THESIS TITLE
SECOND LINE IF NECESSARY

by

AUTHOR NAME

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Master of Science

Queen’s University
Kingston, Ontario, Canada
September 2003

Copyright © Author Name, 2003
Abstract

This is my abstract.
Acknowledgments

Blah blah blah.
Contents

Abstract i
Acknowledgments ii
Contents iii
List of Tables v
List of Figures vi

1 Introduction 1
1.1 Section ........................................... 1
  1.1.1 SubSection .................................. 1
1.2 Motivation ...................................... 1
1.3 Problem ........................................ 2
1.4 Objective ...................................... 2
  1.4.1 Hypothesis .................................. 2
1.5 Contributions .................................. 2
1.6 Organization of Thesis ......................... 2

2 Background 3
2.1 Conformance Checking ......................... 3
  2.1.1 Multiple Definitions ....................... 3

3 Alloy 5
3.1 The Alloy Language ............................. 5
  3.1.1 Example .................................... 6

4 Embee: User Perspective 7
  4.0.2 Phase 1: High-Level Static Mapping ......... 8
List of Tables

5.1 Running times for each phase and total running time of Embee . . . . 13
B.1 Example arity calculations . . . . . . . . . . . . . . . . . . . . . . 17
B.2 Estimate of Boolean formula size . . . . . . . . . . . . . . . . . . . . . 20
B.3 Test series for evaluating the running time of conformance checking . 21
List of Figures

4.1 Excerpt of high-level static mapping file .......................... 8
4.2 Visualization of tree ...................................................... 8
4.3 Visualization of tree after deletion of root (error or omission) . . . 9
5.1 Sample specification and implementation ............................ 11
B.1 Equations to compute arity of relations ......................... 18
Chapter 1

Introduction

1.1 Section

1.1.1 SubSection

SubSubSection

Paragraph

SubParagraph

1.2 Motivation

...the current de facto standard being the Unified Modeling Language (UML) [1]...
1.3 Problem

1.4 Objective

Note: These are the section headings that I decided to use. Check out several recent theses to decide how you want to lay out your introduction (and conclusion) chapters.

1.4.1 Hypothesis

1.5 Contributions

1.6 Organization of Thesis

We proceed by introducing conformance checking and discussing related work in the next chapter. We discuss the Alloy language and the Alloy Analyzer tool in Chapter 3. Chapter 4 describes our Embee tool, from the user’s perspective, with several running examples. Implementation details and the analysis of the tool are presented in Chapter 5. Chapter 6 concludes and outlines future work.
Chapter 2

Background

2.1 Conformance Checking

2.1.1 Multiple Definitions

- checking “whether an implementation conforms to some given design” [3]
- ensuring “that the actual software (the detailed design and code) conforms to the architecture” [2]
- etc. etc.

For our research, we are adopting the following definitions of conformance checking:

**Conformance checking is the process of comparing...**

We further refine our definition with the following caveats:

1. Our version of conformance checking .
2. We distinguish between checking and ensuring...

Don’t forget to discuss related work!!!
Chapter 3

Alloy

3.1 The Alloy Language

Alloy is...

Quantifiers  There are five quantifiers available in Alloy:

<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>all x : e</td>
<td>F</td>
</tr>
<tr>
<td>some x : e</td>
<td>F</td>
</tr>
<tr>
<td>no x : e</td>
<td>F</td>
</tr>
<tr>
<td>sole x : e</td>
<td>F</td>
</tr>
<tr>
<td>one x : e</td>
<td>F</td>
</tr>
</tbody>
</table>

Signatures and Fields

The simple signature sig A {} introduces A as a basic type with a set of atoms of that type. A refers to the set of atoms; the type is inferred by Alloy and cannot be referenced explicitly.

    sig A {}
CHAPTER 3. ALLOY

sig B {
    f : A
}

3.1.1 Example

An excerpt from an Alloy specification of a singly-linked list is presented in Listing 3.1.

Listing 3.1 Excerpt of a simple Alloy specification for a singly-linked list

sig Node {
    next : option Node
}
sig List {
    first : Node
}

{ all n : Node | n in first.*next
  no n : Node | n in n.^next
}
Chapter 4

Embee: User Perspective

Listing 4.1 Alloy specification of a singly-linked list using only binary relations

module List

  sig Node {
    next : option Node
  }

  sig List {
    first : Node
  }

  fact NodeInOneList {
    all n : Node | one l : List | n in (l.first).*next
  }

  fact NoCycle {
    all n : Node | n ! in n.^next
  }

  fun Show() {}

  run Show for 4
4.0.2 Phase 1: High-Level Static Mapping

Phase 1 simply generates the default static mapping and presents it to the user, as shown in Figure 4.1(a). We have modified the map file as shown in Figure 4.1(b).

List = List
List$first = List.first
Node = Node
Node$next = Node.next

(a) Default static mapping

List = SimpleList
List$first = SimpleList.first
Node = Node
Node$next = Node.next

(b) Modified static mapping

Figure 4.1: Excerpt of high-level static mapping file before and after modification

Figure 4.2 shows the tree before and after deletion, with correctly implemented code. Figure 4.3 on the next page shows the tree after deletion of the root, when the root = n2 statement is not executed.

Figure 4.2: Visualization of tree before and after correct deletion of the root node
Figure 4.3: Visualization of tree after deletion of the root node, with an error of omission in the code. Node_7 represents the temporary node in the swapNodes() method.
Chapter 5

Embee: Implementation and Analysis

Listing 5.1 Excerpt from StateDumperThreads.java, showing how to connect to a second virtual machine executing the target code. In this example, the target class is referenced by javaClassName. The JPDA classes can be accessed by including the tools.jar archive in the program’s classpath; this archive is found in the Java installation’s lib directory

```
//com.sun.jdi.connect.LaunchingConnector;
... 
LaunchingConnector connect = Bootstrap.virtualMachineManager().defaultConnector();
Map connectorArguments = connect.defaultArguments();
Connector.Argument main = (Connector.Argument) connectorArguments).get("main");
main.setValue(javaClassName);
...
VirtualMachine vm = connect.launch(connectorArguments);
```
5.1 Complexity and Performance

5.1.1 Definition of Terms

The following terms...:

- **scope**: The maximum number of objects...
- **$R$**: The number of relations...
- **$r_i$**: The $i^{th}$ relation in the specification, $1 \leq i \leq R$.
- **arity($r_i$)**: The arity of relation $r_i$...
- **$N$**: The total number...

Given the calculated arities of a particular specification’s relations, and the scope at a specific breakpoint, Equation 5.1 can be used to determine $N$.

$$N = S \times scope + \sum_{i=1}^{R} scope^{arity(r_i)}$$  \hspace{1cm} (5.1)
The combined complexity of all four steps is

\[ O(N) + O(nN) + O(N^2) + O(F) \]

Again, these steps are completed once for every breakpoint in the target program’s execution; therefore, the overall upper bound becomes

\[ b \times O(N) + b \times O(nN) + b \times O(N^2) + b \times O(F) = O(bN + bnN + bN^2 + bF) \]

The vector \([x_0 \ x_1]\) represents the two possible atoms of type \(X\). With our naming scheme, \(x_0\) represents \(X_0\) and \(x_1\) represents \(X_1\). The binary relation itself is represented by a two-dimensional bit matrix where a 1 in position \([i,j]\) means that there is a mapping between the \(i^{th}\) atom of \(X\) and the \(j^{th}\) atom of \(Y\):

\[
\begin{bmatrix}
  r_{00} & r_{01} \\
  r_{10} & r_{11}
\end{bmatrix}
\begin{bmatrix}
  X_0 \rightarrow Y_0 & X_0 \rightarrow Y_1 \\
  X_1 \rightarrow Y_0 & X_1 \rightarrow Y_1
\end{bmatrix}
\]

Now, consider a fact stating that relation \(r\) is total, i.e.,

all \(x : X \mid \text{some } y : Y \mid x.r = y\)

The CNF formula for our example fact, in scope 2, is

\[
\neg((x_0 \land r_{00}) \lor (x_1 \land r_{10})) \land \neg((x_0 \land r_{01}) \lor (x_1 \land r_{11})) \land
\neg(\neg((x_0 \land r_{00}) \lor (x_1 \land r_{10})) \lor ((x_0 \land r_{01}) \lor (x_1 \land r_{11})))
\]

\[
\neg(\neg((x_0 \land r_{00}) \lor (x_1 \land r_{10})) \lor ((x_0 \land r_{01}) \lor (x_1 \land r_{11})))
\]
Table 5.1 contains...

Table 5.1: Running times for each phase and total running time of Embee

<table>
<thead>
<tr>
<th>Object Model</th>
<th>Scope</th>
<th>Number of Breakpoints</th>
<th>Test Case</th>
<th>Running Time (m:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Phase 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>First 16</td>
<td>Last 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List</td>
<td>20</td>
<td>20</td>
<td>0:07</td>
<td>0:32</td>
</tr>
<tr>
<td>Graph</td>
<td>20</td>
<td>19a</td>
<td>0:07</td>
<td>1:27</td>
</tr>
<tr>
<td>Tree</td>
<td>20</td>
<td>20</td>
<td>0:04</td>
<td>1:20</td>
</tr>
</tbody>
</table>

* Breakpoints occur after the addition of each edge, i.e., the first breakpoint does not occur until the second node is added.

...upper bound on Embee’s performance:

$$\text{upper bound is } \begin{cases} O(bN^2) & \text{if } scope \leq 16 \\ O(bF) & \text{if } scope > 16 \end{cases}$$
Chapter 6

Summary and Conclusions

6.1 Summary

6.2 Future Work

6.3 Conclusion
Bibliography


Appendix A

Alloy Analyzer

A.1 Documentation

Package: alloy.api

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlloyRunner</td>
<td>This class provides</td>
</tr>
<tr>
<td></td>
<td>To do an analysis...</td>
</tr>
<tr>
<td>analyzeCommand</td>
<td>Run the actual...</td>
</tr>
<tr>
<td>prepareSpec</td>
<td>Parse...</td>
</tr>
<tr>
<td>translateCommand</td>
<td>Translate...</td>
</tr>
</tbody>
</table>
Appendix B

Additional Analysis

B.1 Calculation of Arity

Examples of arity calculations are shown in Table B.1. These calculations can be performed using either the equations listed in Figure B.1 on the next page...

Table B.1: Example arity calculations

<table>
<thead>
<tr>
<th>relation $r_i$</th>
<th>$arity(r_i)$</th>
<th>arity equations used</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f : A$</td>
<td>2</td>
<td>(B.1), (B.2a)</td>
</tr>
<tr>
<td>$f : \text{option A}$</td>
<td>2</td>
<td>(B.1), (B.2a)</td>
</tr>
<tr>
<td>$f : A \to A$</td>
<td>3</td>
<td>(B.1), (B.2b), (B.3a)</td>
</tr>
<tr>
<td>$f : A \to ? B$</td>
<td>3</td>
<td>(B.1), (B.2b), (B.3a)</td>
</tr>
<tr>
<td>$f : A \to B \to C$</td>
<td>4</td>
<td>(B.1), (B.2b), (B.3c), (B.3a)</td>
</tr>
<tr>
<td>$f : A \to B ? \to ! C$</td>
<td>4</td>
<td>(B.1), (B.2b), (B.3a), (B.3c)</td>
</tr>
<tr>
<td>$f : A \to B \to C \to D$</td>
<td>5</td>
<td>(B.1), (B.2b), (B.3c), (B.3a)</td>
</tr>
</tbody>
</table>
Given:

- \( v \) is a variable of type \(<\text{var}>\), i.e., an \( \text{id} \) (identifier)
- \( m \) is a multiplicity expression of type \(<\text{multexpr}>\)
- \( r \) is a relation of type \(<\text{relation}>\), i.e., \( r = v : m \)
- \( e_1, e_2, \ldots \) are expressions of type \(<\text{expr}>\) in \( m \)
- \( x \) is an optional set multiplicity modifier of type \(<\text{setmult}>\)
- \( y, z \) are optional relation multiplicity modifiers of type \(<\text{mult}>\)

The arity equations are:

\[
\begin{align*}
\text{arity}(r) &= 1 + \text{arity}(m), \text{ where } r \text{ is of the form } v : m \quad (B.1) \\
\text{arity}(m) &= \begin{cases} 
1 & \text{if } m \text{ is of the form } x v \\
\text{arity}(e_1) + \text{arity}(e_2) & \text{if } m \text{ is of the form } e_1 y \rightarrow z e_2 \ (B.2b) \\
\end{cases} \\
\text{arity}(e) &= \begin{cases} 
1 & \text{if } e \text{ is of the form } \text{id} \\
\text{arity}(e_1) & \text{if } e \text{ is of the form } (e_1) \\
\text{arity}(e_1) + \text{arity}(e_2) & \text{if } e \text{ is of the form } e_1 \rightarrow e_2 \ (B.3c) \\
\end{cases}
\end{align*}
\]

Figure B.1: Equations to compute arity of relations

### B.2 Comparison of \( N \)

#### B.2.1 Reasoning about \( N \) in terms of \( n \)

It is possible to determine an upper bound on the size of \( N \), relative to the size of \( n \).

To do this, we re-examine Equation 5.1.

From Equation 5.1, we have:

\[
N = S \times \text{scope} + \sum_{i=1}^{R} \text{scope}^{\text{arity}(r_i)}
\]

In the worst-case, the scope is equal to the total number of objects that exist at
a particular breakpoint, i.e., \( \text{scope} = n \).

\[
N = Sn + \sum_{i=1}^{R} n^{\text{arity}(r_i)}
\]

We can expand the summation to

\[
N = Sn + n^{\text{arity}(r_1)} + n^{\text{arity}(r_2)} + \ldots + n^{\text{arity}(r_R)}
\]

Because...

\[
O(N) = O(Sn) + O(n^{\text{arity}(r_1)}) + O(n^{\text{arity}(r_2)}) + \ldots + O(n^{\text{arity}(r_R)})
\]

We assume that all \( R \) relations in the specification have the same arity, and that this arity is represented by a value \( x \geq 2 \). Therefore...

\[
O(N) = O(Sn) + R \times O(n^x) = O(Sn) + O(Rn^x) \quad (B.4)
\]

Equation B.4 demonstrates...

Because both \( S \) and \( R \) are finite numbers, it is possible to further reduce Equation B.4 to

\[
O(N) = O(n) + O(n^x) = O(n^x) \quad (B.5)
\]

Therefore...
B.3 Estimation of $F$

For example, Table B.2 contains the values of $F$...

Table B.2: Estimate of Boolean formula size, determined by number of Boolean operators (“and”, “or”, “not”)

<table>
<thead>
<tr>
<th>scope</th>
<th>$N$</th>
<th>0 Facts</th>
<th>1 Fact</th>
<th>2 Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>23</td>
<td>34</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>197</td>
<td>657</td>
<td>729</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>671</td>
<td>13,799</td>
<td>15,200</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>1,731</td>
<td>91,435</td>
<td>96,771</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>scope</th>
<th>$N$</th>
<th>Facts</th>
<th>1 Fact</th>
<th>2 Facts</th>
<th>3 Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>185</td>
<td>1,005</td>
<td>1,783</td>
<td>2,181</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
<td>674</td>
<td>66,722</td>
<td>118,250</td>
<td>142,328</td>
</tr>
<tr>
<td>4</td>
<td>88</td>
<td>1,787</td>
<td>635,811</td>
<td>1,153,063</td>
<td>1,319,611</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>scope</th>
<th>$N$</th>
<th>0 Facts</th>
<th>1 Fact</th>
<th>2 Facts</th>
<th>3 Facts</th>
<th>4 Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>39</td>
<td>78</td>
<td>93</td>
<td>103</td>
<td>104</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>367</td>
<td>1,601</td>
<td>2,487</td>
<td>2,629</td>
<td>2,715</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>1,283</td>
<td>38,528</td>
<td>73,472</td>
<td>76,196</td>
<td>76,568</td>
</tr>
<tr>
<td>4</td>
<td>76</td>
<td>3,359</td>
<td>234,595</td>
<td>456,459</td>
<td>466,979</td>
<td>468,087</td>
</tr>
</tbody>
</table>
### B.3.1 Test Series

Table B.3 summarizes the test series for evaluating the running time of conformance checking.

<table>
<thead>
<tr>
<th>Series Name</th>
<th>Example</th>
<th>$S$</th>
<th>$R$</th>
<th>$\text{arity}(r_i)$</th>
<th>Number of Facts</th>
<th>$\text{scope} = n$</th>
<th>$N$</th>
<th>Number of Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1F0</td>
<td>List</td>
<td>2</td>
<td>2</td>
<td>2, 2</td>
<td>0</td>
<td>1,2,...,40</td>
<td>4 - 3,280</td>
<td>40</td>
</tr>
<tr>
<td>E1F1</td>
<td></td>
<td>1</td>
<td></td>
<td>1,2,...,32</td>
<td>1</td>
<td>1,2,...,32</td>
<td>4 - 1,984</td>
<td>32</td>
</tr>
<tr>
<td>E1F2</td>
<td></td>
<td>2</td>
<td></td>
<td>1,2,...,31</td>
<td>2</td>
<td>1,2,...,31</td>
<td>4 - 1,984</td>
<td>31</td>
</tr>
<tr>
<td>E2F0</td>
<td>Graph</td>
<td>2</td>
<td>2</td>
<td>2, 3</td>
<td>0</td>
<td>2,3,...,40</td>
<td>16 - 65,680</td>
<td>39</td>
</tr>
<tr>
<td>E2F1</td>
<td></td>
<td>1</td>
<td></td>
<td>2,3,...,40</td>
<td>1</td>
<td>2,3,...,40</td>
<td>16 - 33,856</td>
<td>39</td>
</tr>
<tr>
<td>E2F2</td>
<td></td>
<td>2</td>
<td></td>
<td>2,3,...,34</td>
<td>2</td>
<td>2,3,...,34</td>
<td>16 - 33,856</td>
<td>33</td>
</tr>
<tr>
<td>E2F3</td>
<td></td>
<td>3</td>
<td></td>
<td>2,3,...,24</td>
<td>3</td>
<td>2,3,...,24</td>
<td>16 - 14,448</td>
<td>23</td>
</tr>
<tr>
<td>E3F0</td>
<td>Tree</td>
<td>2</td>
<td>4</td>
<td>2, 2, 2, 2</td>
<td>0</td>
<td>1,2,...,40</td>
<td>7 - 6,520</td>
<td>40</td>
</tr>
<tr>
<td>E3F1</td>
<td></td>
<td>1</td>
<td></td>
<td>1,2,...,40</td>
<td>1</td>
<td>1,2,...,40</td>
<td>7 - 6,520</td>
<td>40</td>
</tr>
<tr>
<td>E3F2</td>
<td></td>
<td>2</td>
<td></td>
<td>1,2,...,32</td>
<td>2</td>
<td>1,2,...,32</td>
<td>7 - 4,192</td>
<td>32</td>
</tr>
<tr>
<td>E3F3</td>
<td></td>
<td>3</td>
<td></td>
<td>1,2,...,32</td>
<td>3</td>
<td>1,2,...,32</td>
<td>7 - 4,192</td>
<td>32</td>
</tr>
<tr>
<td>E3F4</td>
<td></td>
<td>4</td>
<td></td>
<td>1,2,...,32</td>
<td>4</td>
<td>1,2,...,32</td>
<td>7 - 4,192</td>
<td>32</td>
</tr>
</tbody>
</table>

Total Number of Tests (Conformance Checks) 412
Vita

<table>
<thead>
<tr>
<th>Name</th>
<th>Your Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place and year of birth</td>
<td>Birth Place, Year</td>
</tr>
<tr>
<td>Education</td>
<td>B. Whatever</td>
</tr>
<tr>
<td></td>
<td>Line 2</td>
</tr>
<tr>
<td></td>
<td>University</td>
</tr>
<tr>
<td>Experience</td>
<td>Job 1 Line 1</td>
</tr>
<tr>
<td></td>
<td>Job 1 Line 1</td>
</tr>
<tr>
<td></td>
<td>Organization, Year(s)</td>
</tr>
<tr>
<td></td>
<td>Job 2 Line 1</td>
</tr>
<tr>
<td></td>
<td>Organization, Year(s)</td>
</tr>
<tr>
<td></td>
<td>Job 3 Line 1</td>
</tr>
<tr>
<td></td>
<td>Job 3 Line 2</td>
</tr>
<tr>
<td></td>
<td>Organization, Year(s)</td>
</tr>
<tr>
<td></td>
<td>Job 4 Line 1</td>
</tr>
<tr>
<td></td>
<td>Organization, Year(s)</td>
</tr>
<tr>
<td>Major Awards</td>
<td>Award 1, Year</td>
</tr>
<tr>
<td></td>
<td>Award 2, Year</td>
</tr>
<tr>
<td></td>
<td>Award 3, Years</td>
</tr>
</tbody>
</table>