Compact representation of Hungarian vocabulary with nondeterministic finite automata

Attila Kertész-Farkas[1], Zoltán Fülöp[2], András Kocsor[3]

[1,3] University of Szeged, MTA-SZTE Research Group on Artificial Intelligence,
Aradi vértanúk tere 1, 6720 H-Szeged *
[2]University of Szeged, Department of Computer Science, Árpád tér 2, 6720 H-Szeged
[1]kfa@rgai.inf.u-szeged.hu
{[2]fulop, [3]kocsor}@inf.u-szeged.hu

Keywords. finite automata, minimization, finite automata reduction

In linguistic applications that process finite languages, e.g., speech recognition and speech synthesis, it is worth to use finite automata due to their simple structure and efficiency. Several automata can be created that accept a given language, however it is important to find the possible smallest of them.

In this paper we deal with automata reduction algorithms. Broadly speaking, an automata reduction algorithm is an algorithm that, for a given nondeterministic automaton A accepting a finite language, delivers a smaller (nondeterministic) one that is equivalent with A.

For a deterministin automaton A, we can easily find the unique deterministic automaton A' which is minimal and equivalent with A. However, the number of the states of a small nondeterministic automaton equivalent with A may be exponentially smaller than that of A'. Therefore it is worth looking for reduction algorithms also for nondeterministic automata, even if computing the minimal nondeterministic automaton is known to be an NP-hard problem.

Another point is that the size of an automaton depends not only on the number of its states but also on the number of its transitions. Therefore, for the optimal storage, the automata reduction algorithms must also take care of the number of transitions. We note that, to our best knowledge, the conventional automata reduction algorithms do not consider the number of transitions, and thus in some cases they can even increase the size of the automaton.

In this paper we present a novel heuristic automata reduction algorithm. The algorithm works for arbitrary nondeterministic automata accepting a finite language and it considers both the number of transitions and states during the reduction. In the experiments we found a gain of 20-25% in the numbers of states and transitions.