESSION;, NUMERAL)>
<!ELEMENT RECORDLITERAL (RECORDENTRY+)>
<!ELEMENT RECORDENTRY (IDENTIFIER, %EXPRESSION;)>
<!ELEMENT TUPLELITERAL ((%EXPRESSION;)*)>
<!ELEMENT UPDATEEXPRESSION (%EXPRESSION;, %EXPRESSION;, %EXPRESSION;)>
<!ELEMENT ARRAYPOSITION (%EXPRESSION;)>
<!ELEMENT ARRAYLITERAL (INDEXVARDECL, %EXPRESSION;)>
<!ELEMENT INDEXVARDECL (IDENTIFIER, %INDEXTYPE;)>
<!ELEMENT LAMBDAABSTRACTION (VARDECL+, %EXPRESSION;)>
<!ELEMENT QUANTIFIEDEXPRESSION (QUANTIFIER, VARDECL+, %EXPRESSION;)>
<!ELEMENT QUANTIFIER (#PCDATA)> <!-- FORALL | EXISTS -->
```

```
<!ELEMENT LETEXPRESSION (LETDECLARATIONS, %EXPRESSION;)>
<!ELEMENT LETDECLARATIONS ((LETDECLARATION)+)>
<!ELEMENT LETDECLARATION (IDENTIFIER, %TYPE;, %EXPRESSION;)>
<!ELEMENT SETPREDEXPRESSION (IDENTIFIER, %TYPE;, %EXPRESSION;)>
<!ELEMENT SETLISTEXPRESSION ((%EXPRESSION;)+)>
<!ELEMENT CONDITIONAL (%EXPRESSION;, %EXPRESSION;, %EXPRESSION;)>
<!ATTLIST CONDITIONAL ELSIF (YES|NO) "NO">

<!ELEMENT NUMERAL (#PCDATA)>
<!ELEMENT VARDECL (IDENTIFIER, %TYPE;)>
<!ATTLIST VARDECL CHAIN (YES|NO) "NO">
<!ELEMENT TYPEDECL (IDENTIFIER)>

<!ELEMENT IDENTIFIER (#PCDATA)>

<!ELEMENT ASSERTIONDECLARATION (IDENTIFIER,ASSERTIONFORM,%ASSERTIONEXPRESSION;)>
<!-- ASSERTIONFORM is one of OBLIGATION, CLAIM, LEMMA, THEOREM -->
<!ELEMENT ASSERTIONFORM (#PCDATA)>
<!ELEMENT ASSERTIONPROPOSITION (ASSERTIONOPERATOR,((%ASSERTIONEXPRESSION;)+))>
<!ELEMENT ASSERTIONOPERATOR (#PCDATA)>
<!ELEMENT QUANTIFIEDASSERTION (QUANTIFIER, VARDECL+, %ASSERTIONEXPRESSION;)>
<!ELEMENT MODULEMODELS (%MODULE;, %EXPRESSION;)>
<!ELEMENT MODULEIMPLEMENTS (%MODULE;, %MODULE;)>
<!ELEMENT MODULEREFINES (%MODULE;, %MODULE;)>

<!ELEMENT OBSERVERMODULEDECLARATION (IDENTIFIER, VARDECLS, %OBSERVERMODULE;)>

<!ELEMENT BASEOBSERVERMODULE ( OBSERVEDDECL
                             | INPUTDECL
                             | OUTPUTDECL
                             | GLOBALDECL
                             | LOCALDECL
                             | DEFDECL )+>
<!ELEMENT SYNCHRONOUSOBSERVERCOMPOSITION (%OBSERVERMODULE;, %OBSERVERMODULE;)>
```

```
<!ELEMENT ASYNCHRONOUSOBSERVERCOMPOSITION (%OBSERVERMODULE;, %OBSERVERMODULE;)>
<!ELEMENT MULTISYNCHRONOUSOBSERVER (INDEXVARDECL, %OBSERVERMODULE;)>
<!ELEMENT MULTIASYNCHRONOUSOBSERVER (INDEXVARDECL, %OBSERVERMODULE;)>
<!ELEMENT OBSERVERHIDING (IDENTIFIER+, %OBSERVERMODULE;)>
<!ELEMENT OBSERVERNEWOUTPUT (VARDECL+, %OBSERVERMODULE;)>
<!ELEMENT OBSERVERRENAMING (NEWVARDECLS?, RENAME+, %OBSERVERMODULE;)>
```

Chapter 4

Example

This is a meaningless example whose sole purpose is to test all the nonterminals of the language. We first present the `test.sal` file, followed by the `test.xsal` file that is generated by our current parser. The results have been validated against the `sal.dtd` presented in the earlier chapter.

test.sal

```
test: CONTEXT =
 BEGIN
  % Type Declarations and types
  color: TYPE = {r, g, b};  % scalartype
  list_int: TYPE = DATATYPE null, cons(car: INTEGER, cdr: list_int) END;
  below10: TYPE = [_..10];
  arr: TYPE = ARRAY INTEGER OF ARRAY foo OF REAL;
  funtype: TYPE = [REAL -> INTEGER];
  tuptype: TYPE = [BOOLEAN, [REAL -> REAL], INTEGER];
  rectype: TYPE = [# flag:BOOLEAN, fun:[REAL -> REAL], ctr:INTEGER #];

  % Constant Declarations and expressions
  a: t = b AND ctx!c AND ctx{;0}!d;
  c: ctx{t; 0}!sometype;
```

```
d: ctx2!another_type;
++: INTEGER = 33 / 11;
<<(a, b0: INTEGER): BOOLEAN = a < f(b0);
--(a,bb,c: INTEGER): BOOLEAN = p(a,bb,c);
sel: BOOLEAN = x[10].a.2;
up: [R -> R] = f WITH (17).a.13 := 11;
r: rectype = (# flag := true, fun := LAMBDA (x: REAL): x, ctr := -17 #);
t: tuptype = (true, LAMBDA (x: REAL): x, -17);
a: arr = [[i: INTEGER] [[x: foo] 0]];
fb: BOOLEAN = FORALL (i:INTEGER): p(i);
c: INTEGER = LET r: REAL = 123456 / 345678,
                 i: INTEGER = 234234234234
              IN floor(r) + i;
ifb: BOOLEAN = IF a THEN b ELSE c ENDIF;
ifc: BOOLEAN = IF a THEN b ELSE IF c THEN d ELSE e ENDIF ENDIF;
ifd: BOOLEAN = IF a THEN b
     ELSIF c THEN d
     ELSE e ENDIF;
parexpr: INTEGER = (((3)));

% Context Declarations
foo: CONTEXT = ctx{t;0};

% Module Declarations
m: MODULE =
 BEGIN
  INPUT x: INTEGER
  OUTPUT y, z: INTEGER
  GLOBAL g: REAL
  LOCAL l: REAL
  DEFINITION
   % Simple Definitions
   x = 0;
   y' IN {x: INTEGER | p(x)};
   x[0] = 0;
   x'.a IN {1,2,3};
   y'.2[x].a = 0;
   % Forall Definitions
   (FORALL (n: INTEGER): x[n] = 0; (FORALL (r:REAL): y IN s(r)))
  INITIALIZATION
   x = 0;
   [ % GuardedCommand
     a --> x' = 2; y' = x + x'
   []
     % NamedCommand
     l1: b --> z' = z
   ]
  TRANSITION
   % MultiCommand
   [ ( [] (n: INTEGER) : TRUE --> x'[n] = 3 ) ]
 END;
m0[x:INTEGER]: MODULE = m2 || m3;
m1[x:INTEGER]: MODULE = m2 [] m3;
m2: MODULE = LOCAL a,b,c IN m1;
m3: MODULE = OUTPUT x: INTEGER IN m1;
m4: MODULE = RENAME a[0] TO x, a[1] TO y IN m1;
m5: MODULE = WITH INPUT a: ARRAY INTEGER OF INTEGER
             RENAME a[0] TO x, a[1] TO y IN m1;
m6: MODULE = m0[3];
m7: MODULE = ctx{INTEGER;0}!m[x];
m8: MODULE = (m1 [] m2) || m3;
m9: MODULE = m1 [] (m2 || m3);
m10: MODULE = (|| (i: t): m0[i]) [] ([] (i: t): m0[i]);
a1: OBLIGATION NOT(m1 |- AG(x > 2));
a2: CLAIM AND(m1 |- AG(p(x)), NOT(m2 |- EF(p(x))));
a3: LEMMA FORALL (x: INTEGER): m[x] |- EF p(x);
obs1: OBSERVER_MODULE =
 BEGIN
```

```
    OBSERVED y1, y2: NATURAL,
             pc1, pc2: Control
    LOCAL  pc1, pc2: Control
    OUTPUT y1a : bool,
           y2a : bool,
           y1_y2 : bool
    DEFINITION
      y1a = (y1 = 0);
      y2a = (y2 = 0);
      y1_y2 = (y1 < y2)
   END ;
  om0[x:INTEGER]: OBSERVER_MODULE = m2 || m3;
  om1[x:INTEGER]: OBSERVER_MODULE = m2 [] m3;
  om2: OBSERVER_MODULE = LOCAL a,b,c IN m1;
  om3: OBSERVER_MODULE = OUTPUT x: INTEGER IN m1;
  om4: OBSERVER_MODULE = RENAME a[0] TO x, a[1] TO y IN m1;
  om5: OBSERVER_MODULE = WITH INPUT a: ARRAY INTEGER OF INTEGER
               RENAME a[0] TO x, a[1] TO y IN m1;
  om6: OBSERVER_MODULE = m0[3];
  om7: OBSERVER_MODULE = ctx{INTEGER;0}!m[x];
  om8: OBSERVER_MODULE = (m1 [] m2) || m3;
  om9: OBSERVER_MODULE = m1 [] (m2 || m3);
  om10: OBSERVER_MODULE = (|| (i: t): m0[i]) [] ([] (i: t): m0[i]);

 END
```

test.xsal

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CONTEXT SYSTEM "sal.dtd">
<!--  XML version of test.sal  -->
<CONTEXT>
  <IDENTIFIER>test</IDENTIFIER>
  <PARAMETERS></PARAMETERS>
```

```
<CONTEXTBODY>
  <TYPEDECLARATION>
    <IDENTIFIER>color</IDENTIFIER>
    <SCALARTYPE>
      <SCALARELEMENT>r</SCALARELEMENT>
      <SCALARELEMENT>g</SCALARELEMENT>
      <SCALARELEMENT>b</SCALARELEMENT>
    </SCALARTYPE>
  </TYPEDECLARATION>
  <TYPEDECLARATION>
    <IDENTIFIER>list_int</IDENTIFIER>
    <DATATYPE>
      <CONSTRUCTOR>
        <IDENTIFIER>null</IDENTIFIER>
      </CONSTRUCTOR>
      <CONSTRUCTOR>
        <IDENTIFIER>cons</IDENTIFIER>
        <ACCESSOR>
          <IDENTIFIER>car</IDENTIFIER>
          <TYPENAME>INTEGER</TYPENAME>
        </ACCESSOR>
        <ACCESSOR>
          <IDENTIFIER>cdr</IDENTIFIER>
          <TYPENAME>list_int</TYPENAME>
        </ACCESSOR>
      </CONSTRUCTOR>
    </DATATYPE>
  </TYPEDECLARATION>
  <TYPEDECLARATION>
    <IDENTIFIER>below10</IDENTIFIER>
    <SUBRANGE>
      <NAMEEXPR>_</NAMEEXPR>
      <NUMERAL>10</NUMERAL>
    </SUBRANGE>
```

```
</TYPEDECLARATION>
<TYPEDECLARATION>
  <IDENTIFIER>arr</IDENTIFIER>
  <ARRAYTYPE>
    <TYPENAME>INTEGER</TYPENAME>
    <ARRAYTYPE>
      <TYPENAME>foo</TYPENAME>
      <TYPENAME>REAL</TYPENAME>
    </ARRAYTYPE>
  </ARRAYTYPE>
</TYPEDECLARATION>
<TYPEDECLARATION>
  <IDENTIFIER>funtype</IDENTIFIER>
  <FUNCTIONTYPE>
    <TYPENAME>REAL</TYPENAME>
    <TYPENAME>INTEGER</TYPENAME>
  </FUNCTIONTYPE>
</TYPEDECLARATION>
<TYPEDECLARATION>
  <IDENTIFIER>tuptype</IDENTIFIER>
  <TUPLETYPE>
    <TYPENAME>BOOLEAN</TYPENAME>
    <FUNCTIONTYPE>
      <TYPENAME>REAL</TYPENAME>
      <TYPENAME>REAL</TYPENAME>
    </FUNCTIONTYPE>
    <TYPENAME>INTEGER</TYPENAME>
  </TUPLETYPE>
</TYPEDECLARATION>
<TYPEDECLARATION>
  <IDENTIFIER>rectype</IDENTIFIER>
  <RECORDTYPE>
    <FIELDDECLARATION>
      <IDENTIFIER>flag</IDENTIFIER>
      <TYPENAME>BOOLEAN</TYPENAME>
    </FIELDDECLARATION>
    <FIELDDECLARATION>
      <IDENTIFIER>fun</IDENTIFIER>
      <FUNCTIONTYPE>
        <TYPENAME>REAL</TYPENAME>
        <TYPENAME>REAL</TYPENAME>
      </FUNCTIONTYPE>
    </FIELDDECLARATION>
    <FIELDDECLARATION>
      <IDENTIFIER>ctr</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </FIELDDECLARATION>
  </RECORDTYPE>
</TYPEDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>a</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>t</TYPENAME>
  <APPLICATION INFIX="YES">
    <NAMEEXPR>AND</NAMEEXPR>
    <TUPLELITERAL>
      <APPLICATION INFIX="YES">
        <NAMEEXPR>AND</NAMEEXPR>
        <TUPLELITERAL>
          <NAMEEXPR>b</NAMEEXPR>
          <QUALIFIEDNAMEEXPR>
            <IDENTIFIER>c</IDENTIFIER>
            <CONTEXTNAME>
              <IDENTIFIER>ctx</IDENTIFIER>
            </CONTEXTNAME>
          </QUALIFIEDNAMEEXPR>
        </TUPLELITERAL>
      </APPLICATION>
      <QUALIFIEDNAMEEXPR>
        <IDENTIFIER>d</IDENTIFIER>
        <CONTEXTNAME>
          <IDENTIFIER>ctx</IDENTIFIER>
          <ACTUALPARAMETERS>
            <ACTUALTYPES></ACTUALTYPES>
            <ACTUALEXPRS>
              <NUMERAL>0</NUMERAL>
            </ACTUALEXPRS>
          </ACTUALPARAMETERS>
        </CONTEXTNAME>
      </QUALIFIEDNAMEEXPR>
    </TUPLELITERAL>
  </APPLICATION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>c</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <QUALIFIEDTYPENAME>
    <IDENTIFIER>sometype</IDENTIFIER>
    <CONTEXTNAME>
      <IDENTIFIER>ctx</IDENTIFIER>
      <ACTUALPARAMETERS>
        <ACTUALTYPES>
          <TYPENAME>t</TYPENAME>
        </ACTUALTYPES>
        <ACTUALEXPRS>
          <NUMERAL>0</NUMERAL>
        </ACTUALEXPRS>
      </ACTUALPARAMETERS>
    </CONTEXTNAME>
  </QUALIFIEDTYPENAME>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
```

```
  <IDENTIFIER>d</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <QUALIFIEDTYPENAME>
    <IDENTIFIER>another_type</IDENTIFIER>
    <CONTEXTNAME>
      <IDENTIFIER>ctx2</IDENTIFIER>
    </CONTEXTNAME>
  </QUALIFIEDTYPENAME>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>++</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>INTEGER</TYPENAME>
  <APPLICATION INFIX="YES">
    <NAMEEXPR>/</NAMEEXPR>
    <TUPLELITERAL>
      <NUMERAL>33</NUMERAL>
      <NUMERAL>11</NUMERAL>
    </TUPLELITERAL>
  </APPLICATION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>&lt;&lt;</IDENTIFIER>
  <VARDECLS>
    <VARDECL CHAIN="YES">
      <IDENTIFIER>a</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <VARDECL>
      <IDENTIFIER>b0</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
  </VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <APPLICATION INFIX="YES">
    <NAMEEXPR>&lt;</NAMEEXPR>
    <TUPLELITERAL>
      <NAMEEXPR>a</NAMEEXPR>
      <APPLICATION>
        <NAMEEXPR>f</NAMEEXPR>
        <TUPLELITERAL>
          <NAMEEXPR>b0</NAMEEXPR>
        </TUPLELITERAL>
      </APPLICATION>
    </TUPLELITERAL>
  </APPLICATION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>--</IDENTIFIER>
  <VARDECLS>
    <VARDECL CHAIN="YES">
      <IDENTIFIER>a</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <VARDECL CHAIN="YES">
      <IDENTIFIER>bb</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <VARDECL>
      <IDENTIFIER>c</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
  </VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <APPLICATION>
    <NAMEEXPR>p</NAMEEXPR>
    <TUPLELITERAL>
      <NAMEEXPR>a</NAMEEXPR>
      <NAMEEXPR>bb</NAMEEXPR>
      <NAMEEXPR>c</NAMEEXPR>
    </TUPLELITERAL>
  </APPLICATION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>sel</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <TUPLESELECTION>
    <RECORDSELECTION>
      <ARRAYSELECTION>
        <NAMEEXPR>x</NAMEEXPR>
        <NUMERAL>10</NUMERAL>
      </ARRAYSELECTION>
      <IDENTIFIER>a</IDENTIFIER>
    </RECORDSELECTION>
    <NUMERAL>2</NUMERAL>
  </TUPLESELECTION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>up</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <FUNCTIONTYPE>
    <TYPENAME>R</TYPENAME>
    <TYPENAME>R</TYPENAME>
  </FUNCTIONTYPE>
  <UPDATEEXPRESSION>
    <NAMEEXPR>f</NAMEEXPR>
    <TUPLESELECTION>
      <RECORDSELECTION>
        <APPLICATION>
          <NAMEEXPR>f</NAMEEXPR>
          <NUMERAL>17</NUMERAL>
        </APPLICATION>
        <IDENTIFIER>a</IDENTIFIER>
      </RECORDSELECTION>
      <NUMERAL>13</NUMERAL>
    </TUPLESELECTION>
    <NUMERAL>11</NUMERAL>
  </UPDATEEXPRESSION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>r</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>rectype</TYPENAME>
  <RECORDLITERAL>
    <RECORDENTRY>
      <IDENTIFIER>flag</IDENTIFIER>
      <NAMEEXPR>true</NAMEEXPR>
    </RECORDENTRY>
    <RECORDENTRY>
      <IDENTIFIER>fun</IDENTIFIER>
      <LAMBDAABSTRACTION>
        <VARDECL>
          <IDENTIFIER>x</IDENTIFIER>
          <TYPENAME>REAL</TYPENAME>
        </VARDECL>
        <NAMEEXPR>x</NAMEEXPR>
      </LAMBDAABSTRACTION>
    </RECORDENTRY>
    <RECORDENTRY>
      <IDENTIFIER>ctr</IDENTIFIER>
      <APPLICATION>
        <NAMEEXPR>-</NAMEEXPR>
        <TUPLELITERAL>
          <NUMERAL>17</NUMERAL>
        </TUPLELITERAL>
      </APPLICATION>
    </RECORDENTRY>
  </RECORDLITERAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>t</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>tuptype</TYPENAME>
  <TUPLELITERAL>
    <NAMEEXPR>true</NAMEEXPR>
    <LAMBDAABSTRACTION>
      <VARDECL>
        <IDENTIFIER>x</IDENTIFIER>
        <TYPENAME>REAL</TYPENAME>
      </VARDECL>
      <NAMEEXPR>x</NAMEEXPR>
    </LAMBDAABSTRACTION>
    <APPLICATION>
      <NAMEEXPR>-</NAMEEXPR>
      <TUPLELITERAL>
        <NUMERAL>17</NUMERAL>
      </TUPLELITERAL>
    </APPLICATION>
  </TUPLELITERAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>a</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>arr</TYPENAME>
  <ARRAYLITERAL>
    <INDEXVARDECL>
      <IDENTIFIER>i</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </INDEXVARDECL>
    <ARRAYLITERAL>
      <INDEXVARDECL>
        <IDENTIFIER>x</IDENTIFIER>
        <TYPENAME>foo</TYPENAME>
      </INDEXVARDECL>
      <NUMERAL>0</NUMERAL>
    </ARRAYLITERAL>
  </ARRAYLITERAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>fb</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <QUANTIFIEDEXPRESSION>
    <QUANTIFIER>FORALL</QUANTIFIER>
    <VARDECL>
      <IDENTIFIER>i</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <APPLICATION>
      <NAMEEXPR>p</NAMEEXPR>
      <TUPLELITERAL>
        <NAMEEXPR>i</NAMEEXPR>
      </TUPLELITERAL>
    </APPLICATION>
  </QUANTIFIEDEXPRESSION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>c</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>INTEGER</TYPENAME>
  <LETEXPRESSION>
    <LETDECLARATIONS>
      <LETDECLARATION>
```

```
        <IDENTIFIER>r</IDENTIFIER>
        <TYPENAME>REAL</TYPENAME>
        <APPLICATION INFIX="YES">
          <NAMEEXPR>/</NAMEEXPR>
          <TUPLELITERAL>
            <NUMERAL>123456</NUMERAL>
            <NUMERAL>345678</NUMERAL>
          </TUPLELITERAL>
        </APPLICATION>
      </LETDECLARATION>
      <LETDECLARATION>
        <IDENTIFIER>i</IDENTIFIER>
        <TYPENAME>INTEGER</TYPENAME>
        <NUMERAL>234234234234</NUMERAL>
      </LETDECLARATION>
    </LETDECLARATIONS>
    <APPLICATION INFIX="YES">
      <NAMEEXPR>+</NAMEEXPR>
      <TUPLELITERAL>
        <APPLICATION>
          <NAMEEXPR>floor</NAMEEXPR>
          <TUPLELITERAL>
            <NAMEEXPR>r</NAMEEXPR>
          </TUPLELITERAL>
        </APPLICATION>
        <NAMEEXPR>i</NAMEEXPR>
      </TUPLELITERAL>
    </APPLICATION>
  </LETEXPRESSION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>ifb</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <CONDITIONAL>
    <NAMEEXPR>a</NAMEEXPR>
    <NAMEEXPR>b</NAMEEXPR>
    <NAMEEXPR>c</NAMEEXPR>
  </CONDITIONAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>ifc</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <CONDITIONAL>
    <NAMEEXPR>a</NAMEEXPR>
    <NAMEEXPR>b</NAMEEXPR>
    <CONDITIONAL>
      <NAMEEXPR>c</NAMEEXPR>
      <NAMEEXPR>d</NAMEEXPR>
      <NAMEEXPR>e</NAMEEXPR>
    </CONDITIONAL>
  </CONDITIONAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>ifd</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <CONDITIONAL>
    <NAMEEXPR>a</NAMEEXPR>
    <NAMEEXPR>b</NAMEEXPR>
    <CONDITIONAL ELSIF = "YES">
      <NAMEEXPR>c</NAMEEXPR>
      <NAMEEXPR>d</NAMEEXPR>
      <NAMEEXPR>e</NAMEEXPR>
    </CONDITIONAL>
  </CONDITIONAL>
</CONSTANTDECLARATION>
```

```
<CONSTANTDECLARATION>
  <IDENTIFIER>parexpr</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <TYPENAME>INTEGER</TYPENAME>
  <TUPLELITERAL>
    <TUPLELITERAL>
      <TUPLELITERAL>
        <NUMERAL>3</NUMERAL>
      </TUPLELITERAL>
    </TUPLELITERAL>
  </TUPLELITERAL>
</CONSTANTDECLARATION>
<CONTEXTDECLARATION>
  <IDENTIFIER>foo</IDENTIFIER>
  <CONTEXTNAME>
    <IDENTIFIER>ctx</IDENTIFIER>
    <ACTUALPARAMETERS>
      <ACTUALTYPES>
        <TYPENAME>t</TYPENAME>
      </ACTUALTYPES>
      <ACTUALEXPRS>
        <NUMERAL>0</NUMERAL>
      </ACTUALEXPRS>
    </ACTUALPARAMETERS>
  </CONTEXTNAME>
</CONTEXTDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <BASEMODULE>
    <INPUTDECL>
      <VARDECL>
        <IDENTIFIER>x</IDENTIFIER>
        <TYPENAME>INTEGER</TYPENAME>
  </VARDECL>
</INPUTDECL>
<OUTPUTDECL>
  <VARDECL CHAIN="YES">
    <IDENTIFIER>y</IDENTIFIER>
    <TYPENAME>INTEGER</TYPENAME>
  </VARDECL>
  <VARDECL>
    <IDENTIFIER>z</IDENTIFIER>
    <TYPENAME>INTEGER</TYPENAME>
  </VARDECL>
</OUTPUTDECL>
<GLOBALDECL>
  <VARDECL>
    <IDENTIFIER>g</IDENTIFIER>
    <TYPENAME>REAL</TYPENAME>
  </VARDECL>
</GLOBALDECL>
<LOCALDECL>
  <VARDECL>
    <IDENTIFIER>l</IDENTIFIER>
    <TYPENAME>REAL</TYPENAME>
  </VARDECL>
</LOCALDECL>
<DEFDECL>
  <SIMPLEDEFINITION>
    <NAMEEXPR>x</NAMEEXPR>
    <RHSEXPRESSION>
      <NUMERAL>0</NUMERAL>
    </RHSEXPRESSION>
  </SIMPLEDEFINITION>
  <SIMPLEDEFINITION>
    <NEXTOPERATOR>
      <NAMEEXPR>y</NAMEEXPR>
  </NEXTOPERATOR>
  <RHSSELECTION>
    <SETPREDEXPRESSION>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
      <APPLICATION>
        <NAMEEXPR>p</NAMEEXPR>
        <TUPLELITERAL>
          <NAMEEXPR>x</NAMEEXPR>
        </TUPLELITERAL>
      </APPLICATION>
    </SETPREDEXPRESSION>
  </RHSSELECTION>
</SIMPLEDEFINITION>
<SIMPLEDEFINITION>
  <ARRAYSELECTION>
    <NAMEEXPR>x</NAMEEXPR>
    <NUMERAL>0</NUMERAL>
  </ARRAYSELECTION>
  <RHSEXPRESSION>
    <NUMERAL>0</NUMERAL>
  </RHSEXPRESSION>
</SIMPLEDEFINITION>
<SIMPLEDEFINITION>
  <RECORDSELECTION>
    <NEXTOPERATOR>
      <NAMEEXPR>x</NAMEEXPR>
    </NEXTOPERATOR>
    <IDENTIFIER>a</IDENTIFIER>
  </RECORDSELECTION>
  <RHSSELECTION>
    <SETLISTEXPRESSION>
      <NUMERAL>1</NUMERAL>
      <NUMERAL>2</NUMERAL>
      <NUMERAL>3</NUMERAL>
    </SETLISTEXPRESSION>
  </RHSSELECTION>
</SIMPLEDEFINITION>
<SIMPLEDEFINITION>
  <RECORDSELECTION>
    <ARRAYSELECTION>
      <TUPLESELECTION>
        <NEXTOPERATOR>
          <NAMEEXPR>y</NAMEEXPR>
        </NEXTOPERATOR>
        <NUMERAL>2</NUMERAL>
      </TUPLESELECTION>
      <NAMEEXPR>x</NAMEEXPR>
    </ARRAYSELECTION>
    <IDENTIFIER>a</IDENTIFIER>
  </RECORDSELECTION>
  <RHSEXPRESSION>
    <NUMERAL>0</NUMERAL>
  </RHSEXPRESSION>
</SIMPLEDEFINITION>
<FORALLDEFINITION>
  <VARDECL>
    <IDENTIFIER>n</IDENTIFIER>
    <TYPENAME>INTEGER</TYPENAME>
  </VARDECL>
  <SIMPLEDEFINITION>
    <ARRAYSELECTION>
      <NAMEEXPR>x</NAMEEXPR>
      <NAMEEXPR>n</NAMEEXPR>
    </ARRAYSELECTION>
    <RHSEXPRESSION>
      <NUMERAL>0</NUMERAL>
    </RHSEXPRESSION>
    </SIMPLEDEFINITION>
    <FORALLDEFINITION>
      <VARDECL>
        <IDENTIFIER>r</IDENTIFIER>
        <TYPENAME>REAL</TYPENAME>
      </VARDECL>
      <SIMPLEDEFINITION>
        <NAMEEXPR>y</NAMEEXPR>
        <RHSSELECTION>
          <APPLICATION>
            <NAMEEXPR>s</NAMEEXPR>
            <TUPLELITERAL>
              <NAMEEXPR>r</NAMEEXPR>
            </TUPLELITERAL>
          </APPLICATION>
        </RHSSELECTION>
      </SIMPLEDEFINITION>
    </FORALLDEFINITION>
  </FORALLDEFINITION>
</DEFDECL>
<INITDECL>
  <SIMPLEDEFINITION>
    <NAMEEXPR>x</NAMEEXPR>
    <RHSEXPRESSION>
      <NUMERAL>0</NUMERAL>
    </RHSEXPRESSION>
  </SIMPLEDEFINITION>
  <SOMECOMMANDS>
    <GUARDEDCOMMAND>
      <GUARD>
        <NAMEEXPR>a</NAMEEXPR>
      </GUARD>
      <ASSIGNMENTS>
        <SIMPLEDEFINITION>
        <NEXTOPERATOR>
          <NAMEEXPR>x</NAMEEXPR>
        </NEXTOPERATOR>
        <RHSEXPRESSION>
          <NUMERAL>2</NUMERAL>
        </RHSEXPRESSION>
      </SIMPLEDEFINITION>
      <SIMPLEDEFINITION>
        <NEXTOPERATOR>
          <NAMEEXPR>y</NAMEEXPR>
        </NEXTOPERATOR>
        <RHSEXPRESSION>
          <APPLICATION INFIX="YES">
            <NAMEEXPR>+</NAMEEXPR>
            <TUPLELITERAL>
              <NAMEEXPR>x</NAMEEXPR>
              <NEXTOPERATOR>
                <NAMEEXPR>x</NAMEEXPR>
              </NEXTOPERATOR>
            </TUPLELITERAL>
          </APPLICATION>
        </RHSEXPRESSION>
      </SIMPLEDEFINITION>
    </ASSIGNMENTS>
  </GUARDEDCOMMAND>
  <LABELEDCOMMAND>
    <LABEL>l1</LABEL>
    <GUARDEDCOMMAND>
      <GUARD>
        <NAMEEXPR>b</NAMEEXPR>
      </GUARD>
      <ASSIGNMENTS>
        <SIMPLEDEFINITION>
          <NEXTOPERATOR>
```

```
              <NAMEEXPR>z</NAMEEXPR>
            </NEXTOPERATOR>
            <RHSEXPRESSION>
              <NAMEEXPR>z</NAMEEXPR>
            </RHSEXPRESSION>
          </SIMPLEDEFINITION>
        </ASSIGNMENTS>
      </GUARDEDCOMMAND>
    </LABELEDCOMMAND>
  </SOMECOMMANDS>
</INITDECL>
<TRANSDECL>
  <SOMECOMMANDS>
    <MULTICOMMAND>
      <VARDECL>
        <IDENTIFIER>n</IDENTIFIER>
        <TYPENAME>INTEGER</TYPENAME>
      </VARDECL>
      <GUARDEDCOMMAND>
        <GUARD>
          <NAMEEXPR>TRUE</NAMEEXPR>
        </GUARD>
        <ASSIGNMENTS>
          <SIMPLEDEFINITION>
            <ARRAYSELECTION>
              <NEXTOPERATOR>
                <NAMEEXPR>x</NAMEEXPR>
              </NEXTOPERATOR>
              <NAMEEXPR>n</NAMEEXPR>
            </ARRAYSELECTION>
            <RHSEXPRESSION>
              <NUMERAL>3</NUMERAL>
            </RHSEXPRESSION>
          </SIMPLEDEFINITION>
```

```
            </ASSIGNMENTS>
          </GUARDEDCOMMAND>
        </MULTICOMMAND>
      </SOMECOMMANDS>
    </TRANSDECL>
  </BASEMODULE>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m0</IDENTIFIER>
  <VARDECLS>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
  </VARDECLS>
  <SYNCHRONOUSCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m2</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
    <MODULEINSTANCE>
      <MODULENAME>m3</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </SYNCHRONOUSCOMPOSITION>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m1</IDENTIFIER>
  <VARDECLS>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
  </VARDECLS>
  <ASYNCHRONOUSCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m2</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
    <MODULEINSTANCE>
      <MODULENAME>m3</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </ASYNCHRONOUSCOMPOSITION>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m2</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <HIDING>
    <IDENTIFIER>a</IDENTIFIER>
    <IDENTIFIER>b</IDENTIFIER>
    <IDENTIFIER>c</IDENTIFIER>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </HIDING>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m3</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <NEWOUTPUT>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </NEWOUTPUT>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m4</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <RENAMING>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>0</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>x</NAMEEXPR>
    </RENAME>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>1</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>y</NAMEEXPR>
    </RENAME>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </RENAMING>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m5</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <RENAMING>
    <NEWVARDECLS>
      <INPUTDECL>
```

```
        <VARDECL>
          <IDENTIFIER>a</IDENTIFIER>
          <ARRAYTYPE>
            <TYPENAME>INTEGER</TYPENAME>
            <TYPENAME>INTEGER</TYPENAME>
          </ARRAYTYPE>
        </VARDECL>
      </INPUTDECL>
    </NEWVARDECLS>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>0</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>x</NAMEEXPR>
    </RENAME>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>1</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>y</NAMEEXPR>
    </RENAME>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </RENAMING>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m6</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <MODULEINSTANCE>
    <MODULENAME>m0</MODULENAME>
    <MODULEACTUALS>
      <NUMERAL>3</NUMERAL>
    </MODULEACTUALS>
  </MODULEINSTANCE>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m7</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <MODULEINSTANCE>
    <QUALIFIEDMODULENAME>
      <IDENTIFIER>m</IDENTIFIER>
      <CONTEXTNAME>
        <IDENTIFIER>ctx</IDENTIFIER>
        <ACTUALPARAMETERS>
          <ACTUALTYPES>
            <TYPENAME>INTEGER</TYPENAME>
          </ACTUALTYPES>
          <ACTUALEXPRS>
            <NUMERAL>0</NUMERAL>
          </ACTUALEXPRS>
        </ACTUALPARAMETERS>
      </CONTEXTNAME>
    </QUALIFIEDMODULENAME>
    <MODULEACTUALS>
      <NAMEEXPR>x</NAMEEXPR>
    </MODULEACTUALS>
  </MODULEINSTANCE>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m8</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <SYNCHRONOUSCOMPOSITION>
    <ASYNCHRONOUSCOMPOSITION>
      <MODULEINSTANCE>
```

```
        <MODULENAME>m1</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
      <MODULEINSTANCE>
        <MODULENAME>m2</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
    </ASYNCHRONOUSCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m3</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </SYNCHRONOUSCOMPOSITION>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m9</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <ASYNCHRONOUSCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
    <SYNCHRONOUSCOMPOSITION>
      <MODULEINSTANCE>
        <MODULENAME>m2</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
      <MODULEINSTANCE>
        <MODULENAME>m3</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
    </SYNCHRONOUSCOMPOSITION>
  </ASYNCHRONOUSCOMPOSITION>
</MODULEDECLARATION>
```

```
<MODULEDECLARATION>
  <IDENTIFIER>m10</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <ASYNCHRONOUSCOMPOSITION>
    <MULTISYNCHRONOUS>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TYPENAME>t</TYPENAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
    </MULTISYNCHRONOUS>
    <MULTIASYNCHRONOUS>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TYPENAME>t</TYPENAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
    </MULTIASYNCHRONOUS>
  </ASYNCHRONOUSCOMPOSITION>
</MODULEDECLARATION>
<ASSERTIONDECLARATION>
  <IDENTIFIER>a1</IDENTIFIER>
  <ASSERTIONFORM>OBLIGATION</ASSERTIONFORM>
  <ASSERTIONPROPOSITION>
```

```
    <ASSERTIONOPERATOR>NOT</ASSERTIONOPERATOR>
    <MODULEMODELS>
      <MODULEINSTANCE>
        <MODULENAME>m1</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
      <APPLICATION>
        <NAMEEXPR>AG</NAMEEXPR>
        <TUPLELITERAL>
          <APPLICATION INFIX="YES">
            <NAMEEXPR>&gt;</NAMEEXPR>
            <TUPLELITERAL>
              <NAMEEXPR>x</NAMEEXPR>
              <NUMERAL>2</NUMERAL>
            </TUPLELITERAL>
          </APPLICATION>
        </TUPLELITERAL>
      </APPLICATION>
    </MODULEMODELS>
  </ASSERTIONPROPOSITION>
</ASSERTIONDECLARATION>
<ASSERTIONDECLARATION>
  <IDENTIFIER>a2</IDENTIFIER>
  <ASSERTIONFORM>CLAIM</ASSERTIONFORM>
  <ASSERTIONPROPOSITION>
    <ASSERTIONOPERATOR>AND</ASSERTIONOPERATOR>
    <MODULEMODELS>
      <MODULEINSTANCE>
        <MODULENAME>m1</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
      <APPLICATION>
        <NAMEEXPR>AG</NAMEEXPR>
        <TUPLELITERAL>
```

```
          <APPLICATION>
            <NAMEEXPR>p</NAMEEXPR>
            <TUPLELITERAL>
              <NAMEEXPR>x</NAMEEXPR>
            </TUPLELITERAL>
          </APPLICATION>
        </TUPLELITERAL>
      </APPLICATION>
    </MODULEMODELS>
    <ASSERTIONPROPOSITION>
      <ASSERTIONOPERATOR>NOT</ASSERTIONOPERATOR>
      <MODULEMODELS>
        <MODULEINSTANCE>
          <MODULENAME>m2</MODULENAME>
          <MODULEACTUALS></MODULEACTUALS>
        </MODULEINSTANCE>
        <APPLICATION>
          <NAMEEXPR>EF</NAMEEXPR>
          <TUPLELITERAL>
            <APPLICATION>
              <NAMEEXPR>p</NAMEEXPR>
              <TUPLELITERAL>
                <NAMEEXPR>x</NAMEEXPR>
              </TUPLELITERAL>
            </APPLICATION>
          </TUPLELITERAL>
        </APPLICATION>
      </MODULEMODELS>
    </ASSERTIONPROPOSITION>
  </ASSERTIONPROPOSITION>
</ASSERTIONDECLARATION>
<ASSERTIONDECLARATION>
  <IDENTIFIER>a3</IDENTIFIER>
  <ASSERTIONFORM>LEMMA</ASSERTIONFORM>
  <QUANTIFIEDASSERTION>
    <QUANTIFIER>FORALL</QUANTIFIER>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <MODULEMODELS>
      <MODULEINSTANCE>
        <MODULENAME>m</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>x</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
      <APPLICATION INFIX="YES">
        <NAMEEXPR>p</NAMEEXPR>
        <TUPLELITERAL>
          <NAMEEXPR>EF</NAMEEXPR>
          <TUPLELITERAL>
            <NAMEEXPR>x</NAMEEXPR>
          </TUPLELITERAL>
        </TUPLELITERAL>
      </APPLICATION>
    </MODULEMODELS>
  </QUANTIFIEDASSERTION>
</ASSERTIONDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>obs1</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <BASEOBSERVERMODULE>
    <OBSERVEDDECL>
      <VARDECL CHAIN="YES">
        <IDENTIFIER>y1</IDENTIFIER>
        <TYPENAME>NATURAL</TYPENAME>
      </VARDECL>
  <VARDECL>
    <IDENTIFIER>y2</IDENTIFIER>
    <TYPENAME>NATURAL</TYPENAME>
  </VARDECL>
  <VARDECL CHAIN="YES">
    <IDENTIFIER>pc1</IDENTIFIER>
    <TYPENAME>Control</TYPENAME>
  </VARDECL>
  <VARDECL>
    <IDENTIFIER>pc2</IDENTIFIER>
    <TYPENAME>Control</TYPENAME>
  </VARDECL>
</OBSERVEDDECL>
<LOCALDECL>
  <VARDECL CHAIN="YES">
    <IDENTIFIER>pc1</IDENTIFIER>
    <TYPENAME>Control</TYPENAME>
  </VARDECL>
  <VARDECL>
    <IDENTIFIER>pc2</IDENTIFIER>
    <TYPENAME>Control</TYPENAME>
  </VARDECL>
</LOCALDECL>
<OUTPUTDECL>
  <VARDECL>
    <IDENTIFIER>y1a</IDENTIFIER>
    <TYPENAME>bool</TYPENAME>
  </VARDECL>
  <VARDECL>
    <IDENTIFIER>y2a</IDENTIFIER>
    <TYPENAME>bool</TYPENAME>
  </VARDECL>
  <VARDECL>
    <IDENTIFIER>y1_y2</IDENTIFIER>
    <TYPENAME>bool</TYPENAME>
  </VARDECL>
</OUTPUTDECL>
<DEFDECL>
  <SIMPLEDEFINITION>
    <NAMEEXPR>y1a</NAMEEXPR>
    <RHSEXPRESSION>
      <TUPLELITERAL>
        <APPLICATION INFIX="YES">
          <NAMEEXPR>=</NAMEEXPR>
          <TUPLELITERAL>
            <NAMEEXPR>y1</NAMEEXPR>
            <NUMERAL>0</NUMERAL>
          </TUPLELITERAL>
        </APPLICATION>
      </TUPLELITERAL>
    </RHSEXPRESSION>
  </SIMPLEDEFINITION>
  <SIMPLEDEFINITION>
    <NAMEEXPR>y2a</NAMEEXPR>
    <RHSEXPRESSION>
      <TUPLELITERAL>
        <APPLICATION INFIX="YES">
          <NAMEEXPR>=</NAMEEXPR>
          <TUPLELITERAL>
            <NAMEEXPR>y2</NAMEEXPR>
            <NUMERAL>0</NUMERAL>
          </TUPLELITERAL>
        </APPLICATION>
      </TUPLELITERAL>
    </RHSEXPRESSION>
  </SIMPLEDEFINITION>
  <SIMPLEDEFINITION>
    <NAMEEXPR>y1_y2</NAMEEXPR>
```

```
        <RHSEXPRESSION>
          <TUPLELITERAL>
            <APPLICATION INFIX="YES">
              <NAMEEXPR>&lt;</NAMEEXPR>
              <TUPLELITERAL>
                <NAMEEXPR>y1</NAMEEXPR>
                <NAMEEXPR>y2</NAMEEXPR>
              </TUPLELITERAL>
            </APPLICATION>
          </TUPLELITERAL>
        </RHSEXPRESSION>
      </SIMPLEDEFINITION>
    </DEFDECL>
  </BASEOBSERVERMODULE>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om0</IDENTIFIER>
  <VARDECLS>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
  </VARDECLS>
  <SYNCHRONOUSOBSERVERCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m2</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
    <MODULEINSTANCE>
      <MODULENAME>m3</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </SYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>
```

```
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om1</IDENTIFIER>
  <VARDECLS>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
  </VARDECLS>
  <ASYNCHRONOUSOBSERVERCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m2</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
    <MODULEINSTANCE>
      <MODULENAME>m3</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </ASYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om2</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <OBSERVERHIDING>
    <IDENTIFIER>a</IDENTIFIER>
    <IDENTIFIER>b</IDENTIFIER>
    <IDENTIFIER>c</IDENTIFIER>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </OBSERVERHIDING>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om3</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <OBSERVERNEWOUTPUT>
    <VARDECL>
      <IDENTIFIER>x</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDECL>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </OBSERVERNEWOUTPUT>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om4</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <OBSERVERRENAMING>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>0</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>x</NAMEEXPR>
    </RENAME>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>1</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>y</NAMEEXPR>
    </RENAME>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </OBSERVERRENAMING>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om5</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <OBSERVERRENAMING>
    <NEWVARDECLS>
      <INPUTDECL>
        <VARDECL>
          <IDENTIFIER>a</IDENTIFIER>
          <ARRAYTYPE>
            <TYPENAME>INTEGER</TYPENAME>
            <TYPENAME>INTEGER</TYPENAME>
          </ARRAYTYPE>
        </VARDECL>
      </INPUTDECL>
    </NEWVARDECLS>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>0</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>x</NAMEEXPR>
    </RENAME>
    <RENAME>
      <ARRAYSELECTION>
        <NAMEEXPR>a</NAMEEXPR>
        <NUMERAL>1</NUMERAL>
      </ARRAYSELECTION>
      <NAMEEXPR>y</NAMEEXPR>
    </RENAME>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </OBSERVERRENAMING>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om6</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <MODULEINSTANCE>
    <MODULENAME>m0</MODULENAME>
    <MODULEACTUALS>
      <NUMERAL>3</NUMERAL>
    </MODULEACTUALS>
  </MODULEINSTANCE>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om7</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <MODULEINSTANCE>
    <QUALIFIEDMODULENAME>
      <IDENTIFIER>m</IDENTIFIER>
      <CONTEXTNAME>
        <IDENTIFIER>ctx</IDENTIFIER>
        <ACTUALPARAMETERS>
          <ACTUALTYPES>
            <TYPENAME>INTEGER</TYPENAME>
          </ACTUALTYPES>
          <ACTUALEXPRS>
            <NUMERAL>0</NUMERAL>
          </ACTUALEXPRS>
        </ACTUALPARAMETERS>
      </CONTEXTNAME>
    </QUALIFIEDMODULENAME>
    <MODULEACTUALS>
      <NAMEEXPR>x</NAMEEXPR>
    </MODULEACTUALS>
  </MODULEINSTANCE>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om8</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <SYNCHRONOUSOBSERVERCOMPOSITION>
    <ASYNCHRONOUSOBSERVERCOMPOSITION>
      <MODULEINSTANCE>
        <MODULENAME>m1</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
      <MODULEINSTANCE>
        <MODULENAME>m2</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
    </ASYNCHRONOUSOBSERVERCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m3</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
  </SYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om9</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <ASYNCHRONOUSOBSERVERCOMPOSITION>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
      <MODULEACTUALS></MODULEACTUALS>
    </MODULEINSTANCE>
    <SYNCHRONOUSOBSERVERCOMPOSITION>
      <MODULEINSTANCE>
        <MODULENAME>m2</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
      <MODULEINSTANCE>
        <MODULENAME>m3</MODULENAME>
        <MODULEACTUALS></MODULEACTUALS>
      </MODULEINSTANCE>
    </SYNCHRONOUSOBSERVERCOMPOSITION>
  </ASYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>
<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om10</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <ASYNCHRONOUSOBSERVERCOMPOSITION>
    <MULTISYNCHRONOUSOBSERVER>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TYPENAME>t</TYPENAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
    </MULTISYNCHRONOUSOBSERVER>
    <MULTIASYNCHRONOUSOBSERVER>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TYPENAME>t</TYPENAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
```

```
          </MODULEINSTANCE>
        </MULTIASYNCHRONOUSOBSERVER>
      </ASYNCHRONOUSOBSERVERCOMPOSITION>
    </OBSERVERMODULEDECLARATION>
  </CONTEXTBODY>
</CONTEXT>
```