ACTA DIDACTICA NAPOCENSIA

Volume 12, Number 2, 2019 - DOI: 10.24193/adn.12.2.6

THE CREATION AND APPLICATION OF A QUESTION BANK FOR AN INTRODUCTORY LOGIC MODULE

Mária BAKÓ, László ASZALÓS

Abstract: The expansion of the LMS system triggered the spread of MCQs as an addition, or even substitute for traditional assessment methods. Our goal was to create a question bank, which could be used to evaluate learning at advanced cognitive levels, and aside from examination purposes it could also function as a tool for student practice. In this article, we briefly describe how this question bank was created, and how we generated the questions and all corresponding incorrect answers. In our research we also examine which categories of the MATH taxonomy the question types appearing in our exams correspond to. Grouped by question types, we compare the students' practice and exam results. Accessing this data from the Moodle system, we analyse whether students perform the exam at least with the same results as the practice exercises.

Key words: MCQ, logic, MATH taxonomy

1. Introduction

By now, all students – from kindergarten to higher education – belong to generation Z and can be treated as digital natives who were born into a digital world that inevitably affects their daily lives (Prensky, 2001; D. Oblinger & J. Oblinger, 2005; Palfrey & Gasser, 2008; Prensky, 2001). Nonetheless, many scientific studies have revealed that not all students have the same high level of technological competence (Hosein, Ramanau & Jones, 2010; Jones & Healing; 2010; Hargittai 2010; Schulmeister 2010). Empirical evidence has disproved the assumption that students born in the digital era – thanks to the technology-rich environment – are able to make the most of technology and can use technological applications at high levels (Hosein, Ramanau & Jones, 2010; Jones & Healing, 2010; Kennedy, Krause, Judd, Churchward, & Gray 2008).

However, considering the explosive progress of different technologies and the emergence of life long learning as a result of the continuous advances of the market economy, higher education institutions have realised that they need to implement significant educational reforms. Although the existence of digital natives as a generation is highly questioned, institutions have tried to introduce technologies corresponding to the cognitive learning patterns of digital natives in the educational environment. As a result, the use of Learning Management System (LMS) is widespread in both traditional education and distance learning. In Hungary, the emergence of LMS systems coincided with the shift of university education into mass training, where there are a large number of students all with very different levels of knowledge. LMSs have a wide range of opportunities for sharing course content: handouts, student works, short quizzes or online exams; and can be used as a communication platform. The steady growth in student numbers, increased availability of computer networks, and the proliferation of LMS systems have led to the increased use of multiple-choice questions (MCQs) as an addition or even a substitute for traditional assessment methods.

Some researchers do not support the use of MCQs, arguing that they only encourage memorisation and lexical knowledge and do not stimulate (or test) high cognitive processes (Airasian, 1994; Scouller, 1998). However, other researchers claim that this depends on what the tests are, as it is also possible to evaluate learning at higher cognitive levels by appropriate tests (Cox, 1976; Johnstone & Arnbusaidi, 2000).

Moodle was introduced in 2015 at the Faculty of Informatics of the University of Debrecen. With the inclusion of this LMS, we have the opportunity to use MCQs for practice tests and exams. Our goal

Received February 2019.

Cite as: Bakó, M. & Aszalós, L. (2019). The creation and application of a question bank for an Introductory Logic module. *Acta Didactica Napocensia,* 12(2), 61-74, DOI: 10.24193/adn.12.2.6.

was to create a question bank, which could be used to evaluate learning even at higher cognitive levels, and to give students options to practice their knowledge, while reducing the probability of getting the same questions during the exam. The course *Introduction to Logic and Computer Science* is an introductory course in which students acquire many competences they will need in the future (in programming, database management and at many other courses in Computer Science and Mathematics). Here, as students move from high school to university, they are required to know exact definitions and how to implement them, a skill that was not necessarily needed at high school level. When designing the module, we aimed to cover the first three levels of Bloom's taxonomy. This means *understanding* the operations of the algorithms and *applying* them for any formulae, while *remembering* the related items and definitions. We have introduced higher levels of this taxonomy alongside the exercises for our more talented students: we have shown several solution types for certain types of questions, allowing them to choose the best befitting their personal preferences, since only the final result is important at the exam, not the path they take to get there. Of course, throughout the semester we presented all the different such paths and practiced them with the students.

In this article, we briefly describe how this question bank was created, and how we carried out the generation of questions and corresponding incorrect answers, which helped us to create a large number of test question of the same types. With this we produced such a large test bank that our students could use it to practice the same type of questions that would be asked at the exams, but with different data.

Additionally, we also examine how the MATH taxonomy (mathematical assessment hierarchy) – created by G Smith and his colleagues (Smith, Wood, Coupland, Stephenson, Crawford & Ball, 1996) – corresponds to our question types, and compare our students' results on the practice tests and at the exam. We downloaded the statistics from our Moodle systems, and analysed whether students perform the exam at least with the same results as the practice exercises.

2. Theoretical framework

The main goal of education is for students to understand the curriculum, and be able to use the learned material in practice at a later time. This is particularly true for Mathematics, where we cannot see how different ideas connect without first understanding the context, and so it becomes impossible to learn any new material without understanding the basics it builds upon. Over the years teachers have recognised this problem, and have tried to facilitate the understanding of the material with a variety of methods. Many have tried changing their teaching style, explaining the material in a more direct ways instead of the traditional, abstract one. Some tried including more practical examples. Others created better slideshows. All these different approaches are based on the assumption that if Mathematics is explained logically, then students will understand it. However research has shown that students are often more motivated to learn materials that are directly related to the exam. They adapt their learning styles to the exam, and they do whatever is necessary to successfully overcome the obstacle posed by the exam. This means that all the attention we devote to changing our teaching methods is done in vain if we do not also focus on the way we perform examinations (Ramsden, 1992). Therefore the truly interesting question to ask is what do we actually expect of an exam? That is, what does it really mean to know a fact or a concept? Does knowing a concept mean that we can state its definition, or that we can use it in context, or that we can list its synonyms and antonyms? There are different levels of knowledge, but it is difficult to define and systemise these levels. Perhaps the most well known such systemisation attempt is Bloom's taxonomy, which determines the different levels of knowledge complexity (Bloom, Engelhart, Furst, & Hill, 1956). This can be regarded as a hierarchy, a system of levels building on top of each other. It is uncertain if this scheme can be applied to all types of knowledge, however it is a fact that Bloom's taxonomy cannot be ignored if we wish to seriously explore the stratification of knowledge. The original taxonomy was created in 1956 and has been revised by Anderson and Krathwohl in 2001 as seen on Figure 1 (Anderson & Krathwohl, 2001).

Bloom's taxonomy is quite good for structuring assessment tasks, but does have some limitations in mathematical context. Smith et al. (1996) modified Bloom's taxonomy and created the MATH taxonomy (mathematical assessment hierarchy) for the structuring of assessment tasks in Mathematics, presented on Table 1 (Smith, Wood, Coupland, Stephenson, Crawford & Ball, 1996).

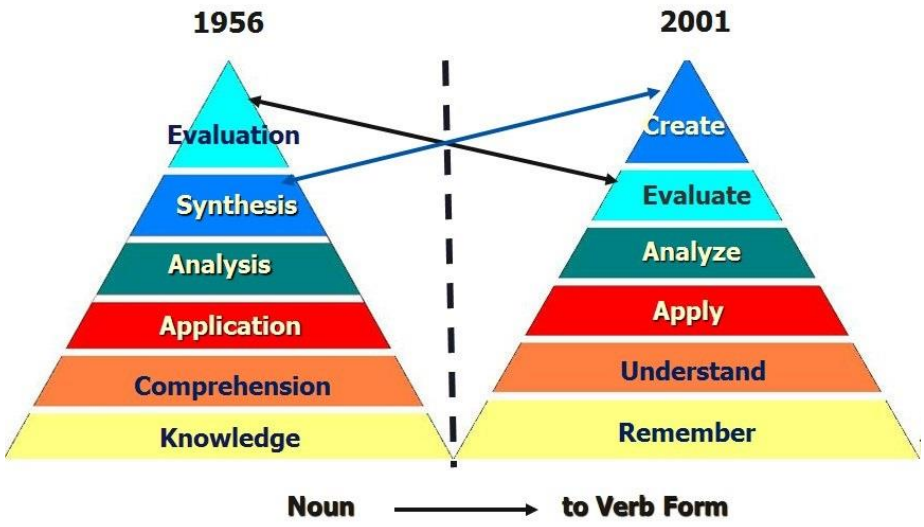

Figure 1. *Bloom's taxonomy* [1]

„Factual knowledge is remembering a specific formula or definition. Examples of comprehension are: understanding the significance of symbols in a formula, recognizing examples and counterexamples of a mathematical object or concept. Routine use of procedures covers algorithms that students would have practiced in class as drill exercises, such as changing the subject of a formula. Information transfer shows the ability to transform information from one form to another – from verbal to numerical, numerical to graphical and so on. It includes taking a general formula and applying it in a specific situation (that goes beyond routine procedures). Applications in new situations tests the ability to choose and apply appropriate methods or information in new situations. Group C categories cover justifying a result, comparisons and implications with justification and evaluation and judgments" (D.'Souza & Wood, 2003).

Table 1. *Exams to assess a range of knowledge and skills*

Group A	Group B	Group C
Factual knowledge (A1)	Information transfer (B1)	Justifying and interpreting (C1)
Comprehension (A2)	Application in new situations (B2)	Implications, conjectures and comparisons (C2)
Rountime procedures (A3)		Evaluation (C3)

3. Technical implementation

Moodle was introduced in 2015 at the University of Debrecen. The module *Introduction to Logic and Computer Science* typically has the worst performance indicator for the first semester, and since it is a foundation module, not completing it significantly slows student progress. Due to mass training, and the large number of resits following failed exams, this module required a considerable effort from the lecturer during the exam period. Moodle and the new 150 seat computer lab on campus provided an ample opportunity to switch to online examinations. Since students adapt their learning style for the

[1] https://thesecondprinciple.com/teaching-essentials/beyond-bloom-cognitive-taxonomy-revised/

exam, in order to achieve the best results we made the exam exercises available in advance so they could be used for preparation. Our goal of course was not for students to simply memorise the correct answers. To avoid this happening, we created an extensive database, with a large number of questions for each question types. The manual compilation of all these questions and corresponding answers requires significant work. Since this module introduces many algorithms, and these algorithms have to be used to solve the questions, it has become partially possible to generate the questions and solutions automatically.

For this, we implemented all the algorithms that were introduced in the lectures and practicals, and applied them to randomly chosen logical formulas. This gave us questions along with the correct answers. Constructing the incorrect answers was a more complicated task. In some cases the correct answers for other questions could be used as incorrect answers, and so could be generated. However there were question types where the incorrect answers had to be created manually.

We created close to 40 short programs, which implement an algorithm each. Adding a randomly generated formula as an input for these programs gives the questions and the output provides the correct answer. In a few cases it was easy to generate incorrect answers as well. Mostly however, answers had to be spoiled manually. Then, the text of the question, the list of correct as well as the list of incorrect answers and a set of explanations to go with the answers was saved into a file. The final version of this file had to be constructed manually, as often the explanations had to be added directly. A separate program generated tests from these files that could be uploaded to Moodle, so that it selected 4 answers for each question such that there would be a mix correct and incorrect answers, and then it formed these for answers into a test question. We ran every program twice, so that one of the runs gave the practice tests and the other the exam tests. Whilst the questions may be identical in both tests, the four answers are unlikely to be an exact match. Therefore we hope that solutions cannot be memorised. Generally, we generated 50-100 questions for each question types.

While only logical tasks had to be implemented we used the programming language Clojure, and the question files were written in the language's own data format. Once the curriculum changed, we added computer science questions and these were generated using the programming language Python and the question files were stored in YAML. The Figure 2 shows how a specific question looks in this file format (which is the more transparent of the two). Here the randomly generated formula is a*b, this is replaced in every question by a different regular expression. The program generated the correct answers, which in this case checked for up to hundreds of random strings whether they fit the regular expression or not. In this case, the strings that did not fit the regular expression could be used as incorrect answers. However, to help students in their practice we manually added a short comment to each of the incorrect answers, so when students selected these Moodle could flag why these were the incorrect answers. The subdirectory *resources* of our github repository contains the data files where the both the correct and incorrect answers are listed for the questions.

```yaml
- question: "Which string match to the regular expression <i>a*b</i>?"
  feedback: "one <i>b</i> needs to follow any number of <i>a</i>s"
  good:
  - answer: ab
  - answer: aab
  - answer: b
  ...
  bad:
  - answer: ba
    hint: "needs to end with <i>b</i>"
  - answer: bab
    hint: "needs to start with <i>a</i>"
```

Figure 2. *Structure of the question file*

4. Material and methods

In this survey we looked at the first year Computer Engineering students from the Faculty of Informatics at the University of Debrecen from the 2017/18 and 2018/19 academic years, taking the module *Introduction to Logic and Computer Science*.

Learning precise and accurate definitions is not required of these students before coming to university, and so their abstraction skills are usually insufficient for university level expectations. Without explicitly saying so, this module offers to fill this gap. Therefore, weekly tests were introduced. These take 10-15 minutes each, and cover both theory and problem solving. For the questions on theory definitions had to be precisely reproduced, and for the problem solving weekly tutorial sheets and online practice tests were given to the students. Whilst the online tests gave immediate feedback, the tutorial solutions became available the weekend before the test.

Those who achieved at least 50% on both parts – or when resitting the test – were allowed to take the exam. The exam was entirely made up of test questions, however selecting the right answers required solving the types of questions seen throughout the semester. These multiple-choice questions could have more than one correct answers.

Wrong answers resulted in point deduction, which partially suppressed guesswork. In the 2018/19 academic year we submitted 15 practice tests into the e-learning system, each assigned to its relevant topic. The number of question types in each test depended on the size of the topic it covered, so for example *Semantic Exercises with Minimally Bracketed Formulas* contained 15 different question types whilst *Automata* only 2. However, it must be noted that whilst for *Automata* there were only two types of questions, the way the tests were created whenever the student restarted the test new questions were generated, with different automata and different answers, only the type of the question remained the same. For the end-of-year exam the questions were uploaded onto a separate exam server with restricted access therefore ensuring the no cheating happens. The test generated for the exam consisted of 33 questions: 10 on theory and 23 practical ones. The types of questions present in the exam came from only 9 out of the total of 15 tests completed over the semester. Basic concepts such as sets and relations were covered in high school, but we wanted to make sure everyone was familiar with them so we repeated them at the beginning of the year. Students were assessed on them during their weekly tests, however we only wanted to focus on the new material in the final exam. So we had 10 questions from *Semantic Exercises with Minimally Bracketed Formulae*, 6 from *First-order Logic* and one each from *Inductive Function Definitions*, *Well-formed Formulae*, *Truth Tables*, *Formation rules of First-order Logic, Regular Expressions, Automata* and *Markov algorithm*. Although every student got the same type of questions, just with different data, thanks to the functionality of Moodle the questions appeared in a different order for every student. 59 out of the 150 students in 2017/18 and 82 out of the 168 students in 2018/19 who registered at the beginning of the year on the electronic education system (Neptun) were able to take the exams based on their results during term. From these two semesters, we examined the practice test and exam results of 141 students. After data cleansing 119 students remained who used the practice tests and took the exam. However, following the 2016/17 academic year, a partial change in curriculum followed on the basis of ministerial decree. Naturally the tests had to correspond to the change in the curriculum, however in the 2017/18 academic year we had 35 students who failed the exam in 2016/17 and were eligible to resit it according to the old curriculum.

Out of the 9 practical questions in the 2017 exam, 4 were not included for those retaking it from the 2016/17 academic year, as these were related to Computer Science which was not present in the previous curriculum. Therefore, out of the 9 practical questions we were only able to compare 5 with the exam results for all 119 students. For the other 4 types of questions we could only consider the data from 84 students sitting the exam under the new system.

5. Result and discussion

The module *Introduction to Logic and Computer Science* starts with the basic concepts of set theory and properties of relations. Since this part of the curriculum is largely a revision of concepts learned in high school, it was not included in the final exam. This is followed by the *Introduction of the Zeroth-*

order Logic. This section was divided into two parts on Moodle, the first deals with semantic knowledge, which contains practice tests on *Inductive Function Definitions* and *Well-formed Formulae.*

The former consists of three questions, and when the test is restarted three new, but similar questions appear. These questions ask about the inductive definitions of zeroth-order logic, as seen on Figure 3. This is done in such a way that it does not ask to repeat the definition learned in class word for word, but asks to use the knowledge of the definition to answer previously unseen questions. As seen in the example, we were interested in the answer to *Which items are part of the inductive function definition that specifies the number of occurrences of the logical constant present in the formula?* Phrasing the question in such a fashion, students can see whether they have understood the essence of the inductive function definition, which they will need later for the programming part of the course when they will be defining and implementing recursive functions.

This question covers the first three levels of Bloom's (new) taxonomy, as the student must remember the definition given in the lesson, and can only apply it in a new situation if they have understood the underlying cases of the definition as well as the inductive steps. According to the MATH taxonomy, this question can be categorised as A2/B1, that is the students learned in the lecture what an inductive definition is and saw a concrete example. At the practical this was recapped, discussed with the supervision of their tutor and they constructed solutions for similar questions. The test verifies the students' knowledge, by checking if they can determine which answers are correct. For this they must understand the concept of inductive definition which corresponds to A2 and demonstrate skills in B1, which is *information transfer*, for example construct a definition for numbers based on the definition given for sets.

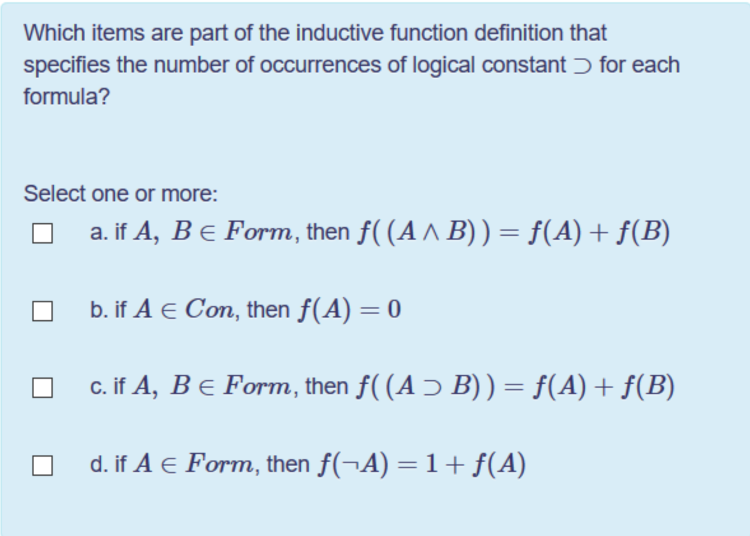

Figure 3. *Sample question on inductive definition*

On the exam, there was one question on inductive function definitions. We examined how the exam results of the 119 students related to the average results achieved on the practice exercises. 83 students (69,7%) achieved at least the same level of results at the exam as in the practice tests, and 36 (30,3%) performed worse at the exam. Amongst these results we can find very extreme cases. We had a student who completed the practice test once and achieved 100% on the exam, and also a student who took the practice test 161 times and still achieved 0% on the exam.

These practice tests are designed for students to evaluate their knowledge and gain confidence for sitting a test. However, they do not replace the material covered in lectures and practicals, even if some students try to get away with missing the lectures. This would require them to acquire a skill based on seeing an example, which corresponds to a very high level C3 in the MATH taxonomy and *Create* in Bloom's, which is not typical for our first year students.

The second test for the topic *Syntax - zeroth-order logic* was on *Well-formed Formulae* from which there was also a question on the exam (see Figure 4). Logical formulas, just like arithmetic terms, have a special structure, which can be given with inductive definitions for example. The students were introduced to these in the lectures and throughout a number of lectures we demonstrated the step-by-step construction of a couple of formulas. This question is best solved backwards, the given formulas must be broken down and it needs to be checked if the structure corresponds to any of the rules. Most incorrect solutions fail at this step, by using unauthorised characters for the operations (which may be accepted elsewhere), or by linking operations in a way that does not appear in our list of rules. This is what students need to understand and solve.

Select the strings that are formulae of the zero-order logical language, where $Con = \{p, q, r\}$! (We now accept only the fully bracketed formulae.)

Select one or more:

☐ a. $(p \wedge (p \equiv (p \supset p)))$

☐ b. $(r + (p + (p\&r)))$

☐ c. $(x \wedge ((y \supset x) \vee y))$

☐ d. $p \supset q$

Figure 4. *Sample question on well-formed formulae*

According to Bloom's taxonomy the example of well formed formula can be catorized into *Apply*, and according to the MATH taxonomy it can be categorized into A2 and A3, as the student must understand the process of constructing the formula, and they must also check whether it conforms to the rules provided by the question or not.

The results of the practice tests and the exams of the 119 students were similar for this type of question. 85 students (71.4%) achieved at least the same level of results at the exam as in the practice tests, and 34 (28.6%) performed worse at the exam. There was a student who took the practice test for this question type only once and achieved 100% on the exam, as well as a student who attempted the practice test 35 times.

The third topic in the curriculum for this module is *Semantics – Zeroth-order Logic*, for which we had two types of tests for both the practice tests and the exam. The most important concept of logic is entailment (logical consequence). By formulating this, the following concepts are introduced: models, valid, satisfiable, unsatisfiable formulae and set of formulae. All of these are verifiable with truth tables. This is often not the fastest method, but since it is omnipotent, by using this foolproof method, students can optimize the amount of knowledge they need to learn. The ability to construct such a table can be checked by submitting the main column of the table (see Figure 5). Although we do not filter out all errors (made in another part of the truth table), this is very likely to indicate the existence of most errors. To solve the qustion, the students had to prepare the truth table on paper, taking care to correctly order the rows of the table, and then copy the main column of the resulting table into the answer field of the question. This task, according to MATH taxonomy, belongs to *Routine use of procedure* in A3, since solving such a task does not require more than the knowledge of the truth functions belonging to the logical connectives, and their precedence. In order to practice the preparation of truth tables, the students were helped by an educational program (Aszalós, L. 2009).

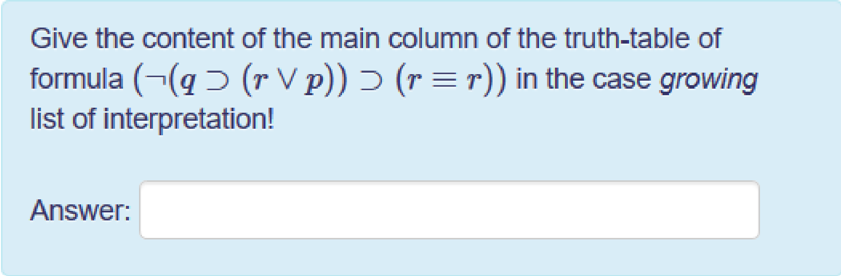

Figure 5. *Sample question on truth tables*

The results of our study show that 75 out of 119 students (63.03%) achieved at least the same level of results at the exam as in the practice tests, and 44 (36.97%) performed worse at the exam. The results for this question are somewhat worse than the previous ones, and this is probably due to the fact that the students inadvertently gave the results in the wrong order (we did draw their attention to this at the exam) so the test did not accept it as correct, even though the truth table made on paper might have been faultless. Again, there was a student who completed the practice test only once and achieved 100% on this question, but the student with the most trials (55) achieved 0% at the exam.

The curriculum also includes a practice test called *Semantic tasks with minimally parenthesed formulas* consisting of 15 questions. Logical consequence is the central concept of logic and we use it in many areas of the real-life conclusions, e.g. at implementations of algorithms (which is very important for our students). So we place great emphasis on this topic in order to help them develop the right competences. Thus, we approach this subject from multiple directions, and check their knowledge with multiple tests, see Figure 6. The importance of this part is indicated by the fact that the exam includes 10 such exercises. The practice test covers the following areas: contradictory set of formulae, satisfiable set of the formulae, valid formulae, model of the formula, model of a set of formulae, logical consequence of a formula and a set of formulae. This section also includes the normal forms received with the application of transformation/rewriting rules (or the use of truth-tables) and minimal normal forms: conjunctive normal form, disjunctive normal form, *product over sum* and *sum over product* maps. These normal forms play a prominent role in the design of electronic circuits and in various artificial intelligence applications.

For the sake of better transparency we left out the unnecesarry parentheses from the formulae following precedence rules, similarly to the customary rules in arithmetic. This adds an extra step for the studens, because they need to mentally put back the deleted parentheses in order to understand the structure of the formula and then get the final result following the well-known method. Therefore, these tasks can be categorized as A2/A3 according to the MATH taxonomy.

The best students are grouped together into a common practical group, and alternative methods are presented to them. They can use any of these at the exams and tests. The best ones can work faster with these alternative methods, making with fewer errors than their peers. Due to the choice of alternative methods at problems of logical consequences and normal forms, the MATH taxonomy level could be B2.

The results to these ten questions on the exam belonging to this topic should be handled together, as the majority of students have solved these using only truth tables. Moodle allows the questions of the exam to be presented to students in a different order and randomly selects the dozens of examples of the same type. Comparing the results of the practice tests and exam of the 199 students for this type of question showed that 98 students (82.4%) achieved at least the same level of results at the exam as in the practice tests, and 21 (17.6%) achieved worse at the exam. The limitations of using truth tables are indicated by the fact that there were no students with perfect results for all ten questions (the best result was 96.8% where the student tried 7 times). From the 119 students only 4 did not complete any of the ten questions.

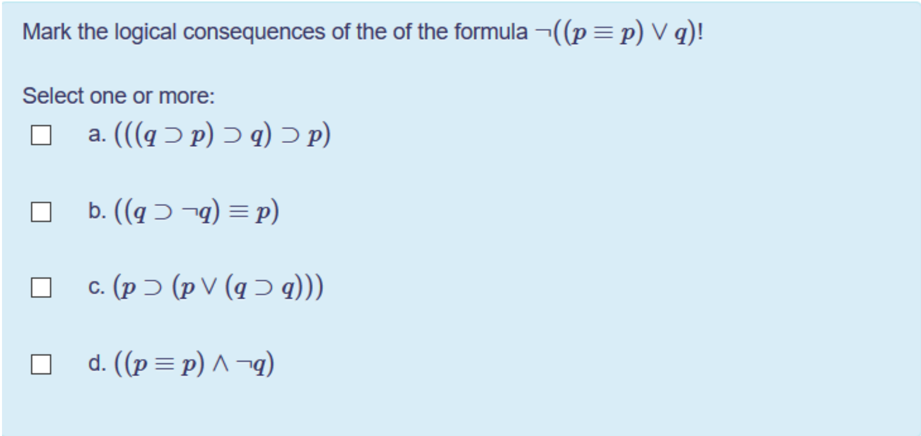

Figure 6. *Sample question on logical consequence*

Since a sufficient number of questions were included for both the practice tests and the exam on this topic, it is worth making a diagram. Figure 7 presents the results of the last exam against the average performance achieved on the practice tests. This figure also shows the trend line, which indicates a weak relationship (R^2 = 0.11 is low due to high variance). The points on the left indicate the students who took part in the exam but did not practice online. They only logged onto the portal to see the types of the tests without attempting to solve them.

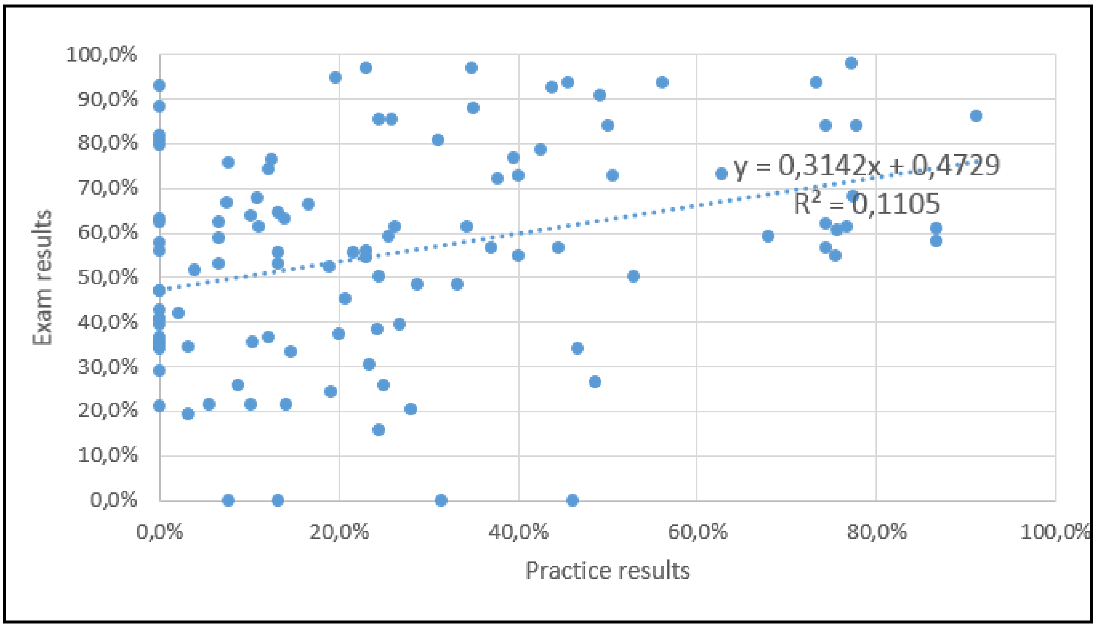

Figure 7. *Comparison of results of semantical problems*

It is possible to compare the number of attempts made on the practice tests with the results of the exam (see Figure 8). Practice tests should be considered as a reference point, which allows the student to see whether they are able to solve a specific type of question. For this reason, being successful the first or second time round, there is no need to solve any more questions. However, if the test is unsuccessful, completing a new test will not help to achieve a better result without looking at the theoretical basics, the exercises solved during the practicals, or if neither of these help then asking the instructor for help.

Zeroth-order logic can be used in many areas of Mathematics and Computer Science, but it also has significant limitations. Therefore, in Mathematics and in Artificial Intelligence we also use first-order languages, where we have constants, variables, quantifiers, properties, relationships and functions. Basically, the (properly constructed) definitions and theorems of the zeroth-order logic can be listed again here, because the basic concepts do not change, they only occur in a different form. We do not have enough time during a semester to introduce a specific first-order calculus to our students; we need to halt the module at discussing the prenex form, which is a special normal form. To be able to construct the prenex form of a formula, students have to acquire the skill to perform a number of equivalent transformations, but here they also met with different constraints, which require the introduction of the concept of bounds and free variables. Whilst this is relatively easy for everyone to do in the natural language, it may be hard to do in a formal way. However, later during the programming stage, the constraints and visibility of the variables are described using the same concept.

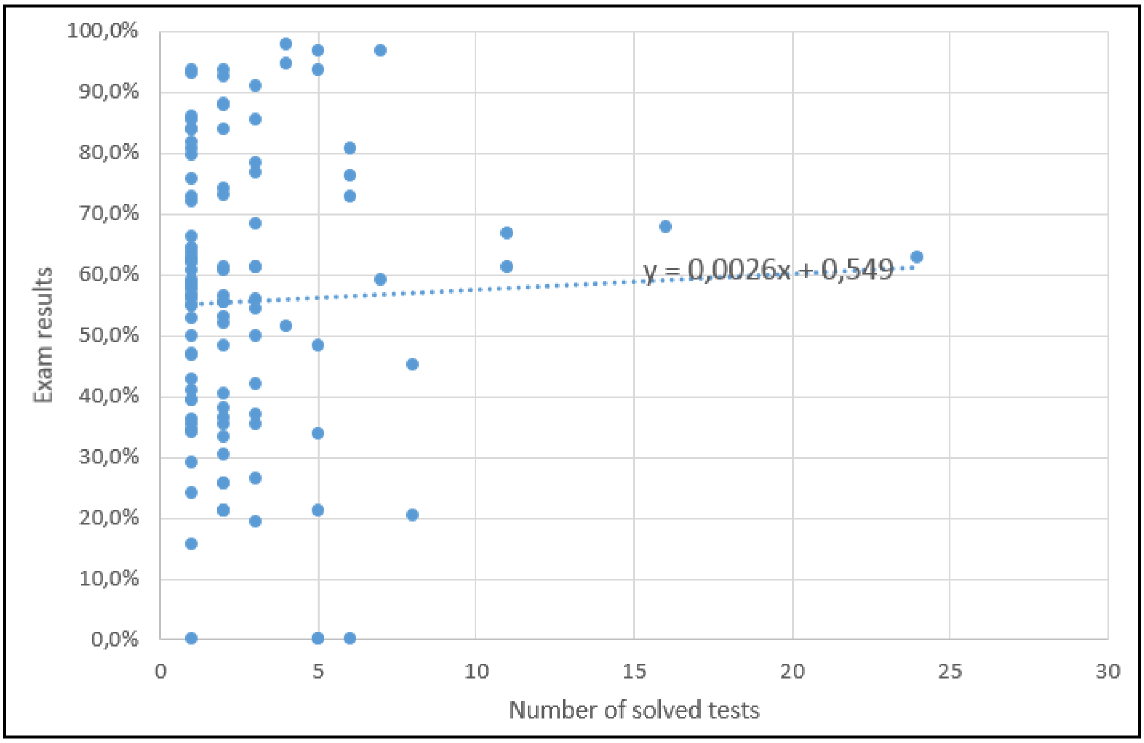

Figure 8. *Comparison of number of attempts at the practice testa and the final results*

Whether a variable is free or bounded, the substitutability of a variable, the result of its substitution, and the creation of the prenex form through such transformations are governed by simple, inductive rules. In the MATH taxonomy, these tasks fit into A2/A3, as the students have to interpret the formula containing the minimum number of parentheses, and next have to apply the specific algorithm. The practice test includes 5 first-order logic-related questions, whilst the exam test has 6, and the sixth, new question is about standardizing the variables apart, which basically summarizes the other tasks.

From the 119 students, 80 (67.2%) achieved at least the same level of results at the exam as in the practice tests, and 39 (32.8%) achieved worse at the exam.

According to the new curriculum we created several new questions types, and in one of them creativity plays the main role. Here the students need to decide whether a particular first-order formula is valid or not, see Figure 9. Typically, such a question can be disproved using a counterexample, such as for example: it is not true *that all natural numbers are odd or all natural numbers are odd*. However, saying that *every natural number is even or odd* is valid. Although there are some basic properties (even, odd, prime, etc.) and relationships (bigger, multiple-of, etc.) that we can use to try and find a counterexample – provided that one exists – the majority of students have failed this question, because there is no guideline that can guarantee a successful answer. Not including any unnecessary parentheses corresponds to category A2 in the MATH taxonomy, choosing the right method corresponds to B2, and evaluating the model invented by the student corresponds to C1. For the new curriculum we have data on 84 students, out of which 49 (58.3%) achieved at least the same level of results at the exam as in the practice tests and 35 (41.7%) achieved worse at the exam.

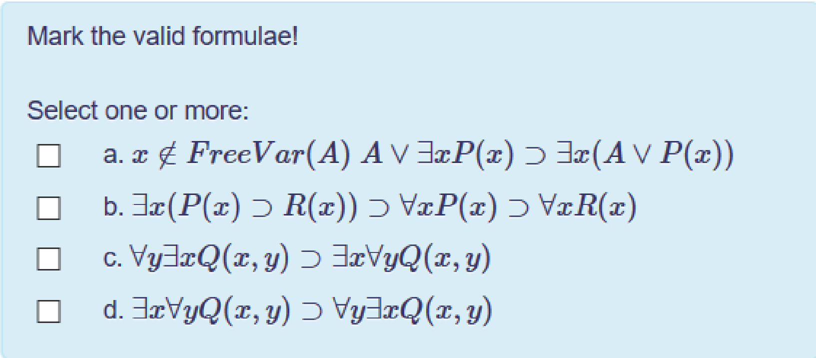

Figure 9. *Sample question on first-order valid formulae*

The emergence of Computer Science in the curriculum has manifested itself in three small sub-areas: regular expressions, finite deterministic automa and Markov algorithm. For the latter, if we had given the students a creative task, then we could have had worse results than for valid first-order formulas, following our experiences from the practical lessons. Therefore, we considered the Markov algorithm as a minimalist form of algorithms, which the students need to follow and simulate, so it is an A3 exercise. Here 73 out of 84 students achieved at least the same level of results at the exam as in the practice tests and 11 achieved worse at the exam. When pairing regular expressions (see Figure 10), or pairing the regular expressions and automatons, after interpreting the specific notations (A2) and transferring between different systems of notations (B1) the student can use the practiced standard method (A3). On the other hand, for the later pairing students could choose from alternative solution methods (B2). Thus, 66 students (62 for the other question) achieved at least the same level of results at the exam as in the practice tests and while 18 (or 22, respectively) achieved worse at the exam.

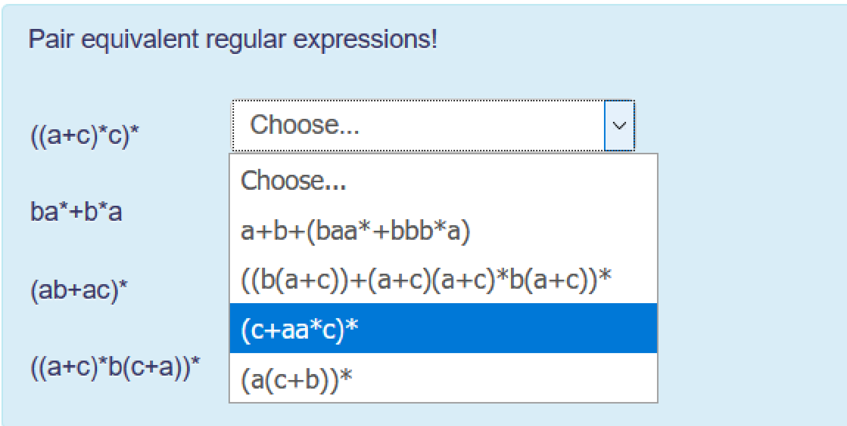

Figure 10. *Sample question on regular expressions*

Naturally, following from our objectives specified above, one of our goals was to help students correctly acquire all definitions. To do this, the official definitions were supplemented with the wrong answers written by students from the previous years, and the exact definitions had to be selected from this mixture. In addition to the 23 practical tasks, 10 theoretical questions were introduced, all of which were of the MCQ type, so students did not have to type much to solve such tasks, and checking the answers could be automated. For these questions, no practice tests were given, but the handout contained all definitions and theorems. The students simply had to recall the learned material (A1).

The exam consisted of 33 questions: 30 MCQ tests, 1 short answer test (Figure 5) and 2 pairing tests (Figure 10). Since this module is a foundation course for freshmen, where the aim is to aquire basic logical knowledge, as well as to meet the university requirements (aquire exact knowledge and interpretation of definitions and theorems), in our opinion these tests are appropriate for examination purposes. At the end of the second year, we meet these students again during the module *Foundations of Artificial Intelligence*. Here, creativity, analysis, and evaluation (which are at the top of the Bloom pyramid) have a much greater role to play, although we also describe various algorithms. MCQ tests are still used to evaluate the latter, while the former are tested in the form of an essay. These essays have to be corrected manually and adding comments to each of them can take several days. However, our freshmen are saved this waiting thanks to the automated marking, so they know their results as soon as they submit their answers.

6. Conclusion

In this article, we presented the material used to create our practice tests and exam scripts used for the module *Introduction to Logic and Computer Science*. We presented our students' results on various topics, comparing the results achieved on the practice tests with those achieved at the exams for each person. Although Figures 7 and 8 show large variation in this data, a weak correlation can be shown between the results. Although there were students who successfully passed the exam by ignoring the practice tests, they were the exception to the rule. As a counterpoint, there was a student who had *solved* unimaginably many tests, but their results were almost random and in the end did not pass the exam. The majority of our students however, did not use the practice tests for learning, but for controlling their acquired knowledge and competence, as the modern equivalent of traditional exercise books. Log files show the result of our students' optimization. Whilst the tests are paper-based (during the semester in our case), they do not use the online tests to check their knowledge, only their notes, downloadable handouts and solutions. As soon as they have the opportunity to take the exam – by fulfilling the required preconditions – preparation for the exam begins. This is manifested in the replacement of the toolkit, starting with the use of online tests. In our research, we examined how the questions of the exam are structured in terms of evaluation, and we have found that all questions cover the classification groups A and B according to MATH taxonomy. In case of one question – determining that a first-order formula is valid, or not – the test belongs to category C1 *(Justifying and interpreting)*. Comparing the results of the practice tests and exam downloaded from the Moodle system, we found that apart from two tests, more than 70% of our students achieved at least the same level of results at the exam as in the practice tests. The biggest difference was in the forementioned category C1, where only 58% of our students achieved at least the same level of results at the exam as in the practice tests.

Unfortunately, our students do not typically ask help from the lecturers, even though this could likely improve their performance.

When we asked for the opinion of our students in a questionnaire, they asked for a detailed, step-by-step presentation of the solutions for each question types. This guide is being prepared and will be available to the students next semester.

These tests were originally designed for Hungarian students. However, the English translations of the tests have also been completed – in fact, the programs that generate them have been translated to English – so they are also being introduced in our English-language course.

References

Airasian, P. W. (1994). Classroom assessment (2nd edn), New York, McGraw-Hill Humanities.

Anderson, L., & Krathwohl, D. A. (2001). Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York: Longman.

Bloom, B. S. (1969). Some theoretical issues relating to educational evaluation. In R. W.Tyler (Ed.), Educational evaluation: new roles, new means: the 63rd yearbook of the National Society for the Study of Education (part II) 69(2), pp. 26-50). Chicago, IL: University of Chicago Press.

Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; & Krathwohl, D. R.(1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York, David McKay Company.

Cox, K. R. (1976) How did you guess? Or what do multiple choice questions measure? Medical Journal of Australia, 1, 884–886.

Dede, C. (2004). Planning for neomillennial learning styles: implications for investments in technology and faculty. In D. Oblinger, & J. Oblinger (Eds.), Educating the net generation (pp. 15.1–15.22), Educause.

D'Souza, S. M., & Wood, L. N. (2003). Designing assessment using the MATH taxonomy. In L. Bragg, C. Campbell, G. Herbert, J. Mousely (eds.), Mathematics Education Research: Innovation, Networking, Opportunity. Proceedings of the 26th Annual Conference of MERGA Inc., (pp. 294-301), Deakin University, Geelong, Australia.

Fishbein, M. , & Ajzen, I. (1975), Belief, attitude, intention and behavior: an introduction to theory and research. Publisher: Reading, MA: Addison-Wesley.

Hargittai, E. (2010). Digital Na(t)ives? Variation in internet skills and uses among members of the "Net Generation". Sociological Inquiry, 80(1), 92–113.

Hosein, A.; Ramanau, R. and Jones, C. (2010, July 26-29). Are all net generation students the same? The fre-quency of technology use at university. Paper presented at the IADIS E-Learning Conference, Freiberg, Germany.

Johnstone, A. H., & Ambusaidi, A. (2000). Fixed response: what are we testing? Chemistry Education: Research and Practice in Europe, 1(3), 323–328.

Jones, C., Ramanau, R., Cross, S. J., & Healing, G. (2010). Net generation or digital natives: Is there a distinct new generation entering university? Computers in Education, 54 (3), 722–732.

Kennedy, G. E., Krause, K.-L., Judd, T. S., Churchward, A., & Gray, K. (2008). First year students' experiences with technology: Are they really digital natives? Australasian Journal of Educational Technology, 24 (1), 108–122.

Mcnaught, C., Lam, P., & Ho, A. (2009). The digital divide between university students and teachers in Hong Kong. In Atkinson, R. J. & McBeath, C. (Eds) Same places, different spaces. Proceedings ascilite Auckland of 26th Annual ascilite International Conference,(pp. 654–664). Auckland, Publisher: The University of Auckland, Auckland University of Technology, and Australasian Society for Computers.

Oblinger, D., & Oblinger, J. (2005). Is it age or IT: first steps towards understanding the net generation. In D. Oblinger, & J. Oblinger (Eds.), Educating the Net Generation (pp. 2.1–2.20), Educause,

Palfrey, J., & Gasser, U. (2008). Born digital: Understanding the first generation of digital natives. New York: Basic Books.

Prensky, M. (2001). Digital natives, digital immigrants: do they really think differently? On the Horizon, 9(6), 1–6.

Ramsden, P. (1992). Learning to Teach In Higher Education London, Routledge.

Smith, G., Wood, L., Coupland, M., Stephenson, B., Crawford, K. & Ball, G. 1996. Constructing mathematical examinations to assess a range of knowledge and skills. International Journal of Mathematical Education in Science and Technology, 27(1), 65-77.

Scouller, K. (1998). The influence of assessment method on students' learning approaches: multiple choice question examination versus assignment essay, Higher Education, 35, 453–472.

Authors

Mária BAKÓ, University of Debrecen, Faculty of Economics and Business, Debrecen (Hungary). E-mail: bako.maria@econ.unideb.hu

László ASZALÓS, University of Debrecen, Faculty of Informatics, Debrecen (Hungary). E-mail: aszalos.laszlo@inf.unideb.hu