Chapter 2
Fuzzy Logic in Student Modeling

Abstract The significant development of the e-learning systems has changed the ways of teaching and learning. In nowadays, everyone can have access to e-learning systems from everywhere. Therefore, the e-learning systems have to adapt the learning material and processes to the needs of each individual learner. However, learning and student's diagnosis are complex processes, which deal with uncertainty. A solution to this is the use of fuzzy logic, which is able to deal with uncertainty and inaccurate data. This chapter explains how fuzzy logic can be used to automatically model the learning or forgetting process of a student, offering adaptation and increasing the learning effectiveness in Intelligent Tutoring Systems. In particular, it presents a novel rule-based fuzzy logic system, which models the cognitive state transitions of learners, such as forgetting, learning or assimilating. The operation of the presented approach is based on a Fuzzy Network of Related-Concepts (FNR-C), which is a combination of a network of concepts and fuzzy logic. It is used to represent so the organization and structure of the learning material as the knowledge dependencies that exist between the domain concepts of the learning material.

2.1 Introduction

Over the past decade, the rapid development of computer and Internet technologies has affect a variety of fields of the human's everyday life. Such a field is the education. The ways of teaching and learning have been changed and the e-learning systems and processes have been developed significantly. E-learning systems offer easy access to knowledge domains and learning processes from everywhere for everybody at any time. As a result, users of web-based educational systems are of varying backgrounds, abilities and needs. Therefore, the e-learning systems and applications have to offer dynamic adaptation to each individual student.

Adaptation is performed through the student model. In particular, the student model is a core component in any intelligent or adaptive tutoring system that is responsible for identifying and reasoning the student's knowledge level, misconceptions, abilities, preferences and needs. The student model represents many

K. Chrysafiadi and M. Virvou, *Advances in Personalized Web-Based Education*,
Intelligent Systems Reference Library 78, DOI 10.1007/978-3-319-12895-5_2

of the student's features, such as knowledge and individual traits, so as to be accessible for offering adaptation (Brusilovsky and Millán 2007). The adaptive and/or personalized educational system consults the student model and delivers the learning material to each individual learner with respect to her/his personal characteristics.

However, student modeling in many cases deals with uncertainty. Learning and student's diagnosis are complex. They are defined by many factors and are depended on tasks and facts that are uncertain and, usually, unmeasured. One possible approach to deal with this is fuzzy logic, which was introduced by Zaheh (1965) as a methodology for computing with words in order to handle uncertainty. It encounters the uncertainty problems that are caused by incomplete data and human subjectivity (Drigas et al. 2009). Chrysafiadi and Virvou (2012) have showed that the integration of fuzzy logic into the student model of an ITS can increase learners' satisfaction and performance, improve the system's adaptivity and help the system to make more valid and reliable decisions. Consequently, fuzzy logic techniques are able to analyze the students' knowledge level, needs and behavior and to make the right decision about the instructional model that has to be applied for each individual learner.

The issue of fuzzy logic and how it can be used in student modeling are presented in the remainder of this chapter. In particular, an overview of the fuzzy logic theory and fuzzy sets are described. Also, applications of fuzzy logic in student modeling are presented. Furthermore, the use of fuzzy logic in the representation of the knowledge domain of an adaptive and/or personalized tutoring system is described. In addition, a novel rule-based fuzzy logic system for modeling automatically the learning or forgetting process of a student is presented. Finally, a brief discussion and the conclusions drawn from this work are presented.

2.2 An Overview of Fuzzy Logic

Fuzzy logic was introduced by Zadeh (1965) to encounter imprecision and uncertainty. It deals with reasoning that is approximate rather than fixed and exact. It is a precise logic of imprecision and approximate reasoning (Zadeh 1975, 1979). In other words, fuzzy logic is able to reason and make rational decisions in circumstances of imprecision, uncertainty, human subjectivity, incomplete information and deficient computations (Zadeh 2001).

The basic element of the fuzzy logic theory is the fuzzy set. A fuzzy set describes a characteristic, thing, fact or state. For example, 'novice' is a fuzzy set that describes the student's knowledge level, 'young' is a fuzzy set that describes the person's age, 'cold' is a fuzzy set that describes the environment's temperature, 'tall' is a fuzzy set that describes the person's height, 'loud' is a fuzzy set that describes the sound's intensity, 'close' is a fuzzy set that describes the distance between two objects. The fuzzy sets that describe an element have no concrete limits (Fig. 2.1).

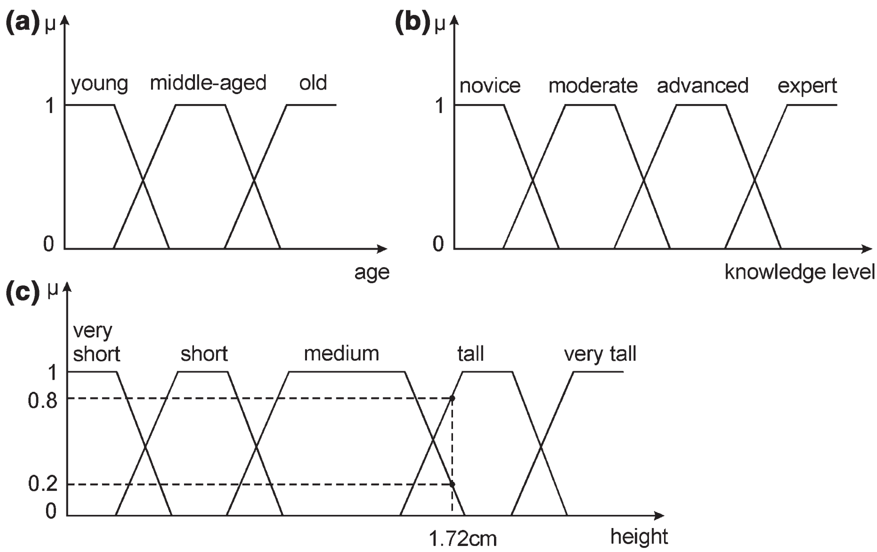

Fig. 2.1 Fuzzy sets and their partitions. **a** Fuzzy sets for age; **b** Fuzzy sets for knowledge level; **c** Fuzzy sets for height

Fuzzy logic variables have a truth-value that ranges in degree between 0 and 1. That value declares the degree in which the particular variable belongs to a fuzzy set. For example, if x is a fuzzy logic variable that describes the student's knowledge level and its value is 0.6 for the fuzzy set 'novice', then it means that the particular student is considered to be 60 % novice. This value is called degree of membership or membership value and is symbolized with μ. A fuzzy logic element can belong to two adjacent fuzzy sets at the same time, but with different membership degrees. For example, if a person's height is 1.72 cm, then according to the fuzzy sets that are depicted in Fig. 2.1c, the particular person is considered to be 80 % tall (the membership degree for the fuzzy set 'tall' is 0.8) and 20 % medium (the membership degree for the fuzzy set 'medium' is 0.2).

Taking into account the above, the definition of a fuzzy set follows (Fig. 2.2). Let S be a set of values that represent an element (i.e. S = {1.20, …, 2.10} for

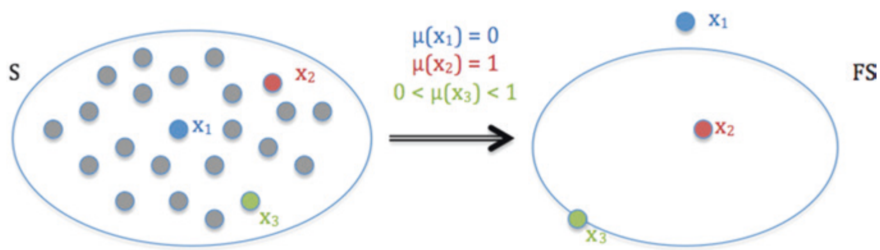

Fig. 2.2 Definition of fuzzy set

height; S = {1, 2, 3, …, 120} for age; S = {0, 1, 2, …, 100} for grades) and $x \in S$. In other words, x is a particular value that belongs to the set S. A fuzzy set FS is a pair (x, $\mu(x)$), where $x \in S$ and $\mu(x)$: $S \to [0, 1]$. In other words, for each $x \in S$, there is a value $\mu(x)$ between 0 and 1, which declares the membership degree of x to the fuzzy set FS.

- If $\mu(x) = 0$, then x is not included in FS
- If $\mu(x) = 1$, then x is fully included in FS
- If $0 < \mu(x) < 1$, then x is partially included in FS

2.2.1 Type-1 Fuzzy Sets

This first approach of fuzzy sets theory, which points that the value of the membership function of a fuzzy set can range between 0 and 1, is called type-1 fuzzy sets. Two common examples of a membership function of type-1 fuzzy sets are depicted in Fig. 2.3. Type-1 fuzzy sets have been criticized about their ability to handle uncertainty. It has been advocated that it is not reasonable to use an accurate membership function for something uncertain. Type-1 fuzzy sets used in conventional fuzzy systems cannot fully handle the uncertainties that are present in intelligent systems (Castillo and Melin 2008). To handle these uncertainties, Lotfi Zadeh (1975) proposed a more sophisticated kind of fuzzy sets theory that is called type-2 fuzzy sets (Mizumoto and Tanaka 1976; Mendel 2001).

2.2.2 Interval Type-2 Fuzzy Sets

The concept of a type-2 fuzzy set was introduced first by Zadeh (1975) as an extension of the type-1 fuzzy set. In particular, the membership function of a general type-2 fuzzy set is three-dimensional (Fig. 2.4):

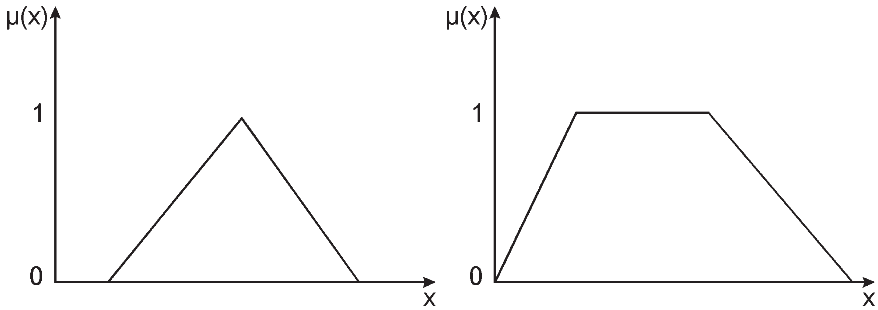

Fig. 2.3 Examples of type-1 fuzzy sets

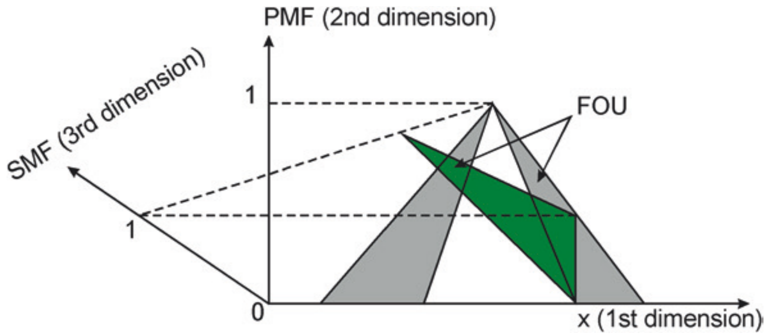

Fig. 2.4 The membership function of a general type-2 fuzzy set

- **1st dimension**: the primary variable x (e.g. age, height, grade, temperature)
- **2nd dimension**: the primary membership function (PMF), which is a function and not just a value between 0 and 1.
- **3rd dimension**: the secondary membership function (SMF), which is the value of the membership function at each point on its two-dimensional domain that is called its footprint of uncertainty (FOU). The value of SMF is, also, range between 0 and 1.

Using type-2 fuzzy logic can reduce the amount of uncertainty in a system. This is happened due to the fact that type-2 fuzzy logic offers better capabilities to handle linguistic uncertainties by modeling vagueness and unreliability of information (Liang and Mendel 2000). Such sets are useful in circumstances where it is difficult to determine the exact membership function for a fuzzy set, as in modeling a word by a fuzzy set.

When the value of the third dimension is the same (e.g. 1) everywhere, then the type-2 fuzzy set is called interval type-2 fuzzy set. For an interval type-2 set the SMF is ignored and only the FOU is used to describe it. The more (less) area in the FOU the more (less) is the uncertainty (Mendel 2001). The FOU represents the blurring of a type-1 membership function. It is completely described by its two bounding functions (Fig. 2.5): (i) a lower membership function (LMF) and (ii) an upper membership function (UMF).

2.2.3 Rule-Based Fuzzy Logic System

Type-2 fuzzy sets are finding very wide applicability in rule-based fuzzy logic systems (FLSs). The operation of FLSs is based on rules. The rules are expressed as a collection of IF-THEN statements (e.g. If George's grade at mathematics is 65/100, then he is classified to moderate students). Fuzzy sets are associated with the terms that appear in the antecedents (IF-part) or consequents (THEN-part) of rules. For example in the example "if George's grade at mathematics is 65/100, then he is classified to moderate students", the fuzzy set 'moderate' appears in

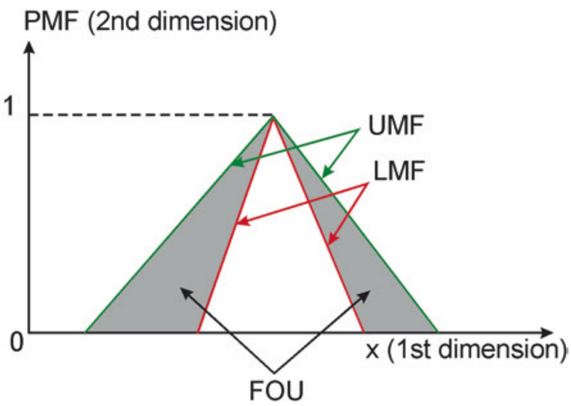

Fig. 2.5 The membership function of an interval type-2 fuzzy set

the consequents, while in the example "if the temperature indicates cold, then the heater must be switched on", the fuzzy set 'cold' appears in the antecedents. Membership functions are used to describe these fuzzy sets.

Experts construct the rules of a FLS considering their experience or data that have been extracted from experiments or surveys. Therefore, the knowledge and data that are used to construct the rules of a FLS are uncertain. This uncertainty leads to rules that have uncertain antecedents and/or consequents, which in turn translates into uncertain corresponding membership functions (Karnik et al. 1999). This uncertainty can be handled using type-2 fuzzy sets.

A type-2 FLS is depicted in Fig. 2.6. Two steps are required to go from an interval type-2 fuzzy set to a number:

- **Type-reduction**: in this step an interval type-2 fuzzy set is reduced to an interval-valued type-1 fuzzy set. This is achieved using particular algorithms. There are a comparable number of type-reduction methods (Mendel 2001).
- **Defuzzification**: In this step the centroid of the type-reduced set is computed. In particular, the average of the two end-points of the finite interval of numbers, which has been come off the process of type-reduction, is calculated. In other words, defuzzification maps the type-1 FS that came of the type-reduction step.

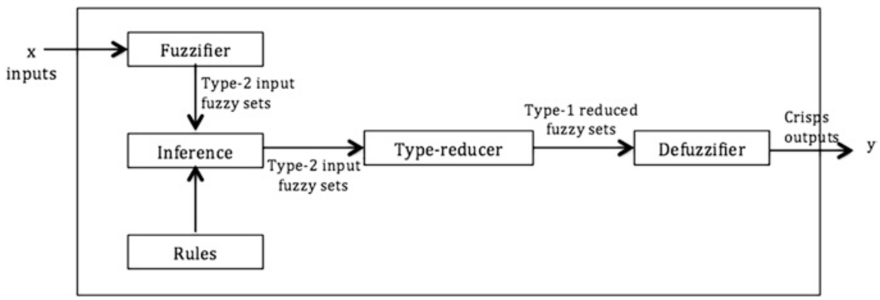

Fig. 2.6 A type-2 rule-based fuzzy logic system

2.2.4 Applications of Fuzzy Logic

The ability of fuzzy logic to handle the uncertainty, imprecise and incomplete data, and information that is characterized by human subjectivity makes it useful in many human-centric fields. Mendel (2007) has categorizes the applications of fuzzy logic in: approximation; clustering; control; databases; decision making; embedded agents; health care; hidden Markov models; neural networks; noise cancellation; pattern classification; quality control; spatial query; wireless communications. In addition, fuzzy set theory has been applied in education and educational systems. The applications of fuzzy logic in the educational field can be categorized into:

- **Grading systems**: Fuzzy logic is used to define the grade (as a letter, as a number, or as a percentage) that characterized the student's level of achievement. Examples of fuzzy applications in grading systems are the researches of (Bai and Chen 2006a, 2006b, 2008; Biswas 1995; Cheng and Yang 1998; Echauz and Vachtsevanos 1995; Law 1996; Wang and Chen 2006; Wilson et al. 1998).
- **Student's evaluation**: It includes an overall assessment of the student's learning. In particular, it is a complex process that includes student's performance, abilities, skills and learning characteristics. Some of the fuzzy logic applications in the process of the student's evaluation, which appear in the literature, are the following: (Chang and Sun 1993; Chen and Lee 1999; Ma and Zhou 2000; Nykänen 2006; Weon and Kim 2001).
- **Learning adaptation**: Learning and teaching are complex processes that have to consider each individual student's characteristics and abilities in order to be effective. The educational systems have to adapt dynamically to each individual learner's needs and abilities. Many researchers (Alves et al. 2008; Jili et al. 2009; Jurado et al. 2008; Kosba et al. 2003; Suarez-Cansino and Hernandez-Gomez 2008) have used fuzzy logic for providing learning and teaching adaptation.

2.2.4.1 Applications of Fuzzy Logic in Student Modeling

The aim of the adaptive and/or personalized tutoring systems is to readjust each time the instructional process and the teaching strategy considering the student's needs and abilities. This operation is based on human subjectivity and conceptualizations. That is the reason for the need of fuzzy logic. Therefore, there are many researchers that have used fuzzy logic techniques in student modeling to deal with uncertainty in the student's diagnose. For example, Xu et al. (2002) have used fuzzy models to represent a student profile in order to provide personalized learning materials, quiz and advices to each student. Furthermore, Kavčič (2004a) have succeeded to provide personalization of navigation in the educational content of InterMediActor system through the construction of a navigation graph and the adoption of fuzzy logic into student reasoning. A fuzzy-based student model has been applied, also, by Stathacopoulou et al. (2005) to a discovery-learning environment that aimed to help students to construct the concepts of vectors in physics and mathematics. The particular fuzzy-based student model allows the diagnostic model to some extent imitate teachers in diagnostic students' characteristics, and equips the intelligent learning environment with reasoning capabilities that can be further used to drive pedagogical decisions depending on the student learning style. Moreover, Jia et al. (2010) have applied fuzzy set theory to the design of an adaptive learning system in order to help learners to memory the content and improve their comprehension. Also, Goel et al. (2012) have used a fuzzy student model for facilitating the student reasoning process, which is based on imprecise information coming from the student-computer interaction, and predicting the degree of error that a student is possible to make in the next attempt to a problem. In addition, Salim and Haron (2006) have provided a personalized learning environment that exploit pedagogical model and fuzzy logic techniques. Other educational systems that have incorporated fuzzy logic techniques into the student model are: F-CBR-DHTS (Tsaganoua et al. 2003); TADV (Kosba et al. 2003, 2005) and DEPTHS (Jeremić et al. 2012).

2.3 Fuzzy Logic for Knowledge Representation

The knowledge domain module is one of the most major modules of an Intelligent Tutoring System (ITS). The knowledge domain representation is the base for the representation of the learner's knowledge, which is usually performed as a subset of the knowledge domain. It contains a description of the knowledge or behaviors that represent expertise in the subject-matter domain the ITS is teaching. In other words, the knowledge domain module is responsible for the representation of the subject matter taking into account the course modules, which involve domain concepts. The particular module has been introduced in ITS but its use has been extended to most current educational software applications that aim to be adaptive and/or personalized.

To enable communication between system and learner at content level, the domain model of the system has to be adequate with respect to inferences and relations of domain entities with the mental domain of a human expert (Peylo et al. 2000). Therefore, the knowledge domain representation in an adaptive and/or personalized tutoring system is an important factor for providing adaptivity. The appropriate approach for knowledge representation makes easier the selection of the appropriate educational material satisfying the student's learning needs. The most common used techniques of knowledge domain representation in adaptive tutoring systems are hierarchies and networks of concepts.

A hierarchical knowledge representation is usually used in order to specify the order in which the domain concepts of the learning material have to be taught (Chen and Shen 2011; Siddara and Manjunath 2007; Vasandani and Govindury 1995), and can be implemented through trees (Kumar 2005; Geng et al. 2011). For example, in INMA, which is a knowledge-based authoring tool for music education, the knowledge domain is described in terms of hierarchies (Virvou et al. 2006). Also, Siddappa et al. (2009) have developed a multilevel hierarchical model for the representation of knowledge domain of an intelligent tutoring system for numerical method (ITNM). This multilevel hierarchical model was based on various aptitude levels of students. An example of hierarchical representation is depicted in Fig. 2.7.

Hierarchies give information about the order in which the learning material should be taught, but they do not clearly depict the relations among the domain concepts. The network of concepts gives this kind of information. In a network of concepts, nodes represent concepts and arcs represent relations between concepts (Fig. 2.8). Many adaptive tutoring systems, such as Web-PTV (Tsiriga and Virvou 2003a, 2003b), DEPTHS (Jeremić et al. 2009) and IDEAL (Khamis 2011) use a network of concepts for representing the knowledge domain. However, in a network of concepts the relations between concepts are restricted to “part-of”, “is-a” and prerequisite relations. They do not depict how the knowledge of a domain concept may be affected by the knowledge of another concept. They do not give answers to the questions: “If a student learns the concept C_i, which will be her/his knowledge level of the depended domain concept C_j?”; “If the student’s knowledge of concepts C_i improves, how will be affected her/his knowledge of the

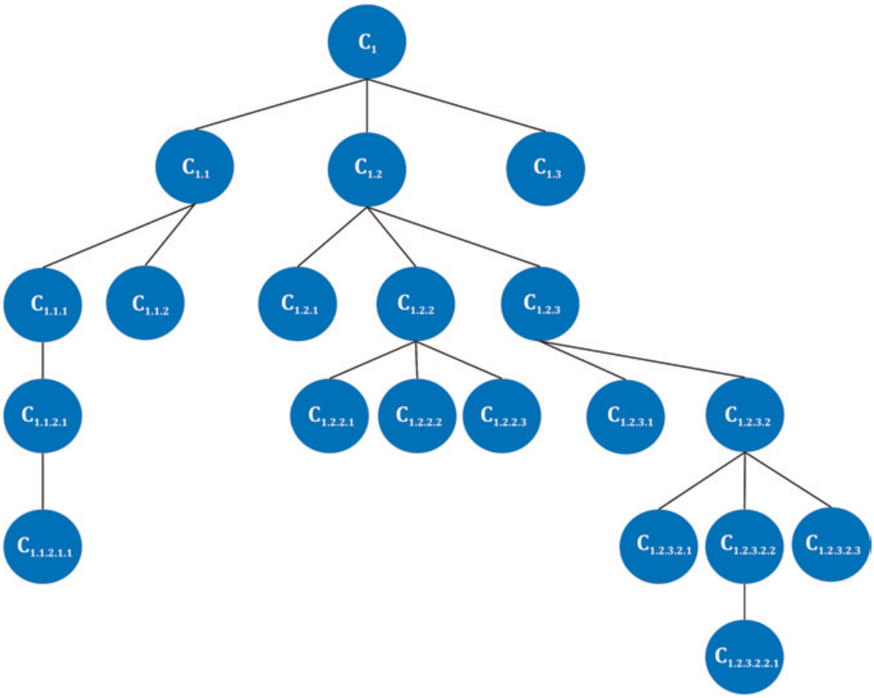

Fig. 2.7 A hierarhical tree

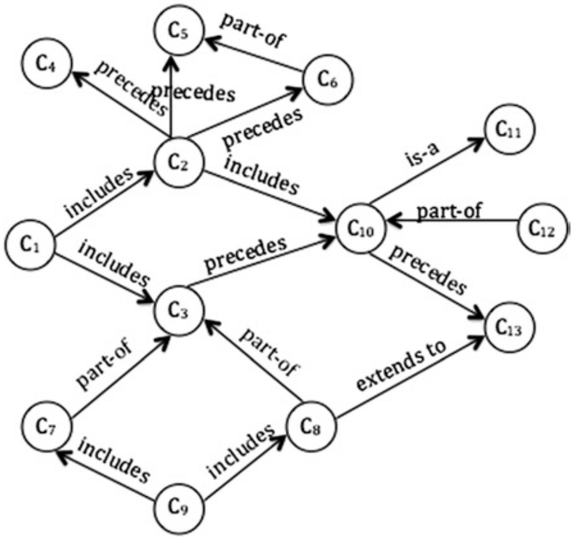

Fig. 2.8 A network of concepts

depended concept C_j?"; "If the student has misconceptions on the domain concept C_i, how will be affected her/his knowledge level of the depended concept C_i?".

The domain concepts that constitute the learning material are not independent from each other. The student's knowledge level of a domain concept usually is affected by her/his knowledge level of other related domain concepts. For example, a new domain concept may be completely unknown to the learner but in other circumstances it may be partly known due to previous related knowledge of the learner. On the other hand, domain concepts, which were previously known by the learner, may be completely or partly forgotten. Hence, currently they may be partly known or completely unknown. Therefore, the knowledge representation approach has to allow the system to recognize either the domain concepts that are already partly or completely known for a learner, or the domain concepts that s/he has forgot, taking into account the learner's knowledge level of the related concepts. Therefore, the representation of dependencies between the domain concepts of the learning material includes imprecise and uncertain information. As a result an effective solution for handling this uncertainty is to use fuzzy logic techniques in the representation of the knowledge domain.

A fuzzy logic application, which is used to model the behavior of complex systems (Leon et al. 2011) and emphasizes the connections and dependencies between the system's elements, is the Fuzzy Cognitive Map (FCM). Fuzzy Cognitive Maps (FCMs) constitute a way to represent real-world dynamic systems; in a form that corresponds closely to the way humans perceive it (Papageorgiou 2011; Papageorgiou and Iakovidis 2013). They are able to incorporate experts' knowledge (Papageorgiou and Salmeron 2012; Salmeron 2009; Salmeron et al. 2012) and approach representation of knowledge by emphasizing the connections and the structure (Lin 2007). A FCM illustrates the whole system as a combination of concepts and the various relations that exist between its concepts (Azadeh et al. 2012; Song et al. 2011; Stula et al. 2010). They are inference networks, using cyclic directed graphs, for knowledge representation and reasoning (Fig. 2.9). In particular, A FCM consists of nodes (N_1, N_2, … N_n), which represent the important elements of the mapped system, and directed arcs, which represent the causal relationships between two nodes (N_i, N_j). The directed arcs are labeled with fuzzy values (f_{ij}) in the interval [−1, 1] that show the "strength of impact" of node N_i on node N_j. If f_{ij} has a positive value, then it indicates that node N_i affects positively node N_j. In other words, the positive value on the directed arc that connects N_i with N_j, means that the increase of the value of N_i leads to the increase of the value of N_j, or the decrease of the value of N_i leads to the decrease of the value of N_j. Otherwise, If f_{ij} has a negative value, then it indicates that node N_i affects negatively node N_j. In other words, the negative value on the directed arc that connects N_i with N_j, means that the increase of the value of N_i leads to the decrease of the value of N_j, or the decrease of the value of N_i leads to the increase of the value of N_j. Therefore, a FCM is a cognitive map whose relations between the nodes can be used to compute the "strength of impact" of these elements. This property of FCM makes it able to predict, to make decisions, to generate a more accurate description of a difficult situation and to explain behaviors, actions and situations (Codara 1998). That is the reason of their extensive use in a wide range of applications (Craiger et al. 1996; Kosko 1999; Miao and Liu 2000; Rodriguez-Repiso et al. 2007; Stylios and Groumpos 2004). Furthermore, according to Papageorgiou (2011), in the past decade, FCMs have gained considerable research interest and are widely used to analyze causal systems such as system control, decision-making, management, risk analysis, text categorization, prediction etc. However, the contribution of FCMs to the knowledge representation of an adaptive tutoring system has not been discussed before.

Taking into account the above, there is the need to represent the knowledge dependency relations between the individual domain concepts of the domain knowledge. In particular, the knowledge dependencies that exist between the domain concepts of the learning material, as well as their "strength of impact" on each other have to be represented. A solution to this is to use a combination of a network of concepts with Fuzzy Cognitive Maps. In this way, a new approach of domain knowledge representation derives. That new approach is called Fuzzy Related-Concept Network (FR-CN).

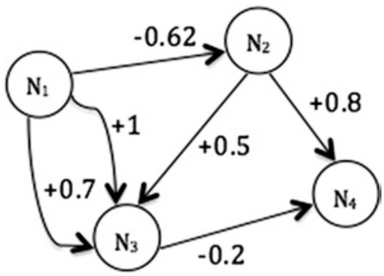

Fig. 2.9 A fuzzy cognitive map

2.3.1 Knowledge Domain Representation Using a Fuzzy Related-Concept Network

A Fuzzy Related-Concepts Network is a network of concepts, which depicts, also, the knowledge dependencies that exist between the domain concepts of the learning material. Therefore, it illustrates so the structure of the learning material, as the concepts' knowledge dependencies. Particularly, it represents the fact that the knowledge level of a domain concept is increased when the knowledge level of a related topic improves, as well as the fact that the knowledge level of a domain concept is decreased when the knowledge level of a depended topic is not satisfactory. The Fuzzy Related-Concepts Network (Fig. 2.10) consists of: nodes, which depict the domain concepts of the learning material, and directed arcs, which represent relations between the concepts of the learning material.

The relations that exist between the concepts of the learning material depict so the order in which the domain concepts have to be delivered and the structure of the learning material, as the knowledge dependencies. In particular, there are three type of relations between the concepts: "precedes" that declares the order in which each domain concept of the learning material has to be taught

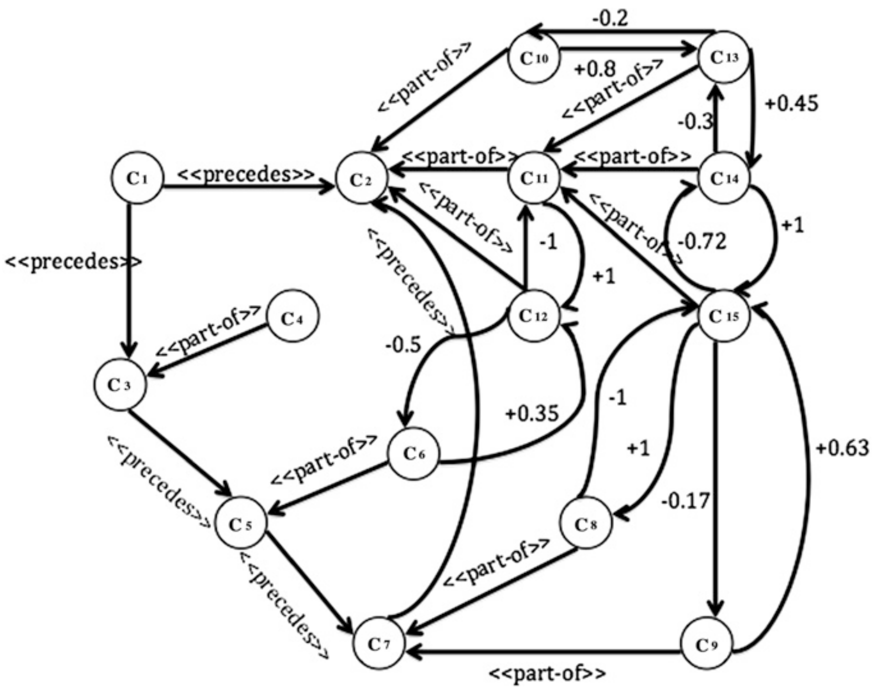

Fig. 2.10 A fuzzy related-concepts network

(for example, in Fig. 2.10 the domain concept C_3 is delivered to the learner before the domain concept C_5); "part-of" that declares that a concept belongs to another concept (for example, in Fig. 2.10 the domain concept C_2 in includes the domain concepts C_{10}, C_{11} and C_{12}); the dependence relation that declares that the knowledge level of a domain concept is affected by the learner's knowledge level on another related concept (For example, in Fig. 2.10 the knowledge level of the domain concepts C_{14}, C_8 and C_9 is affected by the learner's knowledge level on the concept C_{15}).

The dependence relations allow the tutoring system to identify how the knowledge level of a concept is affected by the learner's knowledge level on other related concepts. A dependence relation is characterized by the symbol '+' or the symbol '−' and a number (strength of impact). The symbol depicts the order in which the two related concepts are delivered to the learner. If the symbol '+' is labeled on the arc that connects C_i with C_j with direction from C_i to C_j ($C_i \rightarrow C_j$), then it denotes that C_i is taught before C_j. Otherwise, if the symbol that is labeled on the particular directed arc is the symbol '−', then it denoted that that C_j is taught before C_i. The numbers that are labeled on the directed arcs depict the degree at which the knowledge level of a domain concept is affected regarding the knowledge level of its related domain concepts. In other words, they depict the "strength of impact" of a domain concept on a related concept. The particular numbers are only positive. This is happened due to the fact that the increase of the knowledge level of a domain concept leads to the increase of the knowledge level of a depended domain concept, and the decrease of the knowledge level of a domain concept leads to the decrease of the knowledge level of a depended domain concept. Therefore, the numbers of the directed arcs that depict the knowledge dependencies belong to the interval (0, 1]. For example, in Fig. 2.10, the value '+0.8' that is labeled on the directed arc, which connects C_{10} with C_{13} ($C_{10} \rightarrow C_{13}$), denotes that the concept C_{10} is delivered to the learner before the concept C_{13} and the "strength of impact" of C_{10} on C_{13} is 0.8. Similarly, the value '−0.72' that is labeled on the directed arc, which connects C_{15} with C_{14} ($C_{15} \rightarrow C_{14}$), denotes that the concept C_{15} is delivered to the learner after the concept C_{14} and the "strength of impact" of C_{15} to C_{14} is 0.72.

The arcs in the FR-CN, which represent the domain concepts' dependencies of the knowledge domain, are bidirectional. Furthermore, the value of the arc $C_i \rightarrow C_j$ is not essentially equal to the value of the arc $C_j \rightarrow C_i$. This is happened due to the fact that changes on the knowledge level of C_i may affect the knowledge level of C_j in a different degree than changes on the knowledge level of C_j affect the knowledge level of C_i. It has to be clear that the value 1 on the directed arc that connects two dependent domain concepts does not mean that the two dependent concepts are the same. It implies that if a learner knows a domain concept of a section, s/he may know a related concept of another section at the same degree. The percentage of increase or decrease of the knowledge level of a domain concept that occurs due to changes on the knowledge level of another concept related with this domain concept is defined by experts of the knowledge domain.

Therefore, a FR-CN that is used to represent the knowledge domain of the learning material is a 6-tuple (C, ORD, PART, IMPACT, KL, f), where:

- $C = \{C_1, C_2, \ldots C_n\}$ is the set of concepts of the knowledge domain.
- ORD: $(C_i, C_j) \rightarrow \{0, 1\}$ is a matrix, which denotes that the concept C_i is delivered to the learner before the concept C_j (the value of the corresponding matrix's cell—line i, column j—is 1). If the value of the corresponding matrix's cell is 0, then it denotes that there is no "precedes" relation between the two domain concepts.
- PART: $(C_i, C_j) \rightarrow \{0, 1\}$ is a matrix, which denotes that the concept C_i is part-of the concept C_j (the value of the corresponding matrix's cell—line i, column j—is 1). If the value of the corresponding matrix's cell is 0, then it denotes that there is no "part-of" relation between the two domain concepts.
- IMPACT: $(C_i, C_j) \rightarrow w_{ij}$ is a matrix, where w_{ij} is a weight of the directed arc from C_i to C_j, which denotes the "strength of impact" of the concept C_i on the concept C_j (the value w_{ij} is inserted in the cell that corresponds to line i and column j). If $w_{ij} = 0$, then it denotes that C_i and C_j are not knowledge related concepts.
- KL is a function that at each concept C_i associates the sequence of its activation degree. In other worlds, $KL_i(t)$ indicates the value of a concept's knowledge level at the moment t.
- f is a transformation function. For the definition of the transformation function the following limitation has to be taken into account. Only the knowledge level of the most recently read concept affects the knowledge level of a domain concept, each time. The reason for this is the fact that the learner's knowledge level is affected either by the new knowledge that s/he has obtained, or by the knowledge that s/he has forgot, each time. Consequently, the KL value of a concept is affected only by the KL value of the most recently read concept, regarding the weight of the directed arc that connects them. Therefore, the transformation function for a FR-CN, which is used to represent the knowledge domain of the learning material, is defined as: $KL_i(t + 1) = f(KL_i(t) \pm w_{ji}{*}p_j{*}KL_i(t)/100)$, where p_j is the percentage of the difference on the value of the knowledge level of the most recently read concept C_j, with $p_j = (KL_j(t + 1) - KL_j(t)){*}100/KL_j(t)$. Also, the + is used in case of increase and the − is used in case of decrease.

For example, the matrixes ORD (Table 2.1), PART (Table 2.2) and IMPACT (Table 2.3) for the FR-CN that depicts in Fig. 2.10 are the following:

At the ORD matrix the value of the cell ORD [i, j], which corresponds to the line i and column j, can be 1, although there is no a direct arc in the corresponding FR-CN that connects the node-concept C_i with the node-concept C_j and declares "precedes" relation between the particular concepts. The reason for that is the fact that an indirect relation of type "precedes" can be exist between the particular concepts. For example, in the FR-CN of Fig. 2.10, the concept C_3 precedes the concept C_2 due to the fact that the concept C_7 precedes the concept C_2 and the

Table 2.1 ORD

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
C1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
C4	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
C5	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1
C6	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1
C7	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1
C8	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1
C9	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1
C10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.2 PART

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
C9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
C10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
C11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
C12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
C13	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
C14	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
C15	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0

concept C_3 precedes the concept C_7. Therefore, ORD [3, 2] = 1. Similarly, C_4 precedes C_8 because C_4 is part-of the concept C_3, which precedes the concept C_7 whose part is the concept C_8. C_3 precedes C_7 due to the fact that C_3 precedes C_5, which precedes C_7. As a result, ORD [4, 8] = 1.

Table 2.3 IMPACT

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0	0	0	+0.35	0	0	0
C7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+1
C9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+0.63
C10	0	0	0	0	0	0	0	0	0	0	0	0	+0.8	0	0
C11	0	0	0	0	0	0	0	0	0	0	0	+1	0	0	0
C12	0	0	0	0	0	−0.5	0	0	0	0	−1	0	0	0	0
C13	0	0	0	0	0	0	0	0	0	−0.2	0	0	0	+0.45	0
C14	0	0	0	0	0	0	0	0	0	0	0	0	−0.3	0	+1
C15	0	0	0	0	0	0	0	−1	−0.17	0	0	0	0	−0.72	0

2.3.1.1 Application of FR-CN for the Representation of the Knowledge Domain of the Programming Language 'C'

An application of Fuzzy Related-concepts Networks to a real situation is needed to understand the above, described approach for knowledge domain representation. That is the aim of the particular section, in which the description of the knowledge domain of a programming tutoring system is presented. In particular, the knowledge domain of the programming tutoring system is the programming language 'C'. The aim of the particular tutoring system is to teach learners so the principles and structures of the programming language 'C', as the logic of programming. So, the learning material includes not only expressions, operations and statements of the programming language 'C', but also it includes algorithms, like calculating sums, averages and maximums or minimums. Thereby, the learning material is decomposed in domain concepts which concern declarations of variables and constants, expressions and operators, input and output expressions, the sequential execution of a program, the if, if-else and if-else if statements, the iteration statements (for loop, while loop, do…while loop), sorting and searching algorithms, arrays, functions (Table 2.4).

Learners of programming languages have different backgrounds and their knowledge of a concept of the programming language, which they are taught, is subject to change. A new concept may be completely unknown to the learner but in other circumstances it may be partly or completely known due to previous related knowledge of the learner. For example, if a learner already knows an algorithm (e.g., calculating the sum of integers in a 'for' loop), there is no need to learn another similar algorithm (e.g., counting in a 'for' loop). Similarly, if a learner knows a programming structure (e.g., one-dimensional arrays), it is easier to understand another

Table 2.4 Learning material of the programming language ‘C’

C_1. Basics	$C_{1.1}$. Constants and variables	C_5. Iteration structure Unknown no of loops	$C_{5.1}$. While statement
	$C_{1.2}$. Assignment statement		$C_{5.2}$. Calculating sum in a while loop
	$C_{1.3}$. Arithmetical operators		$C_{5.3}$. Counting in a while loop
	$C_{1.4}$. Comparative operators		$C_{5.4}$. Calculating avgr in a while loop
	$C_{1.5}$. Logical operators		$C_{5.5}$. Calculating max/min in a while loop
	$C_{1.6}$. Mathematical functions		$C_{5.6}$. Do…while statement
	$C_{1.7}$. Input-output statements		
C_2. Sequence structure	$C_{2.1}$. A simple program structure	C_6. Arrays	$C_{6.1}$ One-dimensional arrays
			$C_{6.2}$. Searching
C_3. Conditional structures	$C_{3.1}$. If statement		$C_{6.3}$. Sorting
	$C_{3.2}$. If…else if		$C_{6.4}$. Two-dimensional arrays
	$C_{3.2.1}$ Methodology of finding max/min		
	$C_{3.3}$. Nested if		$C_{6.5}$. Processing per row
			$C_{6.6}$. Processing per column
C_4. Iteration structure Concrete no of loops	$C_{4.1}$. For statement		$C_{6.7}$. Processing of diagonals
	$C_{4.2}$. Calculating sum in a for loop	C_7. Sub-programming	$C_{7.1}$. Functions
	$C_{4.3}$. Counting in a for loop		
	$C_{4.4}$. Calculating avgr in a for loop		
	$C_{4.5}$. Calculating max/min in a for loop		

programming structure (e.g., multidimensional arrays), so this new structure should not be considered as being completely unknown to the learner. On the other hand, domain concepts, which were previously known by the learner, may be completely or partly forgotten. For example, if a learner has difficulties in calculating a sum in a ‘while’ loop, her/his knowledge of the previous domain concept of “calculating a sum in a ‘for’ loop” has eroded. Therefore, there is the need to represent the knowledge dependencies that exist between the domain concepts of the learning material of the programming language. This is achieved using Fuzzy Related-Concepts Network. The FR-CN for the knowledge domain of the programming language ‘C’ that is described in Table 1.7 is depicted in Fig. 2.11. Tables 2.5, 2.6 and 2.7 are a part of the matrixes ORD, PART, IMPACT of the FR-CN of Fig. 2.11 correspondingly. The whole matrixes are presented in the Appendix A.

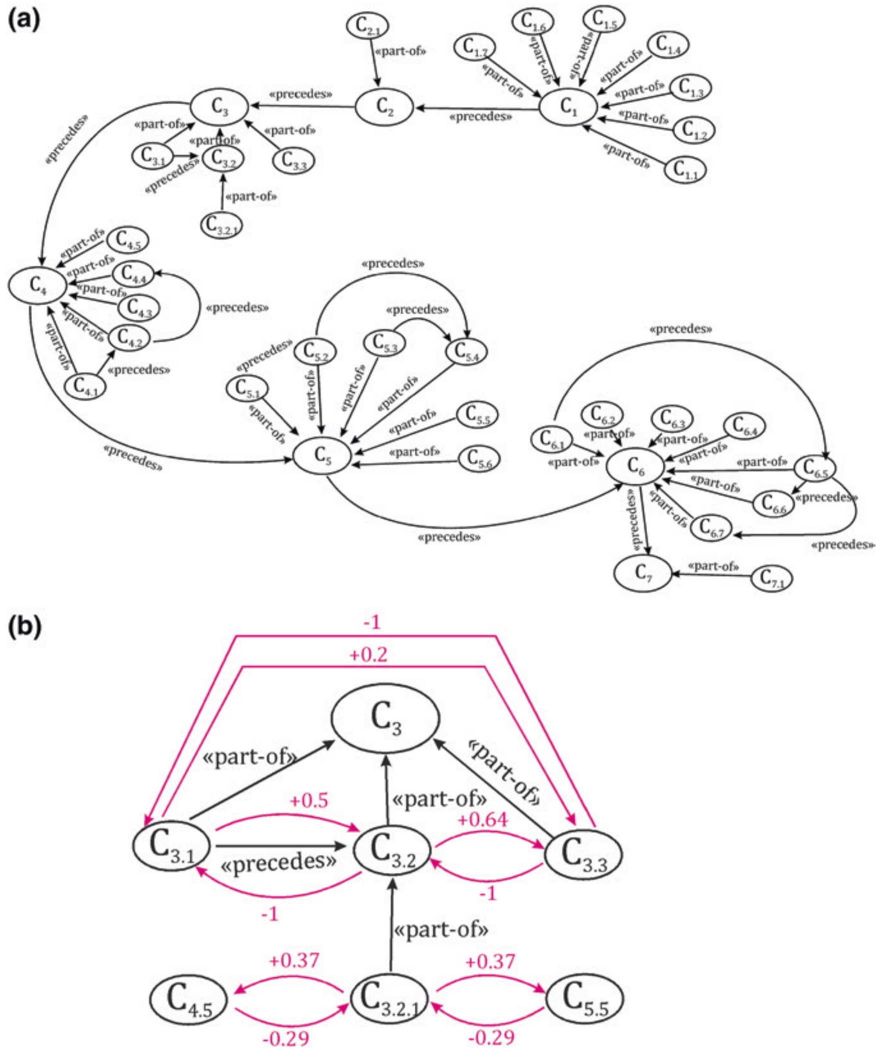

Fig. 2.11 The FR-CN of the knowledge domain of the programming language 'C' (it is decomposed in four graphs). **a** The "precedence" and "part-of" relations of the FR-CN; **b** The knowledge dependence relations for the domain concepts of the section 3; **c** The knowledge dependence relations for the domain concepts of the section 6; **d** The knowledge dependence relations for the domain concepts of the sections 4 and 5

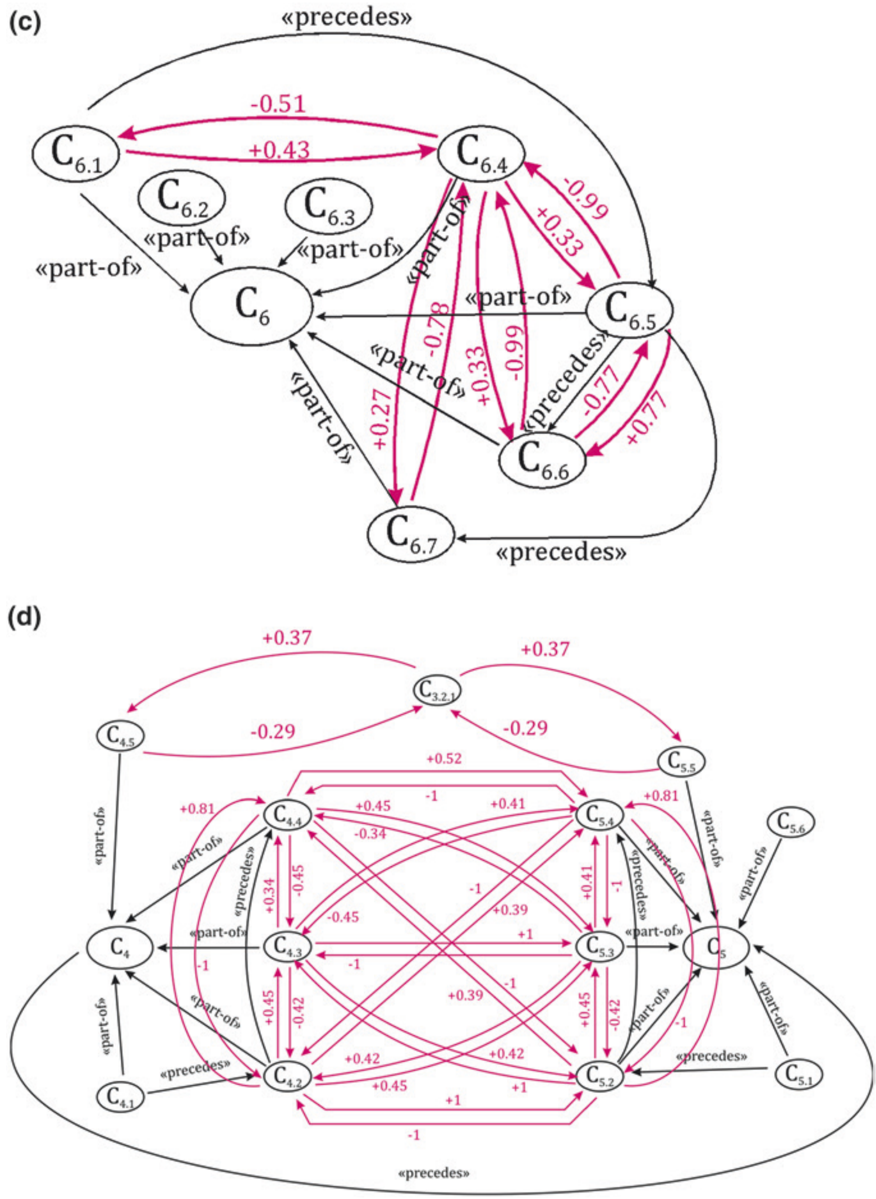

Fig. 2.11 (continued)

The value 1 on the directed arc that connects two dependent domain concepts of the FR-CN implies that if a learner knows a domain concept, then s/he may know a related domain concept at the same degree. For example, if a learner has been tested and found to have known the “for” loop and the “while” loop and this

Table 2.5 A sample of the ORDER matrix of the FR-CN of Fig. 2.11

	C_1	$C_{1.1}$	$C_{1.2}$	$C_{1.3}$	$C_{1.4}$	$C_{1.5}$	$C_{1.6}$	$C_{1.7}$	C_2	$C_{2.1}$	C_3	$C_{3.1}$	$C_{3.2}$	$C_{3.2.1}$	$C_{3.3}$
$C_{1.7}$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
C_2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
$C_{2.1}$	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
C_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$C_{3.1}$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Table 2.6 A sample of the PART matrix of the FR-CN of Fig. 2.11

	C_1	$C_{1.1}$	$C_{1.2}$	$C_{1.3}$	$C_{1.4}$	$C_{1.5}$	$C_{1.6}$	$C_{1.7}$	C_2	$C_{2.1}$	C_3	$C_{3.1}$	$C_{3.2}$	$C_{3.2.1}$	$C_{3.3}$
$C_{2.1}$	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
C_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$C_{3.1}$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
$C_{3.2}$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
$C_{3.2.1}$	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Table 2.7 A sample of the IMPACT matrix of the FR-CN of Fig. 2.11

	C_4	$C_{4.1}$	$C_{4.2}$	$C_{4.3}$	$C_{4.4}$	$C_{4.5}$	C_5	$C_{5.1}$	$C_{5.2}$	$C_{5.3}$	$C_{5.4}$	$C_{5.5}$	$C_{5.6}$
$C_{4.1}$	0	0	0	0	0	0	0	0	0	0	0	0	0
$C_{4.2}$	0	0	0	+0.45	+0.81	0	0	0	+1	+0.45	+0.39	0	0
$C_{4.3}$	0	0	−0.42	0	+0.34	0	0	0	+0.42	+1	+0.41	0	0
$C_{4.4}$	0	0	−1	−0.45	0	0	0	0	+ 1	+0.45	+0.52	0	0
$C_{4.5}$	0	0	0	0	0	0	0	0	0	0	0	+1	0

learner knows how to calculate sum in a "for" loop, s/he will also know how to calculate sum in a "while" loop, since the methodology is the same.

Experts on programming have defined so the domain concepts of the learning material, as their relations ("precedence", "part-of", "knowledge dependence"). In particular, ten professors of computer programming, whose experience counts 12 years at least, are responsible for the definition and structure of the knowledge domain. They were, also, asked to determine, empirically, the knowledge dependencies that exist between the defined domain concepts of the learning material, as well as their "strength of impact" on each other. The FR-CN that is depicted in Fig. 2.11 has been mapped according to the mean of the experts' answers (due to its complexity, it has been decomposed in four graphs).

The information that is derived from the above matrixes concerns:

- The order in which the domain concepts of the leaning material have to be delivered.
- Which domain concepts belong to another general domain concept of the learning material.
- The knowledge dependencies that exist between the domain concepts of the learning material and their "strength of impact".

For example, the domain concept C_1 is delivered before concept C_2 and concept $C_{4.2}$ is delivered before the domain concept $C_{4.4}$. That is derived from the values of the cells ORDER [1, 9] (Table 1.8a) and ORDER [18, 20] (Table 1.8b), which are 1 both. On the other hand, the ORDER [18, 21] = 0 (Table 1.8b) denotes that the concept $C_{4.2}$ is not necessary to be taught before the concept $C_{4.5}$. Furthermore, $C_{3.2.1}$ belongs to the concepts C_3 and $C_{3.2}$ as PART [14, 11] = 1 and PART [14, 13] = 1 (Table 2.1a). In addition, the learner's knowledge level on the concept $C_{4.4}$ affects the particular learner's knowledge level on the previously delivered concepts $C_{4.2}$, $C_{4.3}$, $C_{5.2}$, $C_{5.3}$ and $C_{5.5}$. This information is derived from the matrix IMPACT. In particular, the values IMPACT [20, 18] = −1 and IMPACT [20, 19] = −0.45 (Table 2.2b) denote that the knowledge level of concept $C_{4.4}$ affects the knowledge level of $C_{4.2}$ and $C_{4.3}$, and its "strength of impact" on $C_{4.2}$ and $C_{4.3}$ are 1 and 0.45 correspondingly. Similarly, the values IMPACT [20, 24] = + 1, IMPACT [20, 25] = + 0.45 and IMPACT [20, 26] = +0.52 (Table 2.2b) denote that the knowledge level of concept $C_{4.4}$ affects the knowledge level of the following concepts $C_{5.2}$, $C_{5.3}$ and $C_{5.5}$, and its "strength of impact" on the particular concepts are 1, 0.45 and 0.52 correspondingly. However, the value IMPACT [20, 21] = 0 (Table 2.2b) denote that the knowledge level of concept $C_{4.4}$ does not affect the knowledge level of the concept $C_{4.5}$.

2.4 A Novel Rule-Based Fuzzy Logic System for Modeling Automatically the Learning or Forgetting Process of a Student

Learning is not a "black or white" process. The definition of the learner's knowledge level is a moving target. In other words, it is not a straightforward task to define for each learner which concepts are unknown, known or assimilated and at what degree. The particular process is confronted with uncertainty and human subjectivity. One possible approach to deal with this is fuzzy set techniques, with their ability to naturally represent human conceptualization. That is the reason for the integration of fuzzy logic techniques into the student model.

Fuzzy logic is the solution for recognizing and modeling the increase and/or decrease of the learner's knowledge level on a domain concept in relation with her/his performance on other related domain concepts of the learning material. In particular, the presented rule-based fuzzy logic module is responsible for identifying and updating the student's knowledge level of all the concepts of the knowledge domain. Its operation is based on the Fuzzy Related-Concepts Network that is used to represent the structure of the learning material and the dependencies that exist between the domain concepts. It uses fuzzy sets to represent the student's knowledge level and a mechanism of rules over the fuzzy sets, which is triggered after a change has occurred on the student's knowledge level of a domain concept. This mechanism updates the student's knowledge level of all related with this concept, concepts. With this approach the alterations on the state of student's knowledge level, such as forgetting or learning are represented.

The presented rule-based fuzzy logic module includes the following three steps:

Step 1 **Definition of the fuzzy sets:**

In the particular step, the definition of the fuzzy sets, which represent the learner's knowledge level on a domain concept (i.e. {"Unknown", "Known", "Learned"} or {"Unknown", "Insufficiently Known", "Known", "Learned", "Assimilated"}), is carried out. Fuzzy sets are used to characterize the changeable learner's knowledge level. Therefore, FS_1, FS_2, …, FS_n are the defined fuzzy sets, for the educational adaptive system.

Step 2 **Definition of the membership functions:**

In the particular step, the membership functions of the determined fuzzy sets FS_1, FS_2, …, FS_n is defined. The membership functions (Fig. 2.12) are defined as follows (x indicates the learner's degree of success on a particular domain concept; x_{i-1}, x_i, x_{i+1}, x_{i+2} are thresholds that indicate particular degrees of success like 0, 50, 100):

$$\mu_{FS1} = \begin{cases} 1, & x \le x_1 \\ 1 - \frac{x-x_1}{x_2-x_1}, & x_1 < x < x_2 \\ 0, & x \ge x_2 \end{cases}$$

$$\forall i \neq 1 and i \neq n \mu_{FSi} = \begin{cases} \frac{x-x_{2i-3}}{x_{2i-2}-x_{2i-3}}, & x_{2i-3} < x < x_{2i-2} \\ 1, & x_{2i-2} \le x \le x_{2i-1} \\ 1 - \frac{x-x_{2i-1}}{x_{2i}-x_{2i-1}}, & x_{2i-1} < x < x_{2i} \\ 0, & x \le x_{2i-3} \, or \, x \ge x_{2i} \end{cases}$$

$$\mu_{FSn} = \begin{cases} \frac{x-x_{2n-3}}{x_{2n-2}-x_{2n-3}} & x_{2n-3} < x < x_{2n-2} \\ 1 & x_{2n-2} \le x \le x_{2n-1} \\ 0 & x \le x_{2n-3} \end{cases}$$

The knowledge level of a domain concept changes in a continuous way. Meaning that the knowledge level of a domain concept usually passes gradually from the

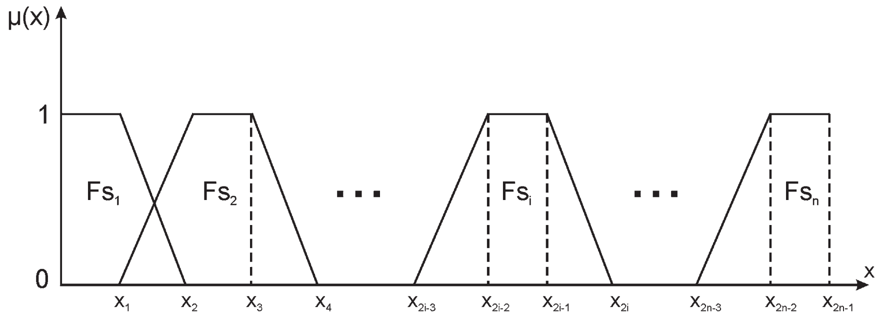

Fig. 2.12 The membership functions μ_{FSi}

unknown state to the learned and assimilated state. Membership values correspond to percentages of the offered knowledge in a way that they cover 100 % of it, at any time. This gives a more natural and understandable way of representation. For example, it would be non-intuitive to say that domain concept "A" is 0.5 (50 %) Insufficiently Known and 0.6 (60 %) Known for a student, given that 0.5 plus 0.6 gives 1.1 (110 %). So, the sum of the concept's percentage of different knowledge levels has to be 100 %, or 1 if the membership value of a concept to a knowledge level category is from 0 to 1. So, the following expression stands:

$$\mu_{FS_1} + \mu_{FS_2} + \mu_{FS_3} + \ldots + \mu_{FS_n} = 1$$

Therefore, a set (μ_{FS1}, μ_{FS2}, μ_{FS3}, … ,μ_{FSn}) is used express the student knowledge of a domain concept.

Step 3 **Definition of the fuzzy rules:**

When there is a dependency between two domain concepts, then the knowledge level of the one domain concept can affect the knowledge level of the other domain concept. More specifically, the following are taken into account:

- Considering the knowledge level of C_i, the knowledge level of its following domain concept C_j is increased or decreased.
- Considering the knowledge level of C_j, the knowledge level of its prerequisite domain concept C_i is increased or decreased.

Consequently, the student model expands when a change on the knowledge level of a domain concept causes increase on the knowledge level of the related concepts, or it is minimized when a change on the knowledge level of a domain concept causes decrease on the knowledge level of the related concepts with this concept.

In this document, D is defined to represent the knowledge dependency between two domain concepts. The symbolism $\mu_D(C_i, C_j)$ is used to represent the "strength of impact" of C_j on C_i and the symbolism $\mu_D(C_j, C_i)$ is used to represent the "strength of impact" of C_i on C_j. The values of $\mu_D(C_i, C_j)$ and $\mu_D(C_i, C_j)$ are the values of the arcs that depict the "knowledge dependencies" relations between the concepts of the learning material in the FR-CN of the knowledge domain (Sect. 3.1).

Concerning two domain concepts C_i and C_j where C_i is taught before C_j, the knowledge level of the concepts can change according to the following rules. These rules depict how the changes on the knowledge level of the domain concepts of the learning material for a student occur, revealing her/his learning state. In particular, they reveal if s/he learns or not or if s/he forgets. If the knowledge level of a concept is decreased, then the system infers that the student does not learn. If the knowledge level of a previously taught concept is decreased, then the system infers that the student forgets. If the knowledge level of a concept is increased, then the system infers that the student learns, and if the knowledge level of all the related concepts is improved continuously, then the system infers that the student assimilates the learning material.

The rules are based on Kavčič's (2004b) work. That work models mainly how the student's knowledge level of the prerequisites concepts that the student had read previously, is improved when s/he performs better in following concepts. In this way Kavčič's work deals only with how learning progresses. In her work there are no rules that imply the possible decrease of knowledge via the student's forgetting of some previously learned concepts. Moreover, another important problem that is not dealt with in Kavčič's work is the fact that in static educational systems, students are often required to repeat previously known concepts thought the following chapters. However, this practice is quite generic and does not take into account individual features of a student such as how fast they learn or how well they remember previously taught concepts. As such, educational systems do not adapt their pace on individual students. In view of the above, in the presented rule-based fuzzy module, Kavčič's rules have been expanded to deal with the above problems. The rules with these novelties that lead to the dynamic personalization of teaching are presented below. In the following rules, FS_x, FS_y are fuzzy sets that represent knowledge levels with $FS_x < FS_y$, and KL() denotes the "Knowledge Level of".

- ***Based on updates of the KL(C_i), the KL(C_j) is improved according to:***

R1: If the same fuzzy sets are active for both C_i and C_j, then KL(C_j) = FS_x with

$$\mu_{FS_y}(C_j) = \max[\mu_{FS_x}(C_i), \mu_{FS_x}(C_i) * \mu_D(C_i, C_j)$$

where FS_x is the last active fuzzy set. Subtract the value (new $\mu_{FSx}(C_j)$—previous $\mu_{FSx}(C_j)$) from the others $\mu_{FSy}(C_j)$ (FSy < FSx) sequentially until $\sum \mu_{FSi} = 1$.

R2: If KL(C_j) = FS_x and KL(C_i) = FS_y, then KL(C_j) = FS_y with

$$\mu_{FS_y}(C_j) = \mu_{FS_y}(C_i) * \mu_D(C_i, C_j)$$

- ***Based on updates of the KL(C_i), the KL(C_j) is deteriorated according to:***

R3: If KL(Cj) = FS_n, then
if $\mu_{FS1}(C_j) + \mu_{FS2}(C_j) + \cdots + \mu_{FSn-1}(C_j) < \mu_{FSi}(C_i) * \mu_D(C_i, C_j)$, where $i < n$, then the corresponding value is subtracted by $\mu_{FSn}(Cj)$
else it does not change.

R4: If KL(C_j) = FS_y and KL(C_i) = FS_x, then KL(C_j) = FS_x with

$$\mu_{FS_x}(C_j) = \mu_{FS_x}(C_i) * \mu_D(C_i, C_j)$$

- ***Based on updates of the KL(C_j), the KL(C_i) is improved according to:***

R5: If the same fuzzy sets are active for both C_i and C_j, then KL(C_j) = FS_x with

$$\mu_{FS_x}(C_i) = \max[\mu_{FS_x}(C_i), \mu_{FS_x}(C_j) * \mu_D(C_i, C_j)]$$

where FS_x is the last active fuzzy set. Subtract the value (new $\mu_{FSx}(C_i)$—previous $\mu_{FSx}(C_i)$) from the others $\mu_{FSy}(C_i)$ (FSy < FSx) sequentially until $\sum \mu_{FSi} = 1$

R6: If KL(C_i) = FS_x and KL(C_j) = FS_y, then KL(C_i) = FS_y with

$$\mu_{FS_y}(C_i) = \mu_{FS_y}(C_j) * \mu_D(C_j, C_i)$$

- ***Based on updates of the KL(C_j), the KL(C_i) is deteriorated according to:***

  R7: *If KL(Ci) = FS_n with* μ_{FSn}(Ci) = 1*, then it does not change*

  R8: The formula $x_i = \left(1 - \mu_D\left(C_i, C_j\right)\right) * x_i + \min[\mu_D\left(C_i, C_j\right) * x_i, \mu_D\left(C_i, C_j\right) * x_j]$, where x_i and x_j are the values of the criterion, which determines the fuzzy sets that are active each time for C_i and C_j respectively, is used (for the calculation of previous x_i, the membership value of the upper active fuzzy set is used). Then, using the new x_i, the KL(C_i) is determined, calculating the membership functions.
- **Limitation:** $\sum \mu_{FSi} = 1$

2.4.1 Integration of the Fuzzy Rules

The application of the fuzzy rules of the step 3 that was described above deals with the problem of estimating wrongly the knowledge level of a domain concept. In particular, consider the fuzzy sets {"Uknown", "Known", "Well-Known", "Learned"} and the set of their membership functions (μ_{Un}, μ_K, μ_{WK}, μ_L) that represent the student's knowledge level of a domain concept. Let's the domain concept C_i to be 100 % 'Learned' and the "strength of impact" of C_i on the following concept C_j to be 0.3. The knowledge level of C_j is 100 % 'Unknown'. According to the rule R2, the knowledge level of C_j will become 30 % 'Learned'. However, that it means that the rest 70 % of the concept C_j is 'Known'? The answer is no. The rest 70 % of the C_j can be 'Unknown', 'Known', 'Well-Known' or 'Learned', or different parts of it can belong to a different fuzzy set (i.e. 10 % 'Unknown', 20 % 'Known' and 40 % 'Well-Known'). In addition, let's the set that describes the knowledge level of the domain concept C_i to be (0.8, 0.2, 0, 0) (e.g. 80 % 'Unknown' and 20 % 'Known' → KL(C_j) = 0.2 'Known') and the "strength of impact" of C_i on its following concept C_j to be 0.6. The knowledge level of C_j is 20 % 'Learned'. According to the rule R4, the knowledge level of C_j will become 60 % 'Known'. However, that it means that the rest 40 % of the concept C_j is 'Uknown'? The answer is no. It can be any of the above fuzzy sets.

A solution to this problem is to keep data for each domain concept of the learning material concerning the different part of the particular concept that can be affected be other related concepts. In such a way, the system can be informed each time about the knowledge level of each separate part of the particular domain concept and it is able to draw conclusions about the learner's knowledge level on the overall domain concept. For example, according to the Fig. 2.10 (Sect. 3.1) the domain concept C_{12} is affected by both concepts C_{11} and C_6. Initially is KL(C_6) = KL(C_{11}) = KL(C_{12}) = 100 % 'Uknown'. During the learning process, the concept C_6 is delivered to the learner firstly. The learner's knowledge level on the particular concept becomes 20 % 'Well-Known' and 80 % 'Known' (KL(C_6) = 20 % Well-Known). According to the rule R2, the learner's knowledge level on the domain concept C_{12} will become 7 % 'Well-Known' and 28 % 'Known'. The other part, however, of C_{12} is not affected by C_6. So, its knowledge level remains 'Unknown'. Therefore, C_{12} is 7 % 'Well-Known', 28 % 'Known' and 65 % 'Unknown'. As a result, the system will advise the learner to read C_{12}. Also, according to R6, the learner's knowledge level of C_{11} will become 7 % 'Well-Known' 28 % 'Known' and 65 % 'Unknown' because C_{12} affects C_{11} with "strength of impact" 1 (Fig. 2.10). Then, the concept C_{11} is delivered to the learner. The learner's knowledge level on the particular concept becomes 40 % 'Learned' and 60 % 'Well-Known' ($KL(C_{11}) = 40$ % Learned). According to R2 is $KL(C_{12}) = 40$ % Learned (40 % 'Learned' and 60 % 'Well-Known'), due to the fact that the "strength of impact" of C_{11} on C_{12} is 1. Therefore, the system will consider that the concept the learner knows C_{12}, and it will not advise her/him to read the particular concept. In addition, C_{12} affects C_6. The "strength of impact" of the particular knowledge dependency is 0.5. Therefore, according to the rule R6, the learner's knowledge level of C_6 will become 20 % 'Learned' and 30 % 'Well-known'. However, because the previous knowledge level of C_6 was 20 % 'Well-Known' and 80 % 'Known', the system will consider that the rest 50 % of C_6 remains 'Known'. Thereby, although the learner's knowledge level on C_6 has been improved, the system will advise the learner to revise the domain concept C_6.

2.4.2 Application of the Presented Rule-Based Fuzzy Logic System in a Programming Tutoring System

In this chapter an application of the presented rule-based fuzzy logic system is described. In particular, the presented rule-based fuzzy logic system is used to model the cognitive states of learners of the programming language 'C'.

Step 1 **Definition of the fuzzy sets:**

The defined fuzzy sets are the following:

- **Unknown (Un)**: the degree of success in the domain concept is from 0 to 50 %.
- **Moderate Known (MKn)**: the degree of success in the domain concept is from 40 to 70 %.
- **Known (Kn):** the degree of success in the domain concept is from 60 to 80 %.
- **Learned (L)**: the degree of success in the domain concept is from 75 to 90 %.
- **Assimilated (A):** the degree of success in the domain concept is from 85 to 100 %.

Step 2 **Definition of the membership functions:**

The membership functions of the fuzzy sets Un, MKn, Kn, L and A are depicted in Fig. 2.13 and are the following (x indicates the learner's degree of success on a particular domain concept):

$$\mu_{Un} = \begin{cases} 1, & x \leq 40 \\ 1 - \frac{x-40}{10}, & 40 < x < 50 \\ 0, & x \geq 50 \end{cases}$$

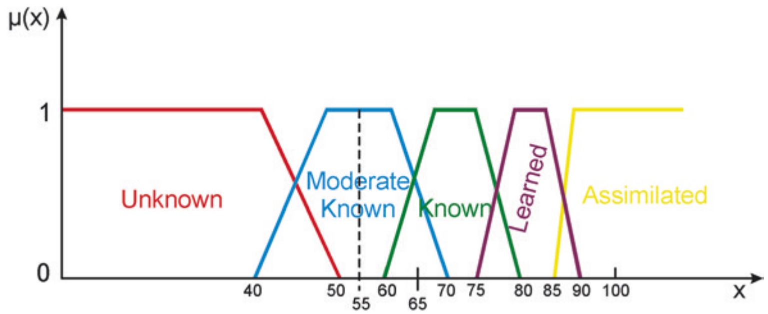

Fig. 2.13 The membership functions of the fuzzy sets of the programming tutoring system for 'C'

$$\mu_{MKn} = \begin{cases} \frac{x-40}{10}, & 40 < x < 50 \\ 1, & 50 \le x \le 60 \\ 1 - \frac{x-60}{10}, & 60 < x < 70 \\ 0, & x \le 40\, or\, x \ge 70 \end{cases}$$

$$\mu_{Kn} = \begin{cases} \frac{x-60}{10}, & 60 < x < 70 \\ 1, & 70 \le x \le 75 \\ 1 - \frac{x-75}{5}, & 75 < x < 80 \\ 0, & x \le 60\, or\, x \ge 80 \end{cases}$$

$$\mu_{L} = \begin{cases} \frac{x-75}{5}, & 75 < x < 80 \\ 1, & 80 \le x \le 85 \\ 1 - \frac{x-85}{5}, & 85 < x < 90 \\ 0, & x \le 75\, or\, x \ge 90 \end{cases}$$

$$\mu_{A} = \begin{cases} \frac{x-85}{5}, & 85 < x < 90 \\ 1, & 90 \le x \le 100 \\ 0, & x \le 85 \end{cases}$$

Therefore, a set (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_{L}, μ_{A}) is used to express the student knowledge of a domain concept.

Experts on programming and teachers of the programming language 'C' have defined the limits of each fuzzy set. In particular, they were asked to determine the lower and higher values of the degree of success that characterize a domain concept as 'Unknown', 'Moderate Known', 'Known', 'learned' and 'Assimilated'. The mean values of their answers consist the base for the definition of the limits of the presented fuzzy sets.

Step 3 **Definition of the fuzzy rules:**

Concerning two domain concepts C_i and C_j where C_i is taught before C_j, the knowledge level of the concepts can change according to the following rules ($\mu_D(C_i, C_j)$ and $\mu_D(C_j, C_i)$ indicate the "strength of impact" of C_i on C_j and of C_j on C_i correspondingly. Their values are the values of the arcs that depict the "knowledge dependencies" relations between the concepts of the learning material in the FR-CN (Sect. 3.1.1 Fig. 2.11)):

- ***Based on updates of the KL(C_i), the KL(C_j) is improved according to:***

Subtract the value (new $\mu_x(C_j)$—previous $\mu_x(C_j)$) from the others $\mu_y(C_j)$ sequentially until $\mu_{Un} + \mu_{MKn} + \mu_{Kn} + \mu_L + \mu_A = 1$, where x = {MKn, Kn, L, A} and y = {Un, MKn, Kn, L} with y < x.

R1: If the same fuzzy sets are active for both C_i and Cj, then:
– If $KL_A(C_j) > 0$: $\mu_{\mathrm{A}}\left(C_j\right) = \max\left[\mu_{\mathrm{A}}\left(C_j\right), \mu_{\mathrm{A}}(C_i) * \mu_D\left(C_i, C_j\right)\right]$
– Else If $KL_L(C_j) > 0$: $\mu_L\left(C_j\right) = \max\left[\mu_L\left(C_j\right), \mu_L(C_i) * \mu_D\left(C_i, C_j\right)\right]$
– Else If $KL_{Kn}(C_j) > 0$: $\mu_{\mathrm{Kn}}\left(C_j\right) = \max\left[\mu_{\mathrm{Kn}}\left(C_j\right), \mu_{\mathrm{Kn}}(C_i) * \mu_D\left(C_i, C_j\right)\right]$
– Else If $KL_{MKn}(C_j) > 0$: $\mu_{\mathrm{MKn}}\left(C_j\right) = \max\left[\mu_{\mathrm{MKn}}\left(C_j\right), \mu_{\mathrm{MKn}}(C_i) * \mu_D\left(C_i, C_j\right)\right]$

R2:

(a) If KL(C_j) = Un and KL(C_i) = MKn, then KL(C_j) = MKn with

$$\mu_{\mathrm{MKn}}\left(C_j\right) = \mu_{\mathrm{MKn}}(C_i) * \mu_D\left(C_i, C_j\right)$$

(b) If KL(C_j) = Un and KL(C_i) = Kn, then KL(C_j) = Kn with

$$\mu_{\mathrm{Kn}}\left(C_j\right) = \mu_{\mathrm{Kn}}(C_i) * \mu_D\left(C_i, C_j\right)$$

(c) If KL(C_j) = Un and KL(C_i) = L, then KL(C_j) = L with

$$\mu_{\mathrm{L}}\left(C_j\right) = \mu_L(C_i) * \mu_D\left(C_i, C_j\right)$$

(d) If KL(C_j) = Un and KL(C_i) = A, then KL(C_j) = A with

$$\mu_{\mathrm{A}}\left(C_j\right) = \mu_{\mathrm{A}}(C_i) * \mu_D\left(C_i, C_j\right)$$

(e) If KL(C_j) = MKn and KL(C_i) = Kn, then KL(C_j) = Kn with

$$\mu_{\mathrm{Kn}}\left(C_j\right) = \mu_{\mathrm{Kn}}(C_i) * \mu_D\left(C_i, C_j\right)$$

(f) If KL(C_j) = MKn and KL(C_i) = L, then KL(C_j) = L with

$$\mu_{\mathrm{L}}\left(C_j\right) = \mu_L(C_i) * \mu_D\left(C_i, C_j\right)$$

(g) If KL(C_j) = MKn and KL(C_i) = A, then KL(C_j) = A with

$$\mu_{\mathrm{A}}\left(C_j\right) = \mu_{\mathrm{A}}(C_i) * \mu_D\left(C_i, C_j\right)$$

(h) If KL(C_j) = Kn and KL(C_i) = L, then KL(C_j) = L with

$$\mu_L\left(C_j\right) = \mu_L(C_i) * \mu_D\left(C_i, C_j\right)$$

(i) If $KL(C_j) = Kn$ and $KL(C_i) = A$, then $KL(C_j) = A$ with

$$\mu_A(C_j) = \mu_A(C_i) * \mu_D(C_i, C_j)$$

(j) If $KL(C_j) = L$ and $KL(C_i) = A$, then $KL(C_j) = A$ with

$$\mu_A(C_j) = \mu_A(C_i) * \mu_D(C_i, C_j)$$

- ***Based on updates of the KL(C_i), the KL(C_j) is deteriorated according to:***

R3: If KL(Cj) = A, then
– if $\mu_{Un}(C_j) + \mu_{MKn}(C_j) + \mu_{Kn}(C_j) + \mu_L(C_j) < \mu_x(C_i) * \mu_D(C_i, C_j)$, where x = {Un, MKn, Kn, L}, then the corresponding value is subtracted by μ_A(Cj)
– else it does not change.

R4:

(a) If $KL(C_j) = L$ and $KL(C_i) = Kn$, then $KL(C_j) = Kn$ with

$$\mu_{Kn}(C_j) = \mu_{Kn}(C_i) * \mu_D(C_i, C_j)$$

(b) If $KL(C_j) = L$ and $KL(C_i) = MKn$, then $KL(C_j) = MKn$ with

$$\mu_{MKn}(C_j) = \mu_{MKn}(C_i) * \mu_D(C_i, C_j)$$

(c) If $KL(C_j) = L$ and $KL(C_i) = Un$, then $KL(C_j) = Un$ with

$$\mu_{Un}(C_j) = \mu_{Un}(C_i) * \mu_D(C_i, C_j)$$

(d) If $KL(C_j) = Kn$ and $KL(C_i) = MKn$, then $KL(C_j) = MKn$ with

$$\mu_{MKn}(C_j) = \mu_{MKn}(C_i) * \mu_D(C_i, C_j)$$

(e) If $KL(C_j) = Kn$ and $KL(C_i) = Un$, then $KL(C_j) = Un$ with

$$\mu_{Un}(C_j) = \mu_{Un}(C_i) * \mu_D(C_i, C_j)$$

(f) If $KL(C_j) = MKn$ and $KL(C_i) = Un$, then $KL(C_j) = Un$ with

$$\mu_{Un}(C_j) = \mu_{Un}(C_i) * \mu_D(C_i, C_j)$$

- ***Based on updates of the KL(C_j), the KL(C_i) is improved according to:***

R5: If the same fuzzy sets are active for both C_i and Cj, then:
– If $KL_A(C_i) > 0$: $\mu_A(C_i) = \max[\mu_A(C_i), \mu_A(C_j) * \mu_D(C_j, C_i)]$
– Else If $KL_L(C_i) > 0$: $\mu_L(C_i) = \max[\mu_L(C_i), \mu_L(C_j) * \mu_D(C_j, C_i)]$
– Else If $KL_{Kn}(C_i) > 0$: $\mu_{Kn}(C_i) = \max[\mu_{Kn}(C_i), \mu_{Kn}(C_j) * \mu_D(C_j, C_i)]$
– Else If $KL_{MKn}(C_i) > 0$: $\mu_{MKn}(C_i) = \max[\mu_{MKn}(C_i), \mu_{MKn}(C_j) * \mu_D(C_j, C_i)]$

Subtract the value (new $\mu_x(C_i)$—previous $\mu_x(C_i)$) from the others $\mu_y(C_i)$ sequentially until $\mu_{Un} + \mu_{MKn} + \mu_{Kn} + \mu_L + \mu_A = 1$, where x = {MKn, Kn, L, A} and y = {Un, MKn, Kn, L} with y < x.

R6:

(a) If $KL(C_i) = Un$ and $KL(C_j) = MKn$, then $KL(C_i) = MKn$ with

$$\mu_{\mathrm{MKn}}(C_i) = \mu_{\mathrm{MKn}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(b) If $KL(C_i) = Un$ and $KL(C_j) = Kn$, then $KL(C_i) = Kn$ with

$$\mu_{\mathrm{Kn}}(C_i) = \mu_{\mathrm{Kn}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(c) If $KL(C_i) = Un$ and $KL(C_j) = L$, then $KL(C_i) = L$ with

$$\mu_{\mathrm{L}}(C_i) = \mu_L\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(d) If $KL(C_i) = Un$ and $KL(C_j) = A$, then $KL(C_i) = A$ with

$$\mu_{\mathrm{A}}(C_i) = \mu_{\mathrm{A}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(e) If $KL(C_i) = MKn$ and $KL(C_j) = Kn$, then $KL(C_i) = Kn$ with

$$\mu_{\mathrm{Kn}}(C_i) = \mu_{\mathrm{Kn}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(f) If $KL(C_i) = MKn$ and $KL(C_j) = L$, then $KL(C_i) = L$ with

$$\mu_{\mathrm{L}}(C_i) = \mu_L\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(g) If $KL(C_i) = MKn$ and $KL(C_j) = A$, then $KL(C_i) = A$ with

$$\mu_{\mathrm{A}}(C_i) = \mu_{\mathrm{A}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(h) If $KL(C_i) = Kn$ and $KL(C_j) = L$, then $KL(C_i) = L$ with

$$\mu_L(C_i) = \mu_L\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(i) If $KL(C_i) = Kn$ and $KL(C_j) = A$, then $KL(C_i) = A$ with

$$\mu_{\mathrm{A}}(C_i) = \mu_{\mathrm{A}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

(j) If $KL(C_i) = L$ and $KL(C_j) = A$, then $KL(C_i) = A$ with

$$\mu_{\mathrm{A}}(C_i) = \mu_{\mathrm{A}}\left(C_j\right) * \mu_D\left(C_j, C_i\right)$$

- ***Based on updates of the $KL(C_j)$, the $KL(C_i)$ is deteriorated according to:***

R7: *If $KL(C_i) = A$ with* $\mu_A(Ci) = 1$*, then it does not change.*

R8: The formula $x_i = \left(1 - \mu_D\left(C_i, C_j\right)\right) * x_i + \min[\mu_D\left(C_i, C_j\right) * x_i, \mu_D\left(C_i, C_j\right) * x_j]$, where x_i and x_j are the degree of success, which determine the fuzzy sets that are active each time for C_i and C_j respectively, is used (for the calculation of previous x_i, the membership value of the upper active fuzzy set is used). Then, using the new x_i, the $KL(C_i)$ is determined, calculating the membership functions.

- ***Limitation* L1:** $\mu_{Un} + \mu_{MKn} + \mu_{Kn} + \mu_L + \mu_A = 1$.

2.4.2.1 Examples of Operation

The above described rule-based fuzzy logic system was used in a postgraduate program in the field of informatics at the University of Piraeus in Greece. It was used in order to offer dynamically personalized e-training in computer programming and the language C. At the beginning, all the domain concepts of the learning material were considered to be 'Unknown' for the learners. At the next interactions, the system delivered to them the appropriate learning material for each individual student's needs by adapting instantly to the learner's individual learning pace. The KL value of each domain concept was determined by the results of the tests. There were two kinds of tests: (i) the tests that corresponded to each individual domain concept of the learning material (practice tests), (ii) the final tests that corresponded to the sections of the learning material (they included exercises of a variety of domain concepts). In particular, each time the learner read a domain concept, s/he had to complete a corresponding practice test. When, the learner had completed successfully all the practice tests of the domain concepts of a section (e.g. iterations with concrete number of loops, arrays, sub-programming), then s/he had to complete the final test of the section. If s/he succeeded to the final test, then s/he transited to a next section. Otherwise, s/he had advised to revise some domain concepts. Representative examples of the system's implementation follow.

- **Example 1**

George had learned the sections 1 (domain concepts 1.1 to 1.7) and 2 (domain concept 2.1) and she was taught the domain concepts of the section 3 (domain concepts 3.1 to 3.3) (Interaction I of Table 2.8). He read the concept $C_{3.1}$. Then, he was examined in the particular domain concept and succeeded 78 %. According to the above, the value of the defined membership functions for concept $C_{3.1}$ become $\mu_{Un} = 0$, $\mu_{MKn} = 0$, $\mu_{Kn} = 0.4$, $\mu_{L} = 0.6$ and $\mu_{A} = 0$. According to the FR-CN (Fig. 2.11) the concept $C_{3.1}$ affects the following concepts $C_{3.2}$ and $C_{3.3}$ with "strength of impact" 0.5 and 0.2 correspondingly. Consequently, applying the fuzzy rule R2 (b) and (c), KL($C_{3.2}$) becomes 20 % 'Known' and 30 % 'Learned'. The rest 50 % of the particular concept remains 'Unknown' (Interaction II of Table 2.8). Similarly, applying the same rules, KL($C_{3.3}$) becomes 8 % 'Known' and 12 % 'Learned'. The rest 80 % of the particular concept remains 'Unknown' (Interaction II of Table 2.8). Therefore, although concepts $C_{3.2}$ and $C_{3.3}$ are not completely unknown to George, the system advises him to read them.

- **Example 2**

Kate had learned the sections 1 (domain concepts 1.1 to 1.7), 2 (domain concept 2.1), 3 (domain concepts (3.1 to 3.3) and the concepts 4.1, 4.5 and 5.5 (Interaction I of Table 2.9). She read the concept $C_{4.2}$ to improve her knowledge level. Then, she was examined in the particular domain concept and succeeded 86 %. According to the above, the value of the defined membership functions for concept $C_{4.2}$ become $\mu_{Un} = 0$, $\mu_{MKn} = 0$, $\mu_{Kn} = 0$, $\mu_{L} = 0.8$ and $\mu_{A} = 0.2$.

Table 2.8 George's progress

Domain concepts	Learner's knowledge	
	Interaction I (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_L, μ_A)	Interaction II (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_L, μ_A)
1.1 Constants and variables	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.2 Assignment statement	(0,0, 0, 0.08, 0.92)	(0,0, 0, 0.08, 0.92)
1.3 Arithmetic operators	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.4 Comparative operators	(0,0, 0,0.08, 0.92)	(0,0, 0,0.08, 0.92)
1.5 Logical operators	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.6. Mathematic functions	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.7 Input-output statements	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
2.1 A simple program's structure	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
3.1 If statement	(1, 0, 0, 0, 0)	(0, 0, 0.4, 0.6, 0)
3.2 If…else if	***(1, 0, 0, 0, 0)***	***(0.5, 0, 0.2, 0.3, 0)***
3.2.1 Finding max, min	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
3.3 Nested if statement	***(1, 0, 0, 0, 0)***	***(0.8, 0, 0.08, 0.12, 0)***
4.1 For statement	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
4.2 Calc. sum in a for loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
4.3 Counting in a for loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
4.4 Calc. avrg in a for loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
4.5 Calc. max/min in a for loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.1 While statement	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.2 Calc. sum in a while loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.3 Counting in a while loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.4 Calc. avrg in a while loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.5 Calc. max/min in a while loop	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.6 Do…until	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.1 One-dimension arrays	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.2 Searching	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.3 Sorting	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.4 Two-dimensions arrays	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.5 Processing per rows	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.6 Processing per column	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.7 Processing of diagonals	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
7.1 Functions	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)

According to the FR-CN (Fig. 2.11) the concept $C_{4.2}$ affects the following concepts $C_{4.3}$, $C_{4.4}$, $C_{5.2}$, $C_{5.3}$ and $C_{5.4}$ with "strength of impact" 0.45, 0.81, 1, 0.45 and 0.39 correspondingly. Consequently, applying the fuzzy rule R2 (c) and (d), KL($C_{4.3}$) becomes 36 % 'Learned' and 9 % 'Assimilated'. The rest 55 % of the particular concept remains 'Unknown' (Interaction II of Table 3.2). Similarly, applying the same rules, KL($C_{4.4}$) becomes 64.8 % 'Learned' and 16.2 % 'Assimilated' (the rest 19 % of the particular remains 'Unknown'), KL($C_{5.2}$) becomes 80 % 'Learned'

Table 2.9 Kate's progress

Domain concepts	Learner's knowledge	
	Interaction I (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_L, μ_A)	Interaction II (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_L, μ_A)
1.1 Constants and variables	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.2 Assignment statement	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.3 Arithmetic operators	(0, 0, 0, 0.02, 0.098)	(0, 0, 0, 0.02, 0.098)
1.4 Comparative operators	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.5 Logical operators	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.6. Mathematic functions	(0, 0, 0, 0.12, 0.88)	(0, 0, 0, 0.12, 0.88)
1.7 Input-output statements	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
2.1 A simple program's structure	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
3.1 If statement	(0, 0, 0, 0.3, 0.7)	(0, 0, 0, 0.3, 0.7)
3.2 If…else if	(0, 0, 0, 0.4, 0.6)	(0, 0, 0, 0.4, 0.6)
3.2.1 Finding max, min	(0, 0, 0, 0.1, 0.9)	(0, 0, 0, 0.1, 0.9)
3.3 Nested if statement	(0, 0, 0, 0.4, 0.6)	(0, 0, 0, 0.4, 0.6)
4.1 For statement	(0, 0, 0, 0.73, 0.27)	(0, 0, 0, 0.73, 0.27)
<u>4.2 Calc. sum in a for loop</u>	<u>(1, 0, 0, 0, 0)</u>	<u>(0, 0, 0, 0.8, 0.2)</u>
4.3 Counting in a for loop	***(1, 0, 0, 0, 0)***	***(0.55, 0, 0, 0.36, 0.09)***
4.4 Calc. avrg in a for loop	***(1, 0, 0, 0, 0)***	***(0.19, 0, 0, 0.648, 0.162)***
4.5 Calc. max/min in a for loop	(0, 0, 0, 0.67, 0.33)	(0, 0, 0, 0.67, 0.33)
5.1 While statement	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
5.2 Calc. sum in a while loop	***(1, 0, 0, 0, 0)***	***(0, 0, 0, 0.8, 0.2)***
5.3 Counting in a while loop	***(1, 0, 0, 0, 0)***	***(0.55, 0, 0, 0.36, 0.09)***
5.4 Calc. avrg in a while loop	***(1, 0, 0, 0, 0)***	***(0.61, 0, 0, 0.312, 0.078)***
5.5 Calc. max/min in a while loop	(0, 0, 0, 0.67, 0.33)	(0, 0, 0, 0.67, 0.33)
5.6 Do…until	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.1 One-dimension arrays	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.2 Searching	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.3 Sorting	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.4 Two-dimensions arrays	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.5 Processing per rows	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.6 Processing per column	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.7 Processing of diagonals	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
7.1 Functions	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)

and 20 % 'Assimilated', KL($C_{5.3}$) becomes 36 % 'Learned' and 9 % 'Assimilated' (the rest 55 % of the particular concept remains 'Unknown') and KL($C_{5.4}$) becomes 31.2 % 'Learned' and 7.8 % 'Assimilated' (the rest 61 % of the particular concept remains 'Unknown') (Interaction II of Table 3.3). Therefore, the increase of Kate's knowledge level on $C_{4.2}$ improves automatically her knowledge level

Table 2.10 Nick's progress

Domain concepts	Learner's knowledge	
	Interaction I (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_L, μ_A)	Interaction II (μ_{Un}, μ_{MKn}, μ_{Kn}, μ_L, μ_A)
1.1 Constants and variables	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.2 Assignment statement	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.3 Arithmetic operators	(0, 0, 0, 0.02, 0.098)	(0, 0, 0, 0.02, 0.098)
1.4 Comparative operators	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.5 Logical operators	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
1.6. Mathematic functions	(0, 0, 0, 0.12, 0.88)	(0, 0, 0, 0.12, 0.88)
1.7 Input-output statements	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
2.1 A simple program's structure	(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1)
3.1 If statement	(0, 0, 0, 0.3, 0.7)	(0, 0, 0, 0.3, 0.7)
3.2 If…else if	(0, 0, 0, 0.4, 0.6)	(0, 0, 0, 0.4, 0.6)
3.2.1 Finding max, min	(0, 0, 0, 0.1, 0.9)	(0, 0, 0, 0.1, 0.9)
3.3 Nested if statement	(0, 0, 0, 0.4, 0.6)	(0, 0, 0, 0.4, 0.6)
4.1 For statement	(0, 0, 0, 0.73, 0.27)	(0, 0, 0, 0.73, 0.27)
4.2 Calc. sum in a for loop	***(0, 0, 0, 0.8, 0.2)***	***(0, 0, 1, 0, 0)***
4.3 Counting in a for loop	***(0, 0, 0, 0.6, 0.4)***	***(0, 0, 0, 1, 0)***
4.4 Calc. avrg in a for loop	***(0, 0, 0, 0.7, 0.3)***	***(0, 0, 1, 0, 0)***
4.5 Calc. max/min in a for loop	(0, 0, 0, 0.67, 0.33)	(0, 0, 0, 0.67, 0.33)
5.1 While statement	(0, 0, 0, 1, 0)	(0, 0, 0, 1, 0)
<u>5.2 Calc. sum in a while loop</u>	<u>(0, 0, 0, 0.8, 0.2)</u>	<u>(0, 0, 1, 0, 0)</u>
5.3 counting in a while loop	***(0, 0, 0, 0.6, 0.4)***	***(0, 0, 0.45, 0.15, 0.4)***
5.4 Calc. avrg in a while loop	***(0, 0, 0, 0.7, 0.3)***	***(0, 0, 0.81, 0, 0.19)***
5.5 Calc. max/min in a while loop	(0, 0, 0, 0.67, 0.33)	(0, 0, 0, 0.67, 0.33)
5.6 Do…until	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.1 One-dimension arrays	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.2 Searching	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.3 Sorting	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.4 Two-dimensions arrays	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.5 Processing per rows	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.6 Processing per column	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
6.7 Processing of diagonals	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)
7.1 Functions	(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0)

on other related domain concepts, also. Indeed, the fact that the knowledge level of concept $C_{5.2}$ became automatically from 100 % 'Unknown', 80 % 'Learned' and 20 % 'Assimilated', without Kate read it, is particular important. This change triggers the system to infer that $C_{5.2}$ is already known for Kate.

- **Example 3**

Nick had learned the sections 1 (the domain concepts 1.1 to 1.7), 2 (the domain concept 2.1), 3 (the domain concepts 3.1 to 3.3), 4 (the domain concepts 4.1 to 4.5) and some domain concepts 5.1 to 5.5 of the section 5 (Interaction I of Table 2.10). He revised the concept $C_{5.2}$. During the revision, he was examined in the particular domain concept and succeeded 73 %. According to the above, the value of the defined membership functions for concept $C_{5.2}$ become $\mu_{Un} = 0$, $\mu_{MKn} = 0$, $\mu_{Kn} = 1$, $\mu_L = 0$ and $\mu_A = 0$. According to the FR-CN (Fig. 2.11) the concept $C_{5.2}$ affects the preceding concepts $C_{4.2}$, $C_{4.3}$, $C_{4.4}$ and the following concepts $C_{5.3}$ and $C_{5.4}$ with "strength of impact" 1, 0.45, 0.81, 0.45 and 0.81 correspondingly. Consequently, applying the fuzzy rule R8 is: $x_{4.2} = (1 - 1) * 86 + \min[1 * 86, 1 * 73] = 73.$ That degree of success corresponds to the fuzzy set 'Known' with $\mu_{Kn} = 1$. (Interaction II of Table 3.4). Similarly, applying the same rule, KL($C_{4.3}$) becomes 100 % 'Learned', and KL($C_{4.4}$) becomes 100 % 'Known' (Interaction II of Table 3.4). Furthermore, according to the rules R3 and R4 (a), KL($C_{5.3}$) becomes 45 % 'Known', 15 % 'Learned' and 40 % 'Assimilated' and KL($C_{5.4}$) becomes 70 % 'Known' and 30 % 'Assimilated' (Interaction II of Table 2.10).

2.5 Conclusions and Discussion

Learning is a complicated process. It cannot be accurately said that a learner knows or does not know a domain concept. For example, a new domain concept may be completely unknown to the learner but in other circumstances it may be partly known due to previous related knowledge of the learner. On the other hand, domain concepts, which were previously known by the learner, may be completely or partly forgotten. Hence, currently they may be partly known or completely unknown. In this sense, the level of knowing cannot be accurately represented. Finally, the teaching process itself changes the status of knowledge of a user. This is happened due to the fact that a learner accepts new concepts while being taught. Furthermore, the learner's knowledge is a moving target. The knowledge level of a domain concept is increased when the student's performance is improved. Alternatively, it is decreased when the student forgets. Improvement of the knowledge level of a domain concept should lead to the increase of the knowledge level of all the related concepts (prerequisite and following), with his concept. Similarly, poor performance on a domain concept should lead to decrease of the knowledge level of all the related concepts with this concept.

In view of the above, an effective adaptive tutoring system has to be responsible for tracking cognitive state transitions of learners with respect to their progress or non-progress. The alterations on the state of student's knowledge level are not linear. They deal with uncertainty. Thus, a solution to represent these is fuzzy logic. Therefore, the target of this section was to develop a rule-based fuzzy logic system, which models the cognitive state transitions of learners, such as forgetting, learning

Table 2.11 Correlation of an e-shop and an adaptive e-learning system concerning the presented rule-based fuzzy logic system

	E-shop	E-learning
Nodes	Products	Domain concepts
Arcs	Preferences' dependencies	Knowledge dependencies
Fuzzy sets	Descriptions of a preference (e.g. 'uninterested', 'interested', 'liked', 'preferred')	Descriptions of knowledge level (e.g. 'unknown', 'insufficiently known', 'known', 'learned')
Changeable states	Preferences	Knowledge level

or assimilating. The presented rule-based fuzzy logic system identifies and updates each time the student's knowledge level not only for the current concept, which is delivered to the learner, but also for all the related concepts with this concept. To achieve that, the system considers either the learner's performance or the knowledge dependencies that exist between the domain concepts of the learning material. In the particular rule-based fuzzy logic system, fuzzy sets are used in order to describe how well each individual domain concept is known and learned. Furthermore, it uses a mechanism of rules over the fuzzy sets, which is triggered after any change of the value of the knowledge level of a domain concept and updates the values of the knowledge level of all the related domain concepts with that. Therefore, the educational system, which has integrated the particular rule-based fuzzy logic system, is able to makes dynamic decisions on how the teaching syllabus is presented to the learner to fit his/her personal needs and learning pace.

The operation of the system is based on the knowledge domain representation that is implemented through a Fuzzy Related-Cognitive Network. This kind of knowledge domain representation helps to manage to represent either the order in which the domain concepts of the learning material have to be taught and organized, or the knowledge dependencies that exist between the domain concepts. This is significant because the knowledge level of a domain concept increases or decreases due to changes on the knowledge level of a related domain concept. The design of the learning material and the definition of the individual domain concepts that it includes, are based on the knowledge and experience of domain experts. Furthermore, the contribution of domain experts is significant for the definition of the knowledge dependencies that exist among the domain concepts of the learning material and their "strength o impact" on each other.

The presented rule-based fuzzy logic system is applicable to systems, in which the user's changeable state and/or preferences are affected by the existing dependencies among the system's elements (like concepts, preferences, events, choices). Thereafter, the particular system could be implemented in adaptive systems other than adaptive tutoring system. For example, it could be used in an e-shop, where the preference of an online shopper for particular products can be used in order to guess and propose her/him other products that the user is likely to be interested in. In the Table 2.11 the correlation of an e-shop and an adaptive e-learning system is presented concerning the particular rule-based fuzzy logic system (Table 2.11).

http://www.springer.com/978-3-319-12894-8

Advances in Personalized Web-Based Education
Chrysafiadi, K.; Virvou, M.
2015, XVII, 156 p. 58 illus., Hardcover
ISBN: 978-3-319-12894-8