UDC 004.934

MARKOVSKYI O.,
SHEVCHENKO O.,
HUMENIUK I.

HASH SEARCH ORGANIZATION IN E-DICTIONARIES USING BLOCK CIPHERS

The article is devoted to the problem of developing high-speed electronic dictionaries for systems computer translation.

The method of organizing a high-speed electronic dictionary based on an ideal hash addressing, where the cryptographic cipher block acts as a hash transformation is proposed. This method was developed taking into account the multilevel memory structure of modern computer systems.

It was testified theoretically and experimentally that the proposed organization of electronic dictionaries guarantees at least twice higher search rate compared to known technologies.

Keywords: hash-search, e-dictionary, contextual search, perfect hash-addressing, computer translation.

1. Introduction

Over the past two decades, the significant changes in research and production work centers in the world has bees occurred. Until recently, there was the tendency of transfers of production strenth to the countries of the Far East. But nowadays these countries are achieving main positions in various fields of science and, above all, in technological researches.

This rules to a further increase of exchange scientific and technical information of East and West. But, there is one significant obstacle to the flow of further increasing information sharing. This obstacle is the language barrier between East and West. Mainly, it is caused by the linguistic differences between East and West languages [1].

Experience has shown that the most promising way to overcome the language barrier in the question of scientific and technical information exchange is the usage of highly-efficient computer translation technologies.

Progress in the field of computer technologies made the background for the successful realization of this approach [2]. The recent progress in artificial intelligence make possible to reach the semantic relevancy of translation, made by computer translation systems. These technologies are based on the analysis of a vast amount of translation options, which make necessary the widespread usage of electronic dictionaries. In such order, as more efficient electronic dictionaries are, as more efficient work of machine translation tools provides. Thus, is necessary to develop some new methods of contextual search in electronic dictionaries.

Therefore, the scientific task of increasing the speed of searching for words in electronic dictionaries is relevant at the present stage of developing information technologies.

2. Problem statement

To reach this aim, an analysis of electronic dictionaries` models has proceeded. The model of context-search dictionary requires the storage of sets of contextual language constructs to each search key-word of the dictionary. In this order, there are two types of information that could be stored in electronic dictionary: keywords used for the search, and the associated with them data. The size stored keywords is totally much smaller than the size of the associated data.

The performance of electronic dictionaries is highly dependent on a compromise between search speed rate and the size of the number of search key-words is an electronic dictionary [3].

Analysis of modern translation systems shows that the semantic adequacy of translation hugely depends on the amount of context information [4]. On the other hand, modern translation systems require

fast search speed electronic dictionaries. As a result, the improvement of computer translation requires finding the compromise between search speed and the size of the dictionaries.

3. The purpose and objectives of the study

The purpose of the study is to enhance the word search rate in e-dictionaries, as parts of computer-assisted translation systems, by reducing the number of swapping cycles proceeded while accessing to the dictionary stored data.

Therefore, the main objective of the study is finding the way to reduce the number of swaps while accessing e-dictionary stored data.

4. Organization of electronic dictionary based on perfect hash-addressing

Potentially, hash addressing is the fastest key-search technology. Hash addressing provide access to any keyword data in a fixed time. This time is approximately equal to time of one hash transformation.

When hashing the position of the element corresponding to the keyword k is calculated as h(k), where h is some hash-function.

The number h(k) is the hash value of the keyword k. The hash values of two different keys can match. This means that a collision has occurred [5].

Collisions are the most significant disadvantage against using hashing in electronic dictionaries organization. This situation is due to the fact that conflict resolution requires a large number of resources and is time consuming, which is unacceptable in the context of using hashing as a technological solution for high-speed dictionaries. It is well known fact that the probability of collisions and the effectiveness of their resolution largely depend on the coefficient φ of memory filling [6].

As already mentioned, the information corresponding to the keyword k is placed at the address, which is calculated as a hash of the keyword h(k). Thus, the coefficient under the coefficient φ means the ratio of the cells of the address hash keywords to the total number of memory cells allocated for the organization of the dictionary.

Another feature of hashing is that the result of a hash function for two close keys will be different. In the context of using hashing in dictionaries, this means that came-root words will be addressed to different parts of memory [7]. This can reduce the efficiency of using hashes in dictionaries.

This problem can be solved by preliminary analysis of the searched word or language construction. This analysis can occur with the involvement of means of stemming, morphosyntactic analysis, root extraction [8]. Thus, the search keys will be generalized matches of the initial input data.

The analysis of the considered features of work of electronic dictionaries, use of hashing and work of systems of computer translation has led to the decision in which it is offered to divide process of search of relevant translation of a keyword into two stages. The first stage is to actually search for contextual keyword data. The second stage is the selection of the most successful translation performed by the computer translation system.

To solve the problem of collisions, it is proposed to use perfect hash addressing. as perfect hash transformation, it is proposed to use a symmetric encryption algorithms. Thus, the searched keyword is fed to the input of the cipher block, and the corresponding ciphertext acts as a hash address.

This solution is due to the possibility of using hardware implementation of symmetric encryption algorithms for faster execution of hash transforms that will speed up the work of the dictionary in general. Using this advantage is easy on a practical level because almost all modern computer systems have built-in cryptoprocessors that implement standardized symmetric cipher.

In practice, the main limitation is the time of selection of such hash transformation, which for a given array of input keywords will be perfect hash transformation. The estimate of the time T required to find the ideal hash transformation can be expressed by the following formula (1):

$$T = t \cdot \varpi \cdot M, \quad (1)$$

where t — is the time of performing h(k), ϖ - mathematical expectation of number of samples, M - number of samples.

It is known that mathematical expectation ϖ depends on the probability that collisions will not occur before current key selection stage and the probability that collision will occur at the current key selection stage [9].

The results of the estimation of coefficient φ that could be achived by selecting the perfect hash-transformation for T (hours) are shown in the Table 1. These results were got by using formula 4 and they are based on the fact that DES algorithm was used as hash transformation and it was performed on the Cortex-M4. According to Cortex-M4 documentation [10], time t = 7.6·10-8 sec. The T values are given for estimation and can be reduced by γ times by using γ processors.

Table 1
The dependence of coefficient φ on time T and number α of keywords

T, hours	Coefficient φ for α of keywords			
	1000	10000	50000	100000
10	0.2210	0.0594	0.0271	0.0184
50	0.2308	0.0615	0.0265	0.0189
100	0.2361	0.0642	0.0281	0.0196
500	0.2462	0.0675	0.0289	0.0210

Based on results presented in Table 1, it can be concluded that the memory efficiency is small if perfect hash addressing is used in real-size dictionaries. To overcome this drawback, it is proposed to place contextual information related to keywords in the hash memory.

It is proposed to fill the hash memory in two stages. Firstly, memory is allocated for links, at addresses formed by perfect hash-transformation of keywords. After that, the context information is placed in the gaps between the primary links. Thus primary links and the context data can be read into the cache memory in minimum number of swapping cycles.

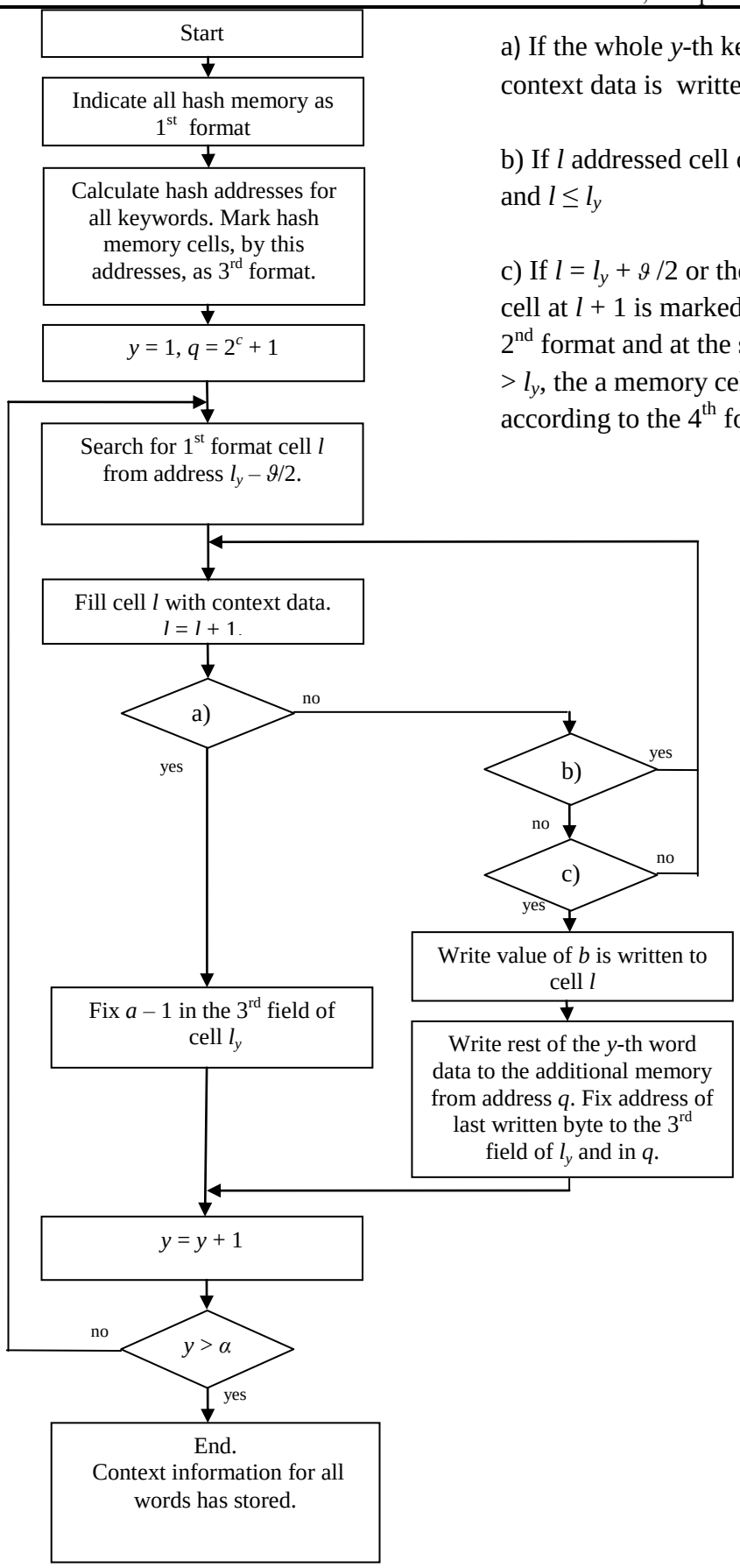
It is proposed to divide memory into two zones: a hash memory and an additional memory. The first zone stores the primary address links and context data that can be read into the cache in a single swap. The additional memory is provided for storing rest of context information.

It is proposed to realize this idea by using four formats of organizing data in the memory cell. The 1st format is proposed to be used to indicate free memory cells. Cells of these formats have the marker S1 in the first byte. The 2nd format has no markers and it is the format of cells that contain some payload data. The 3rd format is used for storage the store primary references. The cells of this format contain three fields. The first field is marker S2 of size 1 byte. The second field holds an address link to the beginning of the contextual information of a certain word; the third field is the address of the last memory cell of the corresponding context information. The 4th format of memory cells has marker S3 and are used for storage an address reference to the continuation of the context data of a keyword in the additional memory.

Every memory cell contains c/4 + 1 bytes, where c is size of address in bits. Markers can be implemented as symbols, which are not used in dictionaries.

The dictionary is filled with keywords and their corresponding contextual data according to the algorithm, the graphical representation of which is shown in Fig.1. Primary links of α keywords are l1, l2,…, lα, ϑ – swapping buffer size.

Search for contextual data of a given keyword is performed according to the algorithm, the graphical representation of which is shown in Fig.2 It should be noted that the algorithm involves recovering contextual data stage of analysis which is performed by the computer translation system.

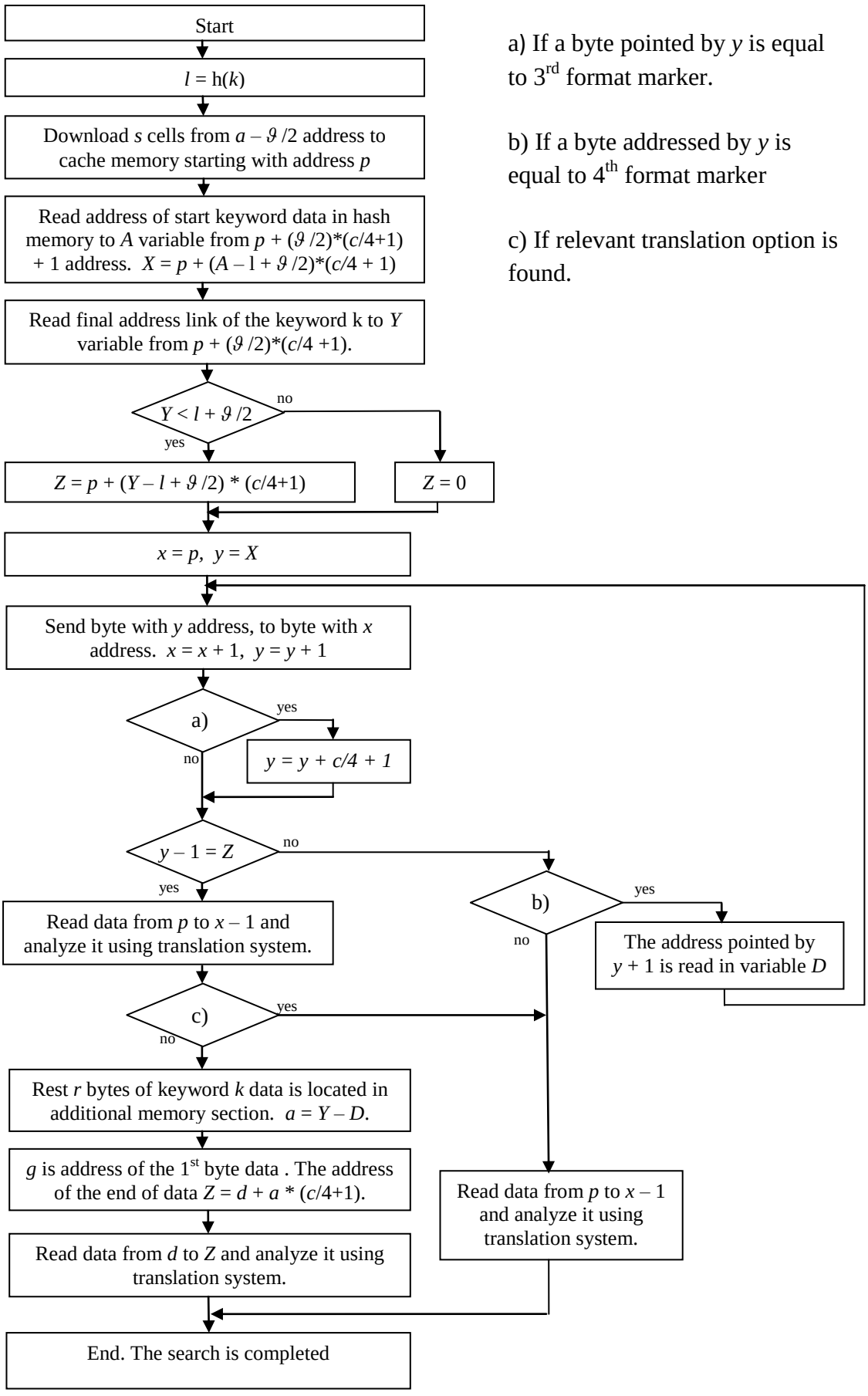


a) If the whole *y*-th key-word context data is written

b) If *l* addressed cell of 2nd format and $l \le l_y$

c) If $l = l_y + \vartheta/2$ or the memory cell at $l + 1$ is marked as cell of 2nd format and at the same time $l > l_y$, the a memory cell is marked according to the 4th format.

Fig.1. Algorithm of organization and allocation data in a dictionary.



a) If a byte pointed by *y* is equal to 3rd format marker.

b) If a byte addressed by *y* is equal to 4th format marker

c) If relevant translation option is found.

Fig.2. Algorithm of the search for keywords context data in a dictionary.

5. Results

The main criteria for estimating the performance of an electronic dictionary, in terms of usage it as an element of computer translation systems, are the speed of context search in the electronic dictionary and the level of memory usage.

In the modern computer systems, the search time Tv is calculated using the following formula (2):

$$T_v = \kappa \cdot \tau_\eta + \tau_n, \quad (2)$$

where κ — is the number of swapping cycles, τη — is the of one performing one swapping cycle , τп — is the time of the context search.

The time τп depends on the complexity of translation algorithm and can be different for different translation systems. Time τη of one swapping cycle is proportional to buffer size ϑ.

Analysis of the modern electronic dictionaries shows that the time τη is much longer than the time τn required to find the most appropriate translation option from stored data. As a result, the speed of the electronic dictionary search is determined by the number of swapping cycles.

On this basis, time τη in can be estimated in terms of the average number η of swap cycles required to access to the keyword context information.

The level of redundancy in memory use can be estimated as follows. If μ is the average amount of contextual information of one key word, then the general amount of linguistic information in electronic dictionary is product μ and α.

All electronic dictionaries contain a some service data. This means that the size of real dictionaries is always greater than the size of linguistic information itself.

The index Ω of the redundancy of memory use in electronic dictionaries can be determined according to the following formula (3):

$$\Omega = \frac{\varsigma}{\mu \cdot \alpha}, \quad (3)$$

where ζ - is size of whole dictionary.

For the experimental study of the efficiency of the proposed electronic dictionary, a statistical modeling software complex has been developed.

In the framework of the simulation, the number α = 10000, the average μ = 400 bytes. To determine these parameters, a statistical study of the translated and interpreted dictionaries of computer terms [4] was carried out.

The first part of experimental study was aimed to determine influence of coefficient φ on main performing criteria. Results of experimental study of dependence between coefficient φ and average amount η of swapping cycles needed to access to all context data of keyword are shown on Fig.3.

The fig.3 clearly proves that the amount η of swapping cycles directly depends on coefficient φ.

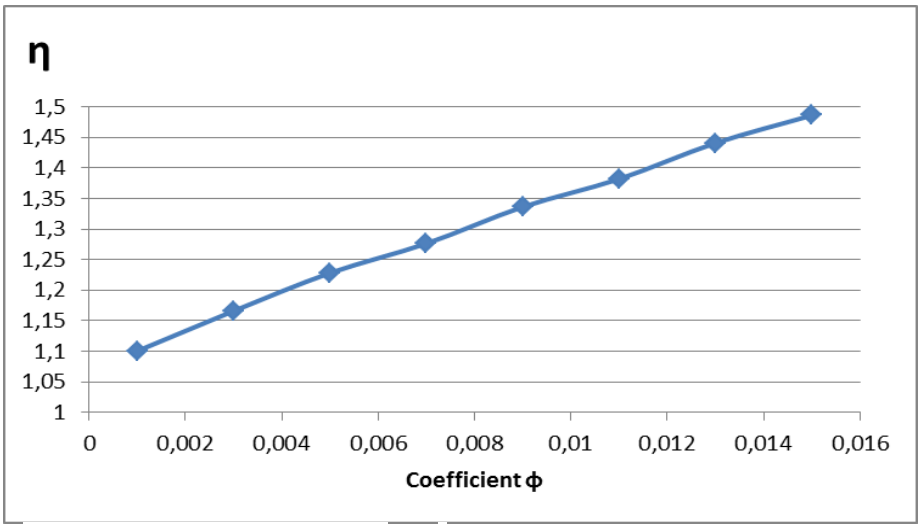


Fig.3. The dependence of amount η of swapping cycles on coefficient φ.

The second part of experimental study was aimed to determine influence of coefficient φ on value of load factor λ . Load factor λ is equal to the ratio of the filled cells of the hash memory to the total hash memory. Results of this study are shown on Fig.4.

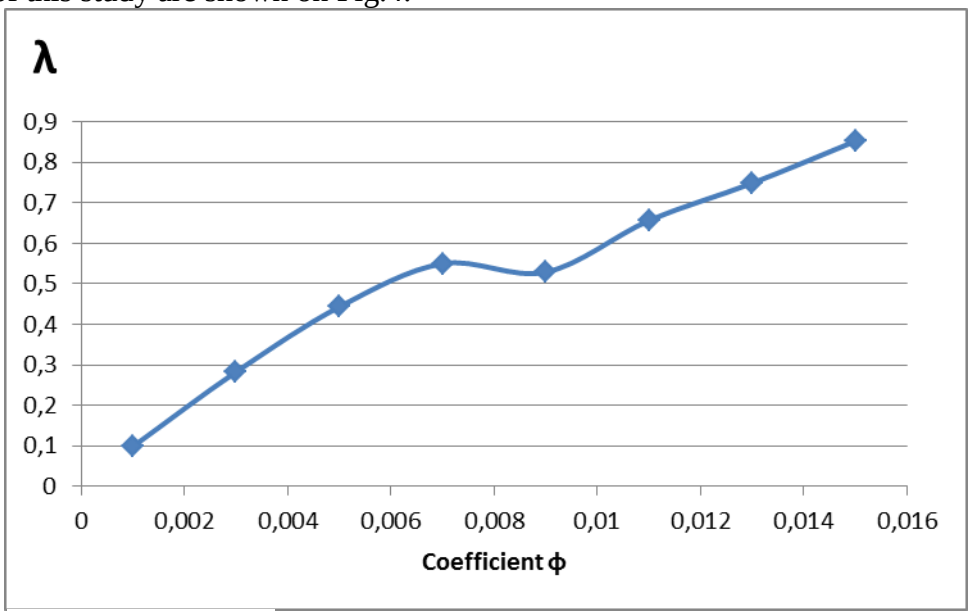


Fig.4. The dependence of the hash memory full load factor λ on coefficient φ.

Fig. 3 and Fig.4 demonstrate that with an increase in the coefficient φ, the redundancy of memory use decreases. But, it increases the number η of swapping cycles during which all the context information of a certain word can be loaded into cache memory.

For instance, if coefficient φ = 0.012, hash memory is totally filled on 65% and the amount of swaps which must be executed to load whole context data is 1.38. The indicator Ω is about 1.6.

The main advantage of the proposed electronic dictionaries is speed rate of access to contextual information of the key-word.

It is advisable to evaluate the work of such an organization of the electronic dictionary by comparing it with known developments according to the selected efficiency criteria.

Comparative performance analysis requires determining the number of η swap cycles for which certain contextual information is accessed.

For tree-based electronic dictionaries, this number depends largely on the type of tree. For balanced binary trees, the average number of memory accesses is $\log_2\alpha$. For other types of trees, this value is greater.

When searching in the contextual information tree for a certain word, a chain of memory accesses is performed. In particular, for an electronic dictionary based on a balanced binary tree with the storage of contextual information in its nodes, the adjacent addresses of the chain of appeals are stored in memory at a distance exceeding the size of swap buffer. Accordingly, the number η of swap cycles is approximately equal to the number of memory accesses and for α = 10000, is about 13,3. Such kind of dictionaries works at least 9.1598 times slower than the proposed dictionary.

For tree-based e-dictionaries with spaced link tree and contextual information storage, the access chain passes without swapping within a subtree that does not exceed the size swap buffer. In particular, for α = 10000 and the payload data is $4 \cdot 10^6$ bytes, each node stores two links and a word router, which occupy a total of 16 bytes. For the accepted size s of the swap buffer, the subtree contains about 28 nodes, every 4 requests swap the next subtree. Thus, access to information requires 4 swaps during the search and one for direct transportation of contextual information. Compared to the proposed dictionary, it works in 3,45 times slower.

For electronic dictionary based on hash search with collisions resolved by probing and spaced storage of addresses and contextual information, finding and loading in the cache memory of contextual information of a certain word is carried out, on average, in 2,8 cycles of swapping. It is in 2.03 times slower comparing with proposed dictionary.

Evaluation of memory efficiency is based on a comparison of redundancy of memory usage.

For dictionaries based on the tree with the preservation of contextual data in its nodes, the service information consists of address links to the descendants of the node. In this example, when using a balanced binary tree, two address links occupy 6 bytes and, accordingly, the average number of bytes of the node is μ + 6. Thus, the indicator Ω is equal to $\frac{\mu + 6}{\mu} = 1{,}015$.

In electronic dictionaries with spaced preservation of the link tree and contextual information, in each of its nodes there are a word-router and two address links. In this case, for the given example, the total volume of service information is $\alpha \cdot 16 = 16 \cdot 10^3$ bytes. Thus, the indicator is equal to $(16 \cdot 10^3 + 4 \cdot 10^6)/ 4 \cdot 10^6 = 1{,}004$.

For electronic dictionaries based on hash search with collisions and spaced storage of addresses and contextual information, hash memory cells containing the keyword and links to its contextual data. In this example, the size of such a cell is 13 bytes, o for φ = 0,7, the total volume of the dictionary is $13 \cdot 10000 / 0{,}7 + 4 \cdot 10^6$. Thus, the indicator Ω of memory usage is 1,047.

Analysis of this data convincingly shows that the proposed organization of the dictionary provides the highest search speed compared to known methods. The resulting effect is achieved due to slightly lower memory efficiency. The proposed approach is quite justified in the current trends of cheaper hardware memory.

6. Conclusions

As a result of research, the new method based on perfect hash-search has been proposed.

In order to increase the efficiency of using hash-addressing, it is proposed to store data for contextual translation between the hash addresses of key-words. These hash-addresses are followed by address links associated with the data, which are needed for translation. This method of storing data is adapted for the multilevel memory architecture of modern computer systems.

The efficiency of the proposed method was studied both theoretically and experimentally. It has been testified that the proposed method doubles the search speed compared to traditional methods of organizing electronic dictionaries. The cost of increasing the speed of searching for words was paid by increasing the amount of used memory.

The developed method of organizing data in electronic dictionaries can be used as a component of highly efficient computer translation systems.

References

1. Pastor V. Searching Techniques in Electronic Dictionaries: A Classification for Translators / V.Pastor, A. Ampago //International Journal of Lexicography.- 2010.- Vol.23.- № 23.- P.307-354.
2. Marchuk U.N. Computational linguistics. –AST, Vostok-Zapad, 2001.-165
3. Agapova N.A. On the principles of creating an electronic dictionary of the linguocultorological type: to the problem statement / N.A. Agapova , N.F.Kartofeleva // Vestnik Tomskogo gosudarstvenogo universiteta. № 382 -2014.- .6-10.
4. Jongejan B. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. / B. Jongejan B. and H.Dalianis // Proceeding of the ACL-2009, Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics of the Asian Federation of Natural Language Processing, Singapore. – 2009. - P. 145-153
5. Johnson R. Using finite state transducers for making efficient reading compression dictionary / R Johnson, L Antonsen, T Trosterud // Proceeding of 19-th Nordic Conference of Computational Linguistics- NODALIDA 2013.- Olso, Norway.- 2013.- P.75-87.
6. Fuertes-Olivera P.A. E- Lexicography: the internet, digital initiatives and lexicography / P.A. Fuertes-Olivera, H. Bergenholtz.- London. Continuum International Published Group.- 2011 – 282 P.
7. Markovskyi A.P. Interactive template method of computer translation of scientific and technical publications / A.P Markovskyi, O.M. Mykolayivna, Fan Chunlei // Visnik of NTUU “Igor Sykorsky KPI” Informatika, upravlinnia ta obchisliuvalna tekhnika - 59. - 2013 - 86-97.

8. Vidrin D.V., Polyakov V.N. Implementation of an electronic dictionary using n grams/ D.V. Vidrin, V.N. Polyakov//Shtuchnyi intellect. № 4 - 2002. -.180-183.
9. Fuertes-Olivera P.A. Wiktionary as a prototape of collective free multiple-language internet dictionary/ P.A. Fuertes-Olivera // The functional theory of lexicography and electronic dictionary.- 2009.- P.99-108.
10. Ball L.H. Heuristic evaluation of e-dictionary / L.H. Ball, Bothma T.J.D. // Library Hi Tech, - 2018.- Vol. 36. - № 2.- P. 319-339.