# KNOWLEDGE LEARNING TECHNOLOGY FOR INTELLIGENT TUTORING SYSTEMS

Taran T.A., Sirota S.V.

*Abstract: In this work we suggest the technology of creation of intelligent tutoring systems which are oriented to teach knowledge. It is supposed the acquisition of expert's knowledge by using of the Formal Concept Analysis method, then construction the test questions which are used for verification of the pupil's knowledge with the expert's knowledge. Then the further tutoring strategy is generated by the results of this verification.*

*Keywords: Computer tutoring, Formal Concept Analysis.*

## Introduction

The conception of the up today's computer tutoring systems is based on teaching knowledge which represented like a pack of the facts and rules. In [3] the learning type's systematization was suggested accordant to cognitive levels used in studying. There are four cognitive levels marked: (1) creative, (2) analogy-generalization, (3) explanation and (4) programming. The 4-th level means the training to solve certain type of tasks, in the 3-d level the studying goes by explanation, the aidless work with exercises and problems needs the analogy-generalization, the highest, creative, level is supposed the pupil to ideate himself the concepts and relations between them. The interaction between the pupil and the tutoring system can hit different cognitive levels. It is clear that using of all four levels is optimal strategy of teaching. But up to day tutoring systems are not able to interact on creative level [3].

In [9] two basic directions of the studying process are marked: the concept learning and training. The concept learning has next stages:

- The description of objects attributes, quality and quantity characteristics of the objects and processes in the domain;
- The eduction of concepts;
- The definition of the relations between concepts;
- The definition of dependences on the sets of attributes and characteristics of objects and processes;

In concept learning tutoring systems the cardinal problems are representation of the facts, description of the attributes and relations and rigorous definition of the concepts to be memorized. The memorizing is the main procedure pupil to do. The further action of the system is directed to monitoring of knowing the main concepts of domain. The training started when the pupil have got certain set of knowledge. Then we create some problems to make pupil use his knowledge for solving. Often the pupil is suggested the gradually more difficult tasks. During the solving the pupil can return to the material studied and get some help refilling his knowledge.

In present tutoring systems the most attention is paid to the second stage. The training subsystems are most active component while during concept learning the developers are sated with passive presentation. As researches showed [8], the pupil's knowledge must to reach certain level to avoid antagonism for further tutoring.

In this paper we suggest the tutoring systems technology of interactive conceptual learning including the test of pupil's knowledge. This technology is embodied in software tools. The main features of this software are: the extraction of expert knowledge; the formalization of the domain's concepts; the construction of the knowledge base; the semi automatic production of the test questions; the construction of the pupil's cognitive model; the estimation of the pupils' knowledge; the creation of the tutoring strategy.

The main task of the primary learning is habituation with the domain concepts. For the formalization of these concepts we use the Formal Concept Analysis.

Such approach represents the formal concept like as a pair <intent; extent> where the extent represents the set of objects and the intent is the set of their attributes. For the concept extraction we use the formal context for the corresponding part of the domain. The formal context is represented by the table <object; attribute> in witch every attribute is marked if it is the proper of certain object. For every formal concept maximal nested full submatrix corresponds. The set of concepts is ordered by the relation superconcept - subconcept and forms a full lattice. For testing of the pupil we use the set of questions witch is generated automatically using this lattice. The purpose of testing is to estimate how the pupil grasped the concepts and relations between them. The test questions are mainly of the closed type. The main problem of their recomposing is a choice of destructors. For this we suggest to use the nearest to the concept testing conceptual lattice elements. The new conceptual lattice is creating using the pupils' answers. This lattice represents the domain in the pupils mind. Then we compare it with the master lattice. The differences between them are used to generate a tutoring strategy.

## The Concept Presentation in Tutoring System

The method of Formal Concept Analysis is used for knowledge representation of problem domain. It was suggested by R. Wille [4, 5] and at the present time is successfully applying in the problems of data-mining and computer tutoring [1, 2, 10]. The backbone is follows. Let us consider the set of objects $V$ and the set of attributes $A$ with an arbitrary relation $I \subseteq V \times A$, such that $pIa$, where $p \in V$, $a \in A$, if and only if $a$ is the attribute of object $p$. Then $K = (V, A, I)$ is called the *formal context.* The binary matrix defines the correspondence of objects and attributes. Let define the correspondence [6]:

$$P' := \{y \in A \mid xIy \text{ for all } x \in P\}, \text{ for } P \subseteq V,$$

$$G' := \{x \in V \mid xIy \text{ for all } y \in G\}, \text{ for } G \subseteq A.$$

Then the pairs $(P, G)$ satisfying $P \subseteq V$, $G \subseteq A$, $P' = G$, $G' = P$ are called *the formal concepts* of the formal context $K = (V, A, I)$. The set of objects $P$ amounts the *extent* of concept and the set of all their attributes amounts its *intent.* Every object $p \in P$ have all attributes from subset $G$. So, the formal concept is the set of objects from domain such that every one of them has all attributes from certain subset of attributes of that objects.

The set of formal concepts $(P, G)$, where $P \subseteq V$, $G \subseteq A$, is partially ordered by the relation: $(P_1, G_1) \le (P_2, G_2)$, if $P_1 \subseteq P_2$ and $G_2 \subseteq G_1$, and form a complete lattice $L(K)$, called *the concept lattice* of the context $K$ [4]. The pair $(P_1, G_1)$ is called the subconcept of the concept $(P_2, G_2)$, and the pair $(P_2, G_2)$ is called the superconcept of the concept $(P_1, G_1)$.

The concept lattice can be represented by the line diagram (Hasse diagram) in which every node of the concept lattice is corresponded by the concept from context. The dual isomorphism on the concept lattice reflects the inverse between the intent and extent of the concepts: the bigger is extent the less is intent

Example. The formal context "Geometry Figures" is represented in the Table 1. This context is based on the set of geometry figures and the set of their attributes. Maximal nested full submatrix corresponds for every concept. For example, marked submatrix in Table 1 corresponds to formal concept {<*Triangle*, *Tetragon* (*quadrangle*), *Pentagon*, *Hexagon*>, <*Vertexes*, *Area*, *Sides*, *Angle*>} witch can be defined like a "*polygon*". Hereby formal concept is the set of objects from domain such that every one of them has all attributes from some subset of domain attributes.

Table 1. Context «Geometry Figures»

| | Has Vertexes | Has Length | Is Line | Has Area | Has Sides | Has Angle |
|---|---|---|---|---|---|---|
| Point | | | | | | |
| Straight Line | | | x | | | |
| Half Line | x | | x | | | |
| Straight Line Segment | x | x | x | | x | |
| Angle | x | | | | x | x |
| Circle | | | | x | | |
| Circumference | | x | x | | | |
| Curve Line | | x | x | | | |
| Poly Line | x | x | x | | x | x |
| Triangle | x | | | x | x | x |
| Tetragon (quadrangle) | x | | | x | x | x |
| Pentagon | x | | | x | x | x |
| Hexagon | x | | | x | x | x |

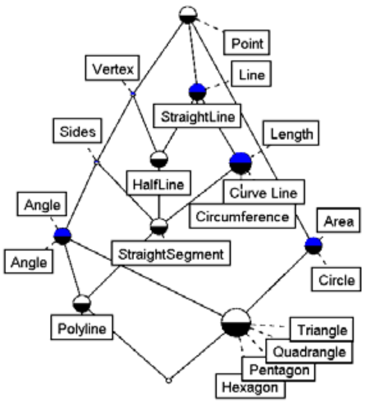



Fig. 1. Conceptual lattice of the context «Geometry figures»

The diagram of conceptual lattice is represented in a Figure 1. By this lattice we can easy find which attributes corresponds to proper object. One concept *(P, G)* corresponds to each node on the diagram. *P* includes the labels of all objects directly underlying, and *G* includes labels of all attributes lying directly above of the given node. The node on the diagram is marked by the label *(p, a)*, where *p* includes the labels of objects, and *a* – labels of attributes directly appropriate to the given node. For example, in Fig. 1 the node with a label «*angle*» and the attribute «*has angle*» corresponds to the concept {<*Angle, Poly line, Triangle, Tetragon (quadrangle), Pentagon, Hexagon*>, <*Vertexes, Sides, Angle*>}. In this context the concept with the empty set of attributes {<*Point*>,$\varnothing$} corresponds to the lattice unit. The point is a primary concept which can not be defined by the set of properties (attributes).

On a concept lattice it is easy to find the *common* attributes for several objects and the *unique* attributes corresponding only to one of objects. For example objects «*triangle*» and «*circle*» has one common attribute "*area*" Likewise easy to find all objects that have certain attribute. For example we can see that all of "Straight Line", "Half Line", "Straight Line Segment", "Circumference", "Curve Line", "Poly Line" are "lines"

One more important result is that the construction of a concept lattice defines the partial order on a set of objects and attributes. Partial ordering of objects and attributes allows us to reveal the dependence between them. Let $K = (V, A, I)$ is a formal context, $X \subseteq A$, $Y \subseteq A$. Then $X \rightarrow Y$, т.е. *X implies Y,* or the set of attributes *Y depends* on the set of  of attributes *X,* if all objects from $P \subseteq V$, that have attributes from *X,* also have all attributes from *Y,* i.e. $X' \subseteq Y'$ (or $Y \supseteq X'$). In this case the concept containing *X* stands lower of the concept containing *Y* on the diagram of a concept lattice.

For example we can see on the diagram that the property "*to have a angle*" is followed by the properties "*to have sides*" and "*to have a vertex*", and property «*to have a length*» is followed by the property «*to be a line*».

The described above features of the concept lattices allow to arrange the pupil's knowledge about subject domain and from other side allow to semi automate the process of the test questions production for the checkout of a pupil's knowledge.

## Representation of the Subject Domain Knowledge

The suggested tools environment uses material is represented by traditional means of hypermedia such as texts, pictures, animation. For the automation of the tutoring and knowledge checkup process we need the structures reflected the connection between the parts of material studied and connections between the main concepts inside every part. The common structure of the subject domain is represented by the ontology. The main structure witch corresponds the knowledge by the fragments is the semantic networks. The construction of the semantic network is process to be difficult formalized. Mainly supposed, that the expert is to do it. He describes the connections between the concepts and objects of the subject domain by the means of graphics visualization [7]. Using the formal contexts allows automating partially the semantic network construction. It is easier to preset the objects and their properties and then using concept lattice to define the main concepts and relations between them. The same process can be used for the checkout of the pupil's knowledge. Constructing his conceptual lattice the pupil reflects the concepts system like he ideates it himself. The main criteria of the full retention of the material are the isomorphism between the conceptual model of the pupil and the master model created by expert.

Transformation of the concept lattice into the semantic network is under next rules. We define the context objects like primary concepts of the semantic network. Then single-place predicates-attributes like *PredicateName*(*X*) *we* transform into two-place predicates like *Function*(*PredicateName, X*), where the *Function* means belonging *X* to the class, type, set or describes other connection between object and attribute (the sets of the objects and attributes can intersect i. e. the same essence can be an object and an attribute. (In our example such essence is a "*angle*"). Than we specify the predicates: every variable is replaced by the object from the predicate truth set according the formal context matrix.

For illustration let us see example above. In context «Geometry Figures» every attribute is a single-place predicate defined on preset set of objects – Geometry figures. Case in point "*X is a Line*", "*X has a Area*" etc. The predicates truth sets marked in a Table 1. We transform all this predicates into two-place predicates *Is(X, Y)* and *Has(X, Y)*. There are only two in our case. Decomposition of the lattice connections allows to construct hierarchy semantic network (fig. 2).

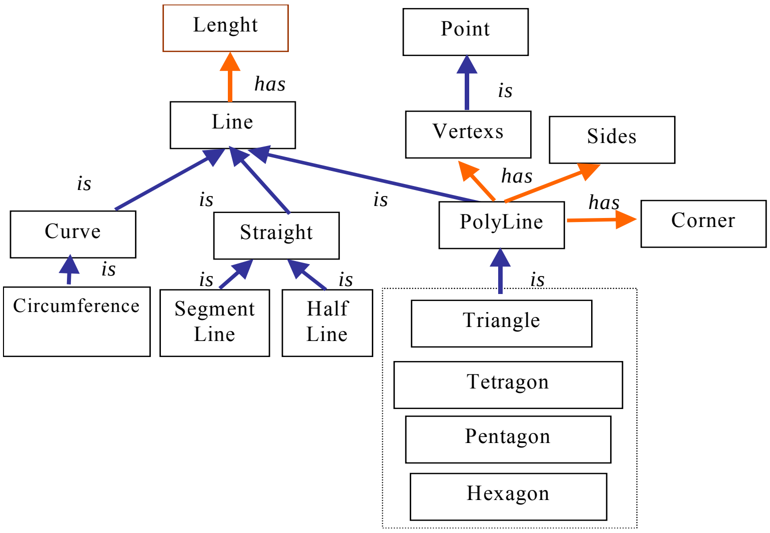



Fig. 2 Hierarchical semantic network for the Context «Geometry Figures»

The data input is realized by creating and filling the context tables. It is possible to organize the row and column titles like hyperlinks to pages that describe the definitions of objects, backbones of their properties and explanations the attributes.

Some objects can have attributes only if the attributes of higher level exist. This allows extracting the attributes of lower level into separate context linked with the main context by the attribute of higher level. Only objects with this attribute are included in it. For example the figures attributed "*Has area*" can be separated to the new context. From the other side some objects can have additional attributes not immanent to other objects. In this case nested context can be created. For example nested context for the object "*Triangle*" will has attribute "*Equilateral*", "*Isosceles*", "*Right*" etc. As the result the context hierarchy (Hypercontext) can be constructed. This structure has a lot of advances. Hypercontext can be processed whole and partially with free choice of processing deepness levels, separate branches, or single contexts. Thus exclude dismissing of the dependent attributes, if the object does not have determinative attribute of higher level. Often necessity of using quantitative attributes appears. Than we use many-valued contexts [4].

Context processing includes next steps: generating of the concept set, creating the line diagram of the concept lattice, extracting of the implications basis. Resulted knowledge system is used by the next way:

– Concepts are raw data for the test generation;
– Implications are used by expert for checkout of context fullness and for test generation;
– Concept lattice is used by test constructor and choice of the tutoring strategy;
– Line diagram is used for visualization of the domain structure fragment.

The implications proved from context can be differed into three groups.

A) Non common sensitive implications. These are ones having false premise on the set of all context objects. Such implications are excluded from further processing.

B) Right (True) implications witch reflects the domain relations correctly. These are relations where the intent of premise and consequence coincides. The accuracy of such implications is 100%.

C) Plausible implications witch reflects the domain relations correctly, but not for all objects, i.e. the intent of premise is bigger than intent of consequence.

For example in context «Geometry Figures» the implication basis includes the right implications:

1. If the Object «*has sides*», that it «*has vertex*» (7 objects);
2. If the Object «*has angle*», «*has vertex and sides*» and (6 objects);
3. If the Object «*has length*», that it «*is line*» (4 objects);
4. If the Object «*has vertex and area*», that it «*has sides and angle*» (4 objects;
5. If the Object «*is line and has vertex and has sides*», that it «*has length*» (2 objects;

6. If the Object «*is line and has vertex and length*», that it «*has sides*» (2 objects;
7. If the Object «*is line and has area* », that it «*has vertex, length, sides, and angle*» (0 objects).

These implications are apparent, because their accuracy in this context is 100%. Besides this there are plausible implications.

8. If the Object «*has vertex*», that it «*has sides*» (accuracy – 86% ).
9. If the Object «*has vertex and sides*», that it «*has angle*» (accuracy – 83%).

All this implications from B and C are suggested for experts checkout as the questions like: «Is it true that IF <common name objects> <attribute $A_{11}$> & <attribute $A_{12}$> & … & <attribute $A_{1n}$>, THAN <attribute $A_{21}$> & <attribute $A_{22}$> & … & <attribute $A_{2n}$>?». Expert can adopt implication or do not. If the expert can not answer "yes" for this question he must to suggest counter-example from the domain and include it into context, or to correct given context if it has a mistake and reprocess it. The process will be done, when expert accepted all implications.

In the context above implication 7 which is false for any objects is excluded. The implications 1-6 which were apparent are accepted as a rule. Implication 8 is false for one object "*segment*" and implication 9 is false for the object "*half-line*". These implications can be cased as a rule with elimination because the concept «*segment*» is basic for such concepts like «*poly-lines*» and "*Polygon*". "*Half-line*" is basic for such concepts like "*angle*". It is possible to add the predicate - «*x consist of* ». Thus our conceptualization will be more complete.

## Testing of the pupils knowledge

Interactive tutoring environment supposes closed iterative cycle of learning which includes such components like: presentation of new material, testing of the pupils' knowledge, constructing of his cognitive model, its comparison with a master one, generation of the tutoring strategy.

This software can automatically generate the tests witch can be used for estimation of pupil's knowledge about main concepts and relations between them.

Any question consists of premise and subject. The subject determines the set of possible answers (explicatively and implicatively). The premise adjects the subject with instruction using witch it is necessary to choose the right answer. The formal concepts let us to test the knowing of their attributes. By the concept (*P, G*) we can generate two types of questions:

- For given object *P* to define the set of their attributes *G*;
- For given set of attributes *G* to define the set of objects *P* witch have all properties from *G.*

Beside this it is possible to recompose tests by intersection of concepts intents and/or extents.

- What common properties have objects $P_i \cup P_j$;
- What objects have properties $G_i \cap G_j$.

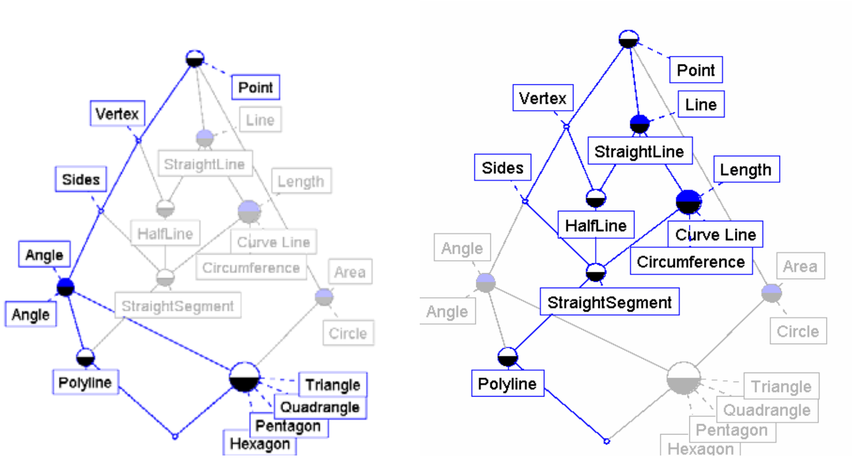
For example, the question can look like: « *What attributes has circumference*? »; « *What objects has an area*? ». The tests like that let us to compose the cognitive model of pupil as the formal context (lattice).

Using implications we can test how the pupil understood the logic and rules of given subject. In the questions composed using implications the rules of the conjunctions elimination are used. For example the question can be like "What *attributes have figures having sides*?" Such questions may be difficult for the pupil. That is why we use not only open form questions but closed form, meaning the questions where one answer from a proposed list must be chosen ('Yes" or "Not"). The exampled question in closed form can be like: "*Is it true that if the object has sides, than it has vertex*?" (answer: «Yes» or «Not»).

The main problem of automatic generation of questions is the choice of destructors i.e. most believable alternatives of answer. Traditionally this problem is solved by the open testing (without the predefined alternatives) and most resent wrong answers become destructors. This solution is quite expensive. That's why we suggest follows.

Let $J(a)$ – ideal, generated by the concept *a*, $a \in L$, where $L$ – initial lattice; $D(a)$ – dual ideal, generated by the concept *a.* It contains all upper elements having path to *a.* For selection of destructors it is necessary to take sequentially concepts $x \in D(a) \setminus \{a\}$ under criteria of minimal distance from *a*, and generate new ideals $J'(x)$. The destructors will be objects from all concepts *y,* where $y \in J'(x) \setminus J(a)$. For attributes testing we generate new dual ideals $D'(x)$ and choose destructors from the set $D(x) \setminus D'(x)$.

For example, we can see on fig. 3,a ideal and dual ideal, generated by the concept ({*Angle, Poly-line, Triangle, Tetragon, Pentagon, Hexagon*}, {*has angle, has sides, has vertex*}).




*a)* *b)*

Fig. 3. Ideals of lattice "Geometry Figures"

The question may be like: «What common properties have *Angle, Poly-line, Triangle, Tetragon, Pentagon, Hexagon?".* To the answer list beside right answers we must add some destructors. Let us construct the ideal generated by the nearest upper concept node (labeled "*sides*"). It includes the object *"segment".* For this node we construct the dual ideal (fig. 3, b). We see that the object *"segment"* has attributes «*is line*» and «*has length*» which belongs to difference between this dual ideal and previous one. This attributes will be taken as destructors.

The questions in which it is necessary to decide if the sentence is correct suppose the answers: "Yes", "No". Questions of this type can be constructed by implications and concepts as well. The questions of this type contains always true sentence. For the construction of the incorrect sentences we can replace some premises between two implications or some attribute sets between concepts. More complete questions suggest arranging the elements by given criteria, combining elements into the groups or to specify the value of … (volume, weight, etc.) of the listed objects". For this type we supposed to use the many-valued contexts. The quantity attributes are contained directly in the concept. That is why this type is not difficult.

## Conclusions

Nowadays computer technologies let us to create intelligent tutoring systems where the knowledge about subject studied meta-knowledge about tutoring control and pupil's knowledge estimation are represented distinctively. Developed technology and software tools using FCA let us not only to visualize concept system of domain but also automate the generating of exercises for acquisition of pupil's knowledge for its testing.

## Bibliography

[1] Burmeister P. Formal Concept Analysis with ConImp: Introduction to basic features // G.Stumme, R.Wille (eds.) Begriffiche Wissensverarbeitung: Methoden und Anwendungen. Springen-Verlag. – 1999.
[2] Kuznetsov S.O., Ob'edkov S.A. Algorithms for the construction of the Set of All Concepts and Their Line Diagram, Dresden, 2000.
[3] Stefanuk V. L. Leaning Level Analysis in Intelligent Tutoring Systems // Japan-CIS Symposium on Knowledge-Based Software Engineering, 1994 (JCKBSE'94). Pereslavl-Zalesski, May 10 - 13, 1994, P. 9 -13.
[4] Wille R. Ganter D. Formal Concept Analysis. Springer –Verlag. Berlin. - 1999.
[5] Wille R. Knowledge Acquisition by methods of Formal Concept Analysis // Preprint № 1238. Technische Hochschule Darmstadt. 1989.
[6] Birkgoff G. Lattice theory. Amer. Math. Soc. Providence / R.I. 1967.
[7] Gavrilova T.A., Choroshevski V.F. Knowledge bases of intelligent systems. Sankt-Petersburg. Piter. 2000 (in Russian).
[8] Golitsin G.A., Petrov V.M. Information – behavior – creation. - M.: Nauka, 1991.
[9] Petrushin V.A. Expert – tutoring systems. – Kiev: Naukova dumka, 1992.
[10] Taran T.A., Sirota S.V. Concept tutoring in intelligent tutoring systems based on formal concept analysis. // J. Artificial Intelligence. Donetsk. №3. 2000. (in Russian).

## Author information

Tatyana A. Taran - National Technical University of Ukraine "KPI" Applied Mathematic Department Av. Peremogy, 37, 03056, Kiev, Ukraine e-mail: taran@pma.ntu-kpi.kiev.ua ; Sergiy V.Sirota - Prosvita publishing Ltd. 46-Shevchenka blwd. Kiev Ukraine Tel/Fax: + 38044 2349523 e-mail: sirota@prosvita.kiev.ua