Challenges of the Reachability Problem in Infinite-State Systems

Paper authored by Wojciech Czerwiński

Some Useful Definitions and Reminders

- Push down automata: is a finite automata with extra memory called stack which helps the automata to recognize **context-free languages**. Basically, they can store more information with the said stack which can hold an unlimited number of symbols.

- **Context Free Languages**: Context-free grammar is a set of recursive rules used to generate patterns of strings. It is usually represented as a 4-tuple of finite set V of variables(that are non-terminal), Σ, the alphabet, R which is the set of production rules and a starting symbol $s \in V$. Context-free language is defined by context-free grammar and accepted by pushdown automata.

- A reminder: Halting Problem is the question of whether or not a program will complete execution (ie, it terminates) for a given input or will it run forever. This is an UNDECIDABLE problem.

- Ackermann Complexity: $O(2^{2^{2^{\cdot^{\cdot^{2}}}}})$

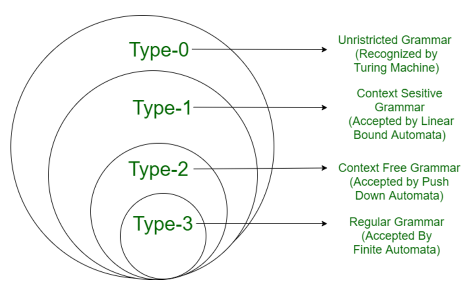

Figure 1: Chomsky Hierarchy of Grammar

1 Introduction

Note: This was a paper presented in a conference so there are a lot of questions instead of answers.

The reachability problem stems from Turing's undecidability of the halting problem. In essence, the problem asks whether, given the initial and final configuration of a system, there exists a sequence of transitions that connects the two. Early undecidability results were established for automata models such as automata with two pushdowns, two counters with zero-tests, and Turing machines. More restricted models, like automata with fewer computational resources, remain a focus of study to determine which conditions lead to decidability.

2 Undecidability of Reachability in Complex Systems

One of the first models proving undecidability was automata with two stacks. The simulation of a Turing machine by an automaton with two stacks shows that its reachability problem is undecidable. Further reductions led to the result that the problem remains undecidable even for automata with two zero-tested counters, a system commonly known as a Minsky machine.
The complexity of simulating two stacks with counters involves using binary encodings and arithmetic on the counters, leading to deep complexity results. Through this reduction, it is possible to simulate one stack with two counters and simulate an automaton with two stacks using only three counters with zero-tests.

3 Decidable Models and Restrictions

Research has shifted focus to models that restrict computational power in ways that render the reachability problem decidable. Several natural restrictions include:

1. Disallowing zero-tests on counters.
2. Allowing counters to hold negative values.
3. Permitting a single zero-tested counter.
4. Combining one stack with counters that are either non-negative or unrestricted in their values.

Each of these restrictions generates models of varying computational complexity, some of which are decidable, while others remain open.

3.1 Vector Addition Systems (VAS)

One well-known decidable model is the Vector Addition System (VAS), where transitions modify the values of counters by fixed increments or decrements without zero-tests. VAS systems, which can be interreducible with Petri nets, have a decidable reachability problem. The complexity of this problem, however, is Ackermann-complete, indicating that its solution may require extremely large computational resources.

For VAS of small dimensions (denoted d-VAS), the complexity increases as the number of counters grows. While VAS with two counters (2-VASS) is known to be PSpace-complete, higher dimensions lead to towering complexity, sometimes stretching beyond Ackermann levels.

4 Open Problems and Challenges

Despite advances in understanding the complexity of reachability in restricted models, significant gaps remain. For example, the gap between known complexity bounds for 3-VASS and higher-dimensional systems is notable. The main open challenges include:

1. Identifying whether 3-VASS has a doubly-exponential shortest path between configurations.
2. Determining if there exist VASS with dimension $d \leq 7$ that exhibit tower-length shortest paths.
3. Proving or disproving the conjecture that fixed VASS systems always have paths of polynomial length.

These questions are fundamental to improving our understanding of computational limits in infinite-state systems.

4.1 Other Extensions and Variants

Another model of interest is the integer VAS (Z-VASS), where counters can take negative values. Though Z-VASS has been shown to have an NP-complete reachability problem, the addition of VASS counters complicates the model further. In particular, combining 2-VASS with multiple Z-counters could lead to new, unknown complexity results.

5 Automata with Zero-Tests and Pushdowns

Automata with two zero-tested counters already have undecidable reachability problems, but one active area of research explores the limits of decidability for systems with restricted zero-tests. A significant result by Klaus Reinhardt shows that VASS with one zero-tested counter, and even hierarchical VASS, have decidable reachability problems. However, the complexity of such systems remains poorly understood, with upper bounds yet to be settled.

Pushdown automata combined with VASS counters (PVASS) represent another difficult class of systems. While the reachability problem for context-free languages (pushdowns) is well-known and polynomial, adding VASS counters introduces considerable complexity, and the decidability of reachability for PVASS remains unresolved.

6 Concluding Remarks

The study of reachability in infinite-state systems is rich with open problems and challenges that have resisted resolution for decades. The undecidability results for more complex models contrast with the decidability of restricted models, showing that even slight variations in system structure can drastically affect computational power.

References

[1] Wojciech Czerwinski, *Challenges of the Reachability Problem in Infinite-State Systems*, 49th International Symposium on Mathematical Foundations of Computer Science,2024.