Electronic Communications of the EASST
Volume 70 (2014)

Proceedings of the
14th International Workshop on
Automated Verification of Critical Systems (AVoCS 2014)

On the Random Structure of Behavioural Transition Systems

Jan Friso Groote, Remco van der Hofstad, and Matthias Raffelsieper

15 pages

Guest Editors: Marieke Huisman, Jaco van de Pol
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

On the Random Structure of Behavioural Transition Systems

Jan Friso Groote[1], **Remco van der Hofstad**[2], **and Matthias Raffelsieper**[3]

[1]J.F.Groote@tue.nl, http://www.win.tue.nl/~jfg
[2]R.W.v.d.Hofstad@tue.nl, http://www.win.tue.nl/~rhofstad
Department of Mathematics and Computer Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[3]raffelsm@ethz.ch
D-MTEC, ETH Zürich, Weinbergstrasse 56/58, 8092 Zürich, Switzerland

Abstract: Random graphs have the property that they are very 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 some practical experiments we show that our new random model is an improvement over the standard model in predicting properties of transition systems representing realistic systems.

Keywords: Random graph, *P*-parallel random transition system, state space size

1 Introduction

Modelling the behaviour of systems is gaining popularity. An unpleasant side effect is that the transition systems of models of realistic systems easily become very large. We ran into such an example while modelling an UART (universal asynchronous receiver/transmitter) for a company called 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 did not occur. 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 edge is present with some probability p, and one where M transitions are chosen out of the N^2 possibilities.

Erdös-Rényi random graphs are a little counterintuitive if it comes to modelling transition systems which 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 to realistic models (e.g., Firewire P1394 protocol [9]) 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.

We develop techniques to estimate the sizes and fanout of the different components. Again, using random simulation we verified that these estimation techniques are correct and precise. More importantly, we estimate the sizes of 'realistic state spaces' and find that these are far better than those we obtain using the 'single threaded' random model. There are also some disadvantages, in particular, the predictions are less stable and the calculational 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 which reflects 'real' state spaces reasonably well and which is sufficiently simple to allow mathematical analysis is really a great asset, because it enables the use of the power of random analysis to substantially increase our generic insight in the behaviour of real systems.

As an illustration of the potential power of the P-parallel model we apply it to the question 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 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 a given confidence that realistic systems are error free.

Acknowledgments. Thanks go to Jesse Goodman, Wil Kortsmit and the referees for remarks, support and discussion. This research was in part supported by NWO and Agentschap NL.

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 [11]. 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 [14] 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 [13] are employed.

Inspired by [14], the authors of [3] present a method to compute the estimated state space size. 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 with 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 $(s, s') \in \rightarrow$ holds, 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 exists 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 model more complex, and therefore harder to use, and less predictive. 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* $T_{1\times 2} = (G, s_0)$ 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 $T_{1\times 2}$ 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 'synchronisation', i.e., a transition in the product transition system which consists of the simultaneous occurrence of transitions in the constituent transition systems. Our random approximations of realistic state spaces do not contain such synchronised transitions.

3 Estimation Based On Duplicates

In this section, a technique is described to estimate the size of a single random transition system $T = (G, s_0)$ with graph $G = (S, \to)$. For this purpose, the process of exploring the state space is analysed. Starting from the initial state s_0, the edges that have their source state in the already explored part of the state space are iteratively explored. Thus, a target state of an edge that is being explored can either be a state that has already been explored previously, or it is a state that is new, i.e., has not been seen previously. Note that the method in this section does not rely on a particular state space exploration strategy, contrary to approach in the next section which requires breadth-first search.

3.1 A stochastic model

We develop a stochastic model where we introduce random variables Y_k, which represent the number of unique states seen after exploring k transitions. Then, the probability distribution of the state space size N after exploring m transitions is

$$\mathbb{P}[N = n \mid \bigwedge_{k=0}^{m} Y_k = i_k],$$

where i_k is the observed number of unique states seen after exploring k transitions. Applying Bayes' law, this probability can be rewritten as follows:

$$\begin{aligned}\mathbb{P}[N = n \mid \bigwedge_{k=0}^{m} Y_k = i_k] &= \frac{\mathbb{P}[N = n \wedge \bigwedge_{k=0}^{m} Y_k = i_k]}{\mathbb{P}[\bigwedge_{k=0}^{m} Y_k = i_k]} \\ &= \frac{\mathbb{P}[\bigwedge_{k=0}^{m} Y_k = i_k \mid N = n]\mathbb{P}[N = n]}{\sum_{r=i_m}^{\infty} \mathbb{P}[\bigwedge_{k=0}^{m} Y_k = i_k \mid N = r]\mathbb{P}[N = r]}\end{aligned} \quad (1)$$

In the above equation (1), the probability $\mathbb{P}[N = n]$ is the so-called *a-priori* probability for the size of the state space. To characterise the probabilities $\mathbb{P}[\bigwedge_{k=0}^{m} Y_k = i_k \mid N = n]$, we use the following recursive identity:

$$\mathbb{P}[\bigwedge_{k=0}^{m} Y_k = i_k \mid N = n] = \mathbb{P}[Y_m = i_m \mid \bigwedge_{k=0}^{m-1} Y_k = i_k \wedge N = n]\,\mathbb{P}[\bigwedge_{k=0}^{m-1} Y_k = i_k \mid N = n] \quad (2)$$

Thus, we need to analyse the first factor at the right-hand side of equation (2) further. In case $m = 0$, then we find that

$$\mathbb{P}[\bigwedge_{k=0}^{0} Y_k = i_k \mid N = n] = \begin{cases} 1 & \text{if } i_0 = 1 \\ 0 & \text{otherwise,} \end{cases}$$

since only the initial state is seen in the beginning.

For $m > 0$, we make a case distinction based on whether the target state of the newly explored edge has been seen before or not. In case $i_m = i_{m-1}$, i.e., the target state has already been seen before, one of the i_{m-1} already seen states has been chosen:

$$\mathbb{P}[Y_m = i_m \mid \bigwedge_{k=0}^{m-1} Y_k = i_k \wedge N = n] = \frac{i_{m-1}}{n} = \frac{i_m}{n} \tag{3}$$

Otherwise, if the target state of the newly explored edge has not been seen before, then $i_m = i_{m-1} + 1$ holds and we have chosen one of the $n - i_{m-1}$ states we have not yet seen, giving the following equality:

$$\mathbb{P}[Y_m = i_m \mid \bigwedge_{k=0}^{m-1} Y_k = i_k \wedge N = n] = \frac{n - i_{m-1}}{n} = \frac{n - i_m + 1}{n} \tag{4}$$

Next, we solve the recurrence in equation (2) using equations (3) and (4). To do so, we introduce variables $q_j^{(m)}$ for $1 \le j \le i_m$ to represent the multiplicity of number j in the sequence $i_0, i_1, \dots, i_m$ of length $m+1$. For example, in the sequence $1,1,2,3,3,4$ it holds that $q_1^{(5)} = 2$, $q_2^{(5)} = 1$, $q_3^{(5)} = 2$, and $q_4^{(5)} = 1$. It should be noted that $\sum_{j=1}^{i_m} q_j^{(m)} = m+1$ always holds. Using these variables, the right hand side of equation (2) becomes

$$\frac{\prod_{j=1}^{i_m}(n-j+1)\prod_{j=1}^{i_m} j^{q_j^{(m)}-1}}{n^{m+1}}. \tag{5}$$

Substituting (5) for (2) in (1) gives the following result, where remarkably enough the variables $q_j^{(m)}$ disappear. Apparently, only the information about the number of unique states (the i_k) and the total number of states explored (which is $m+1$, i.e., the number of target states of explored edges plus the initial state) is required:

$$\mathbb{P}[N = n \mid \bigwedge_{k=0}^{m} Y_k = i_k] = \frac{\left(\prod_{j=1}^{i_m}(n-j+1)\right)\mathbb{P}[N=n]/n^{m+1}}{\sum_{r=i_m}^{\infty}\left(\prod_{j=1}^{i_m}(r-j+1)\right)\mathbb{P}[N=r]/r^{m+1}} \tag{6}$$

We used that the maximally observed number of unique states i_m is a lower bound for the total size N. Often, an upper bound $\overline{n}$ on the total number of states can be obtained. For a relative error of $\frac{1}{K}$ it suffices to take $\overline{n} = i_m(1+\log(K))$. We assume that N is a priori uniformly distributed in the interval $[i_m, \overline{n}]$. In this case, equation (6) reduces to

$$\mathbb{P}[N = n \mid \bigwedge_{k=0}^{m} Y_k = i_k] = \frac{\left(\prod_{j=1}^{i_m}(n-j+1)\right)/n^{m+1}}{\sum_{r=i_m}^{\overline{n}}\left(\prod_{j=1}^{i_m}(r-j+1)\right)/r^{m+1}}, \tag{7}$$

which gives the following expected size of the state space:

$$\mathbb{E}[N \mid \bigwedge_{k=0}^{m} Y_k = i_k] = \frac{\sum_{n=i_m}^{\overline{n}} n\left(\prod_{j=1}^{i_m}(n-j+1)\right)/n^{m+1}}{\sum_{r=i_m}^{\overline{n}}\left(\prod_{j=1}^{i_m}(r-j+1)\right)/r^{m+1}} \tag{8}$$

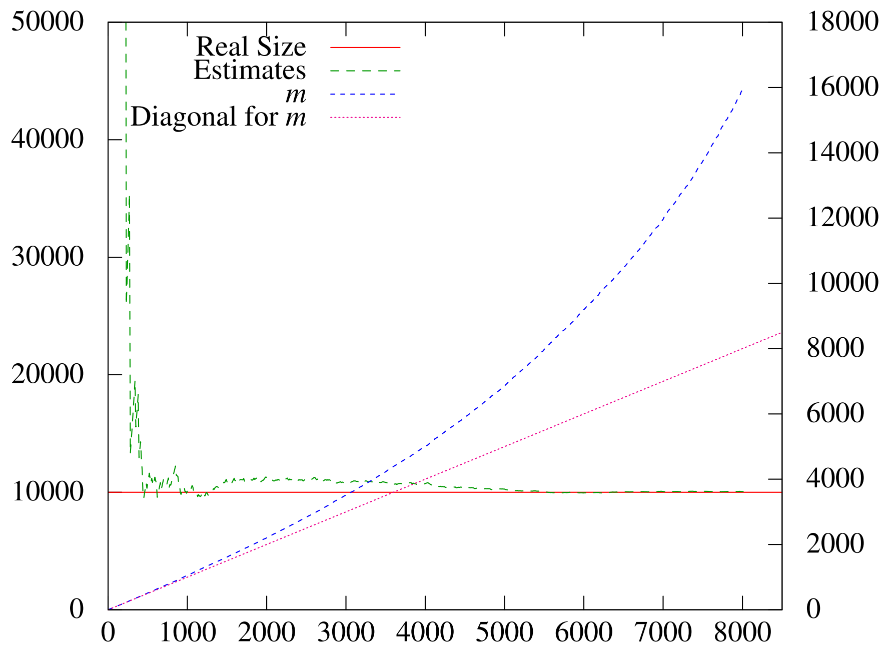

Figure 1: Estimates using equation (8) for an example random transition system with 10,000 states and a fan-out of 2. On the x-axis the number i_m of unique states observed is depicted. On the y-axis, the estimated size is shown at the left, and the number of explored transitions m and the diagonal are shown at the right.

3.2 Simulation experiments

We want to establish the effectiveness of the estimation procedure for the monolithic model by predicting the sizes of a randomly generated state spaces and comparing the results with the actual sizes used to generate them.

For the experiments with the estimation procedure presented in the previous section, we explored the example graphs in a breadth-first fashion, as this is the commonly used strategy of many model checking tools, an example being the tool LPS2LTS contained in the mCRL2 toolset [5, 8]. When used in verbose mode, the tool will output the currently explored number of states and transitions, which therefore allows to apply equation (8) to estimate the total size of the state space.

We present here the results on a random transition system with 10,000 states, each with 2 outgoing transitions. For other results see [7]. After randomly designating an initial state, 8,011 states were reachable. To compute the estimates, we used $\overline{n} = 1{,}000{,}000$ as upper bound on the size of the state space.

The estimation results are shown in figure 1. The x-axis shows the number of unique states seen at a certain point, i.e., the maximal i_m used in that computation. On the y-axis, the result of evaluating equation (8) is depicted. The values of the maximal i_m were increased in steps of 10 and two adjacent points were connected by a straight line segment. It can be observed that the estimation yields results close to the actual value of 10,000 after having observed a few hundred

i_m	11489	21717	51704	102265	150728	188569
m	13889	26669	79388	175029	273051	340608
N	35435	62637	85455	146869	204586	256631
$N_{\text{reachable}}$	25954	45878	62591	107572	149846	187965

Table 1: Estimates for the number of states of the firewire protocol. The actual number of reachable states is 188569. The estimates for N under the assumption that the state space has a random structure are not very accurate.

unique states. After that, it slightly overshoots, but always stays below $11,500$ estimated states. From this we conclude that our estimates are quite accurate.

3.3 Application to the firewire protocol

We use a description of a real time bus access in the firewire or P1394 protocol provided in [9] to observe what happens when we predict the number of states for a realistic protocol assuming that it is randomly generated. The protocol consists of two protocol entities that alternatingly obtain access to a data bus resolving contention conflicts on the way.

The typical values for i_m and m are given in table 1. The estimates for the expected number of states N and reachable states $N_{\text{reachable}}$, cf. equation (12), are also provided. The actual number of reachable states is 188569 and there are 340608 transitions. There is an average fan-out of 1.8.

In table 1 we observe that the estimates for $N_{\text{reachable}}$ while traversing the state space structurally underestimate the actual number of reachable states and they only approach the actual number of states when all states have been traversed. This is quite different from what we observe in figure 1. But this pattern is very similar to what we see in other state spaces representing real systems. From this we conclude that our random model does not represent real systems sufficiently well.

4 Estimates of the Sizes of Product Transition Systems

In order to have a random model that approximates real state spaces better, we study transition systems that are the product of two or more random transition systems in this section. Since we assume that in a product transition system the identity of a component state cannot be recovered, there is no way to obtain the original graphs. If we were able to recover the constituent graphs, we could have applied the technique of section 3 to the components and easily derive a prediction for the size of the whole transition system.

Our estimation technique is completely different from that of section 3. We require that the exploration of a state space is performed in a breadth-first fashion, i.e., first all states at a certain distance from the initial state are considered, before dealing with those at higher distances. This allows to consider *layers* of a transition system $T = (G, s_0)$ with $G = (S, \rightarrow)$. We define the layer $\partial B_T(j)$ at some distance $j \in \mathbb{N}$ by

$$\partial B_T(j) = \{s \in S \mid s_0 \rightarrow^j s \wedge \forall k < j : s \notin \partial B_T(k)\}$$

where $s_0 \rightarrow^j s$ means that state s is reachable from state s_0 in exactly j steps. We also call $\partial B_T(j)$

the boundary ball on intrinsic distance j. Note that by the above definition, every state s reachable from the initial state s_0 is contained in exactly one layer, namely that with the minimal distance j.

The set of those states that have been seen up to some distance $j \in \mathbb{N}$ is defined as $B_T(j) = \bigcup_{i=0}^{j} \partial B_T(i)$ and is called the *ball* of radius j. To obtain the layer $\partial B_T(j)$, only edges from states from the layer $\partial B_T(j-1)$ have to be considered, since otherwise, if there was an edge from a state $s \in \partial B_T(j-k)$ with $k > 1$ to a state $s' \in \partial B_T(j)$, then this would imply $s' \in \partial B_T(j-k+1)$, which would contradict $s' \in \partial B_T(j)$.

4.1 Estimating the size of a single component

We first concern ourselves with the estimation of the size of a single random transition system based on balls and layers. In section 4.2 we use this result to estimate the sizes of the transition systems in parallel products of transition systems. Breadth-first generation of a state space explores edges having their source state in the current layer and adds the target states to the surrounding ball. Thus, to obtain a layer at distance $j+1$, a total of $\lambda|\partial B_T(j)|$ edges are explored where λ is the fan out of each state.

We are interested in the expected size of the reachable state space, which is the same as the size of the ball with maximal radius. Thus, the expected size of the balls should be investigated.

Definition 1 We define $G(j)=\mathbb{E}[|B_T(j)|]$ to be the expected size of the breadth-first graph with states at a distance $\leq j$ from the initial state and $R(j) = \mathbb{E}[|\partial B_T(j)|] = G(j) - G(j-1)$, where $G(-1) = 0$.

We use the convention that the subscript of B_T carries over to G and R. For example, we write $G_{1\times 2}(j)$ for $\mathbb{E}[|B_{T_{1\times 2}}(j)|]$

To analyse the function $G(j)$, we introduce random variables X_i that denote the size of the partial state space after exploring i edges. Let n_i for $i = 1,2,\dots$ denote the number of unique states observed after exploring i edges. Thus, $n_0 = 1$ and for n_{i+1} we have either $n_{i+1} = n_i$ in case the target state of the additionally explored edge was already in the explored part of the state space, or $n_{i+1} = n_i + 1$ if the target state of the edge is new. Since the target state of an edge is picked uniformly at random, the probability of the first case ($n_{i+1} = n_i$) to occur is

$$\mathbb{P}(X_{i+1} = n_{i+1} \mid \bigwedge_{l=0}^{i} X_l = n_l) = \frac{n_i}{N} = \frac{n_{i+1}}{N},$$

which amounts to the probability to pick one of the $n_i = n_{i+1}$ states that were already explored from the total N states. In the second case, where $n_{i+1} = n_i + 1$, the probability is

$$\mathbb{P}(X_{i+1} = n_{i+1} \mid \bigwedge_{l=0}^{i} X_l = n_l) = 1 - \frac{n_i}{N} = \frac{N - n_{i+1} + 1}{N},$$

where a state is picked that is not among the $n_i = n_{i+1} - 1$ already explored states.

The expected value of these random variables can be computed as:

$$\begin{aligned}
\mathbb{E}[X_{i+1}] &= \sum_{k=0}^{\infty} k\mathbb{P}(X_{i+1}=k) \\
&= \sum_{k=0}^{\infty} k\frac{k}{N}\mathbb{P}(X_i=k) + \sum_{k=0}^{\infty} k\frac{N-k+1}{N}\mathbb{P}(X_i=k-1) \qquad (9)\\
&= \sum_{k=0}^{\infty} k\frac{k}{N}\mathbb{P}(X_i=k) + \sum_{k=0}^{\infty} (k+1)\frac{N-k}{N}\mathbb{P}(X_i=k) \\
&= \sum_{k=0}^{\infty} k(1-\frac{1}{N})\mathbb{P}(X_i=k) + \sum_{k=0}^{\infty} \mathbb{P}(X_i=k) \\
&= \sum_{k=0}^{\infty} k(1-\frac{1}{N})\mathbb{P}(X_i=k) + 1 \\
&= \frac{N-1}{N}\mathbb{E}[X_i] + 1. \qquad (10)
\end{aligned}$$

In equation (9) we used the above observations that either an already explored state is reached or a new state was explored. Solving the recurrence equation (10) using the boundary condition that $\mathbb{E}[X_0] = 1$, gives a closed formula:

$$\mathbb{E}[X_i] = N\left(1-\left(\frac{N-1}{N}\right)^{1+i}\right). \qquad (11)$$

This equation can also be used to predict N in table 1 with almost equal estimates for N.

Another use of equation (11) is to derive the number of reachable states s. All reachable states have been explored if there are no other states that have been visited. Then there are λs visits, as for each explored state each outgoing transition is investigated. So, equation (11) becomes

$$s = N\left(1-\left(\frac{N-1}{N}\right)^{1+\lambda s}\right). \qquad (12)$$

From this s can easily be solved, although care is required as there is also a small negative solution for s.

From the observation that $\lambda|B_T(j)|$ edges are explored to obtain $B_T(j+1)$, together with equation (11) the following recursive formula for the function $G(j)$ can be derived, namely the expected size of the ball with radius j for a random transition system, where N and λ denote the total size and the fan-out of each state, respectively:

$$G(j+1) = \mathbb{E}[X_{\lambda|B_T(j)|}] = N\left(1-\left(\frac{N-1}{N}\right)^{1+\lambda G(j)}\right). \qquad (13)$$

Equation (13) gives a means to compute the size N given two consecutive balls, assuming that we have a way to estimate λ as the average fan-out in each state. With more balls available, N, and even λ, can be estimated using the least square method.

4.2 Product Graph Size Estimation

As the estimates of the size of the state spaces are way off if we assume that they are homogenous random state spaces, we are interested in viewing a state space as the product of two random state spaces. We assume that the two components cannot be distinguished in the product graph and therefore we need to formulate a relation between the observable ball and layer sizes of the product graph and the component sizes. For this purpose, we note that due to the definition of the graph product, which only performs a transition in one of the components, a state in the layer of depth j can be reached by either only performing j steps in the first component, or $j-1$ steps in the first component and one step in the second component, etc. Furthermore, steps from different components are independent of each other, i.e., if $(s_1,s_2) \rightarrow_2 (s_1,s_2') \rightarrow_1 (s_1',s_2')$, then also $(s_1,s_2) \rightarrow_1 (s_1',s_2) \rightarrow_2 (s_1',s_2')$, where $\rightarrow_i$ is a step done by component i. Hence, we can re-order the steps such that first all steps in the first component are performed, and then all steps in the second component. Thus, in the product graph, the following equation holds:

$$|\partial B_{T_1\times T_2}(j)| = \sum_{k=0}^{j} |\partial B_{T_1}(k)| \cdot |\partial B_{T_2}(j-k)|. \tag{14}$$

From equation (14) for the expected value $R_{1\times 2}(j)$ of the product graph we derive:

$$R_{1\times 2}(j) = \sum_{k=0}^{j} R_1(k)\cdot R_2(j-k). \tag{15}$$

In the following, the expected sizes of the component layers, R_1 and R_2, are considered in more detail. Let $i \in \{1,2\}$ and recall that $R_i(j+1) = G_i(j+1) - G_i(j)$. Using equation (13), we can also obtain a recursive formula for the component layer sizes:

$$\begin{aligned}
R_i(j+1) &= G_i(j+1) - G_i(j)\\
&= N_i\left[\left(\frac{N_i-1}{N_i}\right)^{1+\lambda_i G_i(j-1)} - \left(\frac{N_i-1}{N_i}\right)^{1+\lambda_i G_i(j)}\right]\\
&= N_i\left(\frac{N_i-1}{N_i}\right)^{1+\lambda_i G_i(j-1)}\left[1-\left(\frac{N_i-1}{N_i}\right)^{\lambda_i(G_i(j)-G_i(j-1))}\right]\\
&= (N_i - G_i(j))\left[1-\left(\frac{N_i-1}{N_i}\right)^{\lambda_i R_i(j)}\right]\\
&= \left(N_i - \sum_{k=0}^{j} R_i(k)\right)\left[1-\left(\frac{N_i-1}{N_i}\right)^{\lambda_i R_i(j)}\right]
\end{aligned} \tag{16}$$

Substituting equation (16) into equation (15) for R_1 and R_2 yields an equation for the observable layer sizes of the product graph, in which the values of N_1, N_2, λ_1, and λ_2 are unknown. Note however that the combined fan-out $\lambda_1+\lambda_2$ can be observed in the product graph, since this is the fan-out of each state in that graph. Therefore, one of the fan-out values can be eliminated, for example by replacing λ_2 with $\lambda - \lambda_1$, where $\lambda = \lambda_1 + \lambda_2$ is the observed constant fan-out in the product graph. By observing the sizes of three layers, three equations can be obtained from which the values of N_1, N_2 and λ can be solved, which is enough to provide the desired estimates. As before if more layers are available, the parameters can be estimated using the least square method. Using random simulation we established that this formula is effective and correct.

It should be noted that the above can easily be extended to more than two components. For example, in case of a product graph consisting of three random transition systems, equation (14) becomes:

$$|\partial B_{T_1 \times T_2 \times T_3}(j)| = \sum_{k=0}^{j} \sum_{l=0}^{j-k} |\partial B_{T_1}(k)| \cdot |\partial B_{T_2}(l)| \cdot |\partial B_{T_3}(j-k-l)|. \tag{17}$$

4.3 Experiments with state spaces of realistic systems

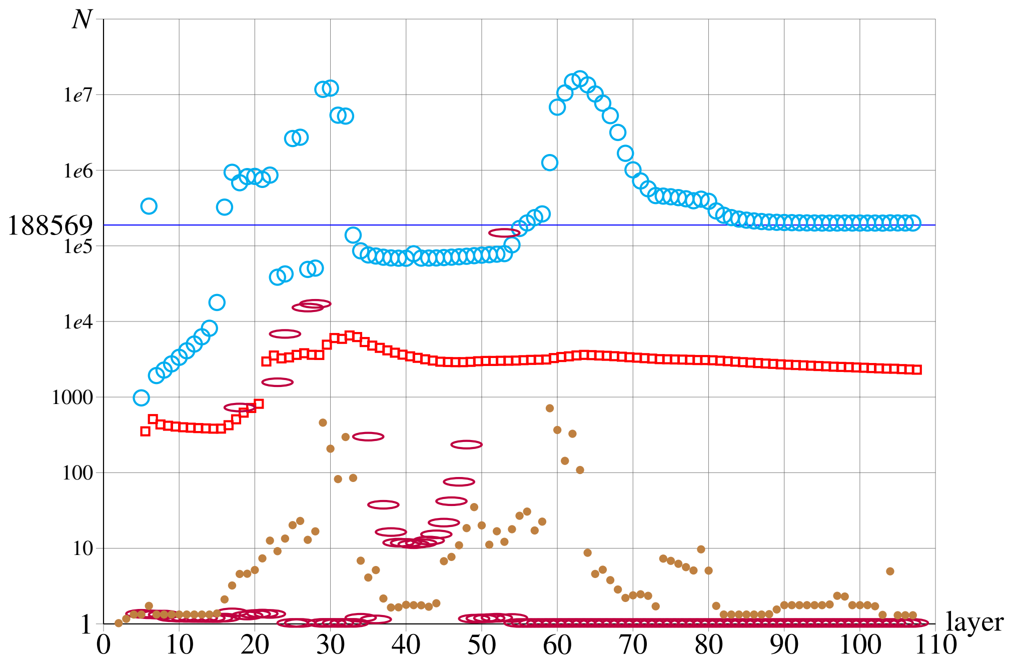

Figure 2: Estimates of the reachable state space of the firewire protocol

We are interested in whether the estimates using product state spaces give a better prediction for the sizes of state spaces that we encounter in practice. In figures 2 and 3 the results are provided for the firewire protocol [9] and the concurrent alternating bit protocol [10]. They are representative for other similar experiments that we have done. The sizes of the layers used in the experiments are provided in [7].

On the x-axis the indices of all layers are indicated. On the y-axis the layer sizes, the estimated number of reachable states and the estimated fan-out are depicted. For the estimated number of states we use a logarithmic scale. The red squares denote the estimates assuming that the state space is a single random state space. The blue circles indicate the estimated number of states assuming that the state space consists of two random parallel systems. The estimated fan-outs for two parallel systems are drawn using purple ellipses and they use a linear scale. In figure 2 $1e5$

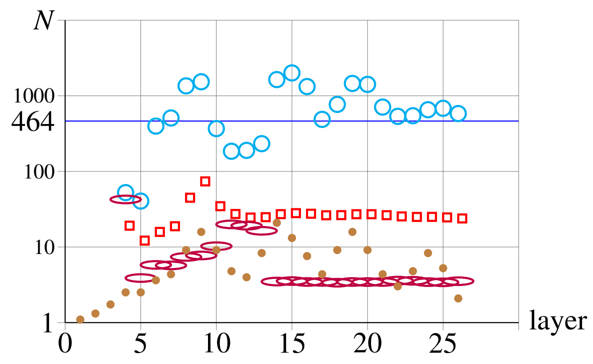

Figure 3: Estimates of the reachable state space of CABP

corresponds to a fan-out of thousand and in figure 3 100 corresponds to a fan-out of 10. The layer sizes are depicted by a brown dot, also on a linear scale. In figure 2 100 corresponds to a layer size of 5000 states. In figure 3 10 corresponds to a layer size of 25. The numbers are calculated using the least square method using Matlab [1] (see [7] for the actual MATLAB code used). To prevent the individual sizes of the components to become very different in the estimations, we add a small penalty $(N_1 - N_2)^2 10^{-6}$ to the squared difference of input data and estimation. The knowledge of the fan-out λ is not used. So, for the blue circles variables λ_1, λ_2 and N_1 and N_2 are estimated. Using these the size of the reachable state space is calculated and put in the figure.

It is obvious that the red boxes structurally underestimate the size of the reachable state space, where the blue circles do quite well. The figures in table 1 and figure 2 for the estimated reachable state space assuming that the state space is not parallel are different (compare $N_{\text{reachable}}$ with the red squares). This difference is due to the different estimation techniques. In section 3 it is explicitly assumed that the number of states cannot be less than the number of observed states (concretely, the lowerbound i_m in equation 6). In the estimation in this section, using the least square method, this information is not used. But the common denominator, namely structural underestimation of the reachable state space, is visible in both cases.

The fact that the obtained blue estimates are doing much better supports the claim that ‘real systems’ behave as parallel non communicating state spaces. That the results contain variations is to be expected. If we look at 3D visualisations of the state space of the firewire protocol, then the two peaks in the blue circles at layers 30 and 63 match very neatly with the disks in this 3D visualisation [6]. It is unexpected that these local peaks are very high, especially because one expects a dampening effect of taking all layers into account. In this light it is also strange that the red boxes are hardly influenced by these two peaks in layer sizes.

5 Estimating the presence of residual bugs

In this section we show that it matters very much if the system has a product graph structure if it comes to the effectiveness of testing for software bugs. It shows that if graphs have a single-threaded random structure, testing for the presence of bugs is virtually impossible, but if graphs have a product structure this is quite feasible.

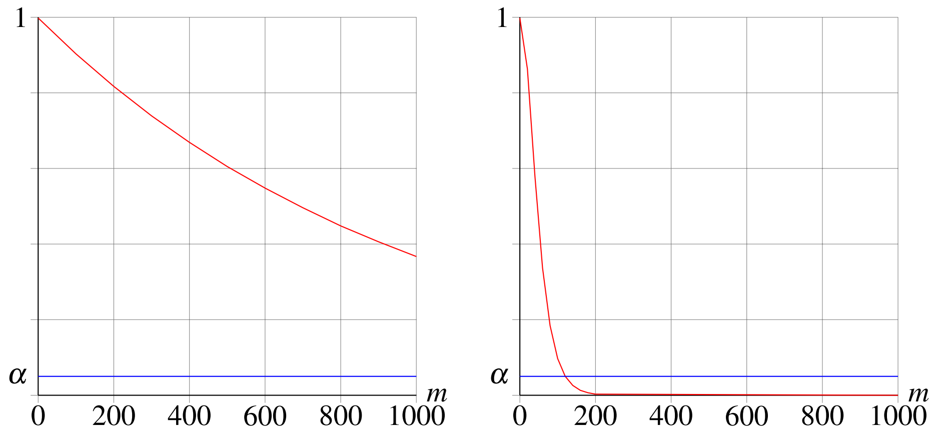

Figure 4: The probability of incorrectly declaring a system error free in G at the left and $G_1,\dots,G_p$ at the right for $p = 3$ and $n = 10$.

Assuming that realistic systems have a parallel structure explains why in practice many bugs are found quite easily by testing. Only rarely bugs are hard to find and such bugs typically occur when a number of components are in very specific states simultaneously. It suggests that bugs in programs must be classified depending on the number of components that must be in a specific state for such a bug to occur.

We calculate how long a successful random walk through the state space must be to conclude with error α that there are no buggy states in the system. For this purpose we introduce two stochastic variables, namely M which represents the number of states in a walk without encountering a bug, and K which is the number of states with a bug. We assume that K is uniformly distributed where there are between 0 and n^p bugs. This upper bound on the number of bugs has little influence on the length of the random walk satisfying small α. So, we could as well assume that we know that there is a small upper bound on the number of states with bugs, without altering the results.

We compare a single random state space G of size n^p with the product of p state spaces $G_1,\dots,G_p$, each of size n. We first calculate the probability that although there are states with bugs in the system, a random walk of size m does not encounter an error state in G. We want this probability to be smaller than α such that if we conclude that the system is free of bugs on the basis of a random walk of a certain size, the probability that this is incorrect is smaller than α. Our estimation assumes that each time we visit a state its outgoing transitions go to random other states, possibly different than in a previous visit. Strictly spoken this is not correct as the outgoing transitions of a state are static, and when we revisit a state in a random traversal, the probability that the outgoing state is also already visited is higher than in our estimation. Taking the already visited states into account is a known difficult problem, and our slightly simplified model is already very useful to heuristically show the effect of the graph structure on testing. We compute

$$
\begin{aligned}
\mathbb{P}[K>0|M=m] &= \frac{\sum_{k=1}^{n^p}\mathbb{P}[M=m|K=k]\,\mathbb{P}[K=k]}{\sum_{k=0}^{n^p}\mathbb{P}[M=m|K=k]\,\mathbb{P}[K=k]}\\
&= \frac{\sum_{k=1}^{n^p}(n^p-k)^m\,\mathbb{P}[K=k]}{\sum_{k=0}^{n^p}(n^p-k)^m\,\mathbb{P}[K=k]}\\
&= 1-\frac{n^{pm}\mathbb{P}[K=0]}{\sum_{k=0}^{n^p}(n^p-k)^m\mathbb{P}[K=k]}\\
&= 1-\frac{n^{pm}}{\sum_{k=0}^{n^p}(n^p-k)^m}<\alpha \qquad (18)
\end{aligned}
$$

At (18) we used that we assume that errors are uniformly distributed over the interval $[0,n^p]$. So, $\mathbb{P}[K=0]$ equals $\mathbb{P}[K=k]$.

For the Cartesian product graph of G_1 to G_p we come to the following estimation where we assume that graph G_i has K_i error states, uniformly distributed from 0 to n. We use the notation $\text{if}(c,x,y)$ to represent x if c holds, otherwise it is y.

$$
\begin{aligned}
&\mathbb{P}[(\textstyle\sum_{i=1}^{p}K_i)>0|M=m]\\
&= \frac{\sum_{k_1=0}^{n}\cdots\sum_{k_p=0}^{n}\text{if}(\sum_{j=1}^{p}k_j{=}0,0,\mathbb{P}[M{=}m|\bigwedge_{i=1}^{p}K_i{=}k_i]\,\mathbb{P}[\bigwedge_{i=1}^{p}K_i{=}k_i])}{\sum_{k_1=0}^{n}\sum_{k_2=0}^{n}\cdots\sum_{k_p=0}^{n}\mathbb{P}[M{=}m|\bigwedge_{i=1}^{p}K_i{=}k_i]\,\mathbb{P}[\bigwedge_{i=1}^{p}K_i{=}k_i]}\\
&= \frac{\sum_{k_i=0}^{n}\cdots\sum_{k_p=0}^{n}\text{if}(\sum_{j=1}^{p}k_j=0,0,(1-\sum_{i=1}^{p}k_i/pn)^m\,\mathbb{P}[\bigwedge_{i=1}^{p}K_i=k_i])}{\sum_{k_1=0}^{n}\sum_{k_2=0}^{n}\cdots\sum_{k_p=0}^{n}(1-\sum_{i=1}^{p}k_i/pn)^m\,\mathbb{P}[\bigwedge_{i=1}^{p}K_i=k_i]}\\
&= 1-\frac{\mathbb{P}[\bigwedge_{i=1}^{p}K_i=0]}{\sum_{k_1=0}^{n}\sum_{k_2=0}^{n}\cdots\sum_{k_p=0}^{n}(1-\sum_{i=1}^{p}k_i/pn)^m\,\mathbb{P}[\bigwedge_{i=1}^{p}K_i=k_i]}\\
&= 1-\frac{1}{\sum_{k_1=0}^{n}\sum_{k_2=0}^{n}\cdots\sum_{k_p=0}^{n}(1-\sum_{i=1}^{p}k_i/pn)^m}<\alpha
\end{aligned}
$$

To obtain the third expression in the derivation above, we observe that a step that does not reach a state with a bug, does not hit such a state in any of the components. A component i has probability $(n-k_i)/n$ to avoid a state with a bug. Assuming that with equal probability each component can do a step, the probability to avoid a state with a bug is $\sum_{i=1}^{p}(n-k_i)/pn$. So, the probability to avoid m times a buggy state is $(1-\sum_{i=1}^{p}\frac{k_i}{np})^m$.

In figure 4 the estimates are depicted for a system with 1000 states versus three systems with 10 states. The difference is quite obvious. In the monolithic system a test run far longer than the number of states must be traversed to declare the system error free with $\alpha=0.05$. With the parallel system a test run of far less than 200 steps is more than sufficient.

Notably, the number of required test runs only increases with the number of states per component. So, for four or more components with each 10 states test runs of approximately 200 are also enough to obtain a certainty of 95% on the freedom of errors, whereas with the monolithic system test runs must have a length of millions of steps to obtain the same effect.

6 Conclusions and future work

Despite the in our view promising experiments showing that P-parallel random transition systems are a good candidate to act as a model for real behaviour, more experimental data is required to determine whether this is really as general as we think. Besides this, there are quite a number of open technical questions. Solving N_i and λ_i in equation (14) is too time consuming, taking hours for all 107 layers of the firewire protocol. It is unclear how to determine the index of parallelism P from experimental data. We fixed P a priori to 2 and 3 [7]. It is unclear how to calculate the probability of a random run in a directed graph to hit a bug (even when $P = 1$).

Understanding the effectiveness of 'highway'-search as mentioned in the introduction is still far beyond reach. This paper only contains a possible single step towards answering it.

References

[1] S. Attaway. Matlab, third edition: a practical introduction to programming and problem solving. Third Edition. Butterworth-Heinemann publishers, Elsevier, 2013.

[2] B. Bollobás. Random Graphs. Cambridge University Press, 2001.

[3] N.J. Dingle and W.J. Knottenbelt. State-Space Size Estimation By Least-Squares Fitting. In *Proceedings of the 24th UK Performance Engineering Workshop (UKPEW'08)*, pages 347–357, 2008.

[4] T.A.N. Engels, J.F. Groote, M.J. van Weerdenburg and T.A.C. Willemse. Search algorithms for automated validation. J. of Logic and Algebraic Programming 78(4), 274-287, 2009.

[5] J.F. Groote et al. The mCRL2 toolset. In *Proceedings of the International Workshop on Advanced Software Development Tools and Techniques (WASDeTT'08)*, 2008.

[6] J.F. Groote and F.J.J. van Ham. Interactive visualization of large state spaces. International Journal on Software Tools for Technology Transfer 8:77-91, 2006.

[7] J.F. Groote, R.W. van der Hofstad, and M. Raffelsieper. On the random structure of behavioural transition systems. Computing Science Report CS-R1401. Department of Mathematics and Computer Science. Eindhoven University, 2014.

[8] J.F. Groote and M.R. Mousavi. Modeling and analysis of communicating systems. The MIT Press, 2014.

[9] S.P. Luttik. Description and formal specification of the link layer of P1394. In: Ignac Lovrek, editor, Proceedings of the 2nd International Workshop on Applied Formal Methods in System Design, University of Zagreb, Croatia, pp. 43-56, June 1997.

[10] S. Mauw and G.J. Veltink (editors). Algebraic specification of communication protocols. Cambridge tracts in theoretical computer science 36. Cambridge University Press. 1993.

[11] J.F. Watson III and A.A. Desrochers. State-space size estimation of Petri nets: A bottom-up perspective. *IEEE Transactions on Robotics and Automation*, 10(4):555–561, 1994.

[12] Wolfram Research Inc. Mathematica 8.0.0.0. See also http://www.wolfram.com.

[13] F.C. Lincoln. Calculating Waterfowl Abundance on the Basis of Banding Returns. *United States Department of Agriculture Circular*, 118:1–4, 1930.

[14] R. Pelánek and P. Šimeček. Estimating State Space Parameters. Technical Report FIMU-RS-2008-01, Faculty of Informatics, Masaryk University, 2008.