http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (2): 348-357, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.348.357

Auto-construction for Knowledge Inheritance Hierarchy of Concepts within NKIMath

Qingtian Zeng
College of Information Science and Technology,
Shandong University of Science and Technology, 266510, Qingdao, China

Abstract: This study introduces the knowledge representation scheme for mathematical concepts and the methods to realize knowledge inheritance between concepts in NKIMath, the mathematical knowledge component of NKI (National Knowledge Infrastructure) in China. Within NKIMath, a concept is represented by a knowledge frame, in which the formal definition of the concept is given by a logical formula in first-order logic. When the knowledge acquisition completed, the knowledge relations between concepts are auto-generated by reasoning, which include concept equivalence, concept subsumption, concept overlapping, concept exclusion and concept weak-correlation. With these relations, a three-level knowledge inheritance hierarchy of mathematical concepts can be constructed from the knowledge base, with which the knowledge can be inherited from one concept to another.

Key words: Knowledge representation, mathematical concept, concept relation, knowledge inheritance, concept hierarchy

INTRODUCTION

Mathematical knowledge representation and acquisition are two major tasks in many mathematical applications, including knowledge-based automated theorem proving, integration of different mathematical software systems, mathematical semantic Web and high-level mathematical instruction (Asperti *et al*., 2001a, b; Zeng *et al*., 2004a, b). Recently, with the development and application of Web technology, mathematical markup language for the content and the context of mathematics toward Web, such as MathML (David *et al*., 2003), Open Math (Abbott *et al*., 1996) and OMDoc (Michael, 2001), has been received more attention and a lot of projects about mathematical domain knowledge base has been started including Mbase (Michael and Franke, 2001; Franke and Kohlhase, 2000), HELM (Asperti *et al*., 2001a; HELM), NOWGLI (Asperti and Wegner, 2002), etc.

In 1999, a long-term research project (called the National Knowledge Infrastructure, or NKI) was initiated in China to develop shareable knowledge bases of different domains and relevant underlying systems. Currently, the NKI contains knowledge from 21 domains, e.g., medicine, biology, history, geography, mathematics, music, ethnology and archaeology (Cao *et al*., 2002, 2004; Cao, 2001; Gu and Cao, 2001) and owns more than 350,000 concepts and 2,000,000 domain assertions. The knowledge of NKI is acquired from encyclopedia, dictionaries, handbooks, textbooks and so on, by semi-automatic and automated knowledge acquisition. NKI has two main purposes, one of which is to provide society-oriented knowledge services by Web, telephone, Email and another of which is to provide KAPI (knowledge application programming interface) for computer systems including language system, digital library, machine translation and so on.

NKIMath is the mathematical knowledge component of NKI (Cao *et al*., 2006; Zeng *et al*., 2003; Zeng *et al*., 2004a, 2006; Zeng, 2005), which is important for building the whole NKI project, not only because mathematics is a useful subject, but also mathematics is the foundation of lots of subjects, such as physics, mechanics and so on. According to the problems that are encountered in designing the mathematical knowledge representation language in NKI and after the discussion of ontological assumptions for mathematical objects, two kinds of formalisms for the representation of mathematical knowledge are provided in Cao *et al*. (2006). One is a description logic in which the range of an attribute can be a formula in some logical language and another is a first order predicate logic in which an ontology represented by the description logic is a part of the logical language Cao *et al*. (2006). The knowledge of NKIMath is acquired from mathematical encyclopedia, mathematical dictionaries, mathematical textbooks and s o on (Zeng *et al*., 2003). Knowledge engineers acquire knowledge from the knowledge sources and store them into the knowledge base (Zeng *et al*., 2004a). In order to manage the knowledge acquired within NKIMath and help the knowledge engineers acquire knowledge, a knowledge management platform, NKIMathE, has been designed and developed (Zeng *et al*., 2006).

To represent and acquire the knowledge of mathematics; especially mathematical concepts, the construction of the knowledge inheritance hierarchy between mathematical concepts, are important to realize knowledge inheritance between concepts. However, it is not easy to directly build the relations between concepts during knowledge acquisition since there are too many concepts. This study introduces the knowledge inheritance mechanism and the construction of knowledge inheritance hierarchy between concepts. Within NKIMath, a concept is represented by a knowledge frame, in which the formal definition of the concept is given by a logical formula in first-order logic (Cao *et al*., 2006). By the type of each concept, the concepts are divided into a set of concept categories. Within a concept category, the relations between concepts are defined. When the knowledge acquisition completed, the knowledge relations between concepts are auto-generated by reasoning, which include concept equivalence, concept subsumption, concept overlapping, concept exclusion and concept weak-correlation. With these relations, a three-level knowledge inheritance hierarchy of mathematical concepts can be constructed from the knowledge base, with which the knowledge can be inherited from one concept to another. Finally, an auto-construction algorithm is given to obtain the knowledge inheritance hierarchy between concepts from the knowledge base.

KNOWLEDGE REPRESENTATION FOR MATHEMATICS CONCEPT WITHIN NKIMATH

First, we present an example, the knowledge frame of the concept *Monoid* in NKIMath is shown in Fig. 1. Within the knowledge frame of *Monoid*, the knowledge is organized as four different components, which are:

- **Concept relation:** Relation IS-A indicates the relation between Monoid and Semi-Group, which can be obtained by knowledge reasoning from Formal Definition (FD) attribute of Monoid and Semi-Group
- **Concept statement:** It presents the Name in German, English and Chinese and InFormal Definition (IFD) in natural language for Monoid, respectively
- **Concept formalization:** The parameters and the FD are defined, which can be used for knowledge reasoning. For example, it can decide whether a given set G and a binary operation op on G can compose a Monoid

```
Defframe monoid: Mathematical concept
{
    IS-A: Semi-Group

    Name: Monoid
          : LN  GR
    Context: Group theory
    IFD: A semi-group (G,op) with a unit e is called a monoid

    Parameter: G
              : Type Set (σ)

    Parameter: op
          : Type Set σ×σ→σ
    FD: Semi-Group (G,op)^Have-Unit (G,op)

    Pattern: (<?G>,<?op> 是一个 <!Name-CN>)
    Pattern: (<?G>,<?op> is a <!Name-EN>)
    Pattern: (<?G>,<?op> ist eine <!Name-GR>)
}
```

Fig. 1: The knowledge frame of concept Moniod

- **Translation pattern:** Three translation patters in Chinese, English and German are given, which serves for auto-generation of multi-lingual IFDs for Monoid

In NKIMath, the knowledge for each concept is represented by a frame embedded with a formula in first-order predicate logic and each frame is composed by a set of slots, where slots are either attributes or relations. The main attributes and relations of the knowledge frames for mathematical concepts include.

Name is an attribute presenting the name for each concept, which is distinguished from other concepts and can be referenced to. In different languages, different words are used to express one same concept. For example, Group in English, Gruppe in German are one same concept. A facet attribute LN specifies which language is used to express the name of the concept. Conforming to the XML recommendation, we use the ISO 639 two-letter country codes (EN for English, FR for French and CN for Chinese. . .) as the LN specification. This standard is also conformed to OMDoc (Michael, 2001).

Parameter is a special attribute to indicate the component of a concept and the methods to use the concept, which is a list of variables with Type facet to specify the type of each variable. For example, Monoid has two parameters, one of which is *G* with the type of Set (σ) and another is op with type $\sigma\times\sigma\rightarrow\sigma$. It means that G and op are two components of Monoid and G and op are also two variables for predicate Monoid (G,op), which is used to determine whether a set and an operation compose a Monoid or not.

Context is an attribute to indicate a branch subject of mathematics, the main purpose of which is convenient for knowledge management. In the NKIMath knowledge management environment (Zeng *et al*., 2006) all the concepts with same Context value will be organized into one same subject. For example, Group, Abelian-Group all have the same Context value Group Theory, thus they belong to the same branch.

IFD (Informal Definition) and FD (Formal Definition) are two attributes to present the definitions of one same concept informally and formally towards different purposes.

IFD presents the user-oriented informal definition of the mathematical concept in natural language, which is used for knowledge querying, knowledge regeneration and knowledge instruction. It corresponds to the CMP element of OMDoc (Michael, 2001).

FD presents the machine-oriented formal definition with a logical formula in first-order logic (Cao *et al*., 2006), which is used for knowledge reasoning and theorem proving. FD is one of the most important attributes for mathematical concepts, which presents the formal intension for each concept and it is the basis of mathematical knowledge formalization. FD corresponds to the FMP element of OMDoc (Michael, 2001).

Pattern is an attribute to present the knowledge translation pattern, which serves for auto-generation of multi-lingual mathematical knowledge base to satisfy knowledge requirements of users with different language backgrounds. The Pattern attribute is one of the innovations for the mathematical concept knowledge representation in NKIMath. By now, we have presented the translation patterns in three kinds of languages including Chinese, English and German and realized to translate NKIMath into English and German automatically (Zeng *et al*., 2006).

Relations IS-A, SUBCATEGORY_OF and INSTANCE_OF give the relations between concepts. During the process of knowledge acquisition, these relations are not presented, which are auto-generated by the reasoning between FDs when knowledge acquisition completed.

Within the knowledge frame of a concept, there are other optional attributes, for example, [PROPOSER] gives the person(s) who first proposed the concept, [TIME_OF_PROPOSAL] gives the time of the first proposal about the concept and [PUBLICATION_OF_PROPOSAL] indicates the source(s) where the concept was first published. All these attributes present some historical knowledge about the concept for educational purpose. Attributes PERIOD, INVERSE_FUNCTION, IMAGE, DOMAIN and FIELD etc. are optional and are generally used to specify a mathematical function.

More details about the knowledge representation for mathematical concepts within NKIMath can be seen in Cao *et al*. (2006). In the following discussions of this paper, the Knowledge Base (KB) is a set of knowledge frames of mathematical concepts and it is assumed that KB is complete and correct. Knowledge errors and abnormities verification and checking methods for NKIMath can be seen in Zeng (2005).

CONCEPT CATEGORY AND CONCEPT RELATION

Here, the type of a mathematical concept is defined first, which is used to divided concepts into categories. Within a concept category, the relations between concepts will be defined, which include concept equivalence, concept subsumption, concept overlapping, concept exclusion and concept weak-correlation.

Concept category

Definition 1: The type of a mathematical concept C is type $(C) = \{\sigma|$ there is an object (instance) $o \in E(C)$ such that the type of o is $\sigma\}$ where $E(C)$ is the set of all the instances of concept C.

The type describes the composition of a mathematical concept. For example, the type of concept Group is $\text{Type(Group)} = \{(\text{Set}(\sigma), \sigma\times\sigma\rightarrow\sigma)|\sigma\in \text{MTSet}\}$, where, MTSet is the set of all types. It indicates that every group is composed of a given set with type $\text{Set}(\sigma)$ and a binary operation $\sigma\times\sigma\rightarrow\sigma$ on the set.

Let ConceptSet be the set of mathematical concepts. According to the types of mathematical concepts, an equivalence relation between concepts can be defined, thus ConceptSet can be divided into a set of equivalence classes. Every concept equivalence class is defined as a concept category.

Definition 2 $\forall C_1, C_2 \in$ ConceptSet, C_1 and C_2 satisfy type equivalence relation iff $\text{Type}(C_1) = \text{Type}(C_2)$, denoted as $C_1 \overset{\text{Type}}{\leftrightarrow} C_2$

Proposition 1: The $\overset{\text{Type}}{\leftrightarrow}$ relation over ConceptSet is an equivalence relation.

Proof: It is easy to show that $\overset{\text{Type}}{\leftrightarrow}$ satisfies reflexivity, symmetry and transitivity, so $\overset{\text{Type}}{\leftrightarrow}$ is an equivalence relation.

Definition 3: The concept set ConceptSet can be divided into a set of equivalence classes according to $\overset{\text{Type}}{\leftrightarrow}$ each of which is named as a concept category.

Let $\mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be the set of all the concept categories and let $CAT \in \mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be a concept category. With Definition 2, for any concept $C_1, C_2 \in CAT$, C_1 and C_2 satisfy $\mathrm{Type}(C_1) = \mathrm{Type}(C_2)$. TypeLabel(C) is used to label the types of all the concepts in CAT, where C is any concept of CAT.

Based on the usual formalization of algebraic structures in type theory, the types of structures such as Group and Set (i.e., Type(Group) and Type(Set) in our notations) are Σ-types (Luo, 1999, 2008), with the former having the latter as a substructure. We can define a coercion carrier from groups to sets which extracts the type corresponding to the carrier set of a group. For example, G is a Group on set S and the carrier can be defined as Type(carrier(G)) = Type(S), thus all the properties satisfied on S are also satisfied on carrier(G). This kind of relation, such as there is a coercive function from Type(Group) to Type(Set), is defined as coercive subtyping.

Next, we define the inheritance relation between concept categories based on coercive subtyping.

Definition 4: Let $CAT_1, CAT_2 \in \mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be two concept categories. CAT_2 is a subcategory of CAT_1, or CAT_1 is a supercategory of CAT_2, if there is a coercive function from $\mathrm{TypeLabel}(CAT_1)$ to $\mathrm{TypeLabel}(CAT_2)$. Also, it is said that CAT_2 inherits from CAT_1, denoted as $CAT_2 \ll CAT_1$.

If $CAT_2 \ll CAT_1$ and $\mathrm{TypeLbel}(CAT_1) \neq \mathrm{TypeLabel}(CAT_2)$, CAT_2 is a proper subcategory of CAT_1 and denoted as $CAT_2 \ll CAT_1$.

With the properties of coercive subtyping, relation $\ll$ satisfies reflexivity and transitivity and relation $\ll$ satisfies non-reflexivity, non-symmetry and transitivity. Certainly, there are other relations between concept categories according to type theory (Luo, 1999, 2008). In this study, we only pay more attention to the inheritance relation between concepts, so other relations are not discussed here.

Relations between mathematical concepts: Within a concept category, all the concepts have same types, i.e., they have similar compositions. Between these concepts, five kinds of relations can be defined based on concept intensions (or FD knowledge), which are respectively concept equivalence, concept subsumption, concept overlapping, concept exclusion and concept weak-correlation.

Concept equivalence

Definition 5: Let $CAT \in \mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be a concept category. $\forall C_1, C_2 \in CAT$, C_1 and C_2 satisfy the relation of concept equivalence (denoted as $C_1 \equiv C_2$), if $KB, FD(C_1) \vdash FD(C_2)$ and $KB, FD(C_2) \vdash FD(C_1)$.

In mathematics, different concepts can be used to express one same class of instances. Though the names of these concepts are different, their extensions are equal to each other and their intensions are equal in logic. For example, Abelian-group and commutative-group, complemented distributive lattice and distributive complemented lattice, triangle with equal edges and triangle with equal angles, satisfy the relation of concept equivalence, respectively.

Concept subsumption

Definition 6: Let $CAT \in \mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be a concept category. $\forall C_1, C_2 \in CAT$, if:

- $KB, FD(C_2) \vdash FD(C_1)$, then C_1 implies C_2 or C_2 is a subconcept of C_1, denoted as $C_2 \leq C_1$
- $KB, FD(C_2) \vdash FD(C_1)$ and $KB, FD(C_1) \nvdash FD(C_2)$, then C_1 implies C_2 properly, or C_2 is a proper subconcept of C_1, denotation $C_2 < C_1$ or IS-A(C_2, C_1)

With the definition of $\leq$ ($<$), all the properties of (or propositions about) C_1 are satisfied to C_2, which indicates the logical inheritance from C_1 to C_2. It is easy to prove that $\leq$ ($<$) relation satisfies transitivity and for any C_1, $C_2 \in CAT$, $C_2 \leq C_2$ and $C_1 \leq C_2$ if $C_1 \equiv C_2$.

Concept Overlapping

Definition 7: Let $CAT \in \mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be a concept category. $\forall C_1, C_2 \in CAT$, C_1 and C_2 satisfy the relation of concept exclusion (denoted as $C_2 \propto C_1$ or $C_1 \propto C_2$), if there is a formula φ such that $KB, \varphi \vdash FD(C_1)$ and $KB, \varphi \vdash FD(C_2)$, but $KB, FD(C_1) \nvdash FD(C_2)$ and $KB, FD(C_2) \nvdash FD(C_1)$.

It is easy to prove that $\propto$ satisfies reflexivity, symmetry and non-transitivity. Besides, relation $\propto$ satisfies the following proposition.

Proposition 2: Let $CAT \in \mathrm{ConceptSet}/_{\overset{\mathrm{Type}}{\leftrightarrow}}$ be a concept category. $\forall C_1, C_2 \in CAT$ such that $C_3 < C_2$ and $C_3 < C_1$, then $C_2 \propto C_1$.

Proof: If there is a formula φ such that $KB, FD(C_3) \vdash \varphi$, since $C_3 < C_2$ and $C_3 < C_1$, $KB, FD(C_1) \vdash \varphi$ and $KB, FD(C_2) \vdash \varphi$. So, $C_2 \propto C_1$.

The conclusion of Proposition 2 is obvious. Because C_3 is a subconcept of C_1 and C_2, C_3 is one of the overlapping parts of C_1 and C_2. For example, let C_1 be complemented lattice, C_2 be distributive lattice and C_3 be complemented distributive lattice, obviously C_1 and C_2 satisfy concept overlapping.

Concept exclusion

Definition 8: Let $CAT \in \text{ConceptSet}/_{\overset{Type}{\leftrightarrow}}$ be a concept category. $\forall C_1, C_2 \in CAT$, C_1 and C_2 satisfy the relation of concept exclusion (denoted as $C_1 <\approx> C_2$), if there is a formula φ such that:

(1) KB, $FD(C_1) \vdash \varphi$ and KB, $FD(C_2) \vdash \varphi$
(2) For any other formula φ', KB, $\varphi \nvdash \varphi'$ and KB, $FD(C_1) \vdash \varphi'$ iff KB, $FD(C_2) \vdash \neg\varphi'$

Two concepts satisfying the relation of concept exclusion indicates that they are opposite in logic. Relation $<\approx>$ is a non-reflexivity, symmetry and transitivity relation and $\forall C_1, C_2 \in CAT$, C_1 and C_2 cannot satisfy $\equiv$, $\prec$ and $\propto$ if $C_1 <\approx> C_2$. There are lots of concepts in mathematics satisfying the relation of concept exclusion, such as Commutative Group and Non-Commutative Group.

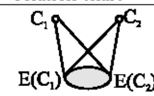
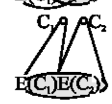
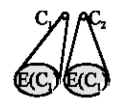
Concept weak-correlation: The four relations between concepts, concept equivalence, concept subsumption, concept overlapping, concept exclusion, indicate two concepts with same types have relations about their logical constraints. If two concepts with same types have no relations about their logical constraints, they are weak-correlation.

Definition 9: Let $CAT \in \text{ConceptSet}/_{\overset{Type}{\leftrightarrow}}$ be a concept category. $\forall C_1, C_2 \in CAT$, C_1 and C_2 satisfy the relation of concept weak-correlation, if they do not satisfy $\equiv$, $\prec$, $\propto$ or $<\approx>$.

Two weak-correlation concepts are consistent only about their structures, but there no any relations about their logical constraints. When the concept categories are divided, it is not expected to put two weak-correlation concepts into one category, which can be overcome by refining the conditions for dividing categories. In this paper, we mainly discuss the inheritance relation to realize knowledge inheritance between concepts, so more details about the conditions for dividing categories are not considered here.

Five kinds of relations between concepts within a concept category are formally defined based on their intensions (or formal definitions). The extension relations between concepts within a concept category can be listed in Table 1.

Table 1: Concept extension relations within a concept category

Concept relation	Extension relation	Extension relation chart
Concept equivalence	If $C_1 \equiv C_2$, then $E(C_2) = E(C_1)$	
Concept subsumption	If $C_2 \leq C_1$, then $E(C_2) \subseteq E(C_1)$ If $C_2 \prec C_1$, then $E(C_2) \subset E(C_1)$	
Concept overlapping	If $C_2 \propto C_1$, then $E(C_2) \cap E(C_1) \neq \phi$ That is, $\exists I \in E(C_2)$ and $I \in E(C_1)$	
Concept exclusion	If $C_1 <\approx> C_2$, Then $E(C_2) \cap E(C_1) = \phi$	
Concept weak-correlation	If C_1 and C_2 are concept weak-correlation, then $E(C_2) \cap E(C_1) = \phi$	

THREE-LEVEL KNOWLEDGE INHERITANCE HIERARCHY OF MATHEMATICAL CONCEPTS

Based on the types of concepts, a given mathematical concept set can be divided into a set of concept categories. All the categories can be organized with the subcategory relation between categories, which can realize structural inheritance between concepts. Within a concept category, the concepts are organized based on the concept subsumption relation, which indicates the logical inheritance between concepts. Every concept includes a class of instances and all the instances satisfy the properties of the concept. Thus, a three-level knowledge inheritance hierarchy between mathematical concepts can be obtained, which is represented in Fig. 2. These three levels in the hierarchy are:

- **Concept category level:** in which every category is represented as a big circle with a broken line and the inheritance relation between categories is represented as a directed arc with a broken line
- **Concept level:** In which every concept is represented as a small circle and the inheritance relation between concepts is represented as a directed arc. The concepts with same types are classified into one same concept category
- **Instance Level:** In which every instance is represented as a small rectangle. The instances with same properties belong to the same concept

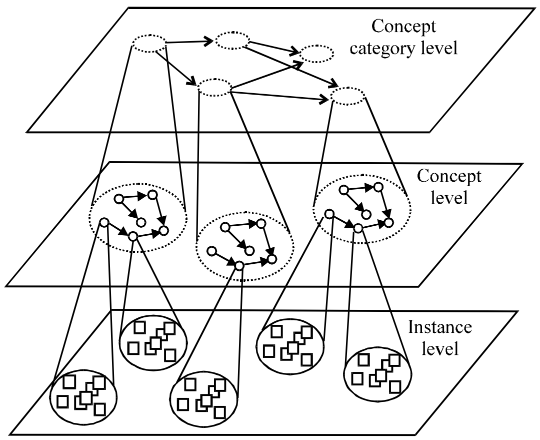


Fig. 2: Three level hierarchy of mathematical concepts

Graph representation for knowledge inheritance hierarchy of mathematical concepts: The concept hierarchy can be represented by a directed graph. The inheritance graph of concept categories and the inheritance graph of concepts are defined respectively as follows.

Definition 10: $G_{CAT} = <V_{CAT}, E_{CAT}, \xi>$ is an inheritance graph between concept categories iff:

- $V_{CAT} = \{CAT \mid CAT \in ConceptSet / \underset{\leftrightarrow}{Type}$ is a concept category$\}$ is the vertex set of G_{CAT}
- $E_{CAT} \subseteq V_{CAT} \times V_{CAT}$ is the directed edge set of G_{CAT}, $\forall v_i, v_j \in V_{CAT}$, $(v_i, v_j) \in E_{CAT}$ iff $v_i \prec\prec v_j$ (i.e., v_i is the proper subcategory of v_j)
- If $(v_i, v_j) \in E_{CAT}$ and f is the coercive function from TypeLabel(v_i) to TypeLabel(v_j), then $\xi(v_i, v_j) = f$ and the edge from v_i to v_j is labeled with f

Based on the definition of the inheritance graph between concept categories, G_{CAT} is a directed graph without any loops. $\forall v_i, v_j, v_k \in V_{CAT}$, $\xi(v_i, v_k) = f \circ g$ if $\xi(v_i, v_j) = f$ and $\xi(v_j, v_k) = g$.

Definition 11: $G = <V, E>$ is a concept inheritance graph iff:

- V = ConceptSet is the set of concepts
- $(v_i, v_j) \in E$ iff $v_i < v_j$

We can represent the concept category inheritance and concept inheritance graph with one combined directed graph. The category vertex is represented by a big circle with a broken line and the inheritance relation between categories is represented by a directed arc with a broken line and every arc is labeled with the coercive function. For every big circle of category, the concept inheritance graph is embedded into it. The concept node is represented by a small circle and the inheritance relation between concepts is represented by a directed arc. Figure 3 shows an example.

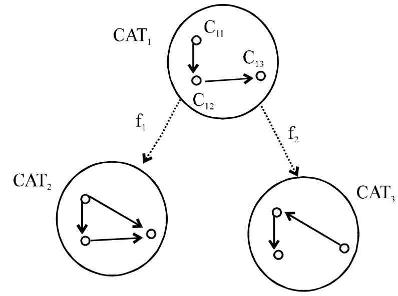


Fig. 3: Directed graph representation for the concept hierarchy

Figure 3, the concept hierarchy is composed by three concept categories (CAT_1, CAT_2 and CAT_3). $\xi(CAT, CAT_1, CAT_2) = f_1$ and $\xi(CAT_1, CAT_3) = f_2$. In each concept category (CAT_1, CAT_2 or CAT_3), the concepts are organized by the inheritance relation.

AUTO-CONSTRUCTION ALGORITHM FOR KNOWLEDGE INHERITANCE HIERARCHY OF MATHEMATICAL CONCEPTS

Auto-construction algorithm: With the analysis for knowledge inheritance hierarchy between mathematical concepts, an auto-construction method for the hierarchy from concept knowledge base can be presented. According to the knowledge representation for the mathematical concepts, the Formal Definition (FD) in the knowledge frame presents the intension and the combination of parameters indicates the type of the concept. We can obtain the concept hierarchy with the knowledge frames of concepts if there are no errors in the frames. The auto-generation algorithm for knowledge inheritance hierarchy between mathematical concepts within NKIMath is presented in Algorithm 1.

Algorithm 1 only presents the framework for auto-generation of knowledge inheritance hierarchy between mathematical concepts within NKIMath. In Step 3 and 4, a reasoning machine is needed to judge the relations between concept categories or concepts.

A running example: In order to show the procedure of Algorithm 1, Table 2 presents the basic knowledge frames for concepts including Group, Ring, Non-associative Ring and so on, which only presents the main information in the knowledge frame, such as name, parameters and FD.

The procedure of auto-generation of the concept hierarchy in Table 2 is presented as in Fig. 4. With the first step, the KF (Knowledge Frame) of Group is read and a node (represented by the little circle in Fig. 4a for Group is produced and a node (represented by the big circle in Fig. 4a for GCC (Concept Category of Group) synchronously. Figure 4b shows the result after reading the KF of Ring. Fig. 4b, Ring is represented by the little circle and RCC (Concept Category of Ring) is represented by the big circle. Since GCC is inherited by RCC, a directed arc is represented from GCC to RCC. Figure 4 shows the whole procedure of the auto-generation of the concept knowledge inheritance hierarchy.

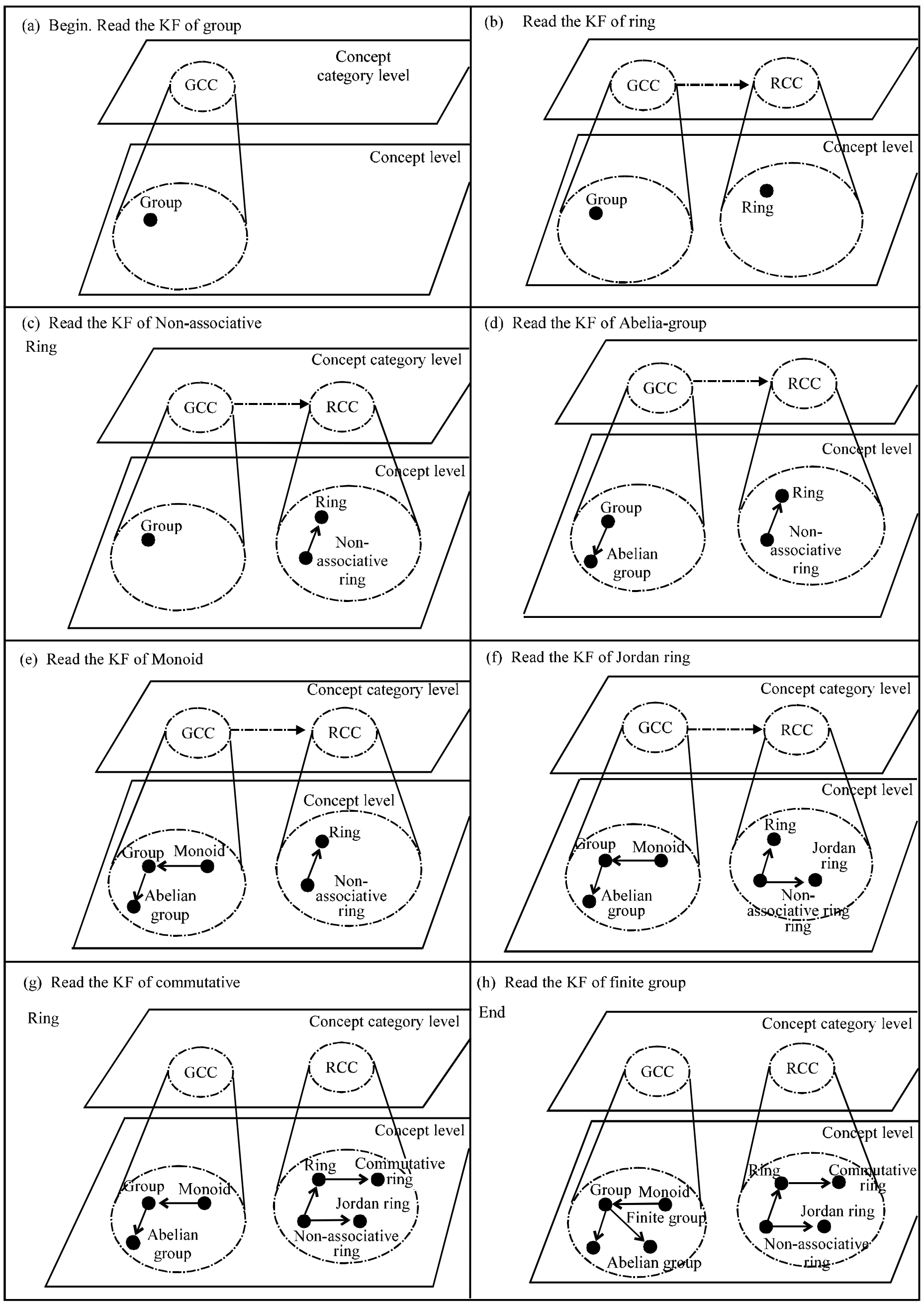


Fig. 4: The procedure of auto-generation of the concept hierarchy

Table 2: Knowledge frames of a set of concepts

Concept	Knowledge frame

Group	Defframe Group: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op
	: Type σ×σ→σ
	FD: Monoid(S, op)^Have_Inverse (S, op)
	}

Ring	Defframe Ring: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op1
	: Type σ×σ→σ
	Parameter: op2
	: Type σ×σ→σ
	FD: Non_associative_Ring (S, op1, op2)^ Associative_Law (S, op2)
	}

Non-associative ring	Defframe Non-associative Ring: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op1
	: Type σ×σ→σ
	Parameter: op2
	: Type σ×σ→σ
	FD: Abelian_Group(S, op1)^Distributive_Law(S, op1, op2)
	}

Abelian-group	Defframe Abelian-Group: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op
	: Type σ×σ→σ
	FD: Group(S, op)^Commutative_Law(S, op)
	}

Monoid	Defframe Monoid: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op
	: Type σ×σ→σ
	FD: Semi_Group(S, op)^Have_Unit (S, op)
	}

Jordan ring	Defframe Jordan Ring: Mathematical concept { Parameter: S : Type Set(σ) Parameter: op1 : Type σ×σ→σ Parameter: op2 : Type σ×σ→σ FD: Non_associative_Ring (S, op1, op2) ^ Commutative_Law (S, op2)}

Commutative ring	Defframe Commutative Ring: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op1
	: Type σ×σ→σ
	Parameter: op2
	: Type σ×σ→σ
	FD: Ring(S, op1, op2)^Commutative_Law (S, op2)
	}

Table 2: Continued

Concept	Knowledge frame

Finite group	Defframe Abelian-Group: Mathematical concept
	{
	Parameter: S
	: Type Set(σ)
	Parameter: op
	: Type
	FD: Group(S, op)^Smaller_Than (Order(S), ∞)
	}

RELATED WORK

Kerber and Kerber (1991) presented a frame-based knowledge representation method for mathematics and the main idea of which comes from object-oriented technology. Kerber and Kerber (1991), every frame is composed by name, a set of slots and fillers. Slot is a set of atom predicates and the filler of every slot is the value of the corresponding atom predicates. Figure 5 shows the knowledge representation for Group in Kerber and Kerber (1991).

Kerber's frame-based knowledge representation for mathematics binds the related knowledge together, thus it is convenient for knowledge organization and maintenance. NKIMath enriched the information within the frame given by Kerber, which can satisfy more requirements of users. For example, IFD (Informal Definition) in natural language can be used for knowledge instruction or knowledge Q-A and knowledge translation patterns can be used for auto-generation of multi-lingual knowledge base to satisfy different requirements of users with different native language. In the frame given by Kerber, it is difficult to construct the knowledge inheritance hierarchy of mathematical concepts directly, so it is also difficult to realize knowledge inheritance between mathematical concepts. Within NKIMath, the knowledge inheritance between concepts can be realized by knowledge reasoning and the auto-construction algorithm for the hierarchy is given.

OMDoc is used for knowledge representation in many mathematical knowledge systems (Michael, 2001) and it will be accepted as the international standard for mathematical document marked language. The knowledge representation of NKIMath follows lots of similarities with OMDoc, for example, the FD and IFD attributes in NKIMath are corresponding to the FMP and CMP elements in OMDoc, respectively. However, the knowledge inheritance between concepts is also not defined by OMDoc. Although, OMDoc can provide hyperlink relation between, it is different from the knowledge inheritance relation between concepts and it

Definition: Group	(Property)
Parameter: G: (τ→o)	//called: carrier
Parameter: +: (G×G→G)	//called: operation
Optional parameter: 0: G	//called: neutral element
Optional parameter: -: G	//called: inverse operation
definational: TRUE	
Superconcepts: (1) associative (G, +)	
(2) ex_neutral_element(G,+,0)	
(3) ex_inverse(G,+,0,-)	
equivalences:	
(1) associative(G,+) ^ ex_left_neutral_element(G,+,0) ^ ex_left_inverse(G,+,0,-)	
(2) associative(G,+) ^ ex_right_neutral_element(G,+,0) ^ ex_rightinverse(G,+,0-)	
Context: Basic algebra	

Fig. 5: Knowledge representation for group with kerber's method

is difficult to realize knowledge inheritance between concepts based on hyperlink relation. The hyperlink relation between concepts is based on the definitions of concepts provide by OMDoc, which is useful to acquire knowledge of concept according to the definition relations to ensure the knowledge complete. A concept-oriented knowledge acquisition method is given and adopted within NKIMath (Zeng *et al.*, 2003).

A knowledge management environment for mathematical concepts within NKIMath, NKIMathE, has been developed to manage the acquired knowledge frames acquired (Zeng *et al.*, 2004a, b). To auto-construct the knowledge inheritance hierarchy between concepts within NKIMath based on the algorithm presented in this study is one of the most important functions of NKIMathE. The knowledge inheritance hierarchy between concepts is convenient for knowledge browse, search and edit. According to the knowledge inheritance hierarchy between concepts, it is easy to discover the backbone about the concepts within NKIMath.

CONCLUSION AND FUTURE WORK

To realize knowledge inheritance between concepts is important to knowledge representation and acquisition of mathematical concepts. In this study, we introduce the knowledge inheritance mechanism between concepts within NKIMath and a three-level knowledge inheritance hierarchy and its auto-construction method are given. There are three main contributions in this study:

- Relation analysis between concepts. Concept category is obtained based on concept types and five kinds of relations between concepts within a concept category are defined
- Three-level knowledge inheritance hierarchy to realize structural and logical knowledge inheritance between concepts
- Graph representation and auto-construction methods for knowledge inheritance hierarchy between concepts

By now, we only consider the relations between concepts based on their formal definitions. In fact, there are many mathematical theorems (lemmas, propositions, etc.) that also build the relations between mathematical concepts. In the future, we will consider the knowledge representation and acquisition for mathematical theorems and have research on how to obtain relations between concepts with the knowledge of theorems. At the same time, a reasoning machine will be designed and developed to realize knowledge reasoning to construct the knowledge inheritance hierarchy automatically.

ACKNOWLEDGMENT

This research is supported partly by the NSFC under Grant No.60603090, 90718011 and 50875158; Science and Technology Development Fund of Shandong Province of China (2010GSF10811); the Excellent Young Scientist Foundation of Shandong Province of China under Grant No. BS2009DX004; the Special Fund for Fast Sharing of Science Paper in Net Era by CSTD (20093718110008); the Open Project of Computer Architecture Lab of ICT, CAS, (No.ICT-ARCH200807); the Research Foundation of Shandong Educational Committee under Grant No. J08LJ77 and the Taishan Scholar Program of Shandong Province.

Algorithm 1: Auto-construction of the knowledge inheritance hierarchy of mathematical concepts

INPUT: Knowledge Frame Set of Concepts
OUTPUT: Concept Knowledge Inheritance Hierarchy
Step 1: Read the knowledge frame KF(C) of concept C;
Step 2: According the parameter definitions in KF(C), to obtain the type of C, denoted as Type(C);
Step 3: If Type(C) is not in the known type set TypeSet, then add Type(C) into TypeSet and add a new category node NewCat into the concept category set V_{Cat}. For all category (denoted as OldCat) in V_{cAT}, if OldCat << NewCat, then add (OldCat, NewCat) into the category inheritance relation set E_{CAT}.
Step 4: If Type(C) is in a known type set TypeSet, then add a new concept nod for C into one concept category (CatV) according Type(C). For all concept (denoted as OldC) in CatV, if Old<C, then add (OldC, C) into the concept inheritance relation set CatE.
Step 5: Output (V_{CAT}, E_{CAT}) and (CarV, CatE)

REFERENCES

Abbott, J., A. Diaz and R.S. Sutor, 1996. A report on openmath: A protocol for the exchange of mathematical information. SIGSAM Bull., 30: 21-24.

Asperti, A. and B. Wegner, 2002. MOWGLI- A new approach for the content description in digital documents. Proceedings of the 9th International Conference on Electronic Resources and the Social Role of Libraries in the Future, Autonomous Republic of Crimea, June 8-16, Ukraine, pp: 215-219.

Asperti, A., L. Padovani, C. Sacerdoti Coen and I. Schena, 2001a. Formal mathematics on the web. Presentation at Crimea 2001. http://www.nbuv.gov.ua/articles/crimea/2001/tom/sec4/Doc24.html.

Asperti, A., L. Padovani, C. Sacerdoti Coen and I. Schena, 2001b. HELM and the Semantic Math-Web. In: Lectures Notes in Computer Science, Boulton, R.J. and P.B. Jackson (Eds.). Springer, Berlin, pp: 59-74.

Cao, C.G., 2001. Medical Knowledge Acquisition from the Electronic Encyclopedia of China. In: Lecture Notes in Computer Science, Ouaglino, S., P. Barabona and S. Anderson (Eds.). Springer Verlag, Berlin, pp: 268-271.

Cao, C.G., Q.Z. Feng, Y. Gao, F. Gu and J.X. Si *et al*., 2002. Progress in the development of national knowledge infrastructure. J. Comput. Sci. Technol., 17: 523-534.

Cao, C.G., H.T. Wang and Y.F. Sui, 2004. Knowledge modeling and acquisition of traditional Chinese herbal drugs and formulae from text. Artificial Intel. Med., 32: 3-13.

Cao, C.G., Y.F. Sui, Y. Sun and Q.T. Zeng, 2006. Representation of mathematical knowledge in national knowledge infrastructure. J. Software, 17: 1731-1742.

David, C., P. Ion, R. Miner and N. Poppelier, 2003. Mathematical markup language (MathML)(Version 2) 2nd Edn. W3C Recommendation 21 October 2003. http://www.w3.org/TR/2003/REC-MathML2-20031021/.

Franke, A. and M. Kohlhase, 2000. System description: MBASE, an open mathematical knowledge base. Proceedings of the 17th International Conference on Automated Deduction, June 17-20, Springer-Verlag, London, pp: 455-459.

Gu, F. and C.G. Cao, 2001. Biological knowledge acquisition from the electronic encyclopedia of China. Proc. 6th Int. Conf. Young Comput. Scientists, 3: 1199-1203.

Kerber, M. and M. Kerber, 1991. Useful properties of a frame-based representation of mathematical knowledge. SEKI Report SR-91-06. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.7937.

Luo, Z.H., 1999. Coercive subtyping. J. Logic Computation, 9: 105-130.

Luo, Z., 2008. Coercions in a polymorphic type system. Mathematical Struct. Comput. Sci., 18: 729-751.

Michael, K. and A. Franke, 2001. MBase: Representating knowledge and context for the integration of mathematical software system. J. Symbolic Comutation, 32: 365-402.

Michael, K., 2001. OMDoc: An open markup format for mathematical documents. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.8086.

Zeng, Q.T., C.G. Cao, Y.F. Sui, J.X. Si, G.G. Tian and H.W. Liu, 2003. Ontology-based knoweldge acquistion and knowledge inheritance mechanism for mathematics. Microelectronics Comput., 20: 19-27.

Zeng, Q.T., C.G. Cao and Y.F. Sui, 2004a. Mathematics on the web. Rep. Inform. Technol., 13: 8-19.

Zeng, Q.T., C.G. Cao, G.G. Tian, Y.F. Sui and H.W. Liu, 2004b. Methods for mathematical concept knowledge representation, acquisition and management within NKIMath. Proceedings of the 2nd International Conference on Knowledge Economy and Development of Science and Technology, Sept. 17-19, Beijing, pp: 27-35.

Zeng, Q.T., 2005. Research on knowledge acquisition and analysis of mathematical concepts. Ph.D. Thesis, Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

Zeng, Q.T., C.G. Cao, H. Duan and Y.Q. Liang, 2006. NKIMathE-A Multi-purpose Knowledge Management Environment for Mathematical Concepts. In: Lecture Notes in Computer Science, Lang, J., F. Lin and J. Wang (Eds.). Springer Verlag, Berlin, pp: 625-636.