Science of Computer Programming 128 (2016) 51–67

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

On the random structure of behavioural transition systems

Jan Friso Groote [a,*], Remco van der Hofstad [a], Matthias Raffelsieper [b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[b] D-MTEC, ETH Zürich, Weinbergstrasse 56/58, 8092 Zürich, Switzerland

ARTICLE INFO

Article history:
Received 8 May 2015
Received in revised form 23 November 2015
Accepted 23 February 2016
Available online 3 March 2016

Keywords:
Random graph
P-parallel random transition system
State space size

ABSTRACT

Numerous properties of random graphs are highly predictable. Even by exploring a small part reliable observations are possible regarding their structure and size. An unfortunate observation is that standard models for random graphs, such as the Erdös–Rényi model, do not reflect the structure of the graphs that describe distributed systems and protocols.
In this paper we propose to use the parallel composition of such random graphs to model 'real' state spaces. We show how we can use this structure to predict the size of state spaces, and we can use it to explain that software bugs are in practice far easier to find than predicted by the standard random graph models. By practical experiments we show that our new probabilistic model is an improvement over the standard model in predicting properties of transition systems representing realistic systems.

1. Introduction

Modelling the behaviour of systems is gaining popularity. For complex real-world systems however the transition systems of their models easily become very large. We ran into such an example while modelling an UART (universal asynchronous receiver/transmitter) for the company NXP. Using highway search [4], a parallel simulation technique far more efficient than random simulation in finding problematic situations, we did *not* find a suspected error. The question that we needed to answer was how large the probability was that the error really was absent. A typical derived question that immediately jumps to mind is to estimate the size of the state space.

In order to answer such questions, one can resort to random graphs [2]. The Erdös–Rényi model is a commonly used model. It has a set S of N states (nodes, vertices) and a set of transitions $\rightarrow$. There are two highly similar variants, one where each conceivable directed edge is present with some probability p, and one where M transitions are chosen uniformly at random out of the $N(N-1)$ possibilities.

Erdös–Rényi random graphs are a little counterintuitive if it comes to modelling transition systems that represent behaviour. Transition systems have an initial state and this initial state has outgoing transitions to states that in general also have outgoing transitions. In the Erdös–Rényi random graph the initial state may not have outgoing transitions (actually with a fairly high probability $e^{-\lambda}$ where λ is the fan-out, i.e., the expected number of transitions leaving a state). Therefore, we choose a slightly different model, where each state has a *fixed* number λ of outgoing transitions each of which goes to randomly selected states of the transition system. All choices are made independently of each other.

Given this model of a random transition system we estimate the size of a transition system by a random walk through the graph. By random simulation we have evidence that these estimates are very good. However, by applying this technique

* Corresponding author.
E-mail addresses: J.F.Groote@tue.nl (J.F. Groote), R.W.v.d.Hofstad@tue.nl (R. van der Hofstad), raffelsm@ethz.ch (M. Raffelsieper).

http://dx.doi.org/10.1016/j.scico.2016.02.006
0167-6423/

to realistic models (e.g., Firewire P1394 protocol [10]) it becomes obvious that the structure of these random graphs is not really a reflection of a 'real state space'.

As an alternative model for the structure of realistic systems, we propose to use the Cartesian product of P parallel random transition systems, reflecting that a realistic system often consists of P more or less independent components. One could not only think of the components as independent parallel processes, but one can also consider the behaviour of subtasks or even variables as potential parallel components. These examples already show that complex systems are not built as one single monolithic piece, but assembled in a compositional fashion from smaller parts.

We develop techniques to estimate the sizes and fan-outs of the different components. Again, using random simulation we have verified that these estimation techniques are effective in retrieving these properties for state spaces that are generated as Cartesian products of random state spaces. More importantly, we estimate the sizes of 'realistic state spaces' and find that these estimates are close to the actual values and in particular far better than those we obtain using the 'single threaded' random graph model. There are also some disadvantages, in particular, the predictions are less stable and the numerical effort for the estimates is higher.

Our experiments provide evidence that P-parallel random transition systems could be a good representation of 'realistic' state spaces. Of course, the state spaces of real applications do not have a random structure. But having a random model that reflects 'real' state spaces reasonably well and that is sufficiently simple to allow mathematical analysis is really a great asset, because it leverages the power of random analysis to substantially increase our generic insight into the behaviour of real systems.

As an illustration of the potential power of the P-parallel model we apply it to the question of how effective testing is. In our experience it is remarkably easy to detect a known error by running a random test. According to the single threaded random graph model this is not possible. The probability of hitting an erroneous state by a random walk is far too small. However, if the error occurs in one of the states of one of the P-parallel components, it is far easier to find. Even stronger, if we know the sizes of the different components, we can come up with small numbers of required test runs to guarantee with high confidence that realistic systems are error free.

Related work As far as we know, there is not much work on the random structure of transition systems representing behaviour. The following is what we are aware of. Estimating the size of a Petri Net's state space has been investigated in [15]. That work makes explicit use of the structure of a Petri Net and is only applicable when the Petri Net is constructed from a set of supported building blocks.

A more general approach was presented in [12] where, as in our work, a state space is seen as a directed graph. However, the authors do not compute an estimate, but instead only classify state spaces into one of three classes: small models, large models, and models that are too large and hence out of reach. In this work classification trees, neural networks and techniques similar to the Lincoln Index [9] are employed.

Inspired by [12], the authors of [3] present a method to compute the estimated size of the state space. There, the observed measure is the size of the breadth-first frontier that is still to be explored in relation to the number of states that have already been explored. By visual inspection, the authors determine that this curve should be approximated by a quadratic function and use least-squares fitting to compute the parameters and thereby an estimate for the state-space size.

2. Random state spaces

In this section we define the basic notions that we employ. We use directed graphs or transition systems without labels, as we do not need the labels in our exposition.

A *state space* is seen as a graph $G = (S, \rightarrow)$, with S being an arbitrary set of *states* (nodes, vertices) and $\rightarrow \subseteq S \times S$ being a multi-set of *transitions* (edges). If $(s, s') \in \rightarrow$, then we generally denote this by $s \rightarrow s'$. For the edges in the set $\rightarrow$ we assume that every state has a fixed degree of outgoing edges, i.e., there is a fixed $\lambda \in \mathbb{N}$ such that $|\{s' \mid s \rightarrow s'\}| = \lambda$ for all $s \in S$ (where $\{s' \mid s \rightarrow s'\}$ is a multi-set) and $|E|$ denotes the size of a multi-set E. One can consider transition systems with a variable fan-out, but this will make the random graph model more complex, and therefore harder to use and of less predictive utility. If $s \rightarrow s'$, then s is called the *source* and s' the *target* state of that edge. In a random state space it is assumed that for every such edge, given its source state s, every other state s' is equally likely to be the target state. Furthermore, we define $N = |S|$ and $M = |\rightarrow|$ to denote the number of states and transitions, respectively.

A tuple $T{=}(G, s_0)$ is called a *random transition system*, where $G{=}(S, \rightarrow)$ is a random state space as described above and $s_0{\in}S$ is an arbitrary, randomly chosen *initial state*. For such a random transition system, only the part reachable from the initial state is of interest, i.e., those states $s'{\in}S$ for which $s_0 \rightarrow^* s'$ holds (where $\rightarrow^*$ denotes the reflexive transitive closure of $\rightarrow$). Note that the number of reachable states is at most N.

This paper considers state spaces being the graph product of two or more random transition systems. Since taking the graph product is associative, we only consider the case of two random transition systems, which can then be repeated for more components. Thus, a *product transition system* $T_{1\times2} = (G, s_0)$ with graph $G = (S, \rightarrow)$ and initial state $s_0 \in S$ is assumed to be composed from two random transition systems $T_1 = (G_1, s_{1,0})$ and $T_2 = (G_2, s_{2,0})$, with $G_1 = (S_1, \rightarrow_1)$, $G_2 = (S_2, \rightarrow_2)$, such that $S = S_1 \times S_2$, $s_0 = (s_{1,0}, s_{2,0})$, and $(s_1, s_2) \rightarrow (s'_1, s'_2)$ iff either $s_1 \rightarrow_1 s'_1$ and $s_2 = s'_2$, or $s_1 = s'_1$ and $s_2 \rightarrow_2 s'_2$.

Note that it is assumed that the states in the product transition system $T_{1\times2}$ are *opaque*, i.e., from a state $s = (s_1, s_2) \in S$ the individual components s_1 and s_2 of the state cannot be recovered. Note also that we do not consider 'synchronisa-