Convergence in Model Checking of Random Kripke Structures

DENIS TIMOFEEV FORNASOV, University of Twente, The Netherlands

ABSTRACT

Randomly generated Kripke structures (graphs with states, initial states and directed edges) are used in model checkers to evaluate benchmarks, this consists of checking the truth of a temporal logic formula, specifically for this research, a Computation Tree Logic (CTL) formula, for the given structure. It has been shown by previous research that, as these graphs grow in size (node/state count), the probability of truth of any given CTL formula converges to a fixed value. This paper focuses on a gap in the current research, which is determining how fast this convergence occurs. The paper demonstrates how the convergence happens very quickly, at graph sizes not exceeding a few hundred nodes even for very complex CTL formulae. This greatly impacts model checking because graph sizes of tens thousands of nodes are typically used. Since it has been shown how to calculate the convergence probabilities by previous research, and we now know that these probabilities are converged to very quickly, this phenomenon can be exploited by model checkers by probabilistically guessing whether a given formula holds for a model instead of actually verifying whether it does. The paper also shows that the convergence happens exponentially, and the pattern of how graph generation parameters (probability of edges occurring, probability of states being labeled as initial) also affect the rate of convergence.

1 INTRODUCTION

1.1 Background and Context

Model checking is an automated method for checking whether a model of a system satisfies certain correctness properties[1]. For this research, a 'model' refers specifically to a Kripke structure[6], which is, in principle, similar to a finite-state automaton: a graph with one or more initial states, nodes that represent reachable states and directed edges which represent transitions between states, but with a focus on interpretation of temporal logics. In general, a temporal logic is any system of rules used to reason about events in terms of time (for example, "I am always hungry" or "I will be hungry until I eat something").

In the field of model checking, it is important to have benchmarks which can be used on a variety of different models to measure the performance of model checkers. In the context of this paper, a benchmark is a logic formula or set of logic formulae used to evaluate a model. Organizations such as the Hardware Model Checking Competition (HWMCC)[7] have developed large numbers of models and benchmarks which are used to evaluate model checkers. However, as the models being tested become larger, logic formulae follow 0-1 and convergence laws[3]. A logic formula adheres to convergence law if, as you increase the size of a random model, the probability of the logic formula holding converges to a fixed value (from 0 to 1). If the probability of the formula holding converges strictly to only 0 or 1, then it adheres to 0-1 law. It has also been found by previous studies that convergence law is followed for single-initial state Kripke structures, while 0-1 law is followed for multiple-initial state Kripke structures[3].

1.2 Problem Statement

While it is known that this convergence of probability occurs and how to calculate it, there is little research to show exactly *when* the convergence occurs. 'When', in this case, refers to how large the model needs to be for the formula's probability of holding to converge. This is important because if we find that the convergence happens relatively 'early' (for example, for model sizes smaller than those used to test benchmarks) this would mean that many of the benchmarks currently being used are not as reliable as expected. On the other hand, if the convergence only happens when the models are incredibly large (larger than any models used in practice), this would mean that presently used model checkers and benchmarks remain unaffected. The HWMCC uses models based on electrical circuits, many of which have thousands and even millions of nodes [8]. So, if convergence for a CTL formula (see Section 2 for detailed explanation of CTL) occurs within merely hundreds of nodes on

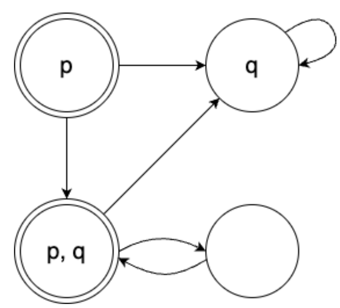

Fig. 1. Example of Kripke Structure satisfying AF(p) and AF(q). AF(p) means: on all paths (starting at initial, double-circled, states), you will eventually reach a state containing the atomic proposition p. This holds trivially for this example because both initial states contain p. For $AF(q)$, one of the initial states already contains q, the other has 2 outgoing transitions, both of which lead to a state with q. The empty state simply means neither p nor q holds in that state.

Author's address: Denis Timofeev Fornasov, University of Twente, The Netherlands.

any random Kripke structure, model checkers partaking in such a competition would be able to calculate the fixed convergence probability of the given CTL formula for the given Kripke structure according to the findings of Dong et al. [3] and, given the large graph sizes used, the actual probability of truth of the CTL formula holding for said Kripke structure would be converged to the fixed value calculated, resulting in the model checker essentially being able to perform a probabilistic guess instead of actually verifying whether the formula holds for the given model. This would enable model checkers to greatly increase their speed while maintaining accuracy, without actually performing the verification or calculation expected of them, working purely on probability.

1.3 Objective and goals

The paper which found the convergence to occur[3] proved its findings for Computation Tree Logic (CTL) and Linear Temporal Logic (LTL)[1]. In this research, the focus is limited to Computation Tree Logic (CTL), which views time as a branching tree of possible futures. Further explanation on CTL follows in Section 2. The objective of this study is to experiment with many randomly generated models and derive a pattern of how large these models need to be for the convergence to occur. This is tested for both single-initial and multiple-initial state models. CTL is chosen due its polynomial time complexity, making it feasible to compute sufficient results within the time frame of this study, as opposed to the PSPACE-complete complexity (requiring exponential time) of LTL[1].

2 TERMINOLOGY AND DEFINITIONS

- Kripke Structure[6]: A mathematical model, in the form of a graph, which represents the behavior of a system. The states of the system are represented by nodes, the transitions between the different states are represented by directed edges, and initial states (labeled as a double-circled node in Figure 1) which represent the points from which we begin to evaluate the behavior of a CTL formula.
- CTL[1]: Extends propositional logic (basic boolean operations over atomic propositions) with temporal operators. For example, the formula $AF(q)$ states that on all paths (A) we will eventually reach (F) a state where atomic proposition q holds. As seen in Figure 1, the formula $AF(q)$ is satisfied, by beginning at either initial state, no matter which transitions we follow, we will eventually reach a state that contains q.
- Random Kripke Structure[4, 5]: Interchangeably referred to as graphs or models in this paper, random Kripke structures are just Kripke structures which are not modeled after an actual system, but have instead been generated following a set of rules. For this research, the Erdős–Rényi[5] strategy for random graph generation is used. The Erdős–Rényi approach consists of selecting a graph size n (number of nodes) and probability $p_{transition}$ of each possible edge between two nodes existing. This process is described in further detail in Section 3.1. The importance in using randomly generated models lies in obtaining very large numbers of different graphs, while not modeling after any existing systems which could introduce unwanted patterns and bias.
- Convergence and 0-1 laws[3]: A convergence law is a property of a temporal logic and a random graph model. A convergence law states that, as the number of nodes in a random model (with the same generation parameters) increases, the probability that the temporal logic formula holds converges to a fixed value. For the case of multiple-initial state models, a 0-1 law is followed, which is the exact same principle but the fixed value is always either exactly 0 or 1.

3 METHODOLOGY AND APPROACH

3.1 Model Generation

As discussed, the models for this research are Kripke structures. The models are generated using the pyModelChecking[2] Python package. This tool was chosen because it is fully open-source, has a vast amount of documentation online and is specifically designed for working with CTL/LTL on Kripke structures. The Kripke structures are generated by setting the graph size n and the probability of transition $p_{transition}$, as per the Erdős–Rényi model. The following additional constraints are also set:

- All atomic propositions each have an independent 50% chance of appearing at any given state.
- Similar to transitions, each state has an independent probability $p_{initial}$ of being labeled as an initial state. For graphs or tables seen in this paper, single-initial state models are described as having $p_{initial} = 0.0$.
- For the sake of totality, if, due to chance, any node in a generated graph does not receive an outgoing edge, the graph is discarded and one is re-generated with the same parameters.

3.2 Model Checking

The pyModelChecking package allows verification of whether a certain CTL formula holds true for the encoded Kripke structure.

Table 1 shows the selected CTL formulae for this research. The formulae follow an increasing complexity, although no strict metric is used to quantify their complexity, simply higher number of atomic propositions and clauses. Since we are testing randomly generated graphs, the formulae are not based on any real behavior of a system, rather they are also random.

3.3 Raw data collection

The research process for getting the raw data (probabilities of truth of different formulae for increasing graph sizes) is broken down in Figure 2. For this paper, the step used in the graph size ranges (the amount by which n is increased when moving onto a larger graph size) is kept at 2 to balance granularity and data generation time. The iteration count, or the number of graphs averaged to obtain the final probability, is 10000 for $\varphi_{1..9}$ and 1000 for φ_{10}, reduced for φ_{10} due to very large computation times.

3.4 Data analysis and statistical techniques

3.4.1 Convergence Point Calculation. The convergence point is calculated by taking the 20-80 inter-percentile range in a window of 10 data points (this equates to a range of 20 nodes since a step of 2 is used for all data collected). If the inter-percentile range is less

φ	CTL
φ_1	$AF\ p$
φ_2	$EF\ (EX\ (r))$
φ_3	$EF\ (p \land EF\ (q))$
φ_4	$EG\ (p \land (q \lor r))$
φ_5	$AG\ (p \to AF\ (q))$
φ_6	$A\ ((p \to q)\ U\ r)$
φ_7	$AG\ (p \to AF\ (EX\ (\neg q)))$
φ_8	$AG\ (p \to AF\ (q \land EX\ (E\ (r\ U\ s))))$
φ_9	$AG\ (EF\ (p \land EX\ (q)) \to EG\ (\neg r \lor A\ (s\ U\ AF\ (t))))$
φ_{10}	$AG\ ((p \to AF\ (q \land EX\ (r \lor E\ (\neg s\ U\ t)))) \lor EG\ (\neg q \land E\ (p\ U\ (s \lor AF\ (r)))))$

Table 1. CTL formulae.

than 0.01 for 4 consecutive windows, the data is deemed to have converged.

3.4.2 Setup for Exponential. In order to demonstrate an exponential relationship, the following steps were taken:

- Find the limit towards which the data is converging. For data that is visibly tending to either 0 or 1, those values are manually selected. For limits which are unclear, SciPy's $curve\_fit$ was performed on the raw data to fit an exponential curve of the following general form:
$$p = L + d * e^{kn}$$
Where p is the probability of truth of φ, L is the limit the curve converges to, d and k are coefficients of the exponential, and n is the number of nodes in the graph.
- If the raw data is converging to an upper limit then we adjust the y-data (probabilities) by performing:
$$adjusted\_probs = L - probs$$
where $probs$ is the list of probabilities of truth.
- If, however, the raw data is converging to a lower limit we adjust the y-data by performing:
$$adjusted\_probs = probs - L$$
- Filter out any non-positive values and take the base 10 logarithm of the filtered $adjusted\_probs$.
- Perform linear regression on the filtered node range vs. the filtered $probs$.

If the relationship is, in fact, exponential, this result should yield a straight line with a downward trend.

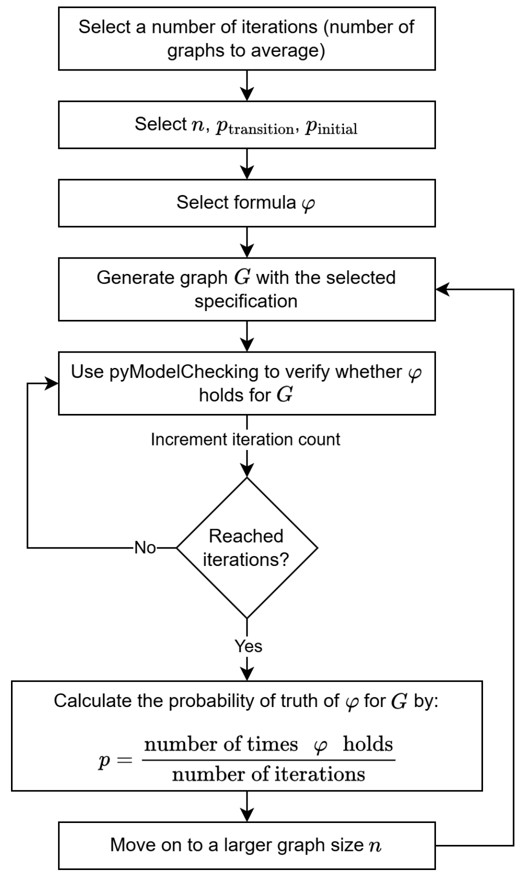

Fig. 2. Flowchart of raw data collection process for a single configuration.

3.4.3 Omission of Values. Due to irregular behavior, which will be explored in Section 4, for smaller graph sizes, some values are omitted before performing curve-fitting and linear regression. Since the raw data is experimental and not perfect, some values may fall above an approximated upper asymptote or below an approximated lower asymptote, these values are also omitted.

4 RESULTS

4.1 Convergence Speed

Looking at Figure 3, we see a clear convergence occurring within merely a couple hundred nodes for both φ_1 and φ_{10}. Notice how, despite the CTL formulae being of vastly differing complexity, the convergence still happens rather quickly in both cases.

As seen in Table 2, all of the CTL formulae converge for relatively small graph sizes, not exceeding a couple hundred nodes. It is also evident from Table 2 and Figure 4 that the formulae have very similar, in some cases even the same, convergence points for single-initial and multiple-initial state models, despite the fact that the values being converged to in those cases are different.

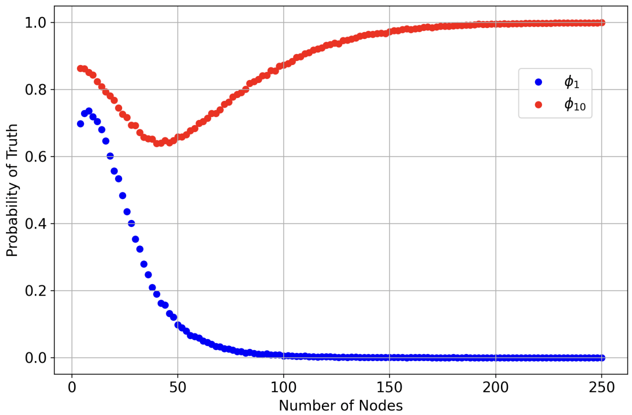

Fig. 3. Probability of Truth vs. Graph Size (Number of Nodes) for φ_1 (blue) and φ_{10} (red), both with $p_{transition} = 0.1$ and $p_{initial} = 0.1$

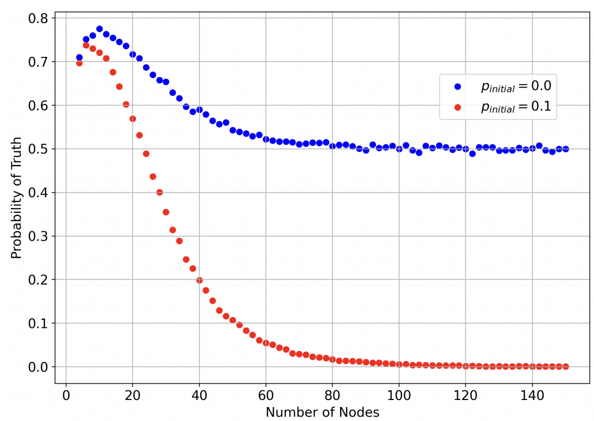

Fig. 4. Probability of Truth vs. Graph Size (number of nodes) for φ_1 with single-initial and multiple-initial state model.

4.2 Exponential Behavior

Although far more noticeable for φ_{10} than φ_1, both curves experience visibly non-exponential behavior at the beginning. This is likely due to the graph size not yet being large enough to demonstrate the behavior of the full CTL formula, with all of its clauses, consistently. By performing linear regression on the base 10 logarithm of the probabilities, following the steps in Section 3.4.2, it becomes clear that the probability of truth does, indeed, follow an (inverse) exponential curve. The data is more sparse towards the end, past the point of convergence, due to many of the probabilities equaling 0 and having to be excluded. The data also becomes increasingly jitter, due to the logarithm producing a more extreme effect on the data as it tends to the limit. Furthermore, the resolution of the results is 10^{-4} for φ_1 and 10^{-3} for φ_{10} due to the 10000 and 1000 graphs being averaged for the two formulae respectively, meaning that the data below the 10^{-4} and 10^{-3} marks, respectively for the two formulae when plotting the logarithm, becomes unreliable. Hence, the best fit line is cut off earlier to fit the consistent exponential section, this cut off is done by eye, not directly at the limit of resolution. The initial non-exponential section is also excluded by eye (notice the x-axis).

Formula	Convergence Point	
	Single-Initial	Multiple-Initial
φ_1	82	98
φ_2	44	48
φ_3	46	52
φ_4	108	78
φ_5	64	64
φ_6	86	100
φ_7	108	110
φ_8	62	62
φ_9	64	64
φ_{10}	174	176

Table 2. Convergence Points found for all selected CTL formulae. All generated models use $p_{transition} = 0.1$, single-initial state models use $p_{initial} = 0.0$ and multiple-initial state models use $p_{initial} = 0.1$. Note that these convergence points are subject to deviation if recalculated due to the relatively weak nature of detecting the convergence (in detail in Subsection 3.4 and the averaging over only 10000 graphs for the sake of computation times for this research.

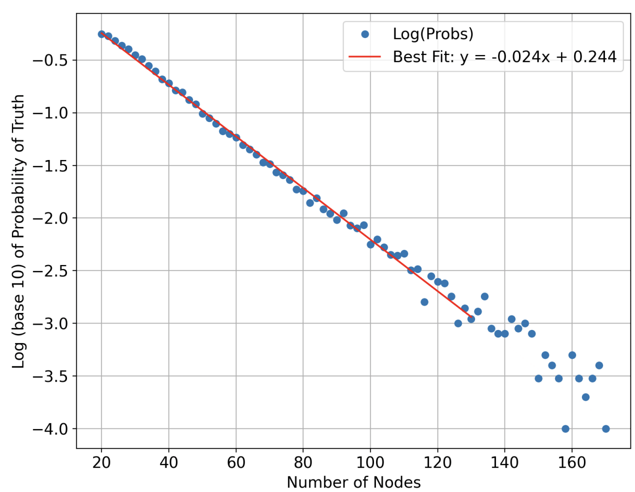

Fig. 5. Log (base 10) of Probability of Truth vs. Graph Size (number of nodes) for φ_1 with non-exponential section cut off by eye. $R^2 = 0.9948$. $RMSE = 0.0574$. Fit for nodes 20 to 130.

4.3 Impact of Generation Parameters

As visible in Figures 7&8, it is clear that the value of $p_{transition}$ has a distinct effect on the convergence: a greater $p_{transition}$ causes a quicker convergence. For φ_1, the actual convergence point does not differ, as the curves seem to merge into one another, however, for φ_{10} the convergence point decreases very noticeably, reducing

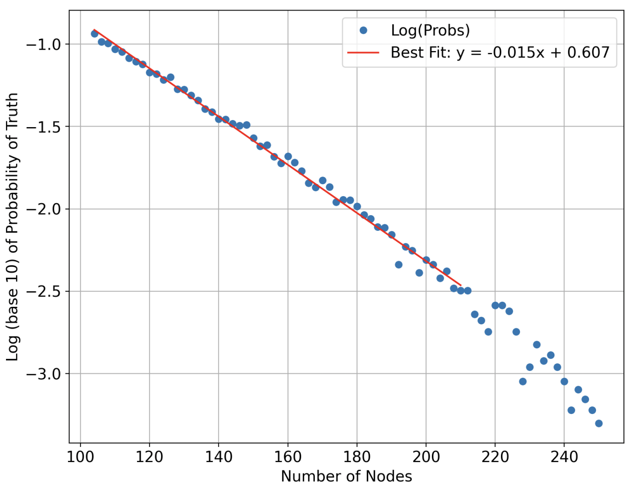

Fig. 6. Log (base 10) of Probability of Truth vs. Graph Size (number of nodes) for φ_{10} with non-exponential section cut off by eye. $R^2 = 0.9940$. $RMSE = 0.0354$.

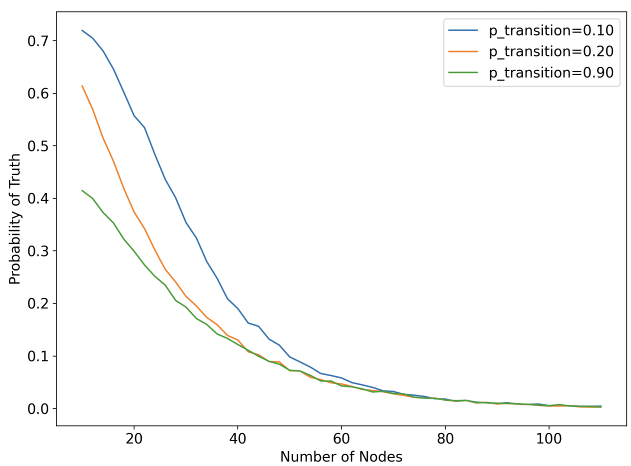

Fig. 7. Probability of Truth vs. Graph Size (number of nodes) for φ_1 with constant $p_{initial} = 0.1$ and different values of $p_{transition}$.

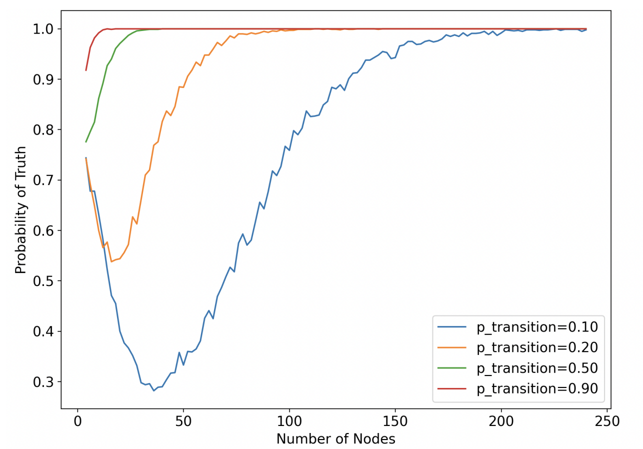

Fig. 8. Probability of Truth vs. Graph Size (number of nodes) for φ_{10} with constant $p_{initial} = 0.1$ and different values of $p_{transition}$.

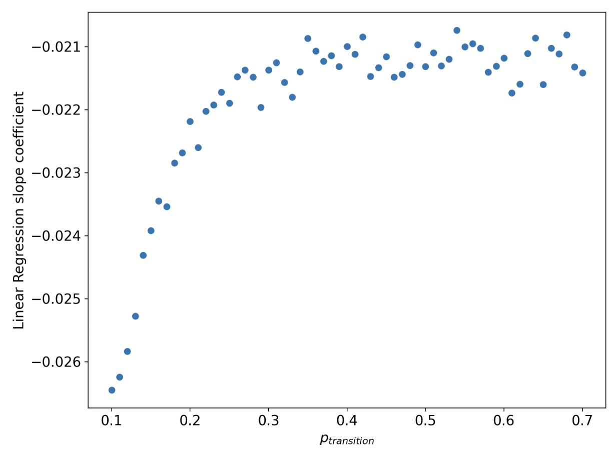

Fig. 9. Linear Regression (of the base 10 log of the probability of truth) slope coefficient vs. increasing $p_{transition}$ for φ_1 with constant $p_{initial} = 0.1$ All slopes fit with linear regression of nodes 20 to 130.

the non-exponential section at the beginning for higher values of $p_{transition}$. Also notice how the convergence curve of φ_1 for $p_{transition} = 0.2$ is actually closer to the curve for $p_{transition} = 0.9$ than to $p_{transition} = 0.1$. This means that some form of a non-linear relationship is also followed for the effect of $p_{transition}$ on the convergence speed.

By plotting the slope coefficients of the linear regression lines fitted from the logarithm of the probabilities for a more fine-grained range of values for $p_{transition}$, seen in Figure 9, we clearly see that a relationship resembling an exponential curve is followed for increasing values of $p_{transition}$. It is, however, important to note the scale of the y-axis, indicating how small the difference actually is in this case. Also notice a misleading factor: seeing as the slopes are negative, the regression lines are more steep for lower values of $p_{transition}$, however, this does not indicate that the probability converges faster for those values. If we look back at Figure 7, we see that the curves already begin at much lower values (around 0.7 for $p_{transition} = 0.1$ but around 0.4 for $p_{transition} = 0.9$). So, despite the curve for $p_{transition} = 0.9$ being closer to the limit being approached, its slope is lower due to the lower starting point.

Looking at Figure 10, we get a further indication that the difference for varying values of $p_{transition}$ is clear, although not very largely impactful for φ_1, even for a very large difference in $p_{transition}$. Further towards the end, the data becomes more jittery and intertwined (notice the darker, overlapping points), the possible reasons for this were discussed at the end of Section 4.2. Although the lines of best fit interpolated from the data intersect very clearly, it is difficult to say at what point they would intersect if the resolution of the results was greatly increased.

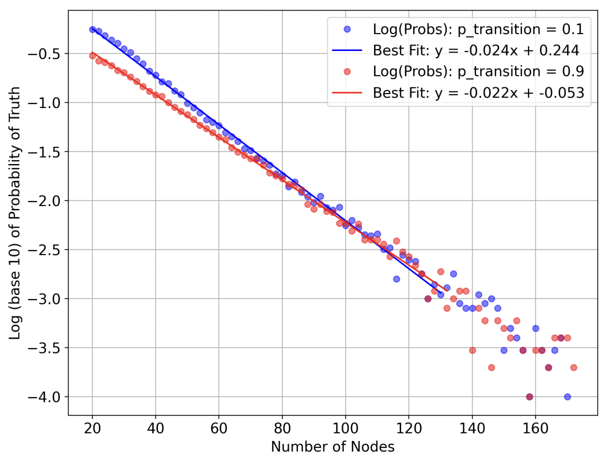

Fig. 10. Log (base 10) of the Probability of Truth vs. Graph Size (Number of Nodes) for φ_1 with constant $p_{initial} = 0.1$ and different values of $p_{transition}$.

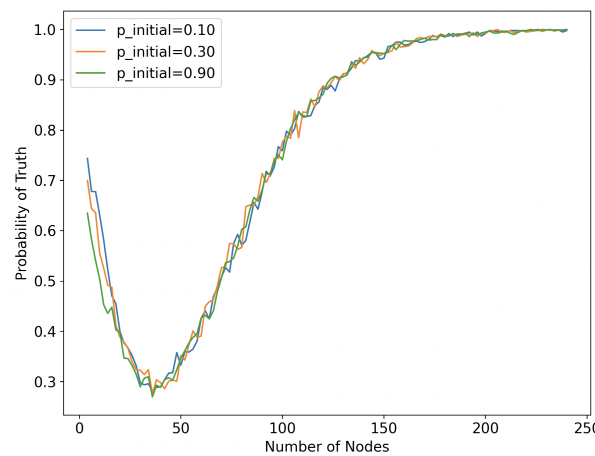

Fig. 12. Probability of Truth vs. Graph Size (number of nodes) for φ_{10} with constant $p_{transition} = 0.1$ and different values of $p_{initial}$.

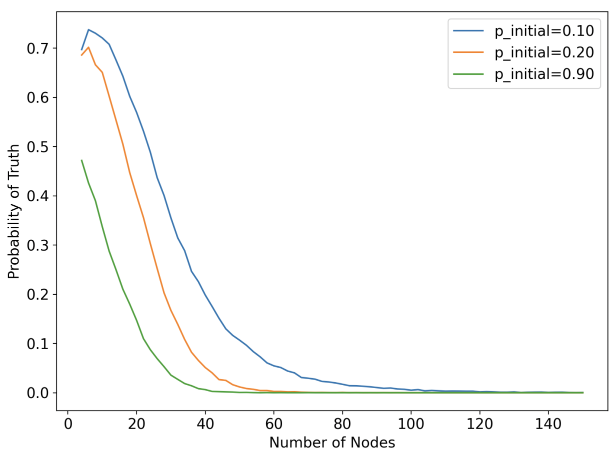

Fig. 11. Probability of Truth vs. Graph Size (number of nodes) for φ_1 with constant $p_{transition} = 0.1$ and different values of $p_{initial}$.

Figure 11 shows the impact of $p_{initial}$, and it clearly has a greater effect on the convergence speed than $p_{transition}$ for φ_1.

Figure 12, on the other hand, shows a completely different effect for φ_{10}. We see no effect whatsoever, especially since the slight deviations are due to lower averaging being used for this formula. The effect of both $p_{transition}$ and $p_{initial}$ varies depending on the specific CTL formula used, but greater values for both parameters appear to either increase the rate of convergence or not affect it.

5 CONCLUSIONS

To conclude, this research demonstrates that the convergence of the probability of truth of CTL formulae in random Kripke structures happens very quickly. Even for CTL formulae with many nested clauses, the convergence occurs within a few hundred nodes. This is seen for both single-initial and multiple-initial state models. Furthermore, the convergence happens exponentially with respect to the number of states (nodes) in the Kripke structure, after an initial period of non-exponential behavior, the size of which depends on the CTL formula and generation parameters used, namely the probability of transitions being added to the graph and the probability of states being labeled as initial. Finally, the magnitude of the effect of the generation parameters differs depending on the CTL formula used, but greater values of $p_{transition}$ and $p_{initial}$ generally increase the rate at which the convergence occurs, or have no effect on it. Since many of the graphs used to evaluate model checkers are of sizes far greater than the convergence points found in this research, when a model checker is given some random model and some CTL formula, it could exploit the convergence phenomenon by calculating the fixed value to which the probability of truth of said CTL formula on said model converges[3], then make a probabilistic guess on whether the formula holds or not, depending on the calculated value, instead of actually verifying the formula's truth (as would be expected of it in the context of benchmarking a model checker). This is problematic because model checkers are able to greatly increase their speed without actually being better model checkers. Moreover, multiple-initial state models give rise to a bigger problem than single-initial state models due to their following of 0-1 law, since, in such a case, a CTL formula on a given model would hold either 100% of the time or 0% of the time if the graph size is beyond the convergence point for that model and formula combination, allowing for a trivial verification of the formula. For single-initial state models (with the probability of some CTL formula holding converging to a value not equal to 0 or 1), the outcome is not as trivial, but could still give some insight or the opportunity for a guess.

6 FUTURE WORK

The convergence phenomenon has great possibility for further research and future work to build upon the findings of this paper and the findings of Dong et al. Future work may include:

- Repetition of the experiments for a wider range of CTL formulae, as well as LTL formulae.
- Repetition of the experiments with larger graph averaging for higher resolution results.
- Repetition of the experiments on the effect of $p_{transition}$ and $p_{initial}$ for a variety of different formulae with more fine-grained ranges of $p_{transition}$ and $p_{initial}$.
- Research into how the complexity of CTL formulae affects the convergence point, with formally defined complexity rather than simply different formula lengths.
- Repetition of the experiments with different random graph generation strategies (other than the Erdős–Rényi[5] approach used in this paper and that of Dong et al.).

REFERENCES

[1] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking.* MIT Press.
[2] Alberto Casagrande. 2024. pyModelChecking. https://pypi.org/project/pyModelChecking/. Accessed: 2025-05-27.
[3] Yanni Dong, Milan Lopuhaä-Zwakenberg, and Mariëlle Stoelinga. 2025. 0-1 Laws for LTL and CTL over Random Transition Systems. Forthcoming. To appear in the 31st International Symposium on Model Checking of Software (SPIN).
[4] Paul Erdős and Alfréd Rényi. 1959. On random graphs I. *Publicationes Mathematicae (Debrecen)* 6 (1959), 290–297.
[5] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5 (1960), 17–61.
[6] Michael Huth and Mark Ryan. 2004. *Logic in Computer Science: Modelling and Reasoning about Systems* (2nd ed.). Cambridge University Press, Cambridge, UK. 2nd edition.
[7] HWMCC organizers. 2025. Hardware Model Checking Competition (HWMCC). https://hwmcc.github.io/. Accessed: 2025-06-16.
[8] Mathias Preiner, Nils Froleyks, and Armin Biere. 2024. HWMCC'24 Benchmarks and Results. https://zenodo.org/records/14156844. Accessed: 2025-06-22.