УДК 004.4
DOI https://doi.org/10.32689/maup.it.2022.3.2

Вячеслав ЛІСКІН
кандидат технічних наук, старший викладач кафедри прикладної математики, факультет прикладної математики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37, Київ, Україна, індекс 03056 (liskinslava@gmail.com)

ORCID: 0000-0002-9418-0633

Сергій СИРОТА
кандидат технічних наук, доцент, доцент кафедри прикладної математики, факультет прикладної математики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37, Київ, Україна, індекс 03056 (sergiy.syrot@gmail.com)

ORCID: 0000-0003-0795-167X

Ольга ЧОЛИШКІНА
кандидат технічних наук, доцент, директор Інституту комп'ютерно-інформаційних технологій, Міжрегіональна Академія управління персоналом, вул. Фрометівська, 2, Київ, Україна, індекс 03039 (greenhelga5@gmail.com)

Viacheslav LISKIN
PhD., Department of Applied Mathematic National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave, Kyiv, Ukraine, postal code 03056 (liskinslava@gmail.com)

Sergiy SYROTA
PhD., Associated professor, Department of Applied Mathematic National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave, Kyiv, Ukraine, postal code 03056 (sergiy.syrot@gmail.com)

Olha CHOLYSHKINA
Candidate of Technical Sciences, Associate Professor, Director of Institute of computer information technologies, Interregional Academy of Personnel Management, Frometivska str., 2, Kyiv, Ukraine, postal code 03039 (greenhelga5@gmail.com)

Бібліографічний опис статті: Ліскін, В., Сирота, С., Чолишкіна, О. (2022). Технологія для автоматизації генерації тестових запитань за допомогою аналізу формальних понять. *Інформаційні технології та суспільство*, 3 (5), 14–19. DOI:

Bibliographic description of the article: Liskin, V., Syrota, S., Cholyshkina, O. (2022). Tekhnolohiia dlia avtomatyzatsii heneratsii testovykh zapytan za dopomohoiu analizu formalnykh poniat [Automated test question generation approach using formal concept analysis]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society*, 3 (5), 14–19. DOI:

AUTOMATED TEST QUESTION GENERATION APPROACH USING FORMAL CONCEPT ANALYSIS

The article is devoted to the E-learning systems content development. In continuation of the previous works of the authors, the technology for the automatic creation of closed-type test questions with multiple correct answers proposed. The proposed technology is based on the set theory model of test question. Grounded Formal Concept Analysis approach for solving this task. This technology helps to create big sets of tests, with the help of which chunk-knowledge will be tested. It was shown, that this approach can help to solve the problem of the e-learning testing students' knowledge and will help to avoid cheating.
Key words: *E-learning, Content, Chunk, Test, Formal Concept Analysis.*

ТЕХНОЛОГІ ДЛЯ АВТОМАТИЗАЦІЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ЗАПИТАНЬ ЗА ДОПОМОГОЮ АНАЛІЗУ ФОРМАЛЬНИХ ПОНЯТЬ

Стаття присвячена розробці контенту систем електронного навчання. У продовження попередніх робіт авторів запропоновано технологію автоматичного створення тестових запитань закритого типу з кількома правильними відповідями. Запропонована технологія базується на теоретико-множинній моделі тестового питання. Обґрунтований підхід аналізу формальної концепції для вирішення цього завдання. Ця технологія дозволяє створювати великі набори тестів, за допомогою яких будуть перевірятися chunk-знання. Показано, що такий підхід може допомогти вирішити проблему перевірки знань студентів електронного навчання та допоможе уникнути списування.
Ключові слова: *Електронне навчання, вміст, чанк, тест, формальний аналіз понять.*

Formulation of the problem. In the context of the COVID-19 pandemic, distance learning systems have been widely developed and used in educational activities. At the present stage of their use, which is accompanied by the constant growth of educational materials and distance courses directly, requirements are put forward for their quality and the ability to adequately assess the results of students.

Testing is an important part of the educational process. A characteristic feature of distance education is the use of information technology in the control of students' knowledge.

A distance course with automated teaching capabilities makes it possible to implement an individual approach to learning, to promote the development of independence, self-discipline, and creative abilities of students, and will provide access to the necessary sources of educational information.

Analysis of the possibilities of traditional testing with a fixed number of tasks and the time to complete them, used in distance learning as the main method of control, led to the conclusion that it is necessary to turn to adaptive testing. By adaptive testing, we mean tests that are formed by the E-learning environment, considering the student's passing of previous tests.

One of technics of adaptive testing was proposed in [1]. Such technics is based on the assumption that the system has a large number of questions related to a given topic but creating such bank of questions is quite laborious. According to the authors of this work, this process can and should be automated.

Denotation of question types and review of existing approaches.

Conventional types of test questions. Let us denote types of test questions:

“Matching questions” provide a list of subjects which corresponds to other subject answers. The respondent must "match" the correct answers with each question.

For example

$$\left|\begin{matrix} What\ time\ of\ year\ is\ January\ ? \\ What\ time\ of\ year\ if\ it\ is\ March? \\ What\ season\ in\ September\ is? \end{matrix}\right| \rightleftharpoons \left|\begin{matrix} summer \\ autumn \\ winter \end{matrix}\right|$$

“Multiple Choice” – in response to a question, the respondent chooses from multiple answers. There are two types of multiple-choice questions – single answer where there is only one correct answer. For example:

January is

- *a summer month*
- *a winter month*
- *an autumn month*

Multiple answer where the student must pick all answers that apply.

The summer month are

- *January*
- *March*
- *September*
- *July*

In “True/False” response to a question, the student selects either True or False.

January is a winter month

- *yes*
- *no*

From simple examples above we understand that all questions where almost about the same. And there is possible generate all possible questions of all types automatically. For this purpose, we should formalize the problem, develop an ontology, and propose an efficient algorithm to do this.

Review of existing models and approaches for automatic test creation.

Most test generators give the opportunity to create question without controlling its content. The question is manually entered, answer options are also entered manually, and scores are assigned for each answer. This approach requires the attention of the course developer as well as the time required [2].

The mostly represented approach, as already mentioned, is based on the development of tools for creating and conducting testing without reference to the meaning of tests and the subject area in which training is performed.

There are also known approaches that try to model the subject area and automatically build and verify test questions based on the obtained models. Also, group questions into tests based on the student's answers.

In previous publications, the authors considered technologies for automating the creation of test questions [3]. The approaches considered were based on the so-called concept of educational chunk.

Chunk is a unit of information which is understanding as one part of the memory, as like an array of letters, words or numbers, which are consider as one knowledge.

Fundamental works describing the subject area of the issues themselves have been known since the 50s of the 20-th century [4].

The logic of questions and answers, also known as erotetic logic, can be approached in different ways. The most general approach treats it as a branch of epistemic logic [5; 6] specially devoted to the problems of the logic of questions and answers.

The general idea of this direction is the construction of sets of queries and responses and finding dependencies on these sets. As well as the study of the fact that set-theory operations with over a set of queries will lead to a match in on sets of responses.

Therefore, it becomes necessary to normalize such issues, that is, to isolate the sets of entities, relations between entities and to clarify the restrictions that are imposed on such sets.

However, according to the authors, there is a need to narrow the mathematical base for the development of applied software which can generate questions.

Formal model of test question. Returning to the example above, we see that in fact the same question can be formulated in different phrases.

What time of year is January? What time of year if it is January? What season is in January?

Therefore, it becomes necessary to normalize such issues, that is, to isolate the sets of entities, relations between entities and to clarify the restrictions that are imposed on such sets.

There are published approaches to formalizing this task by creating an ontology of the question itself [7].

The scheme of one line of a matching question can be represented by a chain: *notation – essence1 – link(relation) – essence2,*

$$Q: < N, E^V, I, E^A >, \tag{1}$$

where, $N = \{n_i\}$ *set of* notations, $E^V=V=\{e_i\}$ – set of essences of subject area wich located on the right side of question $E^A=A=\{e_i\}$ (must be noted that right side can be switched with a left one), $L\ \{l_k\}$– links between essences.

Relation I_k can be considered as a mapping from set E^V to set E^A. Such mapping can be one-to-one, one-to-many, or many-to-many type.

In case one-to-one relation we get so called *strict* maching wich means that one essence from right side coresponds to one and only one on the left side denote $\leftrightarrow$. In case one-to-many relation we get so called *strong* maching of inclusion type wich means that one essence from right side coresponds to more than one on the left side denote $\Lleftarrow$. Respectively relation many-to-many we get *soft* matching denote $\leftrightarrows$.

We can see that in the case of strict compliance, the question will be formed quite simply. Like it was proposed [3] we get some essences from *V*, coresponding essences from *A.* And using sets *N, L,* build oll possible strict matching question. Getting one of essence from *V* and one coresponding essence from *A* than few not corresponding from *A* (distructors) we obtain a set of "multiple choice one anwer" quastions.

In simple cases, it is easy enough to construct sets with relations of this type. But in practice, on existing subject areas that contain more than two dozen entities, it is quite difficult not to miss the characteristics of these relations.

Using Formal Concept Analisys for generating.

Let us consider the set of essences *V* and the set of essences *A* with an arbitrary relation $I \subseteq V \times A$, such that pIa, where $p \in V$, $a \in A$, if and only if *a* is the essence *p* is in relation *I* Then triplet $K = (V, A, I)$ is called the *formal context.* The the relation can be perfommed by binary matrix with rows from *V* and columns from *A*. Let define the connection [8]:

$$P' := \{y \in A \mid xIy \text{ for all } x \in P\}, \text{ for } P \subseteq V$$

$$G' := \{x \in V \mid xIy \text{ for all } y \in G\}, \text{ for } G \subseteq A. \tag{2}$$

Then the pairs (P, G) satisfying $P \subseteq V, G \subseteq A, P' = G, G' = P$ are called *the formal concepts* of the formal context $K = (V, A, I)$. Every object $p \in P$ have all relation *L* to *G.* So, the formal concept is the set of objects from domain such that every one of them have all attributes from certain subset of attributes of that objects.

The set of formal concepts (P, G), where $P \subseteq V, G \subseteq A$, is partially ordered by the relation: $(P_1, G_1) \leq (P_2, G_2)$, if $P_1 \subseteq P_2$ and $G_2 \subseteq G_1$, and form a complete lattice $L(K)$, called *the concept lattice* of the context *K* [9]. The pair (P_1, G_1) is called the subconcept of the concept (P_2, G_2), and the pair (P_2, G_2) is called the superconcept of the concept (P_1, G_1).

For the sets $\wp(V)$ и $\wp(A)$ ($\wp$ denotes that $\wp(X)$ – is set of all subsets *X*) mappings $G \mapsto G'$ и $P \mapsto P'$, are Galois connections having next properties [14]:

1) If $G_1 \subseteq G_2$ than $G_1' \supseteq G_2'$ for any $G_1, G_2 \subseteq V$, i.e. if the set G_1 is included in a set of G_2, than a set G_2' is included in a set G_1';

2) If $P_1 \subseteq P_2$ than $P_1' \supseteq P_2'$ for any $P_1, P_2 \subseteq A$, т.е., i.e. if the P_1 is included in a set P_2, than set P_2' is included to set P_1';

3) $G \subseteq G''$ and $G' = G'''$ for $G \subseteq V$;

4) $P \subseteq P''$ and $P' = P'''$ for $P \subseteq A$.

It follows from properties 1–4 [10], that double applying of operator '' has properties of closure operator, and connections G |→ G' и P |→ P' define the dual isomorphism between complete lattices of closed subsets *V* and *A*. According to properties (1)-(4), concept (*G, P*) also be denoted (P', G') = (G'', G') = (P', P'').

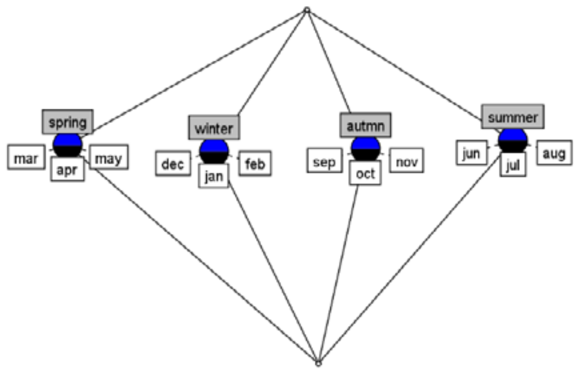
I context $K = (V, A, I)$ set of concepts is preordered by relation [5, 14]:

$$(G_1, P_1) \le (G_2, P_2), \text{if } G_1 \subseteq G_2 \text{ и } P_1 \supseteq P_2. \qquad (3)$$

The concept lattice can be represented by the line diagram (Hasse diagram) in which every node of the concept lattice is corresponded to the concept from context.

Fo examle simple let us consider a siple context with month simple context with months and their relation to seasons. Using free software ConceptExplorer (special thanks to our colegue Serhiy A. Yevtushenko) [11] we built next Hasse diagrams (fig. 1-2)

	spring	summer	autmn	winter
mar	X			
apr	X			
may	X			
jun		X		
jul		X		
aug		X		
sep			X	
oct			X	
nov			X	
dec				X
jan				X
feb				X



(a) **(b)**

Fig. 1. Binary matrix (a) and lattice (b) of Context "Months-seasons"

First step is to fill binary matrix of relation I which means that month relates to season (Fig.1-a). Then we can generate a lattice (Fig.1-b). Conventional top means universal set, bottom empty set, every node means the subset of substances which participate in correct multiple-choice-multiple-answer question while the rest concepts are distractors.

Let us make the context more complex adding, for example, the weather conditions typical for each month (Fig. 2).

	spring	summer	autmn	winter	rain	snow	sun
mar	X				X	X	
apr	X				X		
may	X				X		X
jun		X					X
jul		X					X
aug		X					X
sep			X		X		X
oct			X		X		X
nov			X		X	X	
dec				X		X	
jan				X		X	
feb				X		X	

Fig. 2. Binary matrix of complicated context

The lattice is shown in Fig. 3.

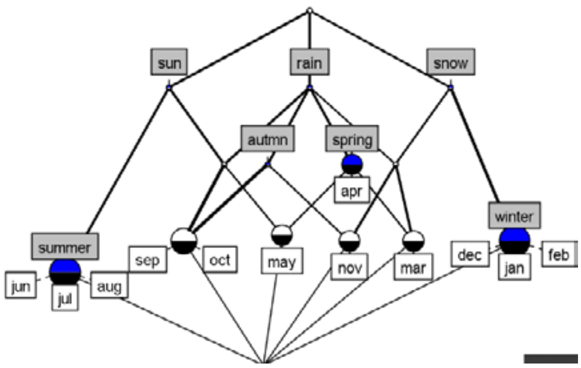


Fig. 3. Lattice of complicated context

How do we read this lattice? Going up from bootom to top by the path of edges we ad every time a substance marked in a bar. When a concept has an addge with other concept it means that consepts participate in one answer. Let us consider the ideal connected with node marred (sep-oct) we found that September and October are autmn months wich can be sunny and rainy (Fig. 4).

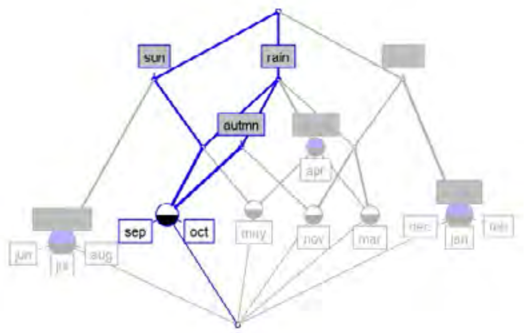


Fig. 4. Ideal generated by concept sep-oct

At same time nolabelled node Fig. 4-b gives us November and March which are connected whith rain and snow.

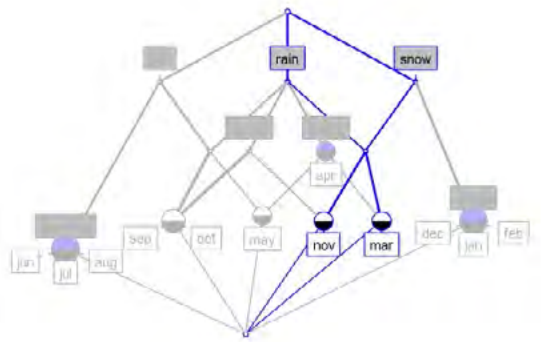


Fig. 5. Ideal generated by nonlabelled concept

Let $J(a)$ – ideal, generated by the concept a, $a \in L$, where L – initial lattice; $D(a)$ – dual ideal, generated by the concept a. It contains all upper elements having path to a. For selection of destructors it is necessary to take sequentially concepts $x \in D(a)/\{a\}$ under criteria of minimal distance from a, and generate new ideals $J'(x)$. The destructors will be objects from all concepts y, where $y \in J'(x)/J(a)$. For attributes testing we generate new dual ideals $D'(x)$ and choose destructors from the set $D(x)/D'(x)$.

Conclusion. As a result of the review of mathematical models and software for creating test questions, it was concluded that it is necessary to continue research to develop technologies for automatic generation of test questions. The existing ontological model of the test question was improved by introducing restrictions on the set of entities that are part of the test question. The types of relations between the entities of the test question were explored. Correspondence between such relations was established for subtypes of generated questions. Proposed clasification for maching in question to *strict, strong, soft* depending of relation type between question and proposed answer. Automatic generation technology of "multiple choice with multiple answer" questions based on Formal Concept Analysis was suggested.

References:

1. Syrota S., Kopychko S., Liskin V. (2020) Ontology-Based Approach for E-learning Course Creation Using Chunks. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol. 938. Springer, Cham https://doi.org/10.1007/978-3-030-16621-2_60
2. Test constructor. Available at: https://onlinetestpad.com/ru/testmaker
3. S.V. Syrota, V.O. Liskin. Development of the test generator for "Moodle" based on ontology. *Eastern-European Journal of Enterprise Technologies*. 2015, 5/2 (77). P. 44–48. doi: 10.15587/ 1729-4061.2015.51334
4. Prior A., Prior M. Erotetic logic. Philosophical review, 1955, 64, 43–59

5. Logic of questions and answers. Available at: pplied-logic/Logic-of-questions-and-answers
6. Nuel D. BelnapJr., and Thomas B. Steel Jr. The logic of questions and answers. Yale University Press, New Haven and London 1976, vii + 209 pp.
7. Liskin V. E-learning Information Technology Based on an Ontology Driven Learning Engine / V. Liskin, S. Syrota // *International Journal of Computer Science and Information Security*. 2017. Vol. 15(8). P. 258–263.
8. B. Ganter Composition and Decomposition in Formal Concept Analysi H.H. Bock (Ed.), Classification and Related Methods of Data Analysis, North-Holland, Amsterdam (1988), pp. 561-566.
9. B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
10. Syrota, S., Liskin, V., & Kopychko, S. (2020). Using Galois Connection to Hierarchical Classifying Chunks in Ontology-Based E-learning System. 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC), 1-5. DOI: 10.1109/saic51296.2020.9239131
11. Concept explorer. Avaible at https://sourceforge.net/projects/conexp/files/ /conexp/