# Test items for and misconceptions of competences in the domain of logic programming

Barbara Linck

Didactics of Informatics and E-Learning, University of Siegen, Siegen, Germany

barbara.linck@uni-siegen.de

**Abstract.** Development of competence-oriented curricula is still an important theme in informatics education. Unfortunately informatics curricula, which include the domain of logic programming, are still input-orientated or lack detailed competence descriptions. Therefore, the development of competence model and of learning outcomes' descriptions is essential for the learning process in this domain. A prior research developed both. The next research step is to formulate test items to measure the described learning outcomes. This article describes this procedure and exemplifies test items. It also relates a test in school to the items and shows which misconceptions and typical errors are important to discuss in class. The test result can also confirm or disprove the competence model. Therefore, this school test is important for theoretical research as well as for the concrete planning of lessons. Quantitative analysis in school is important for evaluation and improvement of informatics education.

**Keywords:** competence, test items, misconceptions, informatics in upper secondary education, logic programming

## 1 Introduction

Development of competence-oriented curricula is still an important theme in informatics education. E.g., the IFIP Working Group 3.2 analyzed various curricula for informatics. They concluded that it is important for curricula to be well designed and to allow a "constructive approach with an open space for interpretation and effective knowledge transfer." [1, p. 189-190]. Hence, the competence orientation in informatics curricula is fundamental. Some frameworks refer to learning outcomes for informatics education e.g., [2-3] explain one of them. However, they do not include competences in logic programming. Logic programming is a theme of ten informatics curricula for upper secondary school in Germany. Unfortunately, these curricula are still input-oriented or lack detailed competence descriptions. For teachers the learning outcomes in informatics curricula are essential. They are the basis for defining which competences the students are supposed to learn. Especially, „During the preparation process the teacher might examine, for example, a section of a GCSE syllabus and consider exactly what are the key concepts and skills to be taught." [4, p. 245] There-

fore, the development of learning outcomes' descriptions is important for the learning process in the domain of logic programming.

A prior research established a competence model for the domain of logic programming [5]. Based on this model, competence descriptions were formulated [6]. The next research step is to develop test items. These items are necessary to measure the described learning outcomes. In addition, this measurement can confirm or disprove the competence model and the competence descriptions. This article describes the method of the items' development, exemplifies chosen test items, and relates the results of a test in school to the items. Furthermore, the answers of the test show which misconceptions and typical errors are important to discuss in class.

## 2 Test items and misconceptions

Regarding to [7, p. 41] a framework, which describes the extent of the domain, is supposed to be a basis for test measurement. This framework delineates which aspects of the content, skills, processes etc. of the domain should be measured. Therefore, the developed competence model with competence descriptions for the domain of logic programming is the basis for the measurement. Moreover, the test format and the target group have to be taken into account while constructing the test. The following sections describe the test format, examples of the test and results out of it.

### 2.1 Method to develop test items

There is a wide variety of test formats. E.g., multiple-choice questions, short answers, coding exercises, essays and filling-the-blank questions are possible. According to Bacon multiple-choice (MC) answers are as reliable as short-answer tests, "but an MC exam may be completed quickly." [8, p. 35] Haladyna et al. [9] presents a revised taxonomy of writing multiple-choice items. Among other things it is important to base each item on important content and to avoid trivial content. Moreover, characteristics of a well-written test are that the content of each item is independent from contents of other items, that the vocabulary is simple for the group of students and that the length of the choices is equal. Furthermore, it is important to minimize the amount of reading, to develop as many effective choices as possible - but three are adequate - and to use typical errors of students to write distractors. Kuechler and Simkin analyzed that multiple-choice questions "enable test takers to better guess correct answers compared to constructed response test" [10, p. 394] for the domain of programming. However, they showed in another article "that carefully constructed MC questions on the topic of basic computer language programming can test a broad range of levels of understanding of that subject." [11, p. 79] Therefore, it is very important to choose the distractors wisely. Furthermore, the students are supposed to decide whether the answers are chosen or not chosen. This allows a distinction between not chosen answers and missing statements. In addition, the test should include open answers as well. A described context, in which the learning outcome is supposed to be applied, is important for the competence orientation [12].

With regard to these demands and to the taxonomy of Anderson and Krathwohl [13] test items were developed and tested in a pretest in school. In addition, the competence descriptions were a basis for the test items. With respect to the pretest, in some cases it was necessary to modify the phrasing of the item. Moreover, appropriate items were chosen out of the collection of the items. This modified test was tested in school again and is described in this article.

### 2.2 Test group and test conditions

Informatics is often a facultative subject in Germany. Therefore, in the most cases only a few students take this course. The test items were tested in three different classes. Three students were in class A, six students in class B and 16 students in class C. Classes A and B were a basic informatics course with three 45-minute lessons per week. Class C was an advanced informatics course with five 45-minute lessons per week.

The test is quantitative, took 45 minutes and was handed out by the teacher. The following sections present and analyze two different test items. Both sections exemplify the test items, the analysis of the test results and the identification of misconceptions as well as typical errors.

### 2.3 Multiple-choice questions: selection of implementation

As shown in [6], one competence is to analyze a given problem and to implement it into the knowledge base. The competence description is: "The learners are able to analyze a given problem. They can implement it into facts and rules. Therefore, they are able to create different possible implementations. In addition, they can compare various implementations and select the best one." [6, p. 8] Item number two tests the comparision and selection of different implementations. The context of the item is that four friends talk about their holidays and in which countries each of them has been on holiday. Four possible implementations (see Table 1, answer 2a-2d) of this situation are given. The students are supposed to select the implementations, which can be used to answer the following two questions:

- Who has been to the United Kingdom?
- How many countries has Florian (one of the friends) been to?

In addition, the students are not supposed to use system predicates e.g. findall/3 to formulate the queries. Especially the students in the classes A and B do not know these system predicates.

The right answer for this task is to choose the last implementation 2d. In Prolog – the logic program that is used in the three classes – the matching of the variables (at the head of the rule to the asked query) requires the knowledge of the position of the variables. In the given implementation two and three, the same country, e.g. the United Kingdom, has different positions. Therefore, they are not possible knowledge bases for answering the two questions. The first implementation 2a can be used to an-

swer the first question. Nevertheless, without the system predicate findall/3 it is not possible for the students to answer the second question.

**Table 1.** Possible implementations

| *chosen* | *not chosen* | | |
|---|---|---|---|
| | | holiday(katrin, united_kingdom).<br>holiday(katrin, france).<br>etc. | 2a |
| | | holiday(katrin, united_kingdom, france, spain).<br>holiday(florian, portugal, united_kingdom, netherlands).<br>etc. | 2b |
| | | holiday_katrin(united_kingdom, france, spain).<br>holiday_florian(portugual, united_kingdom, netherlands).<br>etc. | 2c |
| | | holiday(katrin,[united_kingdom, france, spain]).<br>holiday(florian,[portugual, united_kingdom, netherlands]).<br>etc. | 2d |

The misconception of implementation 2a is that Prolog knows how many facts and rules belong to one predicate. Table 2, 3 and 4 show the distribution of the answers. The students who chose “false” give the right answer since it is not a possible implementation as already mentioned.

With regard to these tables this misconception should be discussed in class. Only 56 % of all students knew that this implementation could not be used to answer the two given questions. The students of the advanced informatics course were only slightly better with 56.25 % than the students of the basic informatics course with 55.56 %. Therefore, this misconception is very important for both courses.

**Table 2.** Answer 2a, all students

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 14 | 56 | 70 |
| right | 1 | 6 | 24 | 30 |
| missing statements | 99 | 5 | 20 | missing statements |
| | total | 25 | 100 | 100 |

**Table 3.** Answer 2a, basic informatics course

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 5 | 55.56 | 100 |
| right | 1 | 0 | 0 | 0 |
| missing statements | 99 | 4 | 44.44 | missing statements |
| | total | 9 | 100 | 100 |

**Table 4.** Answer 2a, advanced informatics course

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 9 | 56.25 | 60 |
| right | 1 | 6 | 37.50 | 40 |
| missing statements | 99 | 1 | 6.25 | missing statements |
| | total | 16 | 100 | 100 |

Moreover, this part of the test item can be used in a following test again. Data analysis presents that the answers 2a and 2d as well as 2c are not too similar and do not have to be changed. E.g., data shows that 24 % of the students chose answer 2a. Therefore, students do not skip the item immediately. This confirms that this item does avoid trivial content and includes typical errors as a distractor. Both demands are important for the development of test items [9]. 40 % of the students, who knew that answer 2d is a possible implementation, did not choose 2a as a right implementation. Hence, the misconception of 2a does not depend on answer 2d. In addition, 48 % of the students, who chose 2b as a wrong answer, did not choose 2a as a right answer. The same occurs with answers 2c and 2a. Therefore, the misconceptions of 2b and 2c are independent of the misconception of 2a. But only 8 % of the students, who chose answer 2c as a wrong answer, did not choose 2b as a wrong answer. As a result, answers 2b and 2c do not show enough reliability. For a repetition of the test one of these two answers should be erased or replaced by another misconception.

### 2.4 Open answer of implementation: connecting predicates with each other

For the implementation of a given problem in facts and rules it is also necessary to use system predicates. Another important competence is to implement a rule, which accesses different predicates. The following item tests both. The described situation is a pizzeria, which has certain pizzas, pastas, desserts, a soup, two salads and a meat dish on its menu. The task is to implement a rule, which creates a menu with an appe-

tizer, a main dish and a dessert. In addition, the rule is supposed to display the price of the menu. This task demands an open answer. Table 5, 6, and 7 show how many students were able to implement a rule, which accesses the three predicates *appetizer*, *main dish* and *dessert*.

**Table 5.** Several predicates, all students

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 1 | 4 | 4.17 |
| right | 1 | 23 | 92 | 95.83 |
| missing statements | 99 | 1 | 4 | missing statements |
| | total | 25 | 100 | 100 |

**Table 6.** Several predicates, basic informatics course

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 1 | 11.11 | 12.5 |
| right | 1 | 7 | 77.78 | 87.5 |
| missing statements | 99 | 1 | 11.11 | missing statements |
| | total | 9 | 100 | 100 |

**Table 7.** Several predicates, advanced informatics course

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 0 | 0 | 0 |
| right | 1 | 16 | 100 | 100 |
| missing statements | 99 | 0 | 0 | missing statements |
| | total | 16 | 100 | 100 |

This analysis shows that 95.83 % of the students understood the described situation as well as the formulation of the tasks and gave the right answer. Therefore, this task does not need to be rephrased. Furthermore, a performance difference between the basic and the advanced informatics course is obvious. All students of the advanced course gained this competence, whereas 87.5 % of the basic course gave the right answer. The latter percentage still confirms the test item. As already mentioned, this

item also includes the competence to use system predicates. Table 8, 9 and 10 show how many students are able to implement the system predicate *is*.

**Table 8.** System predicate, all students

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 18 | 72 | 75 |
| right | 1 | 6 | 24 | 25 |
| missing statements | 99 | 1 | 4 | missing statements |
| | total | 25 | 100 | 100 |

**Table 9.** System predicate, basic informatics course

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 5 | 55.56 | 62.5 |
| right | 1 | 3 | 33.33 | 37.5 |
| missing statements | 99 | 1 | 11.11 | missing statements |
| | total | 9 | 100 | 100 |

**Table 10.** System predicate, advanced informatics course

| *value label* | *value* | *frequency* | *percent* | *valid percent* |
|---|---|---|---|---|
| false | 0 | 13 | 81.25 | 81.25 |
| right | 1 | 3 | 18.75 | 18.75 |
| missing statements | 99 | 0 | 0 | missing statements |
| | total | 16 | 100 | 100 |

According to these tables only 25 % of the students used the system predicate *is* in the right way. 37.5 % of the students in the basic informatics course achieved this competence. Compared to this, only 18.75 % of the students in the advanced informatics course gave the right answer. The test shows that most of the students in the latter course, who gave a wrong answer, mistook the system predicate = for the system predicate *is*. Thus, the misconception between both system predicates should be discussed with more examples in this course.

The result of the test is that the students did not use the system predicate *is* in the correct way. However, this does not imply that these students did not achieve the

learning outcome, which was supposed to be tested. Especially, the output orientation demands to have a closer look at what is tested. This test item shows that these students did not remember the system predicate *is*. The conclusion is not that they cannot use any system predicate correctly. Therefore, another test item should be added to the test. It would be interesting to analyze if the students are able to use a given system predicate in the correct way and, in addition, if they are able to remember as well as use another system predicate.

### 2.5 Test result: misconceptions

As shown in 2.3 the test items use misconceptions as distractors. Furthermore, the open answers in the test include typical errors and can be analyzed as well. 2.4 exemplifies this. With regard to all test items, the students avoided or made the following misconceptions and typical errors (see Table 11). The percentage refers to the valid percentage.

**Table 11.** Misconceptions of logic programming and test results

| *misconception/typical error* | *shown by the students* |
|---|---|
| Only one of the arguments at the head of the rule has to match the query. | 0 % |
| Wrong structure of the clause tree. | 4 % |
| The position of the argument at the head is not important. | 5 % |
| The system answers of *rule(A1,A2)* is *rule(a1,a2)* instead of *A1=a1, A2=a2*. | 12 % |
| Wrong order of the answers (first system answer as first answer in the test). | 29 % |
| Anonyms variable ("_") can give an answer with a concrete instantiation. | 30 % |
| Mistake the meaning of predicate and clauses. | 44 % |
| No point at the end of each fact or rule. | 44 % |
| Mistake the meaning of *is* and =. | 46 % |
| Atoms can be written with a big initial letter. | 68 % |
| Prolog knows how many facts and rules belong to one predicate. | 70 % |

Therefore, the test result can be used to improve the teaching of logic programming. The identified misconceptions and typical errors should be discussed in school.

Another interesting test result is that the students of the advanced informatics course were able to transfer their knowledge. The test included a system predicate to

modify the knowledge base dynamically. Although, the students did not learn the dynamic modification yet, 81.25 % of the students of the advanced course answered this question. 100 % of those students got the structure of the system predicate and the used arguments right. 61.54 % of them knew the system predicate. Most of them used the predicate *delete*, some used the predicate *retract*. It can be assumed that the students transferred the predicate *delete* from another programming language and matched it to the structure of Prolog. This hypothesis would be interesting to test in a further research.

## 3 Conclusion

In summary, this article shows that the distractors of the developed test items are well chosen. Only slight changes should be done for a next test. In some cases one distractor should be erased or replaced. In other cases additional test items can support the testing of the competences. Therefore, this test result can be used to conclude first findings referring to the competence model of logic programming. This will be a next research step.

Based on the test result, this article identifies also misconceptions and typical errors of logic programming. These misconceptions and errors are essential for the learning progress. Students should interact with them critically. This is fundamental for a deeper understanding of the subject. Hence, they are also important for the design of the lessons. The more often a misconception appears, the more it should be discussed in school. Even if a student does not make the typical error her- or himself, the understanding of it supports e.g., the understanding of the structure of facts and rules.

Therefore, this school test is important for theoretical research about competences in logic programming as well as for the concrete planning of lessons. Quantitative analysis in school also referring to other subjects is important for evaluation and improvement of informatics education.

## 4 References


1. Veen, M. van, Mulder, F., Lemmen, K.: What is Lacking in Curriculum Schemes for Computing/Informatics. In: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, pp. 186–190, ACM, New York (2004)
2. Antonitsch, P.: On Competence-Orientation and Learning Informatics. In: Proceedings of 5th International Conference ISSEP, CD-ROM, Bratislava (2011)
3. Micheuz, P.: A Competence-Oriented Approach to Basic Informatics Education in Austria. In: Proceedings of 5th International Conference on Informatics in Schools, Springer, Heidelberg, pp. 43–55 (2011)
4. Webb, M.: Pedagogical Reasoning: Issues and Solutions for the Teaching and Learning of ICT in Secondary Schools, In: Education and Information Technologies 7(3), pp. 237–255, (2002)
5. Linck, B., Schubert, S.: Logic programming in informatics secondary education. In: Proceedings of 5th International Conference ISSEP, CD-ROM, Bratislava (2011)

6. Linck, B.: Competence Descriptions for Informatics Education – using the example of logic programming. In: Proceedings of IFIP-Conference "Addressing educational challenges: the role of ICT (AECRICT)", Manchester Metropolitan University (2012)
7. American Educational Research Association, American Psychological Association, National Council on Measurement in Education: Standards for educational and psychological testing. American Psychological Association, Washington DC (1999)
8. Bacon, D.: Assessing Learning Outcomes: A Comparison of Multiple-Choice and Short-Answer Questions in a Marketing Context. In: Journal of Marketing Education 25(31), pp. 31–36 (2003)
9. Haladyna, T., Dowing, S., Rodriguez, M.: A review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment. In: Applied measurement in education 15(3), pp. 309–334, Lawrence Erlbaum Associates, Inc. (2002)
10. Kuechler, W., Simkin, M.. How Well Do Multiple choice Tests Evaluate Student Understanding in Computing Programming classes? In: Journal of Information Systems Education 14(4), pp. 389–399 (2003)
11. Simkin, M., Kuechler, W.: Multiple-Choice Tests and Student Understanding: What Is the Connection? In: Decision Science Journal of Innovative Education 3(1), pp. 73–97 (2005)
12. Weinert, F.E.: Concept of Competence: A Conceptual Clarification. In: Rychen, D., Salganik, L. (eds.): Defining and Selecting Key Competencies. Seattle, (2001)
13. Anderson, L., Krathwohl, D.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York (2001)