UPPSALA
UNIVERSITET

Institutionen för informationsteknologi
Parosh Aziz Abdulla

Besöksadress:
MIC, Polacksbacken
Lägerhyddvägen 2

Postadress:
Box 337
751 05 Uppsala

Telefon:
018–471 3163

Telefax:
018–51 19 25

Hemsida:
http://user.it.uu.se/~parosh

Epost:
parosh@IT.UU.SE

Department of Information Technology
Parosh Aziz Abdulla

Visiting address:
MIC, Polacksbacken
Lägerhyddvägen 2

Postal address:
Box 337
SE-751 05 Uppsala
SWEDEN

Telephone:
+46 18–471 3163

Telefax:
+46 18–51 19 25

Web page:
http://user.it.uu.se/~parosh

Email:
parosh@IT.UU.SE

Thesis Report – JiŘi Šimáček

1 Background

Model checking has been one of the most successful approaches to *finite-state* program verification. While the original framework is well-suited for reasoning about hardware circuits, it fails to deal with several essential aspects of software, such as infinite data domains, unbounded communication media, timing constraints, dynamic process creation, parameterization (systems with unbounded numbers of components), multi-threading, and dynamically allocated data structures. Therefore, one of the main challenges facing the research community during recent years has been to extend model checking to the context of infinite-state systems. The first steps in that direction have been to develop techniques for model checking of *idealized* models of systems which are infinite in *one* dimension. Examples include timed automata (with real-valued clocks), push-down systems (with unbounded stacks), counter machines (with unbounded counters), lossy channel systems (with unbounded FIFO buffers), and parameterized systems (with unbounded numbers of components).

Despite the rich set of algorithms which have been developed for infinite-state systems, current methods fall short of being able to handle realistic software systems. The main goal of this thesis is to extend the applicability of model checking to an important class of programs, namely those operating on dynamically linked data structures. On the one hand, such data structures are a natural feature in almost all programming languages. On the other hand, they are notoriously difficult to verify since they are unbounded, come often with complicated patterns, and have shapes that change dynamically during the execution of the program. This makes them well beyond existing techniques for the verification of infinite-state systems. The design of heap-manipulating programs is known to be error-prone, and furthermore, errors in the code tend to cause unpredictable behaviors. This makes testing very costly, and classical measures of coverage and testing thoroughness will become inadequate (or at least hard to define) due to the intricate interaction between the program and its data structures. Thus, in this context, model checking appears to be a good complement to classical techniques such as testing and static analysis for achieving a more efficient error analysis procedure.

2 Contributions

The main contribution of the thesis is the development of new techniques for algorithmic verification of programs manipulating complex dynamic data structures. This is based on two ingredients; namely to design symbolic representations that allow to construct sufficiently precise models of program code, and to develop techniques that allow the analysis of such symbolic representations. To that end, the thesis describes several important contributions. Below, I include some highlights:

- The thesis introduces a novel representation formalism for describing heap configurations that arise during the execution of programs. Such configurations take often the form of general graphs, and hence they cannot be represented using known automata-based formalisms, as the latter are restricted to languages that are defined for words or trees. The thesis proposes a novel representation that decomposes a graph into a finite set of trees where each tree represents a portion of the graph whose end-points are defined by a carefully selected set of "significant cells" of the heap. This is a very clever solution to the problem, since it allows to represent each graph canonically using a set of trees (a *forest*).

  The thesis also proposes also a novel model for symbolic representation of infinite sets of forests, namely that of *forest automata*. A forest automaton is a tuple of tree automata each of which generates a set of trees that be composed into a graph. In such a manner we obtain a simple tree-like representation for the much more complicated graph case.

  Finally, in order to make the forest automata symbolic representation useful in verification, the thesis describes how to perform basic operations such post-image computations with respect to program statements on these automata.

- Many symbolic representations used for model checking of infinite-state systems suffer from an "explosion" problem similar to the *state explosion* problem for finite-state model checking. The number and size of generated predicates during reachability analysis grows quickly with the size of the system. Therefore, it is important to develop techniques that make symbolic state-space exploration more efficient. The thesis proposes to use simulation-based reductions of the tree automata that arise during the analysis. More precisely, it proposes to collapse states that are related by the the well-known downward simulation relation for tree automata. For this purpose, the thesis proposes a new method for efficiently computing simulation relations on tree automata. The main idea is to exploit the fact that the transitions are often sparsely labeled by the symbols of

the alphabet, which allows for significant savings of time and space requirements in practical cases.

- One crucial operation in the application of symbolic model checking is the ability to perform language inclusion efficiently. This operation can be a bottleneck in the feasibility of the method sine it is used intensively for checking termination conditions for the underlying fix-point operations. The thesis proposes to extend existing methods that are based on combining anti-chain reasoning with simulation pre-orders. More precisely, it describes how to combine downward simulation with anti-chain techniques so that the method can be applied to tree automata with arbitrary arities as needed by the current application.

- The author has implemented a tree automata library that is freely available and that contains the implementation of basic operations on tree automata. These operations are obviously of general interest since their applicability is not restricted to program verification. The operations in the package can be applied to tree automata with different kinds of representations such as explicit and semi-symbolic representations.

3 Verdict

The quality of the contributions of the thesis is of high international class. The thesis makes several important contributions that allow to extend the applicability of model checking to a new class of programs, namely heap-manipulating programs. As mentioned above, model checking has been mainly applied to finite-state systems, and (more recently) to special cases of infinite-state systems. The class of programs considered in the thesis are known to be extremely difficult to handle with automatic methods. One main source of difficulty is that they operate on graph-like structures and are therefore beyond the capabilities of existing methods. The translation scheme proposed for canonical translation of graphs to sets of trees is ingenious and useful. A strong point of this work is that the methods are carefully explained making them possible to understand for non-specialists. This should not be underestimated considering the fact that several works in the area are only accessible to experts in specific logical frameworks. The thesis also makes several contributions to efficient implementations of fundamental operations on automata, such as simulation, minimization, and language inclusion. This is of independent interest since the applications of such operations go beyond program verification.

The high quality of the works in the thesis is reflected clearly by the excellent venues in which the papers have been published such as CAV, ATVA, TACAS, and FMSD.

UPPSALA
UNIVERSITET

In conclusion, I believe strongly that the doctoral thesis meets the requirements of the proceedings leading to PhD title conferment.

Prof. Parosh Aziz Abdulla