

Technical Perspective

The Equivalence Problem for Finite Automata

By Thomas A. Henzinger and Jean-François Raskin

FORMAL LANGUAGES AND automata are fundamental concepts in computer science. Pushdown automata form the theoretical basis for the parsing of programming languages. Finite automata provide natural data structures for manipulating infinite sets of values that are encoded by strings and generated by regular operations (concatenation, union, repetition). They provide elegant solutions in a wide variety of applications, including the design of sequential circuits, the modeling of protocols, natural language processing, and decision procedures for logical formalisms (remember the fundamental contributions by Rabin, Büchi, and many others).

Much of the power and elegance of automata comes from the natural ease with which they accommodate nondeterminism. The fundamental concept of nondeterminism—the ability of a computational engine to guess a path to a solution and then verify it—lies at the very heart of theoretical computer science. For finite automata, Rabin and Scott (1959) showed that nondeterminism does not add computational power, because every nondeterministic finite automaton (NFA) can be converted to an equivalent deterministic finite automaton (DFA) using the subset construction. However, since the subset construction may increase the number of automaton states exponentially, even simple problems about deterministic automata can become computationally difficult to solve if nondeterminism is involved.

One of the basic problems in automata theory is the equivalence problem: given two automata A and B , do they define the same language, that is, $L(A) = L(B)$. For DFA, the equivalence problem can be solved in linear time by the algorithm of Hopcroft and Karp (1971). For NFA, however, the minimization algorithm does not solve the equivalence problem, but

computes the stronger notion of bisimilarity between automata. Instead, the textbook algorithm for NFA equivalence checks language inclusion in both directions, which reduces to checking that both $\overline{L(A)} \cap L(B) = \emptyset$ and $L(A) \cap \overline{L(B)} = \emptyset$. The complementation steps are expensive for NFA: using the subset construction to determinize both input automata before complementing them causes, in the worst case, an exponential cost. Indeed, it is unlikely that there is a theoretically better solution, as the equivalence problem for NFA is PSpace-hard, which was shown by Meyer and Stockmeyer (1972).


As the equivalence problem is essential in many applications—from compilers to hardware and software verification—we need algorithms that avoid the worst-case complexity as often as possible. The exponential blow-up can be avoided in many cases by keeping the subset construction implicit. This can be done, for example, by using symbolic techniques such as binary decision diagrams for representing state sets. Filippo Bonchi and Damien Pous show us there is a better way.

The starting point of their solution is a merging of the Hopcroft-Karp DFA equivalence algorithm with subset constructions for the two input automata.

Much of the power and elegance of automata comes from the natural ease with which they accommodate nondeterminism.

This idea does not lead to an efficient algorithm per se, as it can be shown that the entire state spaces of the two subset constructions (or at least their reachable parts) may need to be explored to establish bisimilarity if the two input automata are equivalent. The contribution of Bonchi and Pous is to show that bisimilarity—that is, the existence of a bisimulation relation between states—can be proved without constructing the deterministic automata explicitly. They show that, instead, it suffices to compute a bisimulation up to congruence. It turns out that for computing such a bisimulation up to congruence, often only a small fraction of the subset spaces need to be explored. As you will see, the formalization of their idea is extremely elegant and leads to an algorithm that is efficient in practice.

The authors evaluate their algorithm on benchmarks. They explore how it relates to another class of promising recent approaches, called antichain methods, which also avoid explicit subset constructions for solving related problems, such as language inclusion for NFA over finite and infinite words and the solution of games played on graphs with imperfect information. In addition, they show how their construction can be improved by exploiting polynomial time computable simulation relations on NFA, an idea that was suggested by the antichain approach.

As this work vividly demonstrates, even classical, well-studied problems like NFA equivalence can still offer surprising research opportunities, and new ideas may lead to elegant algorithmic improvements of practical importance. 

Thomas A. Henzinger is president of the IST Austria (Institute of Science and Technology Austria).

Jean-François Raskin is a professor of computer science at the Université Libre de Bruxelles, Belgium.

Copyright held by authors.