NEW OPERATIONS IN FORMAL LANGUAGE THEORY AND THEIR USE

Martin VÍTEK, Master Degree Programme (5)
Dept. of Information Systems, FIT, BUT
E-mail: xvitek10@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This paper introduces some new language operations discussed in the formal language theory at present. Most importantly, sequential and parallel versions of deletion and insertion are discussed. Algorithms that construct finite automata for these operations are given.

1 INTRODUCTION

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

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 language into every string in language 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 language L_2 between all the letters of string u, before the first letter and after the last letter of string 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 of language 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 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:

$abababa \rightarrow aba = \{baba, abba, abab\}$

$ab \rightarrow aba = \emptyset$

$ba^2 \rightarrow aba = \emptyset$

$aba \rightarrow aba = \{\varepsilon\}$

2.4 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:

$L_1 = \{abababa, aababa, abaabaaba\}$, $L_2 = \{aba\}$

$L_1 \Rightarrow L_2 = \{b, abba, aba, aab, \varepsilon\}$

We obtain this result as the union of the following sets:

$abababa \Rightarrow \{aba\} = \{b, abba\}$

$aababa \Rightarrow \{aba\} = \{aba, aab\}$

$abaabaaba \Rightarrow \{aba\} = \{\varepsilon\}$

3 CLOSURE PROPERTIES, EXHIBITIONS OF 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 automata 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 four 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. All necessary algorithms are in [2].

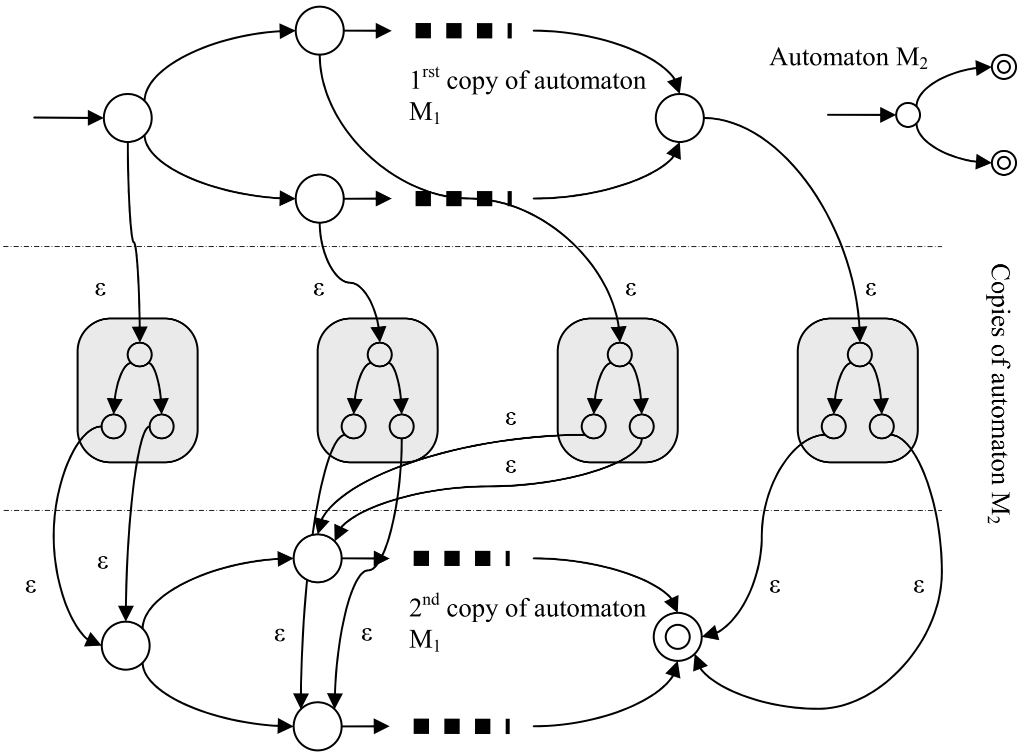

Fig. 1: *Automaton for sequential insertion*

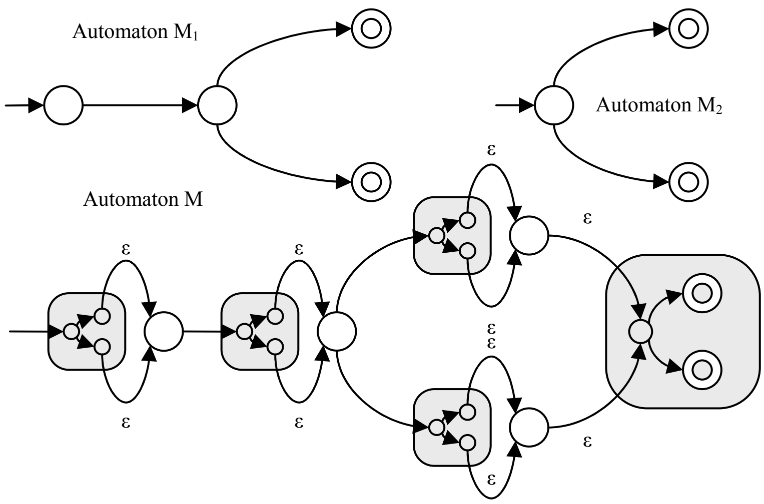

Fig. 2: *Automaton for parallel insertion*

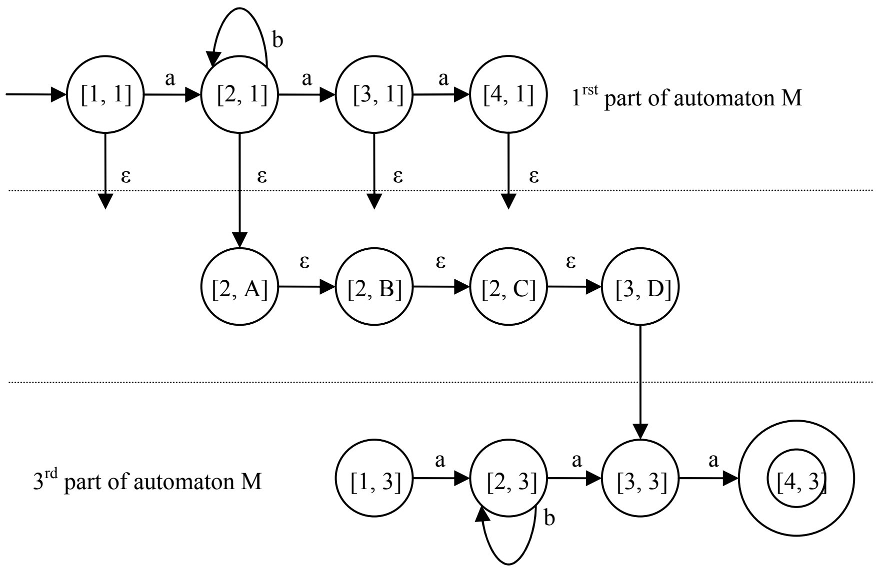

Fig. 3: *Automaton for sequential deletion*

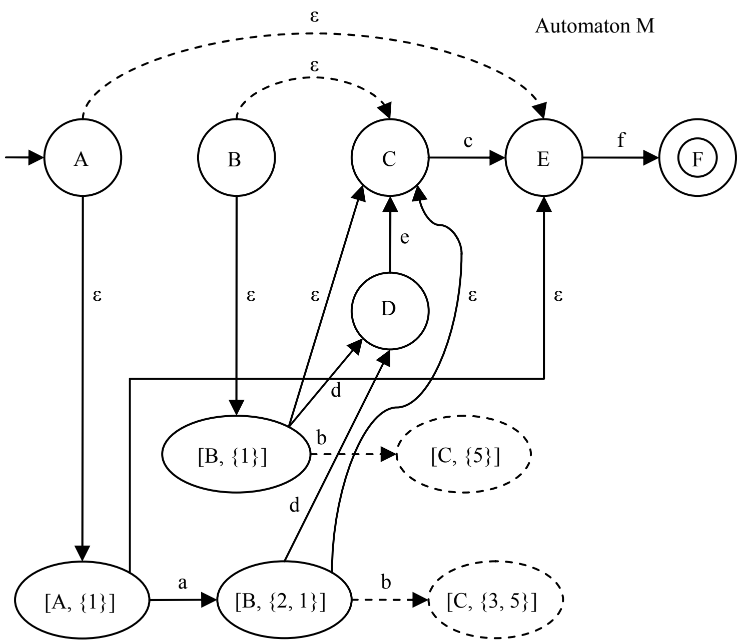

Fig. 4: *Automaton for parallel deletion*

The automaton for SI has 2 copies of M_1. It starts accepting in the first copy. It cannot stop; the first copy doesn't contain any finite state. It must go to the second copy through one copy of M_2 and accept a string from L_2.

The automaton for PI goes after accepting every symbol from string from L_1 and before accepting the first symbol through copy of M_2, where it accepts the whole string from L_2.

In the automaton for SD we move forward in the second part of the automaton using ε–edges and we skip accepting the deleted substring. We must go from the 1rst part to the 3rd part, because the 1rst part doesn't contain any finite state.

The automaton for PD is very complicated and cannot be so shortly described. It has complicated identifiers of its states. The states are identified with a couple, whose first part is identifier of state of M_1, and the second part is a set of identifiers of states of M_2. It handles all possibilities of simultaneous deletion of all the non–overlapping occurrences of strings from L_2.

4 THE USE OF THE NEW OPERATIONS

4.1 THE USE IN COMPUTER SCIENCE

These operations have a large application in computer science, including parallel word processors, parallel compilers, tools for processing of native language, in description and processing of formal languages. If we have programming language, which supports embedded commands of another language, like SQL commands in a C++ program or PHP tags in a HTML file, we can describe it using insertion operations. Another example is in algorithms for searching the substring in a string, substring in a file, and algorithms for searching a file, which contains a word from the list or which does not contain any word from the list. Further applications can these operations find in searching the files attacked by computer viruses and cure of attacked files in this way. Separate area of application of these operations is computer graphics. Some fractals in computer graphics are described by using formal languages and grammars. These operations can bring new interesting figures.

4.2 THE USE IN MEDICINE AND BIOLOGY

Some of these new operations can find their applications in other branches, such as biology. The order of genetic bases in DNA acid represents the genetic information of every live organism. This information can be described using formal resources and these operations can describe for example genetic mutations or genetic information of descendant if this information of his parents is known. So the application is offered in cultivation of animals and plants or in foresight for genetic illnesses of descendants. Another possibility is in transplantations. In future, it will be possible to find the structure of DNA of donor and receiver of organs and formally test, whether the organism of receiver won't refuse the organ. Biologists can describe how viruses change genetic information of cells so that the cell makes no other cells but new copies of virus.

4.3 SOME OTHER USE

The new operations can be used in chemistry in description of large molecules of polymers. They can help in nanotechnology. Using these operations, architects can define layout of trees, paths and benches in parks. In textile industry, they can be used in description of regular patterns.

REFERENCES

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

[2] Češka, M.: Písemné podklady ke kurzu Teoretická informatika 1, FIT VUT Brno

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