Bayesian Networks For Competence-based Student Modeling

Nguyen-Thinh LE & Niels PINKWART

Humboldt Universität zu Berlin
Department of Informatics
Germany

nguyen-thinh.le@hu-berlin.de

Abstract: Adaptive learning technologies have been demonstrating effective by many types of adaptive educational systems (e.g., intelligent tutoring systems, adaptive hypermedia systems, adaptive assessment systems). NMC Horizon Report 2015 predicts that this kind of learning technologies would be deployed widely in higher education in four to five years. Recently, there is a trend of shifting the measurement of student's performance from a knowledge-based model to a competence-based model. However, in literature, most adaptive educational systems employ student's knowledge to build student models. In this paper, we propose to integrate competences in student models for adaptive learning technologies. We use Bayesian Networks to model student's competences in addition to student's knowledge. We describe a case study in the domain of object-oriented programming, which makes use of the proposed competence-based Bayesian Network model.

Keywords: competence, student modeling, adaptive educational systems, adaptive hypermedia systems, intelligent tutoring systems, adaptive assessment systems

1. Introduction

Adaptive learning technologies provide a mechanism to adjust to individual students' needs as they learn. According to NMC Horizon Report (NMC, 2015), adaptive learning technologies are in trend and would be deployed widely in higher education on the horizon of four to five years. In order to be able to adjust instruction according to individual needs, adaptive learning technologies use information about the student to build a student model. In general, information about an individual student can be classified into two categories (Holden, 2013): domain-specific and domain-independent information. Domain-specific information includes e.g., student's knowledge, skills, misconceptions, and problem solving strategies, etc. that reflect the student's state of knowledge or student's ability within a particular domain. Domain-independent information consists of relevant characteristics of an individual student, e.g., learning goals, cognitive aptitudes, affective states, learning preferences (including learning styles and personality), interest, demographics, past performance, behavioral/psychological measures, and personal control beliefs (self-efficacy, locus of control). Among the types of domain-specific information, student's knowledge is employed by most adaptive hypermedia systems and adaptive educational systems for building student models (Brusilovsky & Millán, 2007). Holden (2013, p. 71) stated that the first-generation intelligent tutoring systems mainlly adapted instruction based on student's performance and student's knowledge.

Recently, most curricula on the school level are changing to be competence-based (Drieschner, 2009). That is, a curriculum for each specific subject specifies competences that each student has to acquire after attending a semester or a school year. Thus, progress measurement has been being shifted from a knowledge-based model to a competence-based model. The concept "competence" is diversely defined in literature. In this paper, we adopt the definition of competence developed by Weinert (2001). According to Weinert (2001, p.27, original in German), "a competence is the existence of learnable cognitive abilities and skills which are needed for problem solving as well as the associated motivational, volitional and social capabilities and skills which are needed for successful and responsible problem solving in variable situations". Through this definition, we can see that acquiring knowledge is just a first step to build a competence. In order to build a competence, a student needs to understand a concept, master it through many problem-solving tasks, and apply it in different situations. One may have sufficient knowledge about a domain, but may not have the competence to perform a task in a new situation (e.g., using a new learning environment). The labor market expects that their graduates are not only equipped with sufficient knowledge but also they should have competences in a specific domain[1].

Yet, most existing adaptive educational systems do not support the aspect of developing required competences of students (that is, the capability of applying acquired knowledge in variable situations) as Nitchot and colleagues (Nitchot et al., 2010) reported. Few research works have been developing frameworks for competence-based adaptive educational systems, e.g. Hnida et al. (2014), Nussbaumer et al. (2010), Nitchot et al. (2010). However, these approaches to building competence-based adaptive educational tools use their own specific technique for representing competences.

With respect to the approaches to student and domain modeling, the Bayesian Network approaches has been being used widely in adaptive educational systems. However, most of existing systems deployed Bayesian Networks for modeling student's knowledge (Brusilovsky & Millán, 2007; Desmarais & Baker, 2011). To our best knowledge, no research on modeling student's competences using Bayesian Networks has been attempted.

The goal of this paper is to propose to use Bayesian networks to model competences in addition to knowledge. The remainder of this paper is structured as follows. In Section 2, we review the state of the art of competence-based adaptive educational systems and the Bayesian Network modeling approach. In Section 3, we propose to use Bayesian networks to develop competence-based student and domain models. In Section 4, we describe a study case in the learning domain of object-oriented programming. We summarize our conclusions in Section 5.

2. The State of the Art

2.1. Competence-based adaptive educational systems

In the state of the art, there are few attempts to building competence-based adaptive educational systems. Nitchot and colleagues (Nitchot et al., 2010) proposed to develop a competence-based model, which is exploited to recommend appropriate study materials from the Web. According to Nitchot and colleagues, a competence is composed of a context and intended learning outcomes. Each learning outcome is the composition of capability and subject matter. Capability indicates what the learner will be able to do with the subject matter. The competence-based model deploys a directed acyclic graph of competences. Each node of the graph represents a competence. Each arc shows the relationship between two competences. This competence-based model is used both to model required competences of a specific domain and to model students. The differences between the two graphs (one of the domain model and another one of the student model) indicate the gap of competences that a student needs to acquire.

Hnida and colleagues (Hnida et al., 2014) proposed an ontology-based representation to specify competences in student models, which underlie their adaptive e-learning system. The developed ontology of Hnida and colleagues is relatively complex. Among the important concepts, the ontology includes learning situation, learning activity, assessment, knowledge, trajectory of a learner, competence, ability, etc. This ontology has been developed to be integrated in an adaptive e-learning system.

Nussbaumer and colleagues (Nussbaumer et al., 2010) proposed to apply the competence-based Knowledge Space Theory (CbKST) for representing domain knowledge and the current state of student's competence and knowledge in adaptive learning systems. CbKST is a psychological framework for representing knowledge and competence states of learners. Originally, the CbKST framework (Albert & Lukas, 1999) defines formalism to model domain knowledge in terms of a set of problems. The knowledge state of a student is the set of problems that can be solved by him/her. Between the problems, there may be relationships that represent the prerequisite(s) of each problem. Representing a student's knowledge state as a subset of observable information (solved test

[1] In the industry, people use the term "competency" to refer to description of skills and knowledge that are required for a job or a specific task. Here, in this paper, for simplicity, we use the term "competence".

item/problem) is similar to the overlay approach (Woolf, 2009). Heller et al. (2006) extended this framework with the concept of competence. The theory assumes that a set of fine-grained skills that are required for solving problems exist. If a student has a subset of skills, this subset of skills represents the competence state of this person. The relationship between competence and observable performance is established through mappings between skills and problems of a domain. This competence framework has been proposed for competence-based testing (Nussbaumer et al., 2010). The Knowledge Space Theory indicates which knowledge/competence states can be reached from a given knowledge/competence state based on the surmise relations among items. Surmise relationships represent the prerequisite between problems. These surmise relationships recommend a student to solve simpler problems before moving on to more complex ones. We note that the special feature of student models based on knowledge spaces is that they contain only observable information of the student (e.g., solved test items/problems).

2.2. Bayesian Networks for Student Modeling

In order to allow adaptive support for individual students, adaptive educational systems require a student model for each individual student and a domain model. Student models are necessary to adapt individual instruction (or learning materials) to particular students. The task of student modeling is associated with the problem of uncertainty. For example, after how many successful opportunities should we consider a skill mastered or a competence is acquired? The sources for uncertainty may be various. For instance, a slip can happen when a student accidentally makes an error, or a student may construct a correct solution (or perform a correct action) by chance (Baker et al., 2008). Bayesian network is an AI technique, which can be used for dealing with the uncertainty issue in the task of modeling students (Woolf, 2009).

Bayesian Networks are widely used in adaptive educational systems. Intelligent tutors deploy Bayesian Networks to support classifying and predicting student knowledge, to predict student behavior, to make tutoring decisions, and to determine on which steps students will need help and their probable method for solving problems (e.g., Andes (Gertner et al., 1998), Bayesian Knowledge-Tracing (Desmarais & Baker, 2012), Bayesian Networks for constraint-based tutors (Mayo & Mitrovic, 2001)). For the purpose of adaptive curriculum sequencing, Bayesian Networks can be used to represent skills / knowledge in a domain (e.g., Al-Muhaideb & Menai, 2011). The adaptive curriculum sequencing systems decide among alternatives, within a probabilistic model of student knowledge and goals, to recommend next problem to be solved. Several adaptive testing and assessment systems also deploy Bayesian Networks to recommend individual test items (e.g., Vomlel, 2004; Almond et al., 2015).

While the reviewed competence models have been developed based on specific modeling approaches (e.g. directed acyclic graph of competencies, ontological representation, and Knowledge Space theory), we propose to apply the well-known overlay modeling and Bayesian Network approaches.

Although the Bayesian Network modeling approach is established and has been widely used for domain and student modeling in various adaptive educational systems (Brusilovsky & Millán, 2007; Desmarais & Baker, 2011), most of them employed knowledge-based models. In this paper, we propose to apply the Bayesian Network approach to modeling student’s competences in addition to student’s knowledge. That is, if students have acquired some specific competences, they should be recommended with new other challenging tasks to acquire new competences.

3. Bayesian Networks for Modeling Student’s Competences

In this section, we develop a Bayesian Network model for competence-based student and domain modeling. The process of building Bayesian Networks consists of three steps. The first step is defining the structure of a Bayesian Network, i.e., its topology. This means establishing the nodes and arcs of the network. Typically, for building an expert-centric network, the structure comes from the domain expert who decides which topics or skills are dependent on other skills.

The second step is initializing the estimate values of student knowledge and the assessment of student learning. Initialization of the Bayesian Network greatly influences the manner in which the network is updated on the basis of current students. Given that student interaction with tutors lasts only a short time and experience with a single student is limited, the initialization and updating processes are important in establishing enough information to make inferences about a single current student. Estimates of student knowledge can be initialized in various ways, including educated guesses (experts provide their best guess), average values from past student use, and estimates based on student performances (test results).

The third step is updating probabilities in a Bayesian Network. Using past and current information (based on interaction between students and the intelligent tutor), the Bayesian machine learning technique estimates and updates probabilities of a Bayesian Network.

Bayesian Networks are useful for student modeling due to its high representative power and an intuitive graphical representation. In this paper, we focus on the first step of developing Bayesian Networks: defining the structure of a Bayesian Network.

A Bayesian Network student model consists of nodes and arcs. A node represents a variable that can be related to the learning process, e.g. a unit of knowledge (K), a competence (C), a learning event (P) (i.e., an answer to a test item, a solution of a problem). Each learning event node has a value (e.g., "solved" or "not solved", "correct" or "incorrect"). Each arc of the network represents a relationship, e.g., the arc between the node "knowledge" and the node "learning event" means that the state of knowledge influences the result of the event. We use two types of relationships: relationship between a competence and a knowledge unit and relationship between a knowledge unit and an evidential node. The first relationship type represents the association between a knowledge unit and a competence. The relationship between knowledge unit and an evidential node depicts knowledge that will be acquired if the student has performed a learning event.

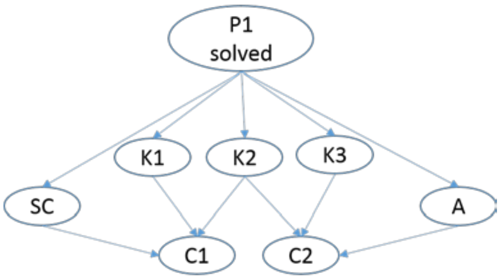

Figure 1. Bayesian Network for competence-based student modeling.

Figure 1 illustrates these two types of relationships. If the problem P1 is solved, it is probable that the student has acquired knowledge K1, K2, and K3. If knowledge units K1, K2 have been acquired, it is probable that the student has the competence C1. If knowledge units K2 and K3 have been acquired, it is probable that the student has the competence C2. This Bayesian Network model for competences includes both knowledge and competences, because according to the psychological framework for competences (Albert & Lukas, 1999; Heller et al., 2006), a student may have both knowledge state and competence state (that is defined by a sub-set of skills).

In addition to modeling cognitive aspects of the student, social competence and affective states of students can also be considered in this model. For example, we assume that a collaborative learning environment requires two or more students to solve a problem collaboratively. If the problem P1 is solved by the collaboration between several students, it is probable that each student has the social competence (SC), and a specific affective state A (e.g., "happiness", because a problem has been solved). These features are considered in this Bayesian Network model, because according to Weinert (2001), in addition to cognitive abilities and skills, students require social and motivational capabilities in order to solve problems successfully. The evidence about student's affective states and social capabilities can be enhanced by e.g., analyzing facial data through a camera (e.g., Kaliouby & Robinson, 2004) and analyzing the collaboration between students through collected interaction data (e.g., Martinez-Maldonado et al., 2013).

The relationships between the nodes in a Bayesian Network can be explained in two directions (Brusilovsky & Millán, 2007): causal direction or diagnostic direction. Considering the relationships between a knowledge unit K and a problem P, relationships can be: 1) modeled in the causal direction ($K \rightarrow P$) or 2) in the diagnostic direction ($P \rightarrow K$). In the case of modeling the relationships between a competence and a knowledge unit, the causal relationship between a specific knowledge K and a competence C (K → C) can be interpreted that knowing a component indicates the possibility to have a competence. In the diagnostic direction (C → K), the relationship shows that having a competence means having associated knowledge and the knowledge node can be used as a measure of how well the student has solved a problem.

Student modeling and domain modeling are two sides of a coin. The function of a domain model is to provide a framework for representing and measuring student's knowledge and competences of a learning domain. The overlay model works according to the following principle (Woolf, 2009). For each domain model concept [2], individual student's model stores some data (e.g., knowledge or competence) that is an estimation of the student's knowledge and competence of this concept. In the original form, an overlay model uses a binary value to represent student's knowledge (mastered or not mastered) as an overlay of the domain model. This original form of overlay models was widely applied in many adaptive educational systems (e.g., Beaumont, 1994; Gonschorek & Herzog, 1995; Pérez et al., 1995). The overlay knowledge models were accepted as a standard technology for student modeling in adaptive educational and non-educational hypermedia systems (Brusilovsky & Millán, 2007) due to the intuitiveness of this approach.

Brusilovsky and colleagues (Brusilovsky et al., 2005) have developed many adaptive educational and hypermedia systems applying the so-called concept-based modeling mechanism for adaptive task sequencing and navigation (e.g., Hsiao et al., 2009). The concept-based models aim to connect domain knowledge with educational materials (educational problems, questions, examples, presentation pages of lessons, etc.). The process of connecting domain knowledge with an educational element is referred to as indexing. That is, each educational element is indexed with a set of underlying concepts. This process consists of two stages: First, concepts are extracted from educational material. Then, on the second stage, extracted concepts are divided into two groups: prerequisites and outcomes. Prerequisites are concepts, which have to be mastered before working with a new educational material, and outcomes are concepts to be learned after working with that educational material. Prerequisite and outcome concepts of each educational element are identified based on a sequence of learning goals defined by the instructor. Note, the notion of prerequisite relationships between problem nodes was initially introduced by Doignon and Falmagne (1985) in the Knowledge Space Theory. Adopting this concept-based mechanism, we add prerequisite relationships between the problem nodes in our Bayesian Network to model a learning domain.

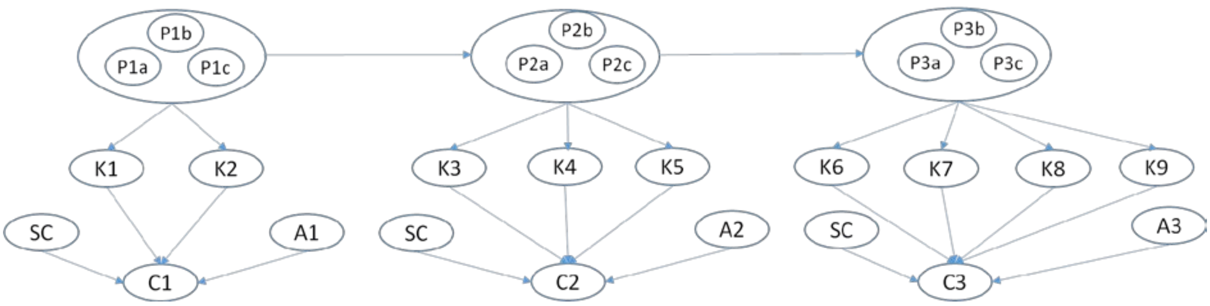

Figure 2. Bayesian Network for domain modeling.

Figure 2 illustrates a Bayesian Network domain model. It shows that solving one of the problems P1 (P1a, P1b, P1c) is the prerequisite for solving the problems P2 (P2a, P2b, P2c). Accordingly, solving one of the problems P2 (P2a, P2b, P2c) is the prerequisite for solving the problems P3 (P3a, P3b, P3c). We also note that the nodes that represent the groups of problems (e.g., P1a, P1b, P1c) and denote that these problems are similar and can be grouped in a same class. In this model, we consider different affective states (A1, A2, A3) of students that may lead to successful problem solving. Several models of affective states for students have been developed (Sabourin et al., 2011; Calvo & D'Mello, 2010). With respect to student's social competence, we just only have one node SC that

[2] The term "concept" may denote coarse-grained chunk of knowledge (e.g., "topic") or fine-grained piece of knowledge (e.g., "knowledge item", "knowledge element", "learning objective")

describes that a student has social capability to solve problems and this social capability is one of the factors leading to acquiring different competences C1, C2 or C3.

4. A Case Study

We plan to apply the proposed competence-based Bayesian Network model in the domain of object-oriented programming. We focus on the cognitive aspect of the students. That is, knowledge and competences that are domain-specific information about a student (Holden, 2013) will be modeled. We adopt the competence structure model for Computer Science education on the higher education level (Le & Müller, 2015). Le and Müller applied a three-step methodology to develop the competence structure model[3]. First, the content of existing Computer Science curricula of selected universities on the basis of the topic categories provided by the ACM/IEEE recommendations is analyzed. Second, a survey is conducted to find out the expectations on graduates in the identified learning domains. Third, results of the survey are combined with the results of the curricula analysis to gain a consolidated competence model in the topic area. Based on the consolidated competence model, ten dimensions, each describes a relevant category for Computer Science, have been identified: fundamental programming concepts, fundamental data structures, algorithm and design, software engineering processes, requirements analysis, software construction and design, development methods, test methods, tools and environment, software project management.

Table 1. Competences for curricula of higher education in Software Engineering

C1	**Fundamental programming concepts**
C1.1	Know & apply fundamental programming constructs
C1.2	Know syntax and semantic
C1.3	Know & apply basic concepts of programming languages
C1.4	Know & use primitive data types
C1.5	Explain & implement small programs
C1.6	Know program execution
C1.7	Know & apply input and output techniques (streams)
C2	**Fundamental data structures**
C2.1	Know object references (pointer)
C2.2	Know & apply fundamental data structures
C2.3	Know & apply graphs and trees
C2.3	Know & apply abstract data types
C3	**Algorithm and design**
C3.1	Know, identify & apply problem solving strategies
C3.2	Know algorithmic strategies
C3.3	Use big O-notation
C3.4	Know & apply traversals
C3.5	Know basic algorithms (sorting & searching)
C3.6	Know & apply recursion
C.37	Awareness and identification of problems

[3] Le & Müller (2015) distinguished between competence structure model and competence level model. The competence structure models can show coherences between the accomplishments of different requirements. Competence level models can describe the stages of competences students should accomplish.

In this paper, we only consider the three dimensions of Computer Science that are related to Software Engineering (Table 1): C1 (Fundamental Programming Concepts), C2 (Fundamental Data Structures), and C3 (Algorithm and Design).

Using these competences, we build a Bayesian Network for the domain of object-oriented programming. Figure 3 illustrates the competence-based Bayesian Network for a part (some competences of the competence dimension C1 "Fundamental programming concepts") of the domain of object-oriented programming. After solving one of the P1 problems (P1a, P1b, or P1c) successfully, the student would have acquired the knowledge e.g., programming constructs "WHILE" and "IF-ELSE", and it is probable that the student has the competence C1.1 (Know and apply fundamental programming constructs). While solutions for P1a are required to use a "WHILE" loop, solutions for P1b and P1c require the "IF-ELSE" construct. Groups of similar programming assignments can be illustrated by the following examples. The sample solutions for these programming assignments are shown in the appendix.

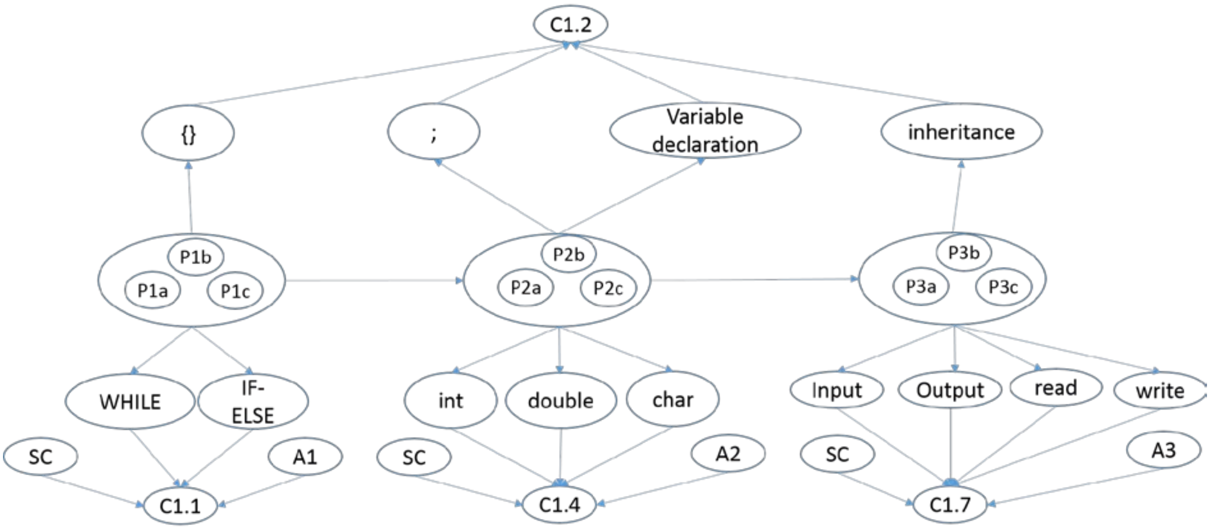

Figure 3. A Bayesian Network for a part of the object-oriented programming domain

P1a: "*write a program that calculates the return after investing an amount of money at a constant yearly interest rate using a WHILE loop.*"

P1b: "*write a program that calculates the return after investing an amount of money at a constant yearly interest rate applying the tail recursion*"

P1c: ""*write a program that calculates the return after investing an amount of money at a constant yearly interest rate applying the naive recursion*"

Similar to the group of programming problems P1, if the student has solved one of the P2 problems (P2a, P2b, or P2c) successfully, the student would have acquired knowledge the primitive data type "int", the primitive data type "double" and the primitive data type "char", and it is probable that the student as mastered the competence C1.4 (primitive data types). Similarly, if the student has solved one of the P3 problems (P3a, P3b, or P3c) successfully, then it means that the student has acquired knowledge of the "input" class, the "output" class, the "read" operator, and the "write" operator. Then, it is probable that the student has mastered the competence C1.7 (Know & apply input and output techniques (streams)). Solving the problem groups P1, P2, and P3 successfully would not only lead to acquiring the competences C1.1, C1.4 and C1.7, knowledge about using braces for beginning and ending a code block ({}), using semicolon, declaring variables, and defining a new instance of a class (using inheritance principle) could also be mastered. As a consequence, it is probable that the competence C1.2 (Know syntax and semantic) will also be acquired.

5. Conclusions and Future Work

In this paper, we have proposed to use Bayesian Network and the overlay modeling approach to build competence-based student models. In addition to knowledge, we adopt competences and relationships between knowledge and competences to be included in student models. Since currently there is no adaptive educational system that uses Bayesian Networks to model relationships between students' competences and learning events, or between competences and knowledge, the proposal of deploying Bayesian Networks to build student models is the contribution of this paper.

As future work, we will extend the Bayesian Network domain model for object-oriented programming with probability for relationships between the nodes. We plan to develop a prototype that uses this Bayesian Network in order to recommend programming assignments according to individual student's knowledge and competences.

Appendix

A sample solution for the programming assignment P1a: "*write a program that calculates the return after investing an amount of money at a constant yearly interest rate using a WHILE loop*."

```
                              Iterative Strategy
double investIterative(double startMoney, int period, double rate){
   double result=startMoney;
   if (period == 0) { result=startMoney;}
   while (period>0) {
      result = startMoney*rate+startMoney;
      period=period-1;
      startMoney=result;
         }
   return result;
}
```

A sample solution for the programming assignment P1b: "*write a program that calculates the return after investing an amount of money at a constant yearly interest rate applying the tail recursion.*"

```
                              Tail Recursive Strategy
double investTR(double startMoney, int period, double rate){
   if (period == 0) {return startMoney;}
         else {return investTR(startMoney+startMoney*rate, period-1, rate);}
}
```

A sample solution for the programming assignment P1c: ""*write a program that calculates the return after investing an amount of money at a constant yearly interest rate applying the naive recursion*"

```
                              Normal Recursive Strategy
double investNR(double startMoney, int period, double rate) {
  if( period == 0) {return startMoney;}
  else {
         return               (investNR(startMoney,               period-1,
   rate)+rate*investNR(startMoney,period-1, rate));
  }
}
```

References

Albert, D., & Lukas, J. (Eds.) (1999). *Knowledge Spaces: Theories, Empirical Research, Applications*. Mahwah, NJ: Lawrence Erlbaum Associates.

Almond, R. G., Mislevy, R. J., Steinberg, L., Yan, D., & Williamson, D. (2015). *Bayesian Networks in Educational Assessment*. Springer Verlag.

Al-Muhaideb, S. & Menai, M. E. (2011). Evolutionary computation approaches to the Curriculum Sequencing problem. *International Journal of Natural Computing*, 10(2), pp. 891-920.

Baker, R.S., Corbett, A.T., & Aleven, V. (2008). More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, pp. 406-415.

Beaumont, I. (1994). User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. *Journal User Modeling and User-Adapted Interaction*, 4(1), pp. 21-45.

Brusilovsky, P., Sosnovsky, S., Yudelson, M., & Chavan, G. (2005). Interactive authoring support for adaptive educational systems. *Proceedings of 12th International Conference on Artificial Intelligence in Education*, pp. 96-103.

Brusilovsky, P. & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In: Brusilovsky, P., Kobsa, A. & Nejdl, W. (Eds.), *The Adaptive Web*, Springer-Verlag, pp. 3-53.

Calvo, R.A. & D'Mello, S. (2010). Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *Journal IEEE Transactions on Affective Computing*, 1(1), pp. 18-37.

Desmarais, M. C. & Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *Journal User Modeling and User-Adapted Interaction*, 22(1-2), pp. 9-38.

Doignon, J.-P., & Falmagne, J. C (1985). Spaces for the Assessment of Knowledge. *International Journal of Man-Machine Studies*, 23(2), pp. 175–196.

Drieschner, E. (2009). Bildungsstandards praktisch: Perspektiven kompetenzorientierten Lehrens und Lernens. VS Verlag für Sozialwissenschaften.

Gertner, A., Conati, C., & VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. *Proceedings of the 15th National Conference on Artificial Intelligence AAAI*, pp. 106-111.

Gonschorek, M. & Herzog, C. (1995). Using hypertext for an adaptive helpsystem in an intelligent tutoring system. *Proceedings of the 7th World Conference on Artificial Intelligence in Education*, AACE, pp. 274-281.

Heller, J., Mayer, B., Hockemeyer, C., & Albert, D. (2006). Competence–based knowledge structures for personalised learning. *International Journal on E–Learning*, 5, pp. 75-88.

Hnida, M., Idrissi, M. K., & Bennani, S. (2014). A formalism of the competency-based approach in adaptive learning systems. *WSEAS Transactions on Information Science and Applications*, vol. 11, pp 83-93.

Holden, H. K. (2013). Understanding current learner modeling approaches. In R. Sottilare, A. Graesser, X. Hu, H. Holden (Eds.), *Design Recommendations for Adaptive Intelligent Tutoring Systems*, Vol. 1 (learner modeling), U.S. Army Research Laboratory.

Hsiao, I., Sosnovsky, S., & Brusilovsky, P. (2009). Adaptive navigation support for parameterized questions in object-oriented programming. *Proceedings of the European Conference on Technology Enhanced Learning*, Springer Verlag.

Kaliouby, R. E & Robinson, P. (2004). Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures. *Proceedings of International Conference on Computer Vision & Pattern Recognition*, pp. 181-200.

Le, N. T. & Müller, K. (2015). A Competency-based Analysis of AI-supported Instructional Approaches for Computer Science. To be appeared in *Special Issue on AI-supported Education in Computer Science, International Journal of Artificial Intelligence in Education (IJAIED).*

Martinez-Maldonado, R., Yacef, K., Dimitriadis, Y., Edbauer, M., & Kay, J. (2013). MTClassroom and MTDashboard: supporting analysis of teacher attention in an orchestrated multi-tabletop classroom. *Proceedings of International Conference on Computer-Supported Collaborative Learning*, pp. 119-128.

Mayo, M. & Mitrovic, A. (2001). Optimising ITS Behaviour with Bayesian Networks and Decision Theory. In *International Journal of Artificial Intelligence in Education*, 12, pp. 124-153.

NMC (2015). *NMC Report Higher Education Edition*. Retrieved on 19/06/2015: http://www.nmc.org/publication/nmc-horizon-report-2015-higher-education-edition

Nitchot, A., Gilbert, L., & Wills, G.B. (2010). Towards a Competence Based System for Recommending Study Materials (CBSR). *Proceedings of the 10th Conference on Advanced Learning Technologies*, pp. 629-631.

Nussbaumer, A., Gütl, C., & Neuper, W. (2010). A Methodology for Adaptive Competence Assessment and Learning Path Creation in ISAC. *Proceedings of the 13th Conference on Interactive Computer Aided Learning*, pp. 1136-1139.

Pérez, T., Gutiérrez, J., & Lopistéguy, P. (1995). An adaptive hypermedia system. *Proceedings of the 7th World Conference on Artificial Intelligence in Education*, AACE, 351-358.

Sabourin, J., Mott, B., & Lester, J. C. (2011). Modeling Learner Affect with Theoretically Grounded Dynamic Bayesian Networks. *Proceedings of the 4th Conference on Affective Computing and Intelligent Interaction*, pp. 286-295, Springer Verlag.

Vomlel, J. (2004). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness, and Knowledge-Based System*, 12(1), pp. 83-100.

Weinert, F. E. (2001): Competencies and Key Competencies: Educational Perspective. *International Encyclopedia of the Social and Behavioral Sciences*, vol. 4, Elsevier, 2433–2436.

Woolf, B. P. (2009). *Building Intelligent Interactive Tutors Student-centered Strategies for Revolutionizing e-learning*, Elsevier.