A Merging States Algorithm for Inference of *RFSAs*[⋆]

Gloria Alvarez[1], Pedro García[2], and José Ruiz[2]

[1] Pontificia Universidad Javeriana - Seccional Cali. Colombia
[2] Universidad Politécnica de Valencia. Spain - DSIC
{galvarez@dsic, pgarcia, jruiz}.upv.es

Abstract. The aim of this paper is to present *MRIA*, a new merging states algorithm for inference of Residual Finite State Automata.

1 Introduction

We propose a non deterministic extension of *RPNI* algorithm, that will be called *MRIA* (**M**erging states **R**esidual finite state automata **I**nference **A**lgorithm). It extends *RPNI* finding inclusion relations among the residuals of the positive sample and merging states when it is possible or just keeping the transitions obtained from the inclusions.

MRIA outputs a *NFA* and converges to a *Residual Finite State Automaton* (*RFSA*) that recognizes the target language and which has the same number of states as the minimal *DFA* for that language. A postprocess can be done to the output automaton in order to obtain a *RFSA* of size between the canonical *RFSA* and the minimal *DFA* for the target language.

Definitions concerning Formal Languages can be found in [2]. Previous work related to *RFSAs* can be found in [1].

2 The *MRIA* Algorithm

Algorithm *MRIA* is a merging states algorithm that on input of a complete sample, outputs a non-deterministic finite state automaton which is consistent with the input.

The algorithm begins building the prefix tree acceptor of the positive sample, whose states are lexicographically ordered ($\ll$). It consists of two loops, the outer one takes care of the first state (q) that is being compared whereas the inner analyzes the states p such that $p \ll q$ in turn.

First it checks if p is *smaller* than q (that is, if the automaton that results of adding to the current one every transition of the form (q', a, q) for every existing transition of the form (q', a, p) does not recognize a negative sample. If q is an initial state, p is also added to I). If the answer is positive it keeps

⋆ Work partially supported by Spanish CICYT under TIC2003-09319-C03-02 and Spanish GVA under GV06/068.

Y. Sakakibara et al. (Eds.): ICGI 2006, LNAI 4201, pp. 340–341, 2006.

the transitions resulting from this fact and if q has not been previously merged to a state previous to p, the comparison is checked in the other way. If both comparisons result positive, the states are merged. Two facts make this merging of states differ from the classical merging done by $RPNI$:

– The merges are not propagated to keep the automaton deterministic.
– A copy of state q is kept until the relations between q and all the previous states have been determined.

At the end of every inner loop, if the state q has not been merged with any previous one, a new hypothesis is emitted. This hypothesis is an automaton whose states are those previous to q and the transitions related to them.

Note that a state is only deleted when it has been compared with the rest of the previous ones. Otherwise, some transitions could not be obtained. That is the use of the boolean variable in the algorithm.

The complexity of the proposed algorithm is $O(n^3)$, being n the sum of the lengths of the input sample.

3 Example

Let $D_+ = \{\varepsilon, 00, 10, 11, 010\}$ and $D_- = \{0, 1, 01, 001\}$. The sequence of hypothesis emitted by *MRIA* and the output of *DeLeTe2* and *RPNI* under the same input are depicted in Fig. 1.

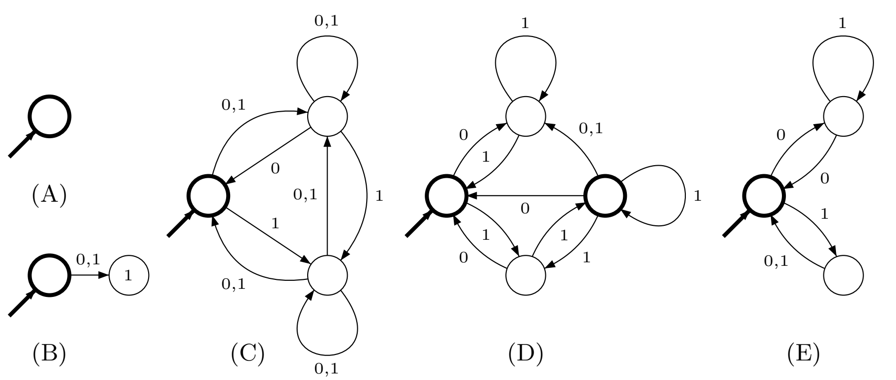

Fig. 1. A,B and C: Sequence of hypothesis emitted by *MRIA* algorithm. D and E hypothesis emitted by *DeLeTe2* and *RPNI* with the same input.

References

1. Denis, F. Lemay, A. and Terlutte, A. *Learning regular languages using RFSAs.* Theoretical Computer Science 313(2), pp 267-294 (2004).
2. Hopcroft, J. and Ullman, J. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley (1979).