Serdica
Bulgariacae mathematicae publicationes

Сердика
Българско математическо списание

A METHOD FOR A COMPUTER-ASSISTED TEST CONSTRUCTION

AVRAM M. ESKENASI, VALENTIN D. SABEV

A method for computer-assisted test-item construction is considered, which is a part of a whole program system for test generation. The user supplies: a) a fixed structure vocabulary comprising natural language words and word groups; b) test frames using a simple language. A program is meant to analyze any given frame and compile a set conditions. Using these conditions and parts of vocabulary another program generates syntactically, linguistically and logically valid test-items. The system involving the method described has been successfully implemented in several Universities in Bulgaria.

1. Introduction. A method for automatic test construction was proposed in 1975 [1] and later extended [2]. The program system based on it has been successfully implemented in several Bulgarian Universities and High schools. As a result some perfection appeared necessary concerning particularly the semantics of the test items generated.

Let us introduce some basic notions referring to [3]. For simplicity we will consider only multiple-choice test-items. A test is a set of items. An item consists of a question and several answers, one being the true and the others wrong, called distracters. E. g.:

Paris is the capital of:

a) Bulgaria

b) France

c) Switzerland.

Among the four problems related to the computer-assisted test construction [4] — item banking, item selection, test output, and item generation — the latter seems to be the most difficult. The method proposed aims at giving a solution to it.

2. Basic concepts. Only a short review of the basic concepts of the method will be given, as they are discussed in details in [1] and [2]. A test is usually designed to verify the student's knowledge and skills in a science or discipline, its base being a set of notions and their characteristics. Consequently, an item is testing the knowledge of a notion and/or its characteristics or the relations between two or more notions. Two basic concepts are suggested for the item generation. The first:

2.1 Vocabulary. The vocabulary consists of lines structured as follows

Notion	**Essence**	**Ownness**	**Function**
Elbe	River	in Germany	—
// EXEC	statement	of Job Control	calling a phase for execution

Two lines are given as an example.

SERDICA Bulgaricae mathematicae publicationes. Vol. 11, 1985, p. 54—58.

Each component of a line is required to meet some morphological and syntactical rules. These rules are precisely defined for the Bulgarian language and will be given only approximately for English. The notion (**N**) must be a noun. The essence (**E**) must also be a noun. The whole subject area (i. e. the set of all notions) is divided into subsets. The common (characterizing) name of such a subset is an essence. Provided, e. g. the area is Disk operating system (DOS) the notions could be distributed into the following subsets: statements, programs, operands, attributes, etc. As each notion belongs to one and only one subset (class) each N has one and only one (**E**). The following construction is possible: **N** is an **E.**

//EXEC is statement. Paris is a capital.

N and **E** are obligatory for each line. A notion can be contained in no more than one line. The ownness (**O**) is optional, but should be recommended just to generate more various test items. The set of all notions is divided in a second way into subsets $O_1, O_2, \ldots, O_k$, whose names are introduced as "ownness" in the vocabulary. For every two O_1, and O_j must be fulfilled:

$$O_i \cap O_j = \emptyset, \quad O_i \subset O_j \text{ or } O_j \subset O_i.$$

The form of **O** is (each component in brackets is optional)

(noun)

preposition noún

(adjective)

For example:

Of Job Control. In Germany. Of the statement//OPTION.

If the area considered is DOS, is mentioned above, a possible division could be:

1	DOS
11	Librarian
12	Job Control
121	Statement//EXEC
122	Statement//OPTION.
. . . . .	

The fourth component of each line is the function (**F**). The form is dependent on the language. For English it could be:

present participle object group

E. g.: calling a phase for execution

F is an optional component, but the variety of the items generated is obviously influenced by its presence.

The vocabulary set forth above is supplied by the user. He can create, delete, and update lines, using a very simple language.

2.2. Frames. The second basic point is the way of generating natural language structures, i. e. test-items. Each structure results from a frame which can be either built-in, or user-defined. A frame consists of constant parts and variables. The latter are replaced by words extracted from the vocabulary when transforming the frame into a concrete test-item. A simple example:

Frame: (**N**) is the (**E**): (1) question

(**O**) answers

Item: Paris is the capital:

(2)

a) of France
b) of Bulgaria

The user frame-specification language consists of three types of objects — constants, variables and mixed.

Object	Syntax	Semantics
Constant	Character string	Natural language words which remain constant when a frame is transformed to a test item.
Variable	**N**, **E**, **O** or **F** in parenthesis, i. e. (**N**) (**E**) (**O**) (**F**)	The symbol will be replaced by the respective component from a vocabulary line where the item is generated.
Mixed	Character string in parenthesis	Natural language words in their basic form which should be coordinated grammatically with the concrete value of a variable.

Mixed type objects are necessary in Bulgarian, Russian, French, etc. Let us have an example in French:

(**N**) est (un) (**E**), appartenant à:

In this case "un" has to get the gender of the concrete value of the **E** in each item generated.

The following restrictions are imposed on the user-defined frames:

R1. Negative questions are not permitted (theory does not recommend them either [3]).

R2. A variable of a given type may appear not more than once in a frame.

R3. Each question-part ($\mathbf{F}_q$) of a frame must contain at least one variable. The same applies to each answer-part ($\mathbf{F}_a$).

R4. Each frame must contain a (**N**).

R1-R4 are quite natural and not so "restrictive" as they might have seemed.

3. Generation. 3.1. Basic conditions. When generating a test-item the following conditions must be satisfied to obtain a syntactically, linguistically and logically valid item:

C1. All necessary variables must be present in the line selected. Remember that **O** and **F** are optional.

C2. Every two answers must be different.

C3. Each distracter must be completely false. An item of the type that follows appears unadmissible:

//EXEC is:
a) a statement
b) a Job Control statement

C4. If the question contains an explanative part, containing variables, these variables are not a subject to testing. Therefore they must be in conformity to any of the answers: Which of the following is a // MTC operand:

a) REWIND b) ACTION

The **N** in every answer must be an "operand".

3.2. Generation of conditions. The above mentioned conditions have to be translated into the terms of the vocabulary in order to be used in the process of the item generation. Therefore an analysis of each frame appears necessary and results in a set of conditions, sufficient to have C1-C4 fulfilled. No problems arise about C1 and C2. E. g., if the frame contains a (**F**), it is sufficient to check whether the vocabulary line used for the item generated contains an **F**. But as far as C3 and C4 are concerned, an analysis on semantic level has proved to be necessary. The following formalism has been introduced. Each variable (**V**) belongs to one and only one of the following 4 classes:

VN (notion variable) — this is the (**N**) variable.

VT (tested variable) — this **V** is subject to the testing.

VE (explanative variable) — this **V** is present in the explanative part of the frame question-part ($\mathbf{F_q}$).

VA (accompanying variable) — each **V** which does not belong to any of the other three classes.

This formalism is transparent to the user. The above definitions are "intuitive". The formal classification rules follow:

Each **N** is a **VN**.

If **N** belongs to $\mathbf{F_q}$, then every **E**, **O** and **F**, belonging to $\mathbf{F_a}$, is a **VT**, and every **E**, **O** and **F**, belonging to $\mathbf{F_q}$, is a **VA**.

If **N** belongs to $\mathbf{F_a}$, then

if **E**, **O** and **F** belong to $\mathbf{F_q}$, then **E** and **O** are **VE**, and **F** — **VT**;

if **E** and **O** belong to $\mathbf{F_q}$, then **E** is a **VE**, and **O** is a **VT**;

if **E** and **F** belong to $\mathbf{F_q}$, then **E** is a **VE** and **F** is a **VT**;

if only one of **E**, **O** and **F** belongs to $\mathbf{F_q}$, then it is a **VT**.

every **E**, **O** and **F**, belonging to $\mathbf{F_a}$, is a **VA**.

Following R1-R4 it is easy to prove that the class of every **V** is defined in a unique way and that every frame contains a **VT**. E. g. in (1) (**N**) is a **VN**, (**E**) is a **VA**, (**O**) is a **VT**. Now we are ready to show what kind of conditions are generated in order to satisfy C1-C4. Let us denote by $\mathbf{N^t}$, $\mathbf{E^t}$, $\mathbf{O^t}$ and $\mathbf{F^t}$ the components of a dictionary line used to generate the true answer and the question, and by $\mathbf{N^d}$, $\mathbf{E^d}$, $\mathbf{O^d}$ and $\mathbf{F^d}$ — the components used to generate distracters.

For C1: If$(\mathbf{O}) \in \mathbf{F_q}$, then $\mathbf{O^t} \neq$ blank is generated; if $(\mathbf{O}) \in \mathbf{F_a}$, then additionally $\mathbf{O^d} \neq$ blank is generated. In complete analogy conditions concerning (**F**) are generated.

For C2: If $\mathbf{F_a}$ contains k variables v^1, v^2, . . ., v^k and p is the number of the answers, then for every two i and j ($i, j \in [1, p]$) at least one r such that $v_i^r \neq V_j^r$ must exist.

For C3: Only **VT**-s are considered. For every such (**E**) and/or (**F**) $\mathbf{E^t} \neq \mathbf{E^d}$ and/or $\mathbf{F^t} \neq \mathbf{F^d}$ are generated. For every (**O**) ($\mathbf{O^t} \neq \mathbf{O^d}$ AND $\mathbf{O^t} \supset \neq \mathbf{O^d}$ AND $\mathbf{O^d} \supset \neq \mathbf{O^t}$ is generated.

For C4: Only **VE**-s (belonging to $\mathbf{F_q}$) are considered. For every such (**E**) and/or (**O**) $\mathbf{E^t} = \mathbf{E^d}$ and/or $\mathbf{O^t} = \mathbf{O^d}$ are generated.

3.3. Linguistic problems. Linguistic problems should be solved with regard to the concrete natural language which has already been complet-

ed for Bulgarian. The following is an attempt to summarize them for most of the European languages:

L1. For some the declination of (**N**) and (**E**) is to be determined. For others — the same applies to the article.

L2. The **V** is to be determined with which every mixed object is to be coordinated.

L3. In some languages prepositions change in form depending on the following word.

3.4. Scheme of the generation process.

(i) Frame analysis

Every frame specified by the user is analysed. The class of every variable (**VN**, **VT**, **VE**, **VA**) is determined. The conditions sufficient to have C1-C4 fulfilled are generated and cataloged with the constants and mixed objects. The same applies to linguistic conditions, if any.

(ii) Test item generation

The user specifies the frame and the (part of) vocabulary to be used, as well as some parameters concerning the number of items in one test variant, number of variants, number of answers in one item, etc. After a frame has been chosen, a vocabulary line for the true answer is selected by using a random value generator. The question and the true answer are generated (if the components of these line meet the conditions previously generated by the frame analysis). Then the vocabulary is searched for lines to generate distracters. Then an exchange of places is performed between the true and one of all answers.

REFERENCES

1. А. Ескенази. Система автоматического генерирования тестов. Сборник докладов НККЛ, Варна 1975. С., 1976, 121—124.
2. A. Eskenasi. A Language for Computer Assisted Test Construction. *Wiss. Z. Techn. Univers. Dresden*, **29**, 1980, 347—348.
3. R. Ebel. Essentials of Educational Measurements. Prentice Hall, 1972.
4. G. Lippey (ed). Computer-assisted Test Construction. Educational Technology Publications, Englewood Cliffs, 1974.

Centre of Mathematics and Mechanics
Sofia 1090 *P. O. Box 373*

Received 31. 1. 1983