INSERTION AND DELETION OPERATIONS

Ing. Martin VÍTEK, Doctoral Degree Programme (1)
Dept. of Information Systems, FIT, BUT
E–mail: vitek@fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

Stringology represents a modern part of the formal language theory, which deals with strings, languages and operations on them. It introduces many new language operations, which can be divided into two groups — insertion and deletion operations. Some of these operations are described in [1]. This paper presents these operations and some their properties. Especially, closure properties are studied here. New algorithms that construct finite automata accepting languages resulting from some of these operations are described here. We actually demonstrate by designing these algorithms, that the family of regular languages is closed under these operations.

1 INTRODUCTION

The language operations that change strings by shuffling or inserting some substrings fulfill an important role in several modern computer science fields, ranging from cryptography through various text algorithms and stringology to DNA computation. Therefore, it comes as no surprise that the formal language theory has recently played a special attention to their investigation (see [1]). The present paper introduces and discusses some more operations of this kind. Specifically, it discusses operations sequential insertion, parallel insertion, scattered sequential insertion, sequential deletion, parallel deletion and scattered sequential deletion.

2 NEW LANGUAGE OPERATIONS

2.1 SEQUENTIAL INSERTION

The result of sequential insertion of string v into string u is a set of strings u, which have in any place inserted the string v. This operation can be generalized to sequential insertion on languages. We obtain the result of sequential insertion of language L_2 into language L_1 by sequentially inserting every string from L_2 into every string in L_1.

EXAMPLE:
$u = cd$, $v = a$
$u \leftarrow v = \{acd, cad, cda\}$

2.2 PARALLEL INSERTION

The parallel insertion of a language L_2 into a string u is a set of strings obtained after inserting strings from L_2 between all the letters of u, before the first letter and after the last letter of u. Parallel insertion of language L_2 into language L_1 is the union of sets obtained after parallel inserting L_2 into all strings from L_1.

EXAMPLE:
$L_1 = \{cd\}$, $L_2 = \{a, b\}$
$L_1 \Leftarrow L_2 = \{acada, acadb, acbda, acbdb, bcada, bcadb, bcbda, bcbdb\}$

2.3 SCATTERED SEQUENTIAL INSERTION

Both previous operations had common property, that the inserted string was inserted in the compact way on one place. But we can also insert the string scattered, so not the whole string but its separate symbols are sparsely inserted. The result of scattered sequential insertion of string v into string u is string u having inserted all symbols of v on arbitrary places respecting their order in v. Scattered sequential insertion of language L_2 into language L_1 is the union of scattered sequential insertion of all strings from L_2 into all strings from L_1.

EXAMPLE:
$L_1 = \{abb\}$, $L_2 = \{cd\}$
$L_1 \leftarrow_s L_2 = \{cdabb, cadbb, cabdb, cabbd, acdbb, acbdb, acbbd, abcdb, abcbd, abbcd\}$

2.4 SEQUENTIAL DELETION

The result of sequential deletion of string v from string u is a set of strings v, from which we have extracted an arbitrary occurrence of the string u. Sequential deletion of language L_2 from language L_1 is the union of sequential deletions of strings from language L_2 from strings from language L_1.

EXAMPLE:
$L_1 = \{abababa, ab, ba^2, aba\}$, $L_2 = \{aba\}$
$L_1 \rightarrow L_2 = \{baba, abba, abab, \varepsilon\}$
We obtain this result as union of the following sets:

$$abababa \rightarrow aba = \{baba, abba, abab\}$$
$$ab \rightarrow aba = \emptyset$$
$$ba^2 \rightarrow aba = \emptyset$$
$$aba \rightarrow aba = \{\varepsilon\}$$

2.5 PARALLEL DELETION

Parallel deletion of language L_2 from string u erases all the non–overlapping occurrences of strings in L_2 from u. No nonempty string from L_2 can appear between any two occurrences of strings from L_2 to be erased. The result can still contain a string from L_2 as the result of catenation of the remaining pieces. Parallel deletion of language L_2 from language L_1 is obtained by parallel deletion of L_2 from all strings in L_1.

EXAMPLE:

$L_1 = \{ababab a, aababa, abaabaaba\}$, $L_2 = \{aba\}$
$L_1 \Rightarrow L_2 = \{b, abba, aba, aab, \varepsilon\}$
We obtain this result as the union of the following sets:

$$abababa \Rightarrow \{aba\} = \{b, abba\}$$
$$aababa \Rightarrow \{aba\} = \{aba, aab\}$$
$$abaabaaba \Rightarrow \{aba\} = \{\varepsilon\}$$

2.6 SCATTERED SEQUENTIAL DELETION

Similarly as scattered sequential insertion we can define sequential deletion in a scattered sense. We do not delete the whole substring v but all its individual symbols in their order in v. Generalized to languages, the result is the union of scattered sequential deletion of all strings from one language from strings of the second language.

EXAMPLE:

$L_1 = \{a^n b^n c^n \mid n \geq 1\}$, $L_2 = \{ab^2c^3\}$
$L_1 \rightarrow_s L_2 = \{a^{n+2}b^{n+1}c^n \mid n \geq 0\}$

3 CLOSURE PROPERTIES, FINITE AUTOMATA

This paper studies also closure properties of these new operations. It has proved the closure of the class of regular languages under these operations. The proof of the closure is built on newly proposed algorithms of construction of finite automata for these operations. These algorithms receive two deterministic finite automata M_1 and M_2 without ε–edges, which accept languages L_1 and L_2, respectively. A new finite automaton M for selected operation is then constructed from them. Automaton M accepts language obtained by insertion or deletion of L_2 into/from L_1. Algorithms for all six basic operations introduced in the previous chapter have been found.

Automata constructed by using these newly designed algorithms are nondeterministic, have many ε–edges, inaccessible and indistinguishable states. Finally they are processed with algorithms of elimination of ε–edges, elimination of nondeterminism and minimalization.

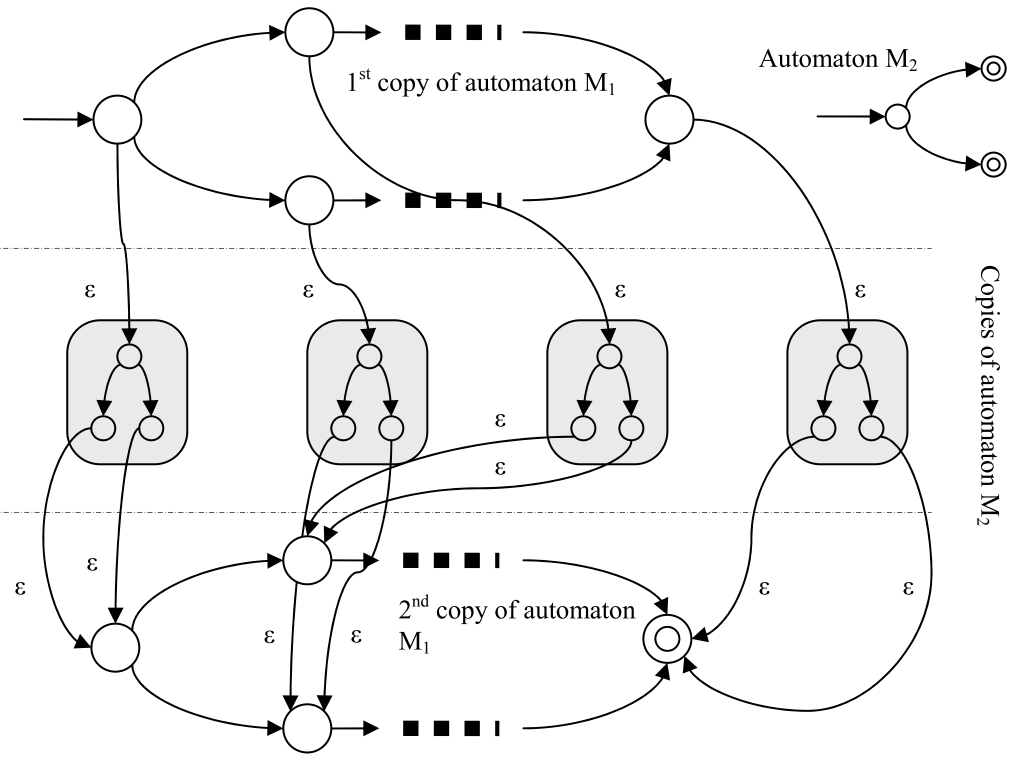

Fig. 1: *Automaton for sequential insertion*

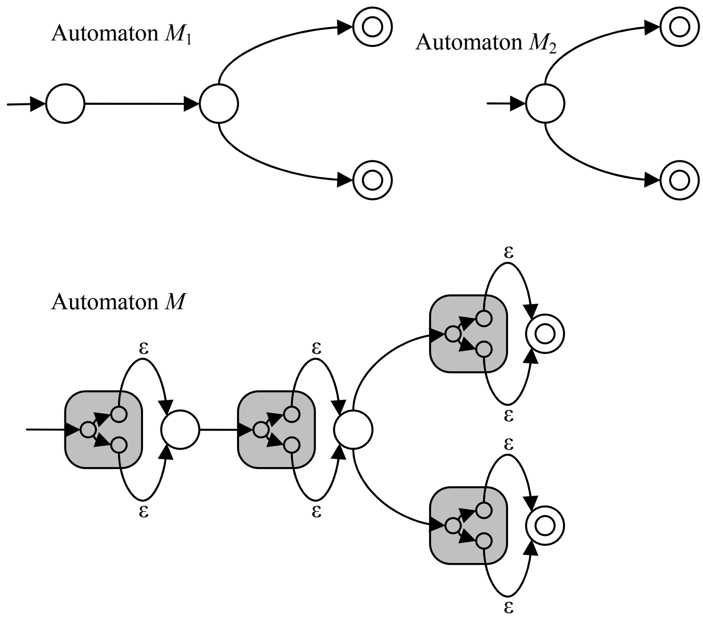

Fig. 2: *Automaton for parallel insertion*

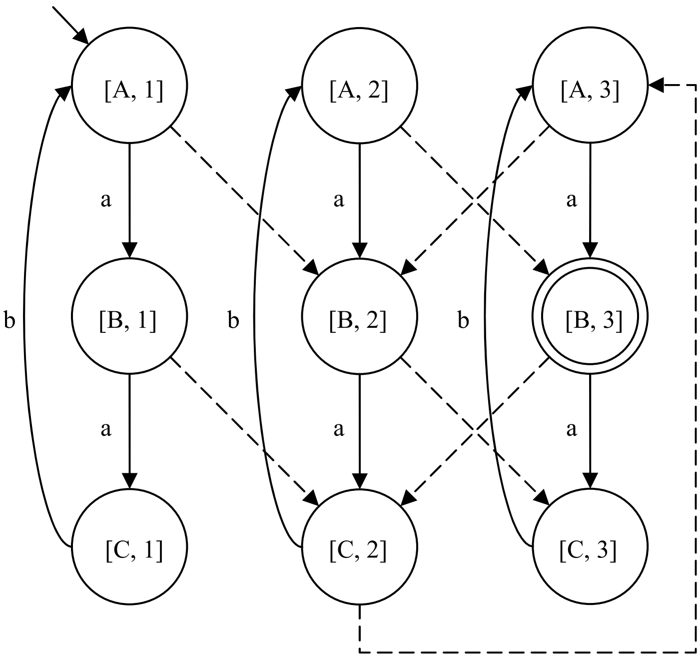

Fig. 3: *Automaton for scattered sequential deletion*

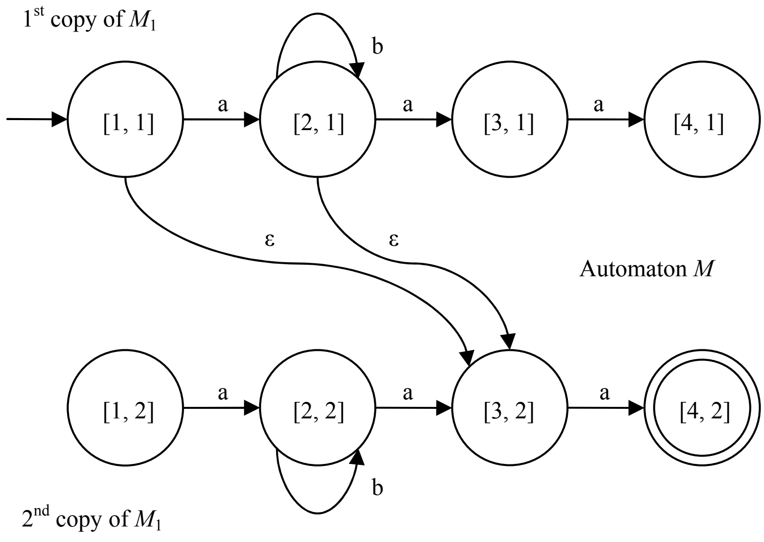

Fig. 4: *Automaton for sequential deletion*

REFERENCES

[1] Kari, L.: On insertion and deletion in formal languages, Turku, Finland, 1991

[2] Kari, L.: Power of Controlled Insertion and Deletion, LNCS 812, Springer, Berlin, 1994

[3] Vítek, M.: Nové operace v teorii formálních jazyků a jejich využití, FIT VUT Brno 2003