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.cs1.sri.com/projects/cara/reports/language-report.ps.gz. In order to be as independent as possible of the choice of operating system or implementation 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 It is very much a work in progress—the tables are quite out-of-date and incomplete; the SAL DTD and the examples are much closer to finished.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

\[
\begin{align*}
a & : \text{INTEGER;} \\
b, c & : \text{INTEGER;}
\end{align*}
\]

These are represented in the SAL DTD as:

\[
\begin{align*}
\text{<CONSTANTDECLARATION CHAIN="YES">} \\
\text{<IDENTIFIER>a</IDENTIFIER>} \\
\text{<VARDECLS)}</VARDECLS> \\
\text{<TPNAME>INTEGER</TPNAME>} \\
\text{</CONSTANTDECLARATION>}
\end{align*}
\]

\[
\begin{align*}
\text{<CONSTANTDECLARATION CHAIN="NO">} \\
\text{<IDENTIFIER>b</IDENTIFIER>} \\
\text{<VARDECLS)}</VARDECLS> \\
\text{<TPNAME>INTEGER</TPNAME>} \\
\text{</CONSTANTDECLARATION>}
\end{align*}
\]

\[
\begin{align*}
\text{<CONSTANTDECLARATION CHAIN="NO">} \\
\text{<IDENTIFIER>c</IDENTIFIER>} \\
\text{<VARDECLS)}</VARDECLS> \\
\text{<TPNAME>INTEGER</TPNAME>} \\
\text{</CONSTANTDECLARATION>}
\end{align*}
\]

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

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

\[
\begin{align*}
\text{<APPLICATION INFIX="YES">} \\
\text{<NAMEEXPR>AND</NAMEEXPR>} \\
\text{<TUPLELITERAL>} \\
\text{<NAMEEXPR>A</NAMEEXPR>} \\
\text{<NAMEEXPR>B</NAMEEXPR>} \\
\text{</TUPLELITERAL>} \\
\text{</APPLICATION>}
\end{align*}
\]
<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>SAL</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context</strong></td>
<td>Identifier {Parameters} : CONTEXT = ContextBody</td>
<td><code>&lt;CONTEXT&gt;</code>&lt;br&gt; Identifier Parameters ContextBody&lt;br&gt;<code>&lt;/CONTEXT&gt;</code></td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>[TypeDecls] TYPE ; {VarDecls}*,</td>
<td><code>&lt;PARAMETERS&gt;</code>&lt;br&gt; TypeDecls VarDecls&lt;br&gt;<code>&lt;/PARAMETERS&gt;</code></td>
</tr>
<tr>
<td><strong>ContextBody</strong></td>
<td>BEGIN Declarations END</td>
<td><code>&lt;CONTEXTBODY&gt;</code>&lt;br&gt; {Declarations}+&lt;br&gt;<code>&lt;/CONTEXTBODY&gt;</code></td>
</tr>
<tr>
<td><strong>Declarations</strong></td>
<td>{Declaration ; }+</td>
<td><code>{Declaration }+</code></td>
</tr>
<tr>
<td><strong>Declaration</strong></td>
<td>TypeDeclaration&lt;br&gt;</td>
<td>ConstantDeclaration&lt;br&gt;</td>
</tr>
<tr>
<td><strong>TypeDef</strong></td>
<td>ScalarType</td>
<td><code>&lt;TYPEDECLARATION&gt;</code>&lt;br&gt; Identifier [TypeDef]&lt;br&gt;<code>&lt;/TYPEDECLARATION&gt;</code></td>
</tr>
<tr>
<td><strong>Scalar Type</strong></td>
<td>{{Identifier}, }*</td>
<td><code>&lt;SCALARTYPE&gt;</code>&lt;br&gt; {Identifier},+&lt;br&gt;<code>&lt;/SCALARTYPE&gt;</code></td>
</tr>
<tr>
<td><strong>Data Type</strong></td>
<td>DATATYPE Constructors END</td>
<td><code>&lt;DATATYPE&gt;</code>&lt;br&gt; Constructors&lt;br&gt;<code>&lt;/DATATYPE&gt;</code></td>
</tr>
<tr>
<td><strong>Constructors</strong></td>
<td>{Constructor}+</td>
<td><code>{Constructor}+</code></td>
</tr>
<tr>
<td><strong>Constructor</strong></td>
<td>Identifier {Accessors }</td>
<td><code>&lt;CONSTRUCTOR&gt;</code>&lt;br&gt; Identifier Accessors&lt;br&gt;<code>&lt;/CONSTRUCTOR&gt;</code></td>
</tr>
<tr>
<td><strong>Accessors</strong></td>
<td>{Accessor}+</td>
<td><code>{Accessor}+</code></td>
</tr>
<tr>
<td><strong>Accessor</strong></td>
<td>Identifier : Type</td>
<td><code>&lt;ACCESSOR&gt;</code>&lt;br&gt; Identifier Type&lt;br&gt;<code>&lt;/ACCESSOR&gt;</code></td>
</tr>
<tr>
<td>Nonterminal</td>
<td>SAL</td>
<td>XML</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>ConstantDeclaration</td>
<td>Identifier [ ( VarDecs ) ] : Type</td>
<td>&lt;CONSTANTDECLARATION&gt; Identifier Type {Expression}</td>
</tr>
<tr>
<td></td>
<td>[ = Expression]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;CONSTANTDECLARATION&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/CONSTANTDECLARATION&gt;</td>
</tr>
<tr>
<td>AssertionDeclaration</td>
<td>Identifier : AssertionForm = AssertionExpression</td>
<td>&lt;ASSERTIONDECLARATION&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AssertionForm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AssertionExpression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/ASSERTIONDECLARATION&gt;</td>
</tr>
<tr>
<td>ModuleDeclaration</td>
<td>Identifier [ ( VarDecs ) ] :</td>
<td>&lt;MODULEDECLARATION&gt;</td>
</tr>
<tr>
<td></td>
<td>MODULE = Module</td>
<td>Identifier {VarDecl}^* Module</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/MODULEDECLARATION&gt;</td>
</tr>
<tr>
<td>ContextDeclaration</td>
<td>Identifier : CONTEXT = ContextName</td>
<td>&lt;CONTEXTDECLARATION&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier ContextName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/CONTEXTDECLARATION&gt;</td>
</tr>
<tr>
<td>ContextName</td>
<td>Identifier { ActualParameters }</td>
<td>&lt;CONTEXTNAME&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier ActualParameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/CONTEXTNAME&gt;</td>
</tr>
</tbody>
</table>
**Types**

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>SAL</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>BasicType</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ArrayType</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FunctionType</td>
</tr>
<tr>
<td><strong>BasicType</strong></td>
<td>BOOLEAN</td>
<td>REAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NZINTEGER</td>
</tr>
<tr>
<td><strong>Subrange</strong></td>
<td>[ Bound .. Bound ]</td>
<td>&lt;TYPENAME&gt;BOOLEAN&lt;/TYPENAME&gt;</td>
</tr>
<tr>
<td><strong>Bound</strong></td>
<td>Unbounded</td>
<td>Expression</td>
</tr>
<tr>
<td><strong>Unbounded</strong></td>
<td>-</td>
<td>&lt;NAMEEXPR&gt; &lt;/NAMEEXPR&gt;</td>
</tr>
<tr>
<td><strong>ArrayType</strong></td>
<td>ARRAY IndexType OF Type</td>
<td>&lt;ARRAYTYPE&gt;</td>
</tr>
<tr>
<td></td>
<td>IndexType ElementType</td>
<td>&lt;ARRAYTYPE&gt;</td>
</tr>
<tr>
<td><strong>IndexType</strong></td>
<td>INTEGER</td>
<td>Subrange</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subrange</td>
</tr>
<tr>
<td><strong>ScalarTypeName</strong></td>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td><strong>TupleType</strong></td>
<td>[ {Type}+ ]</td>
<td>&lt;TUPLETYPE&gt; {Type}+ &lt;/TUPLETYPE&gt;</td>
</tr>
<tr>
<td><strong>FunctionType</strong></td>
<td>[ Type -&gt; Type ]</td>
<td>&lt;FUNCTIONTYPE&gt;</td>
</tr>
<tr>
<td></td>
<td>Type Type</td>
<td>&lt;FUNCTIONTYPE&gt;</td>
</tr>
<tr>
<td><strong>RecordType</strong></td>
<td>[# {FieldDecl},+ #]</td>
<td>&lt;RECORDTYPE&gt;</td>
</tr>
<tr>
<td></td>
<td>{FieldDecl},+</td>
<td>&lt;RECORDTYPE&gt;</td>
</tr>
<tr>
<td><strong>FieldDecl</strong></td>
<td>Identifier : Type</td>
<td>&lt;FIELDDECL&gt;</td>
</tr>
<tr>
<td></td>
<td>Identifier Type</td>
<td>&lt;FIELDDECL&gt;</td>
</tr>
</tbody>
</table>
Expressions

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>SAL</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>NameExpr</td>
<td>NameExpr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NextVariable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Numerul</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>InfixApplication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ArraySelection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecordSelection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TupleSelection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UpdateExpression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LambdaAbstraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QuantifiedExpression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LetExpression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetExpression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ArrayLiteral</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RecordLiteral</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TupleLiteral</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conditional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ParenExpression</td>
</tr>
<tr>
<td>NextVariable</td>
<td>Identifier '</td>
<td>&lt;NEXTOPERATOR&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifier</td>
</tr>
<tr>
<td>Application</td>
<td>Expression Argument</td>
<td>&lt;APPLICATION&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expression Argument</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expression Argument</td>
</tr>
<tr>
<td>Argument</td>
<td>( {Expression} ) *</td>
<td>&lt;TUPLELITERAL&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{Expression} *</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{Expression} *</td>
</tr>
<tr>
<td>Nonterminal</td>
<td>SAL</td>
<td>XML</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
</tbody>
</table>
| InfixApplication | Expression Identifier Expression | `<APPLICATION INFIX="YES">  
  <NAMEEXPR>
  Identifier
  </NAMEEXPR>
  <TUPLELITERAL>
  Expression Expression
  </TUPLELITERAL>
  </APPLICATION>` |
| ArraySelection | Expression[Expression]        | `<ARRAYSELECTION>` Expression Expression     |
| RecordSelection | Expression. Identifier        | `<RECORDSELECTION>` Expression Identifier     |
| TupleSelection | Expression. Numerical         | `<TUPLESELECTION>` Expression Numerical       |
| UpdateExpression | Expression WITH UpdatePosition := Expression | `<UPDATEEXPRESSION>` Expression UpdateSelection Expression     |
| UpdatePosition | { Argument [Expression] .Identifier | Numerical }+ | `See Below` |

*UpdateExpressions* are represented as selections in the DTD. Thus

\[ F \text{ WITH } (x,y).a := 3 \]

becomes

\[
<\text{UPDATEEXPRESSION}>
  <\text{NAMEEXPR}>
    F
  <\text{NAMEEXPR}>
\]
<RECORDSELECTION>
  <APPLICATION>
    <NAMEEXPR>F</NAMEEXPR>
    <TUPLELITERAL>
      <NAMEEXPR>x</NAMEEXPR>
      <NAMEEXPR>y</NAMEEXPR>
    </TUPLELITERAL>
    a
  </APPLICATION>
</RECORDSELECTION>
</UPDATEEXPRESSION>
Expressions (continued)

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>SAL</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>LambdaAbstraction</td>
<td>\texttt{LAMBDA (VarDecls) : Expression}</td>
<td>\texttt{&lt;LAMBDAABSTRACTION&gt; VarDecls Expression&lt;/LAMBDAABSTRACTION&gt;}</td>
</tr>
<tr>
<td>QuantifiedExpression</td>
<td>\texttt{Quantifier (VarDecls) : Expression}</td>
<td>\texttt{&lt;QUANTIFIEDEXPRESSION&gt; VarDecls Expression&lt;/QUANTIFIEDEXPRESSION&gt;}</td>
</tr>
<tr>
<td>LetExpression</td>
<td>\texttt{LET LetDeclarations IN Expression}</td>
<td>\texttt{&lt;LETEXPRESSION&gt; LetDeclarations Expression&lt;/LETEXPRESSION&gt;}</td>
</tr>
<tr>
<td>LetDeclarations</td>
<td>{LetDeclaration}^+</td>
<td>{LetDeclaration}^+</td>
</tr>
<tr>
<td>LetDeclaration</td>
<td>\texttt{Identifier : Type = Expression}</td>
<td>\texttt{&lt;LETDECLARATION&gt; Identifier Type Expression&lt;/LETDECLARATION&gt;}</td>
</tr>
<tr>
<td>SetExpression</td>
<td>\texttt{SetListExpression}</td>
<td>\texttt{SetListExpression}</td>
</tr>
<tr>
<td></td>
<td>\texttt{</td>
<td>SetPredExpression}</td>
</tr>
<tr>
<td>SetPredExpression</td>
<td>{Identifier : Type</td>
<td>Expression }</td>
</tr>
<tr>
<td>SetListExpression</td>
<td>{Expression}^+</td>
<td>\texttt{&lt;SETLISTEXPRESSION&gt; Expression^+&lt;/SETLISTEXPRESSION&gt;}</td>
</tr>
<tr>
<td>ArrayLiteral</td>
<td>\texttt{[ [ IndexVarDecl ] Expression ]}</td>
<td>\texttt{&lt;ARRAYLITERAL&gt; IndexVarDecl Expression&lt;/ARRAYLITERAL&gt;}</td>
</tr>
<tr>
<td>RecordLiteral</td>
<td>\texttt{( # {RecordEntry}^+ # )}</td>
<td>\texttt{&lt;RECORDLITERAL&gt; {RecordEntry}^+&lt;/RECORDLITERAL&gt;}</td>
</tr>
<tr>
<td>RecordEntry</td>
<td>\texttt{Identifier := Expression}</td>
<td>\texttt{&lt;RECORDENTRY&gt; Identifier Expression&lt;/RECORDENTRY&gt;}</td>
</tr>
<tr>
<td>Nonterminal</td>
<td>SAL</td>
<td>XML</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>TupleLiteral</td>
<td>( {Expression}^+ )</td>
<td>(&lt;\text{TUPLELITERAL}&gt;)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{Expression}^+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(&lt;\text{TUPLELITERAL}&gt;)</td>
</tr>
<tr>
<td>Conditional</td>
<td>IF Expression</td>
<td>(&lt;\text{CONDITIONAL}&gt;)</td>
</tr>
<tr>
<td></td>
<td>THEN Expression</td>
<td>(\text{Expression})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{Expression})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{Expression})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(&lt;\text{CONDITIONAL}&gt;)</td>
</tr>
<tr>
<td></td>
<td>{ ELSIF Expression THEN Expression }^*</td>
<td>(&lt;\text{CONDITIONAL} \ \text{ELSIF=&quot;YES&quot;&gt;})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{Expression})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{Expression})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{Expression})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(&lt;\text{CONDITIONAL}&gt;)</td>
</tr>
<tr>
<td></td>
<td>ELSE Expression ENDIF</td>
<td>(&lt;\text{CONDITIONAL}&gt;)</td>
</tr>
<tr>
<td>ParenExpression</td>
<td>( Expression )</td>
<td>See Below</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>SAL</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>\textit{BaseModule}</td>
<td>\textit{BaseModule}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{SynchronousComposition}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{AsynchronousComposition}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{MultiSynchronous}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{MultiAsync}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{Hiding}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{NewOutput}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{Renaming}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{ModuleName}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textit{ParenModule}</td>
</tr>
<tr>
<td>BaseModule</td>
<td>\texttt{BEGIN BaseDeclarations END}</td>
<td>\texttt{&lt;BASEMODULE&gt; BaseDeclarations &lt;/BASEMODULE&gt;}</td>
</tr>
<tr>
<td>BaseDeclarations</td>
<td>{BaseDeclaration}^*</td>
<td>{BaseDeclaration}^*</td>
</tr>
<tr>
<td>BaseDeclaration</td>
<td>\textit{InputDecl}</td>
<td>\textit{InputDecl}</td>
</tr>
<tr>
<td></td>
<td>\textit{OutputDecl}</td>
<td>\textit{OutputDecl}</td>
</tr>
<tr>
<td></td>
<td>\textit{GlobalDecl}</td>
<td>\textit{GlobalDecl}</td>
</tr>
<tr>
<td></td>
<td>\textit{LocalDecl}</td>
<td>\textit{LocalDecl}</td>
</tr>
<tr>
<td></td>
<td>\textit{DefDecl}</td>
<td>\textit{DefDecl}</td>
</tr>
<tr>
<td></td>
<td>\textit{InitDecl}</td>
<td>\textit{InitDecl}</td>
</tr>
<tr>
<td></td>
<td>\textit{TransDecl}</td>
<td>\textit{TransDecl}</td>
</tr>
<tr>
<td>Nonterminal</td>
<td>SAL</td>
<td>XML</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><strong>InputDecl</strong></td>
<td>INPUT VarDecs</td>
<td><code>&lt;INPUTDECL&gt;</code> VarDecs <code>&lt;/INPUTDECL&gt;</code></td>
</tr>
<tr>
<td><strong>OutputDecl</strong></td>
<td>OUTPUT VarDecs</td>
<td><code>&lt;OUTPUTDECL&gt;</code> VarDecs <code>&lt;/OUTPUTDECL&gt;</code></td>
</tr>
<tr>
<td><strong>GlobalDecl</strong></td>
<td>GLOBAL VarDecs</td>
<td><code>&lt;GLOBALDECL&gt;</code> VarDecs <code>&lt;/GLOBALDECL&gt;</code></td>
</tr>
<tr>
<td><strong>LocalDecl</strong></td>
<td>LOCAL VarDecs</td>
<td><code>&lt;LOCALDECL&gt;</code> VarDecs <code>&lt;/LOCALDECL&gt;</code></td>
</tr>
<tr>
<td><strong>DefDecl</strong></td>
<td>DEFINITION Definitions</td>
<td><code>&lt;DEFDECL&gt;</code> Definitions <code>&lt;/DEFDECL&gt;</code></td>
</tr>
<tr>
<td><strong>InitDecl</strong></td>
<td>INITIALIZATION <code>{DefinitionOrCommand}</code>;</td>
<td><code>&lt;INITDECL&gt;</code> <code>{DefinitionOrCommand}</code>; <code>&lt;/INITDECL&gt;</code></td>
</tr>
<tr>
<td><strong>TransDecl</strong></td>
<td>TRANSITION <code>{DefinitionOrCommand}</code>;</td>
<td><code>&lt;TRANSDECL&gt;</code> <code>{DefinitionOrCommand}</code>; <code>&lt;/TRANSDECL&gt;</code></td>
</tr>
<tr>
<td>Nonterminal</td>
<td>SAL</td>
<td>XML</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>DefinitionOrCommand</td>
<td>Definition</td>
<td>[ SomeCommands ]</td>
</tr>
<tr>
<td>Definition</td>
<td>SimpleDefinition</td>
<td>ForallDefinition</td>
</tr>
<tr>
<td>SimpleDefinition</td>
<td>Lhs RhsDefinition</td>
<td></td>
</tr>
<tr>
<td>Lhs</td>
<td>Identifier [?] [Access]*</td>
<td>NextVariable [ArraySelection</td>
</tr>
<tr>
<td>Access</td>
<td>ArrayAccess</td>
<td>RecordAccess</td>
</tr>
<tr>
<td>ArrayAccess</td>
<td>[ Expression ]</td>
<td></td>
</tr>
<tr>
<td>RecordAccess</td>
<td>. Identifier</td>
<td></td>
</tr>
<tr>
<td>TupleAccess</td>
<td>. Numerals</td>
<td></td>
</tr>
<tr>
<td>RhsDefinition</td>
<td>RhsExpression</td>
<td>RhsSelection</td>
</tr>
<tr>
<td>RhsExpression</td>
<td>= Expression</td>
<td>&lt;RHSEXPRESSION&gt; Expression &lt;/RHSEXPRESSION&gt;</td>
</tr>
<tr>
<td>RhsSelection</td>
<td>IN Expression</td>
<td>&lt;RHSSELECTION&gt; Expression &lt;/RHSSELECTION&gt;</td>
</tr>
<tr>
<td>ForallDefinition</td>
<td>( FORALL ( VarDecls ) : Definitions )</td>
<td>&lt;FORALLDEFINITION&gt; VarDecls Definitions &lt;/FORALLDEFINITION&gt;</td>
</tr>
<tr>
<td>Definitions</td>
<td>{Definition} +</td>
<td>{Definition} +</td>
</tr>
<tr>
<td>SomeCommands</td>
<td>{SomeCommand} +</td>
<td>{SomeCommand} +</td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>SAL</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SomeCommand</strong></td>
<td><strong>NamedCommand</strong></td>
<td><strong>NamedCommand</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MultiCommand</strong></td>
</tr>
<tr>
<td><strong>NamedCommand</strong></td>
<td></td>
<td><strong>GuardedCommand</strong></td>
</tr>
<tr>
<td></td>
<td>[Identifier : ]</td>
<td></td>
</tr>
<tr>
<td><strong>GuardedCommand</strong></td>
<td></td>
<td><strong>Guard Assignments</strong></td>
</tr>
<tr>
<td><strong>Guard</strong></td>
<td><strong>Expression</strong></td>
<td><strong>Guard Assignments</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Expression</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Guard</strong></td>
</tr>
<tr>
<td><strong>Assignments</strong></td>
<td></td>
<td><strong>Assignments</strong></td>
</tr>
<tr>
<td></td>
<td>{SimpleDefinition} ;</td>
<td></td>
</tr>
<tr>
<td><strong>MultiCommand</strong></td>
<td></td>
<td><strong>VarDecls SomeCommand</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VarDecls SomeCommand</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MultiCommand</strong></td>
</tr>
<tr>
<td><strong>SynchronousComposition</strong></td>
<td><strong>Module Module</strong></td>
<td><strong>SynchronousComposition</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SynchronousComposition</strong></td>
</tr>
<tr>
<td><strong>AsynchronousComposition</strong></td>
<td><strong>Module Module</strong></td>
<td><strong>AsynchronousComposition</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>AsynchronousComposition</strong></td>
</tr>
<tr>
<td><strong>MultiSynchronous</strong></td>
<td></td>
<td><strong>MultiSynchronous</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MultiSynchronous</strong></td>
</tr>
<tr>
<td><strong>MultiAsynchronous</strong></td>
<td></td>
<td><strong>MultiAsynchronous</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MultiAsynchronous</strong></td>
</tr>
<tr>
<td>Nonterminal</td>
<td>SAL</td>
<td>XML</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Hiding</td>
<td>LOCAL {Identifier}^+ IN Module</td>
<td><code>&lt;HIDING&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>{Identifier}^+ Module</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;/HIDING&gt;</code></td>
</tr>
<tr>
<td>NewOutput</td>
<td>OUTPUT VarDecls IN Module</td>
<td><code>&lt;NEWOUTPUT&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>VarDecls Module</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;/NEWOUTPUT&gt;</code></td>
</tr>
<tr>
<td>Renaming</td>
<td>[WITH NewVarDecls]</td>
<td><code>&lt;RENAME&gt;</code></td>
</tr>
<tr>
<td></td>
<td>RENAME Renames IN Module</td>
<td><code>{RENAMES} Renames Module</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;/RENAMES&gt;</code></td>
</tr>
<tr>
<td>NewVarDecls</td>
<td>{InputDecl</td>
<td>OutputDecl</td>
</tr>
<tr>
<td></td>
<td></td>
<td>`{InputDecl</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;/NEWVARDECLS&gt;</code></td>
</tr>
<tr>
<td>Renames</td>
<td>{Lhs TO Lhs}^+</td>
<td><code>&lt;RENAME&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Lhs Lhs</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;/RENAME&gt;</code></td>
</tr>
<tr>
<td>ModuleName</td>
<td>Name [ {Expression}^+ ]</td>
<td><code>&lt;MODULENAME&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Name \{Expression\}^+</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;/MODULENAME&gt;</code></td>
</tr>
<tr>
<td>ParenModule</td>
<td>( Module )</td>
<td><code>See Below</code></td>
</tr>
</tbody>
</table>
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 | SETPREEXPRESSION)">

<!ENTITY % EXPRESSION "(" NAMEEXPR
   | QUALIFIEDNAMEEXPR
   | NEXTOPERATOR
   | NUMERAL

21
APPLICATION
ARRAYSELECTION
RECORDSELECTION
TUPLESELECTION
UPDATESCALING
LAMBDAABSTRACTION
QUANTIFIEDABSTRACTION
LETEXPRESSION
%SETEXPRESSION;
ARRAYLITERAL
RECORDLITERAL
TUPLETUPLE
CONDITIONAL )";

<!ENTITY % MODULE
"( BASEMODULE
| SYNCHRONOUCOMPOSITION
| ASYNCHRONOUSCOMPOSITION
| MULTISYNCHRONOUS
| MULTASYNCHRONOUS
| HIDING
| NEWOUTPUT
| RENAMING
| MODULEINSTANCE
| OBSERVERMODULE)">

<!ENTITY % OBSERVERMODULE
"( BASEOBSERVERMODULE
| SYNCHRONOUSOBSERVERCOMPOSITION
| ASYNCHRONOUSOBSERVERCOMPOSITION
| MULTISYNCHRONOUSOBSERVER
| MULTIASYNCHRONOUSOBSERVER
| OBSERVERHIDING
| OBSERVERNEWOUTPUT

22
<!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 "(MODULEMODULES|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 (#PCDATA)>  
<!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 (NEWVARDECL?,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 OBServerMODULE (%MODULE;,%OBServerMODULE;)>  
<!ELEMENT SCALARETYPE (SCALARELEMENT+)>  
<!ELEMENT SCALARELEMENT (#PCDATA)>  
<!ELEMENT DATATYPE (CONSTRUCTOR+)>  
<!ELEMENT CONSTRUCTOR (IDENTIFIER,ACCESSOR*)>  
<!ELEMENT ACCESSOR (IDENTIFIER,%TYPE;)>  
<!ELEMENT BASICETYPE (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 recordentry (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 SETPREEXPRESSION (IDENTIFIER, %TYPE; , %EXPRESSION;)>  
<!ELEMENT SETLISTEXPRESSION (%EXPRESSION;+)>  
<!ELEMENT CONDITIONAL (%EXPRESSION;, %EXPRESSION;, %EXPRESSION;)>  
<!ATTLIST CONDITIONAL ELISIF (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 MODULEIMPLEMENTES (%MODULE; , %MODULE;)>  
<!ELEMENT MODULEREFINES (%MODULE; , %MODULE;)>  

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

<!ELEMENT BASEOBSERVERMODULE ( OBSERVEDDECL  
   | INPUTDECL  
   | OUTPUTDECL  
   | GLOBALDECL  
   | LOCALDECL  
   | DEFDDECL )>>  

<!ELEMENT SYNCHRONOUSOBSERVERCOMPOSITION (%OBSERVERMODULE; , %OBSERVERMODULE;)>
<!ELEMENT ASYNCHRONOUSOBERVERCOMPOSITION (%OBSERVERMODULE;, %OBSERVERMODULE;)>
<!ELEMENT MULTISYNCHRONOUSOBSERVER (INDEXVARDECL, %OBSERVERMODULE;)>
<!ELEMENT MULTIASYNCHRONOUSOBSERVER (INDEXVARDECL, %OBSERVERMODULE;)>
<!ELEMENT OBSERVERHIDING (IDENTIFIER+, %OBSERVERMODULE;)>
<!ELEMENT OBSERVERNEWOUTPUT (VARDECL+, %OBSERVERMODULE;)>
<!ELEMENT OBSERVERRENAMEING (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}; % scalar type
  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];
  tuftype: 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
  ELSEIF 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);
ob1: OBSERVER_MODULE =
BEGIN
OBSERVED y\textsubscript{1}, y\textsubscript{2}: NATURAL,
   pc\textsubscript{1}, pc\textsubscript{2}: Control
LOCAL pc\textsubscript{1}, pc\textsubscript{2}: Control
OUTPUT y\textsubscript{1}\textsubscript{a}: bool,
    y\textsubscript{2}\textsubscript{a}: bool,
    y\textsubscript{1}\_y\textsubscript{2}: bool
DEFINITION
    y\textsubscript{1}\textsubscript{a} = (y\textsubscript{1} = 0);
    y\textsubscript{2}\textsubscript{a} = (y\textsubscript{2} = 0);
    y\textsubscript{1}\_y\textsubscript{2} = (y\textsubscript{1} < y\textsubscript{2})
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 = ctxINTEGER;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>
          <TPNAME>INTEGER</TPNAME>
        </ACCESSOR>
        <ACCESSOR>
          <IDENTIFIER>cdr</IDENTIFIER>
          <TPNAME>list_int</TPNAME>
        </ACCESSOR>
      </CONSTRUCTOR>
    </DATATYPE>
  </TYPEDECLARATION>
  <TYPEDECLARATION>
    <IDENTIFIER>below10</IDENTIFIER>
    <SUBRANGE>
      <NAMEEXPR>_</NAMEEXPR>
      <NUMERAL>10</NUMERAL>
    </SUBRANGE>
  </TYPEDECLARATION>
</CONTEXTBODY>
<TYPEDECLARATION>
  <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>
    <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>ctr</IDENTIFIER>
              </CONTEXTNAME>
            </QUALIFIEDNAMEEXPR>
          </TUPLELITERAL>
        </APPLICATION>
      </TUPLELITERAL>
    </APPLICATION>
  </VARDECLS>
</CONSTANTDECLARATION>
<QUALIFIEDNAMEEXPR>
  <IDENTIFIER>d</IDENTIFIER>
  <CONTEXTNAME>
    <IDENTIFIER>ctx</IDENTIFIER>
    <ACTUALPARAMETERS>
      <ACTUALTYPES></ACTUALTYPES>
    </ACTUALPARAMETERS>
  </CONTEXTNAME>
</QUALIFIEDNAMEEXPR>
</APPLICATION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>c</IDENTIFIER>
  <VAR_DECLS></VAR_DECLS>
  <QUALIFIEDTYPE_NAME>
    <IDENTIFIER>sometype</IDENTIFIER>
  </QUALIFIEDTYPE_NAME>
  <CONTEXTNAME>
    <IDENTIFIER>ctx</IDENTIFIER>
    <ACTUALPARAMETERS>
      <ACTUALTYPES>
        <TYPE_NAME>t</TYPE_NAME>
      </ACTUALTYPES>
      <ACTUAL_EXPRS>
        <NUMERAL>0</NUMERAL>
      </ACTUAL_EXPRS>
    </ACTUALPARAMETERS>
  </CONTEXTNAME>
</QUALIFIEDTYPE_NAME>
</CONSTANTDECLARATION>
</CONSTANTDECLARATION>
<IDENTIFIER>d</IDENTIFIER>
</VARDECLS>
<QUALIFIEDTYPENAME>
  <IDENTIFIER>another_type</IDENTIFIER>
  <CONTEXTNAME>
    <IDENTIFIER>ctx2</IDENTIFIER>
  </CONTEXTNAME>
</QUALIFIEDTYPENAME>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>++</IDENTIFIER>
  <VARDECLS>
    <VARDECL CHAIN="YES">
      <IDENTIFIER>a</IDENTIFIER>
      <TYPE>INTEGER</TYPE>
    </VARDECL>
    <VARDECL>
      <IDENTIFIER>b0</IDENTIFIER>
      <TYPE>INTEGER</TYPE>
    </VARDECL>
  </VARDECLS>
  <TYPE>BOOLEAN</TYPE>
</CONSTANTDECLARATION>
<APPLICATION INFIX="YES">
  <NAMEEXPR>&lt;/NAMEEXPR>
  <TUPLELITERAL>
    <NAMEEXPR>a</NAMEEXPR>
    <APPLICATION>
      <NAMEEXPR>b</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>
  </TUPLELITERAL>
</APPLICATION>
</APPLICATION>
</APPLICATION>
<IDENTIFIER>a</IDENTIFIER>
</RECORDSELECTION>
<NUMERAL>13</NUMERAL>
</TUPLESELECTION>
<NUMERAL>11</NUMERAL>
</UPDATEEXPRESSION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
<IDENTIFIER>r</IDENTIFIER>
</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>ct</IDENTIFIER>
<APPLICATION>
<NAMEEXPR>−</NAMEEXPR>
</TUPLELITERAL>
<NUMERAL>17</NUMERAL>
</TUPLELITERAL>
<APPLICATION>
</RECORDDENTRY>
</RECORDLITERAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>t</IDENTIFIER>
</VARDECLS>
<TYPENAME>tuptype</TYPENAME>
<TUPLELITERAL>
  <NAMEEXPR>true</NAMEEXPR>
</LAMBDAABSTRACTION>
<VARDECL>
  <IDENTIFIER>x</IDENTIFIER>
</VARDECL>
<TYPENAME>REAL</TYPENAME>
</APPLICATION>
<APPLICATION>
  <NAMEEXPR>x</NAMEEXPR>
</TUPLELITERAL>
<TYPENAME>17</TYPENAME>
</APPLICATION>
</TUPLELITERAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>a</IDENTIFIER>
</VARDECLS>
<TYPENAME>arr</TYPENAME>
<ArrayLITERAL>
  <INDEXVARDECL>
    <IDENTIFIER>i</IDENTIFIER>
    <TYPENAME>INTEGER</TYPENAME>
  </INDEXVARDECL>
<ARRAYLITERAL>
  <INDEXVARDDECL>
    <IDENTIFIER>x</IDENTIFIER>
    <TYPENAME>foo</TYPENAME>
  </INDEXVARDDECL>
  <NUMERAL>0</NUMERAL>
</ARRAYLITERAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>fb</IDENTIFIER>
  <VARDCLS></VARDCLS>
  <TYPENAME>BOOLEAN</TYPENAME>
  <QUANTIFIEDEXPRESSION>
    <QUANTIFIER>FORALL</QUANTIFIER>
    <VARDCL>
      <IDENTIFIER>i</IDENTIFIER>
      <TYPENAME>INTEGER</TYPENAME>
    </VARDCL>
    <APPLICATION>
      <NAMEEXPR>p</NAMEEXPR>
      <TUPLELITERAL>
        <NAMEEXPR>i</NAMEEXPR>
      </TUPLELITERAL>
    </APPLICATION>
  </QUANTIFIEDEXPRESSION>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>c</IDENTIFIER>
  <VARDCLS></VARDCLS>
  <TYPENAME>INTEGER</TYPENAME>
  <LETEXPRESSION>
    <LETDECLARATIONS>
      <LETDECLARATION>
<IDENTIFIER>i</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>23423234234</NUMERAL>
 </LETDECLARATION>
</LETDECLARATIONS>
<APPLICATION INFIX="YES">
 <NAMEEXPR>+</NAMEEXPR>
<TUPLELITERAL>
<APPLICATION>
 <NAMEEXPR>floor</NAMEEXPR>
<TUPLELITERAL>
 <NAMEEXPR>i</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/>
  <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/>
  <TYPENAME>BOOLEAN</TYPENAME>
  <CONDITIONAL>
    <NAMEEXPR>a</NAMEEXPR>
    <NAMEEXPR>b</NAMEEXPR>
    <CONDITIONAL_ELSEIF>"YES"
      <NAMEEXPR>c</NAMEEXPR>
      <NAMEEXPR>d</NAMEEXPR>
      <NAMEEXPR>e</NAMEEXPR>
    </CONDITIONAL_ELSEIF>
  </CONDITIONAL>
</CONSTANTDECLARATION>
<CONSTANTDECLARATION>
  <IDENTIFIER>parexpr</IDENTIFIER>
  <VARDECLS/>
  <TPNAME>INTEGER</TPNAME>
  <TUPLELITERAL>
    <TUPLELITERAL>
      <TUPLELITERAL>
        <NUMERAL>3</NUMERAL>
      </TUPLELITERAL>
    </TUPLELITERAL>
  </TUPLELITERAL>
</CONSTANTDECLARATION>

<CONTEXTDECLARATION>
  <IDENTIFIER>foo</IDENTIFIER>
  <CONTEXTNAME>
    <IDENTIFIER>ctx</IDENTIFIER>
    <ACTUALPARAMETERS>
      <ACTUALTYPES>
        <TPNAME>t</TPNAME>
      </ACTUALTYPES>
      <ACTUALEXPRS>
        <NUMERAL>0</NUMERAL>
      </ACTUALEXPRS>
    </ACTUALPARAMETERS>
  </CONTEXTNAME>
</CONTEXTDECLARATION>

<MODULEDECLARATION>
  <IDENTIFIER>m</IDENTIFIER>
  <VARDECLS/>
  <BASEMODULE>
    <INPUTDECL>
      <VARDECL>
        <IDENTIFIER>x</IDENTIFIER>
        <TPNAME>INTEGER</TPNAME>
      </VARDECL>
    </INPUTDECL>
  </BASEMODULE>
</MODULEDECLARATION>
<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>

<FORALLDDEFINITION>

<VARDECL>
<IDENTIFIER>n</IDENTIFIER>
<TYPENAME>INTEGER</TYPENAME>
</VARDECL>

<SIMPLEDEFINITION>
<ARRAYSELECTION>
<NAMEEXPR>z</NAMEEXPR>
<NAMEEXPR>n</NAMEEXPR>
</ARRAYSELECTION>

<RHSEXPRESSION>
<NUMERAL>0</NUMERAL>
</RHSEXPRESSION>
</SIMPLEDEFINITION>
</FORALLDEFINITION>

</VARDECL>
  <IDENTIFIER>r</IDENTIFIER>
  <TYPE_NAME>REAL</TYPE_NAME>
</VARDECL>

</SIMPLEDEFINITION>
  <NAMEEXPR>y</NAMEEXPR>
  <RHSEXPRESSION>
    <APPLICATION>
      <NAMEEXPR>s</NAMEEXPR>
      <TUPLE_LITERAL>
        <NAMEEXPR>r</NAMEEXPR>
      </TUPLE_LITERAL>
    </APPLICATION>
  </RHSEXPRESSION>
</SIMPLEDEFINITION>
</FORALLDEFINITION>
</FORALLDEFINITION>
</DEFDECL>

</SIMPLEDEFINITION>
  <NAMEEXPR>x</NAMEEXPR>
  <RHSEXPRESSION>
    <NUMERAL>0</NUMERAL>
  </RHSEXPRESSION>
</SIMPLEDEFINITION>

</SIMPLEDEFINITION>
  <SOMECOMMANDS>
    <GUARDEDCOMMAND>
      <GUARD>
        <NAMEEXPR>a</NAMEEXPR>
      </GUARD>
    </GUARDEDCOMMAND>
  </SOMECOMMANDS>

</INITDECL>

</SIMPLEDEFINITION>

</SIMPLEDEFINITION>

</SIMPLEDEFINITION>

</SIMPLEDEFINITION>

</SIMPLEDEFINITION>

</SIMPLEDEFINITION>

</ASSIGNMENTS>
  <SIMPLEDEFINITION>

49
<NAMEEXPR>n</NAMEEXPR>
</VARDECL>
</GUARDEDCOMMAND>
</GUARD>
</ASSIGNMENTS>
</SIMPLEDEFINITION>
<ARRAYSELECTION>
</NEXTOPERATOR>
</NAMEEXPR>n</NAMEEXPR>
</NEXTOPERATOR>
</NAMEEXPR>n</NAMEEXPR>
</ARRAYSELECTION>
</RHSEXPRESSION>
</SIMPLEDEFINITION>
</ASSIGNMENTS>
</GUARDEDCOMMAND>
</MULTICOMMAND>
</SOMECOMMANDS>
</TRANSDECL>
</BASEMODULE>
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</SYNCHRONOUSCOMPOSITION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</VARDECLS>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</SYNCHRONOUSCOMPOSITION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</MODULEDECLARATION>
<br />
</VARDECLS>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
</VARDECL>
<br />
52
<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>
      <TYPE>INTEGER</TYPE>
    </VARDECL>
    <MODULEINSTANCE>
      <MODULENAME>m1</MODULENAME>
    </MODULEINSTANCE>
</NEWOUTPUT>
<MODULEACTUALS></MODULEACTUALS>
</MODULEINSTANCE>
</NEWOUTPUT>
</MODULEDECLARATION>
<MODULEDECLARATION>
<IDENTIFIER>m4</IDENTIFIER>
<VARDECLS></VARDECLS>
<RENAME>
<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>
</MODULEDECLARATION>
<MODULEDECLARATION>
<IDENTIFIER>m5</IDENTIFIER>
<VARDECLS></VARDECLS>
<RENAME>
<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>
</RENAME>
<RENAME>
   <ARRAYSELECTION>
      <NAMEEXPR>a</NAMEEXPR>
      <NUMERAL>1</NUMERAL>
   </ARRAYSELECTION>
   <NAMEEXPR>y</NAMEEXPR>
</RENAME>
</MODULEINSTANCE>
</RENAMING>
</MODULEDECLARATION>
</MODULEDECLARATION>
<IDENTIFIER>m0</IDENTIFIER>
</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>
      <IDENTIFIER>m9</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>
      </MODULEINSTANCE>
    </SYNCHRONOUSCOMPOSITION>
</MODULEDECLARATION>
<MODULEDECLARATION>
  <IDENTIFIER>m10</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <ASYNCHRONOUSCOMPOSITION>
    <MULTISYNCHRONOUS>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TYPOENAME>t</TYPOENAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
    </MULTISYNCHRONOUS>
  </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>
    </TUPLELITERAL>
  </APPLICATION>
  </APPLICATION>
</MODULEMODELS>
</ASSERTIONPROPOSITION>
<ASSERTIONDECLARATION>
  <IDENTIFIER>a2</IDENTIFIER>
  <ASSERTIONFORM>CLAIM</ASSERTIONFORM>
  <ASSERTIONPROPOSITION>
    <ASSERTIONOPERATOR>AND</ASSERTIONOPERATOR> <ASSERTIONOPERATOR>NOT</ASSERTIONOPERATOR>
  </MODULEMODELS>
  <MODULEINSTANCE>
    <MODULENAME>m1</MODULENAME>
    <MODULEACTUALS></MODULEACTUALS>
  </MODULEINSTANCE>
  <APPLICATION>
    <NAMEEXPR>AG</NAMEEXPR>
  </APPLICATION>
</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>
    <TYPECELL>INTEGER</TYPECELL>
  </VARDECL>
  <MODULEMODELS>
    <MODULEINSTANCE>
      <MODULENAME>m</MODULENAME>
      <MODULEACTUALS>
        <NAMECELL>x</NAMECELL>
      </MODULEACTUALS>
    </MODULEINSTANCE>
    <APPLICATION INFIX="YES">
      <NAMECELL>p</NAMECELL>
      <TUPLELITERAL>
        <NAMECELL>EF</NAMECELL>
        <TUPLELITERAL>
          <NAMECELL>x</NAMECELL>
        </TUPLELITERAL>
      </TUPLELITERAL>
    </APPLICATION>
  </MODULEMODELS>
</QUANTIFIEDASSERTION>
</ASSERTIONDECLARATION>

<OBERVERMODULEDECLARATION>
  <IDENTIFIER>obs1</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <BASEOBERVERMODULE>
    <OBERVERDECL>
      <VARDECL CHAIN="YES">
        <IDENTIFIER>y1</IDENTIFIER>
        <TYPECELL>NATURAL</TYPECELL>
      </VARDECL>
    </OBERVERDECL>
  </BASEOBERVERMODULE>
</OBERVERMODULEDECLARATION>

61
<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>
  </VARDECL>

</OUTPUTDECL>
<TYPENAME>bool</TYPENAME>
</VARDECL>
</OUTPUTDECL>
<DEFDECL>
<SIMPLEDEFINITION>
 <NAMEEXPR>y1a</NAMEEXPR>
 <RHSEXPRESSION>
  <TUPLELITERAL>
   <APPLICATION INFIX="YES">
    <NAMEEXPR>=</NAMEEXPR>
   </APPLICATION>
   <TUPLELITERAL>
    <NAMEEXPR>y1</NAMEEXPR>
    <NUMERAL>0</NUMERAL>
   </TUPLELITERAL>
  </TUPLELITERAL>
 </RHSEXPRESSION>
</SIMPLEDEFINITION>
<SIMPLEDEFINITION>
 <NAMEEXPR>y2a</NAMEEXPR>
 <RHSEXPRESSION>
  <TUPLELITERAL>
   <APPLICATION INFIX="YES">
    <NAMEEXPR>=</NAMEEXPR>
   </APPLICATION>
   <TUPLELITERAL>
    <NAMEEXPR>y2</NAMEEXPR>
    <NUMERAL>0</NUMERAL>
   </TUPLELITERAL>
  </TUPLELITERAL>
 </RHSEXPRESSION>
</SIMPLEDEFINITION>
<SIMPLEDEFINITION>
 <NAMEEXPR>y1 - y2</NAMEEXPR>
</SIMPLEDEFINITION>
<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>
</OBSERVERMODULEDECLARATION>

<VARDECLS>
  <VARDECL>
    <IDENTIFIER>x</IDENTIFIER>
    <TYPE_NAME>INTEGER</TYPE_NAME>
  </VARDECL>
</VARDECLS>

<ASYNCHRONOUSOBSERVERCOMPOSITION>
  <MODULE_INSTANCE>
    <MODULE_NAME>m2</MODULE_NAME>
    <MODULE_ACTUALS></MODULE_ACTUALS>
  </MODULE_INSTANCE>
  <MODULE_INSTANCE>
    <MODULE_NAME>m3</MODULE_NAME>
    <MODULE_ACTUALS></MODULE_ACTUALS>
  </MODULE_INSTANCE>
</ASYNCHRONOUSOBSERVERCOMPOSITION>

</OBSERVERMODULEDECLARATION>

<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om2</IDENTIFIER>
</OBSERVERMODULEDECLARATION>

<VARDECLS/>

<OBSERVERHIDING>
  <IDENTIFIER>a</IDENTIFIER>
  <IDENTIFIER>b</IDENTIFIER>
  <IDENTIFIER>c</IDENTIFIER>
  <MODULE_INSTANCE>
    <MODULE_NAME>m1</MODULE_NAME>
    <MODULE_ACTUALS></MODULE_ACTUALS>
  </MODULE_INSTANCE>
</OBSERVERHIDING>

</OBSERVERMODULEDECLARATION>

<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om3</IDENTIFIER>
</OBSERVERMODULEDECLARATION>
<VARDECL/></VARDECL>
<ObserverNewOutput>
  <VARDECL>
    <IDENTIFIER>x</IDENTIFIER>
    <TYPE_NAME>INTEGER</TYPE_NAME>
  </VARDECL>
  <ModuleInstance>
    <ModuleName>m1</ModuleName>
  </ModuleInstance>
</ObserverNewOutput>
</ObserverModuleDeclaration>
<Identifier>om4</Identifier>
<VARDECLs/></VARDECLs>
<ObserverRenaming>
  <Rename>
    <ArraySelection>
      <NameExpr>a</NameExpr>
      <Numeral>0</Numeral>
    </ArraySelection>
  </Rename>
  <Rename>
    <ArraySelection>
      <NameExpr>a</NameExpr>
      <Numeral>1</Numeral>
    </ArraySelection>
  </Rename>
</ObserverRenaming>
</ObserverModuleDeclaration>
<ModuleInstance>
  <ModuleName>m1</ModuleName>
</ModuleInstance>
</ModuleInstance>
</ModuleInstance>

66
<OBSERVERRENAMEING>
</OBSERVERRENAMEING>
<OBSERVERVERIFICATION>
</OBSERVERVERIFICATION>
<IDENTIFIER>om5</IDENTIFIER>
<VARDECLS></VARDECLS>
<OBSERVERRENAMEING>
<NEWVARDECLS>
<INPUTDECL>
<VARDECL>
<IDENTIFIER>a</IDENTIFIER>
<ARRAYTYPE>
<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>
</MODULEINSTANCE>
</MODULEINSTANCE>
</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>
<MODULEDECLARATION>
<IDENTIFIER>om8</IDENTIFIER>
<VARDECLS></VARDECLS>
<SYNCHRONOUSOBSERVERCOMPOSITION>
<ASYNCHRONOUSOBSERVERCOMPOSITION>
<MODULEINSTANCE>
<MODULENAME>m1</MODULENAME>
<MODULEACTUALS></MODULEACTUALS>
</MODULEINSTANCE>
</ASYNCHRONOUSOBSERVERCOMPOSITION>
</MODULEDECLARATION>
<MODULEDECLARATION>
<IDENTIFIER>om9</IDENTIFIER>
<VARDECLS></VARDECLS>
<ASYNCHRONOUSOBSERVERCOMPOSITION>
<MODULEINSTANCE>
<MODULENAME>m1</MODULENAME>
<MODULEACTUALS></MODULEACTUALS>
</MODULEINSTANCE>
</ASYNCHRONOUSOBSERVERCOMPOSITION>
</MODULEDECLARATION>
<MODULEDECLARATION>
<IDENTIFIER>om9</IDENTIFIER>
<VARDECLS></VARDECLS>
<ASYNCHRONOUSOBSERVERCOMPOSITION>
<MODULEINSTANCE>
<MODULENAME>m1</MODULENAME>
<MODULEACTUALS></MODULEACTUALS>
</MODULEINSTANCE>
</ASYNCHRONOUSOBSERVERCOMPOSITION>
</MODULEDECLARATION>
</MODULEINSTANCE>

</MODULEINSTANCE>

<MODULEINSTANCE>
  <MODULENAME>m3</MODULENAME>
  <MODULEACTUALS></MODULEACTUALS>
</MODULEINSTANCE>

</SYNCHRONOUSOBSERVERCOMPOSITION>
</ASYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>

<OBSERVERMODULEDECLARATION>
  <IDENTIFIER>om10</IDENTIFIER>
  <VARDECLS></VARDECLS>
  <ASYNCHRONOUSOBSERVERCOMPOSITION>
    <MULTISYNCHRONOUSOBSERVER>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TPNAME>t</TPNAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
    </MULTISYNCHRONOUSOBSERVER>
    <MULTIASYNCHRONOUSOBSERVER>
      <INDEXVARDECL>
        <IDENTIFIER>i</IDENTIFIER>
        <TPNAME>t</TPNAME>
      </INDEXVARDECL>
      <MODULEINSTANCE>
        <MODULENAME>m0</MODULENAME>
        <MODULEACTUALS>
          <NAMEEXPR>i</NAMEEXPR>
        </MODULEACTUALS>
      </MODULEINSTANCE>
    </MULTIASYNCHRONOUSOBSERVER>
    <MODULEINSTANCE>
      <MODULENAME>m0</MODULENAME>
      <MODULEACTUALS>
        <NAMEEXPR>i</NAMEEXPR>
      </MODULEACTUALS>
    </MODULEINSTANCE>
  </ASYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>

70
</MODULEINSTANCE>
</MULTIASYNCHRONOUSOBSERVER>
</ASYNCHRONOUSOBSERVERCOMPOSITION>
</OBSERVERMODULEDECLARATION>
</CONTEXTBODY>
</CONTEXT>