An Approach to Probabilistic Symmetry Reduction

Christopher Power
Department of Computing Science
University Of Glasgow
power@dcs.gla.ac.uk

Alice Miller
Department of Computing Science
University Of Glasgow
alice@dcs.gla.ac.uk

Abstract

We present a technique for the automated detection of potential structural and data symmetries from a probabilistic specification language. This approach involves the construction of an extended static channel diagram, a graphical representation of channel-based communication and global variable accesses from a specification defined in our custom made language SPS. This graph is used to compute potential symmetries whose validity is verified against the probabilistic specification. Unlike previous approaches this method can detect arbitrary structural and data symmetries.

1 Introduction

As software becomes more complex the need for development techniques capable of uncovering errors at design time is critical. A model checker accepts two inputs: a specification $\mathcal{P}$, described in a high level formalism and a set of testable properties, $\phi$. A model checker generates and exhaustively searches a finite state model $\mathcal{M}(\mathcal{P})$ to confirm if a property holds, or conversely, report a violation of the system specification. The intuition is that, bugs found in the model will reveal bugs in the system design. However, the application of model checking is limited as the state space of even moderately sized concurrent systems can be too large for state-of-the-art machines to exhaustively search.

Although verification algorithms have a linear run time complexity, this is offset as the number of states in a model grows exponentially as parameters are added. Consequently, research often focuses on techniques to reduce the impact of the state-space explosion. In the probabilistic domain, research into alleviating the state space explosion problem is still in its infancy. Only two techniques in the form of symbolic state storage [5] and partial order reduction [2] have been comprehensively investigated, and recently steps have been taken in the application of symmetry reduction to symbolic storage schemes [6]. This research is of importance due to the additional overhead required by probabilistic verification algorithms. To this end, we are investigating the application of symmetry reduction in the area of probabilistic explicit state model checking.

2 Symmetry Reduction for Probabilistic Model Checking

In model checking, symmetry reduction involves replacing sets of symmetrically equivalent states in a model $\mathcal{M}$ by a single representative, $\text{rep}(s)$, from each equivalence class. The resulting structure $\mathcal{M}'$ is called a quotient structure. For highly symmetric systems, exploring the quotient structure only, can result in a reduction factor exponential to the number of system components. In the probabilistic domain a commonly used structure is a Discrete Time Markov Chain (DTMC).

For a DTMC, $\mathcal{D} = (S, s_0, P)$, a permutation $\alpha : S \rightarrow S$ that preserves the transition relation and set of initial states is termed an automorphism of $\mathcal{D}$. The set of all automorphisms of $\mathcal{D}$ forms a group where the operator is a mapping and is denoted $\text{Aut}(\mathcal{D})$. For $G \leq \text{Aut}(\mathcal{D})$, the orbits of $S$ under $G$ can be used to construct a quotient DTMC $\mathcal{D}_G$. Let $\mathcal{D}$ be a DTMC, and $G$ an automorphism group of $\mathcal{D}$. The quotient structure $\mathcal{D}_G = (S_G, s_0^G, P_G)$ is defined as:

- $S_G = \{ \text{rep}_G(s) : s \in S \}$, where $\text{rep}_G(s)$ is a unique representative of $s^G$
- $s_0^G = \text{rep}_G(s_0)$
- $P_G(\text{rep}_G(s), \text{rep}_G(t)) = \sum_{x \in t^G} P(\text{rep}_G(s), x)$

For a model $\mathcal{D}$ and its quotient model $\mathcal{D}_G$ with respect to a group $G$, $\mathcal{D}, s \models \phi \iff \mathcal{D}_G, \text{rep}_G(s) \models \phi$ [7]. By choosing a suitable symmetry group $G$, model checking can be performed over $\mathcal{D}_G$ instead of $\mathcal{D}$, often resulting in considerable savings in memory and verification time. If automorphisms can be identified in advance, then a quotient structure can be incrementally constructed even if the original structure is intractable.
3 Automated Symmetry Detection

It has been established [4] that there is a correspondence between symmetries in a specification’s underlying communication structure and those in the model. To determine the symmetry present in a system, a structure called a static channel diagram [4], a graphical representation of potential communication within the system, can be generated directly from a model specification. As with most other automated symmetry detection techniques static channel diagrams relate to structural symmetry. However, another form of symmetry, namely data symmetry, can be exploited to increase the effectiveness of model checking.

Specifications often contain large data structures that can be populated by numerous potential values. To capture potential data symmetries in a specification we have extended the definition of a static channel diagram to include nodes for global variables and edges between process identifiers and global variables. An edge between process identifiers and global variable nodes is included if a process can potentially update the variable, and an edge from a global variable node to a process identifier is included if the result of an update may be affected by the value of the variable.

To test our extension a suitable specification language was required. The language would need to be able to define a wide range of probabilistic models, be compatible with existing approaches to automated symmetry detection and have formally defined semantics. To our knowledge the only probabilistic specification language potentially compatible with existing techniques is ProbMela [1]. However, ProbMela [1] is a language with many constructs, making it infeasible to rigorously prove implemented reduction techniques are sound. Therefore, we elected to define our own smaller probabilistic channel based language called, Symmetric Probabilistic Specification (SPS). SPS is tailored to meet all the above criteria and has a fully defined grammar and type system, in addition to providing precise DTMC structure semantics for a specification.

We have created a tool that implements these static channel diagram extensions. A SPS specification is taken as input and its abstract syntax tree constructed. In turn the tree is used to type-check the specification ensuring that variables are used appropriately. If deemed correct the static channel diagram $C(P)$ is generated and saucy [3] used to compute a set of generators for $\text{Aut}(C(P))$. Each of the generators is checked for validity against the specification. These checks are generally conditions on assignments to process id sensitive variables and can be efficiently checked. For an element $\alpha \in \text{Aut}(C(P))$, we say that $\alpha$ is valid (for $P$) if $\alpha(P) \equiv P$.

The complexity of deriving $C(P)$ from $P$ is linear in the size of $P$ but no polynomial time algorithm is known for the calculation of generators for $\text{Aut}(C(P))$. However, saucy [3] was specifically designed to calculate automorphisms of sparse graphs and $C(P)$ tends to be relatively sparse. The performance of the tool is generally very good and a set of valid specification automorphisms can be calculated in under a second.

In current work we are proving the following correspondence theorem. Let $P$ be a channel based probabilistic specification with extended static channel diagram $C(P)$ and associated DTMC structure $D$. Let $\alpha \in C(P)$. If $\alpha$ is valid for $P$ then $\alpha^* \in \text{Aut}(D))$. This theorem will prove the basis for the sound implementation of the first on the fly probabilistic model checker that utilises symmetry reduction to manage the state space.

References


