Implementation of key-value databases by cellular automata with locators [1]

E. E. Gasanov[2], A. A. Propazhin[3]

In this paper, it is shown that key-value databases can be implemented by cellular automata with locators in such a way that the execution time of basic operations, such as search, insert, delete, will not depend on the size of the database and will be equal to the total length of the key and value.

Keywords: Cellular automata with locators, key-value databases.

The key-value database is a popular data storage paradigm now, also called a dictionary. Such a database can be represented as a set of pairs of strings (k, v), where the first string k is called the key and serves as the identifier of the pair, and the second string v is called the value. *String* is a sequence of characters of some alphabet A, ending with a special character 0, called *the string ending character*, and the character 0 does not belong to the alphabet A.

The key-value database supports the following operations:

1) *inserting a pair* (k, v) — an entry with the key k and the value v appears in the database; if an entry with the key k already existed in the database, then the value is replaced with v;

2) *deleting an entry with a key* k — an entry (k, v) is deleted from the database; if there is no entry with the key k in the database, then the database does not change;

3) *searching for an element by key* k — there is an entry (k, v) in the database, and the value v is returned as an answer; if there is no entry with the key k in the database, then the answer is an empty set.

The concept of a cellular automaton with locators was introduced by E. E. Gasanov in [1] and refined by G. V. Kalachev in [2]. The exact definition can be found in the above-mentioned works, but here we give an informal definition of a one-dimensional cellular automaton with one complete locator, which will be used in this work.

A one-dimensional cellular automaton is a set of identical elementary automata located at integer nodes of the real line, and called cells. The

[1] Originally published in *Intellektualnye Sistemy. Teoriya i prilogeniya* (2021) **25**, No. 4, 108-112 (in Russian).

[2] *Gasanov Elyar Eldarovich* — professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems, e-mail: el_gasanov@gmail.com.

[3] *Propazhin Artem Alekseevich* — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems, e-mail: artem.propazhin@mail.ru.

behavior of a cellular automaton is specified by the transition function, namely, the state of the cell at the next moment is uniquely determined by its own state at the current moment and the states of its neighbors. In what follows, we will assume that each cell has exactly two neighbors: the closest to the left and the closest to the right. One of the states of a cell is called an quiescent state, and if the cell and its neighbors are in an quiescent state, then the next moment the cell will remain in an quiescent state. In a cellular automaton with one complete locator, in addition to the alphabet of states, there is a broadcasting alphabet with a commutative semigroup operation specified on it. In this paper, the maximum operation will be used as such an operation. In a cellular automaton with one complete locator, each cell sends a certain signal from the broadcasting alphabet, determined by the broadcasting function, to the air every moment. The value of the cell broadcasting function is determined by the cell's own state, the states of its neighbors and the total broadcasting signal. The total broadcasting signal is formed by summing up the broadcasting signals of all cells with the exception of the signal of this cell using a semigroup operation. In a cellular automaton with one complete locator, the value of the cell transition function is also determined by the cell's own state, the states of its neighbors and the total broadcasting signal.

Let's introduce another entity — *the database user*. We will assume that the database user has the ability to send signals from the broadcasting alphabet to the air, i.e. his broadcasting signal is summarized with broadcasting signals of all automata. And database user can receive total broadcasting signal from the air. We will assume that a cellular automaton with locators, together with the user, implements a key-value database if the broadcasting alphabet is a set consisting of pairs of the form ("command", $A \cup \{0\}$), where "command" takes one of the values: "search", "insert", "delete", "answer", "no answer", "no command" (here "no command" is the minimum element), and the behavior of the cellular automaton is set as follows.

1) The user broadcasts the "search" command and the first character of the key k. Then the user sequentially broadcasts the "no command" command and all the other characters of the key, including the character 0. If there is no entry with the key k, in the database, then on the next clock cycle after the 0 character is given, the cellular automaton broadcasts the command "no answer". If there is an entry (k, v) in the database, then on the next clock cycle after the character 0 is submitted, the cellular automaton broadcasts a pair ("answer", a), where $a \in A$ is the first element of the value v, and in subsequent cycles the cellular automaton sequentially broadcasts all other symbols of the value v, including the character 0.

2) The user broadcasts the "insert" command and the first character of the key k. Then the user sequentially broadcasts the "no command" command

and all the other characters of the key, including the character 0. After that the user sequentially broadcasts the "no command" command and all characters of the value v, including the character 0. As a result, a pair (k, v) appears in the database implemented by the cellular automaton, i.e. if the "search" command and the key k are subsequently submitted to the cellular automaton, then the cellular automaton will return the value of v in response.

3) The user submits the "delete" command and the first character of the key k. Then the user sequentially broadcasts the "no command" command and all the other characters of the key, including the character 0. As a result, the entry with the key k disappears from the database implemented by the cellular automaton, i.e., during the subsequent search for the key k, the cellular automaton returns "no answer".

Theorem 1. *There is a cellular automaton with locators and a user which implements a key-value database, and for which the execution time of the search, insert, delete operations will not exceed the total length of the key and value.*

Here is the idea of proving this theorem.

We will use a one-dimensional cellular automaton with one complete locator, and elementary automata lying in the negative region of the numerical line will not be used. The alphabet of broadcasting has already been described above. The alphabet of states consists of triples of the form ("command", "cell type", $A \cup \{0, *\}$), where "command" takes one of the values: "search", "insert", "delete", "answer"; "cell type" takes one of the values: "commander", "current cell type key", "current cell type value", "unprocessed cell", "receiver for recording"; $*$ is a special character. If an elementary automaton will be marked with the symbol "receiver for recording" then starting from this automaton the next record will be written to the database. At the initial moment, the automaton with the number 0 will be marked with symbol "receiver for recording". Database records will be pairs of key-value strings, and the first character of each record will be marked with the state "commander".

Let's describe the functioning of this automaton. When the insertion begins, all commanders receive a signal from the air in the form ("insert", k_1), where k_1 is the first character of the key. If k_1 matches the value stored in the state of the commander cell, then in the state of the cell next to the right, the command changes to "insert" and the cell type to "current cell type key". On the next clock cycle, the cell next to the right of the commander, also receiving a signal from the air, checks for a match with the stored symbol. After that the current cell changes its type to "unprocessed cell". When a match occurs, in the state of the cell next to the right of current cell, the command changes to "insert" and the cell type to "current

cell type key“ and a similar process occurs to the previous one. The process occurs sequentially up to and including the 0 symbol. The key verification process starts simultaneously with all commanders. If at some point there is a mismatch, then the type of the next cell changes to “unprocessed cell“. If we have reached the 0 symbol, it means that we have found a record with the desired key, and this record should be excluded from the database. Exclusions from the database are achieved by the fact that 0 in the state of the current cell changes to the symbol $*$. As a result, there will be no match in subsequent searches.

Simultaneously with the search for the key, starting from the cell marked with the state “receiver for recording“ the key-value pair is sequentially recorded into the database. When the insert command arrives, the cell in the “receiver for recording“ state becomes the commander and stores the first character of the key k_1 in its state. At the same time, the cell to the right of this cell changes the command to “insert“ and the cell type to “receiver for recording“. Further, the cells that have the “insert“ command and the “receiver for recording“ cell type retain the key symbols and values coming from the air in their state. In this case, the state (“insert“, “receiver for recording“) moves to the next cell on the right. The exception is when the 0 character arrives for the value. At this moment, the next cell on the right switches to the state (“search“, “receiver for recording“).

Deletion is similar to insertion. Only during deletion does the process of writing to the end of the database not begin.

During the search, the key is read in the same way as when inserting. But after reaching the 0 symbol, the replacement of the symbol 0 with $*$ does not occur. The cell next to the right of the cell with the 0 symbol changes the command to “search“ and the cell type to “current cell type value“. Cells in this state send a signal to the air (“answer“, a), where $a \in A$ is the stored value. The current cell changes its type to “unprocessed cell“ after sending the signal. Signals with value symbols are sequentially sent to the air up to and including the signal with the symbol 0.

Список литературы

[1] Gasanov E. E., “Cellular automata with locators”, *Intelligent Systems. Theory and applications*, **24**:2 (2020), 121–133 (In Russian).

[2] Kalachev G. V., “Remarks on the definition of a cellular automaton with locators”, *Intelligent Systems. Theory and applications*, **24**:4 (2020), 47–56 (In Russian).