维普资讯 http://www.cqvip.com

课件知识点操作的形式化研究*)

肖 锟[1] 李晓建[1] 陈世鸿[1,2]
(武汉大学计算机学院 武汉 430072)[1] (武汉大学国家多媒体软件工程技术研究中心 武汉 430072)[2]

摘 要 知识点关系是连接知识点的纽带。教学系统智能性体现的一个重要方面是能够根据知识点之间的关系组织教学内容。受学生学习知识的能力和课时影响等原因,知识点之间的关系会经常发生动态变化。本文通过分析知识点之间的基本关系,形式化知识点的操作语义,通过操作语义检查,能保知识点动态变化情况下知识点间的关系依旧能保持一致性,为课件的动态生成提供理论依据。

关键词 知识点关系,形式化,课件

An Operation Formalization of Knowledge Points in Course-ware

XIAO Kun[1] LI Xiao-Jian[1] CHEN Shi-Hong[1,2]
(School of Computer, Wuhan University, Wuhan 430072)[1]
(National MultiMedia Software Engineering Technology Research Center, Wuhan University. Wuhan 430072)[2]

Abstract The relationship among the knowledge points (KP) is the key for the KP. It is an important for intelligent tutor systems (ITS) to teach students in proper orders according to the knowledge points relationship (KPR). Due to the different teaching purpose, the KPR is often dynamically changed. This paper formalizes an operation semantic of knowledge points to provide a theoretical basis for the dynamic generation course-ware. At the same time, we also analyze the fundamental relationship among the KP in the dynamic evolution to maintain the consistency of an overall KP structure.

Keywords KPR, Formalization, Course-ware

1 引言

知识点关系是连接知识点的纽带,教学系统智能性体现的一个重要方面是能够根据知识点之间的关系组织教学内容。每一知识点具有多种不同类别的相关素材(具有独立含义的教育资源元素),可以组成不同的课件,供程度不同、基础不同、方式不同的学习者学习,这种素材与知识点结合的方式有利于教师个性化、自主性组织教学内容。但目前大多数智能教学系统还不适应以下几方面问题:(1)随着科学发展,教学大纲中规定的知识点存在着合并、分解、增加、取消等现象;(2)不同教学目的的教学大纲对知识点的掌握有不同的要求,在一些情况下,部分知识点内容可以增减;(3)针对不同学习能力的学生,课件的内容应有所变化;(4)受课时影响,教授的知识点会发生变化。以上提出的这些问题都牵涉到知识点动态变化下针对不同的教学知识点,如何组织现有素材,生成符合知识点规范的课件。文[1~4]关注课件结构的分析、设计,文[5]关注知识点网络特性关系的描述,它们对知识点动态变化语义的形式化研究略显不足,未能解决对已入库素材如何自适应知识点变动,重新组织生成新课件的问题。本文从分析知识点之间的基本关系出发,形式化知识关系的基本操作,为课件的自动化生成提供理论依据。

本文第2节叙述知识点之间的关系;第3节阐述知识关系操作的形式化;最后对全文进行总结。

2 知识点关系

知识点之间的关系简称为知识关系,它是随着知识点的变化而改变的。知识点与素材之间的关系是内容与形式的关系,素材是用来描述知识点的,是知识点的载体。一份良好的课件,体现了素材序列与知识关系序列的一致性。

2.1 知识点间的几种关系

通过对知识点的分析知,知识关系一般分为三种:

(1)父子关系

若干知识点有机组合,形成知识点间的层次关系,称为父子关系。又可分为:

1)包含关系

表示某知识点包含若干知识点。设知识点A、B有机组合起来可以形成知识点C,则有 $A\in C, B\in C$,用 Contain(C,A)和 Contain(C,B)表示,又称C为A、B的父知识点,A、B为C的子知识点。同一父知识点之间的子知识点彼此称为兄弟知识点。

2)属于关系

表示某知识点是另一复合知识点的子知识点,是包含关系的逆关系。用 Belonging(A,B)表示知识点A是知识点B的子知识点。

(2)有序关系

若干知识点之间存在一定的前后顺序关系,称为有序关系。有两类有序关系:

1)前导关系

在学习某一知识点之前必须先学习相关的另一知识点,这两者之间的关系即为前导关系。设在学习知识点B前必须学习知识点A,则有 $\neg A \rightarrow \neg B$,用 Pre_relation(A,B)表示。

2)后续关系

*)国家高技术研究发展计划(863计划),编号2002AA111010。**肖 锟** 博士生,主要从事软件工程、知识工程研究;**李晓建** 博士生,从事多媒体技术、知识发现、数据挖掘研究;**陈世鸿** 教授,博士生导师,主要从事软件工程、知识工程、多媒体技术研究。

维普资讯 http://www.cqvip.com

在学习某一知识点之后，由此知识点直接支持的知识点，这两者之间构成后续关系。设知识点 B 为知识点 A 直接支持的知识点，则有 B→A，用 Succ_relation(B,A)表示。

(3)平行关系

两知识点之间无父子关系或有序关系，学习时可不分先后次序，它们之间就构成平行关系。为描述平行关系，引入符号⊥，为一特殊知识点，表示本课程或本课件学习的终止；符号 t 表示从一知识点起至课程学习终止或至另一知识点止的学习序列。知识点 A 和知识点 B 的平行关系，可表述为

1) 绝对平行关系：知识点 A 和知识点 B 为平行关系，如果它们的后续知识点永不相遇，即知识点 A 与知识点 B 均可单独结束课程(课件)学习，则$(\exists t_{A\to\perp}(A\to\perp))\wedge(\exists t_{B\to\perp}(B\to\perp))$，$t_{A\to\perp}$表示从知识点 A 至课程(课件)终止的学习序列，$t_{B\to\perp}$表示从知识点 B 至课程(课件)终止的学习序列，用 A_Par_relation(A,B)表示；

2)相对平行关系：知识点 A 和知识点 B 为平行关系，如果存在知识点 C 为知识点 A 和知识点 B 的共同后续，则$(\exists t_{A\to C}(A\to C))\wedge(\exists t_{B\to C}(B\to C))\wedge(\exists t_{C\to\perp}(C\to\perp))$，用 $O_Par_relation(A,B)$表示。

平行关系不会直接影响教学知识点序列，本文不予以讨论。

2.2 知识关系属性描述

用∑表示课件中全体知识点的集合；'表示某一操作后属性集或属性集变元的变化；字母 β、γ、δ、ζ 表示知识关系属性集，其中 β 为包含关系属性集，γ 为属于关系属性集，δ 为前导关系属性集，ζ 为后继关系属性集；大写字母 A、B、C 表示知识点；X 表示某个属性集，X∈{β、γ、δ、ζ}；A. X 表示知识点 A 的知识关系属性集 X；existence(X,y)表示属性集 X 中含有 y 属性；|X|表示属性集 X 的属性个数；Xj 表示属性集 X 的第 j 个属性(0≤j<|X|)。

在描述知识关系属性前，先给出如下定义。

定义 1(根知识点)　没有父知识点的知识点为根知识点。若 $A\in\sum\wedge\forall_{B\in\sum}\to\neg Contain(B,A)\wedge\neg Belonging(A,B)$，则称知识点 A 为根知识点。

定义 2(同一知识点)　两个知识点所有的知识关系属性均相同，则为同一知识点。若 $A\in\sum,B\in\sum,r\in\{\beta,\gamma,\delta,\zeta\}$，有 $\forall y\forall A.X\exists B.X((A.X=B.X)\wedge(existence(A.X,y)=existence(B.X,y))\wedge\forall y\forall B.X\exists A.X((B.X=A.X)\wedge(existence(B.X,y)=existence(A.X,y))\to A=B$，则称知识点 A、B 为同一知识点。

下面对知识关系属性状态所涉及的一些行为进行描述。令 $f(A,B)\in\{Contain(A,B),Pre_relation(A,B),Succ_relation(A,B),Belonging(A,B)\}$，X、Y 为某个知识关系属性集，$y,y_j\in Y$：

1) invariability(X',X)表示操作前后 X 不发生变化

$invariability(X',X)\leftrightarrow\forall j\{(0\leqslant j<|X|)\to(X_j'=X_j)\}$

2) accession(X,y)表示在 X 中添加 y

$accession(X,y)\leftrightarrow\forall j\{((0\leqslant j<|X|\to(X_j'=X_j))\wedge((j=|X|)\to(X_j'=y))\wedge(|X'|=|X|+1)\}$

3) addition(X,y)表示如果 y∉X，则在 X 中增加 y

$addition(X,y)\leftrightarrow(\neg existence(X,y)\to accession(X,y))\vee(existence(X,y)\to(X'=X))$

4) deletion(X,y)表示如果 y∈X，则在 X 中删除 y

$deletion(X,y)\leftrightarrow(existence(X,y)\to(\forall j(((0\leqslant j<|X|\wedge X_j\neq y))\to(X_j'=X_j)\wedge((j<|X|)\to(X_j'=X_{j+1}))\wedge(|X'|=(|X|-1)))))\vee(\neg existence(X,y)\to X'=X)$

5) copy(A. X,B. X)：表示将 B 的 X 属性集复制到 A

$copy(A.X,B.X)\leftrightarrow\forall j((0\leqslant j<|B.X|\to((existence(B.X,f_j(B,C))\wedge\neg existence(A.X,fj(A,C)))\to addition(A.X,fj(A,C))))$

6) $addition_All(X,Y)$表示如果$(Y\text{-}X\cap Y)\neq\emptyset$，则在 X 中增加元素 $y_i=(Y-X\cap Y)(i=1,2\cdots,|Y|-|X\cap Y|)$

addition_All(X,Y)↔∀j((0≤j<|Y|→(¬existence(X,y_j)→addition(X,y_i))∨(existence(X,y_j)→$X_j'=X_j$)))

3 知识点操作语义

3.1 基本操作语义

知识点的基本操作模式有四种：创建(生成一个新的知识点)、挂接(某知识点成为另一知识点的子知识点)、关联(确定两知识点间的前导和后续关系)、删除(取消此知识点)。

(1)创建空知识点(∅)Create(A)：生成一个新的不含有知识关系的知识点 A，|A. X|=0，X∈{β、γ、δ、ζ}。

(2)挂接知识点 Hang(A,B)：将知识点 A 挂接到知识点 B 上，形成知识点间的包含关系。在知识点 B 的包含属性集中添加包含关系 Contain(B,A)，在知识点 A 的属于属性集中添加属于关系 Belonging(A,B)。

前置条件：

$B\neq\perp\wedge\neg existence(B.\beta,Contain(B,A))\wedge\neg existence(A.\gamma,Belonging(A,B))$

后置条件：

$addition(B.\beta,Contain(B,A))\wedge invariability(B.\zeta',B.\zeta)\wedge invariability(B.\gamma',B.\gamma)\wedge invariability(B.\delta',B.\delta)\wedge addition(A.\gamma',Belonging(A,B))\wedge invariability(A.\zeta',A.\zeta)\wedge invariability(A.\beta',A.\beta)\wedge invariability(A.\delta',A.\delta)$

(3)有序知识点 Link(A,B)：指定知识点 A 必须在知识点 B 前学习，亦知识点 B 是知识点 A 的后续知识点。在知识点 A 的前导属性集中添加前导关系 Pre_relation(A,B)，在知识点 B 的后续属性集中添加后续属性 Succ_relation(B,A)，对 B 为⊥时需单独考虑。

前置条件：

$A\neq\perp\wedge\neg existence(A.\delta,Pre_relation(A,B))\wedge\neg existence(B.\zeta,Succ_relation(B,A))$

后置条件：

1) 当且仅当 A 为课件某学习序列 t 的最后一个可学习知识点↔B=⊥：

若 Pre_relation(A,⊥)，则操作不对知识关系产生影响，否则在 A. δ 中添加属性 Pre_relation(A,⊥)。

$(existence(A.\delta,Pre_relation(A,\perp))\to invariability(A.\gamma',A.\gamma)\wedge invariability(A.\zeta',A.\zeta)\wedge invariability(A.\beta',A.\beta)\wedge invariability(A.\delta',A.\delta))\vee(\neg existence(A.\delta,Pre_relation(A,\perp))\to addition(A.\delta,Pre_relation(A,\perp))\wedge invariability(A.\zeta',A.\zeta)\wedge invariability(A.\beta',A.\beta)\wedge invariability(A.\delta',A.\delta))$

2) B≠⊥

$(addition(A.\delta,Pre_relation(A,B))\wedge invariability(A.\zeta',A.\zeta)\wedge invariability(A.\beta',A.\beta)\wedge invariability(A.\gamma',A.\gamma))\wedge(addition(B.\zeta,Succ_relation(B,A))\wedge invariability(B.\gamma',B.\gamma)\wedge invariability(B.\beta',B.\beta)\wedge invariability(B.$

维普资讯 http://www.cqvip.com

$\delta', B.\delta))$

(4)删除知识点 Delete(A)：删除指定的知识点 A，要求知识点 A 不能有子知识点且不能是其它知识点的前导知识点。知识点 A 删除后，A 的所有前导知识点的前导属性集中删除与 A 相关的前导属性。如前导知识点的后续属性集中未包含有⊥，则需添加⊥属性；A 的所有父知识点的包含属性集中删除与 A 有关的包含属性；

前置条件：

$|A.\delta|=0 \wedge |A.\beta|=0$

后置条件：

对前导知识点：

$\forall_{K\in\Sigma}(existence(K.\delta, Pre_relation(K,A)) \rightarrow ((\neg existence(K.\delta, Pre_relation(K,\perp)) \rightarrow addition(K.\delta, Pre_relation(K,\perp))) \vee (existence(K.\delta, Pre_relation(K,\perp)) \rightarrow invariability(K.\delta', K.\delta))) \wedge deletion(K.\delta', Pre_relation(K,A)) \wedge invariability(K.\zeta', K.\zeta) \wedge invariability(K.\beta', K.\beta) \wedge invariability(K.\gamma', K.\gamma))$

对父知识点：

$\forall_{K\in\Sigma}(existence(K.\beta, Contain(K,A)) \rightarrow (deletion(K.\beta', Contain(K,A)) \wedge invariability(K.\zeta', K.\zeta) \wedge invariability(K.\delta', K.\delta) \wedge invariability(K.\gamma', K.\gamma)))$

3.2 复合操作语义

复合操作分合并知识点和分裂知识点两类操作：

(1)合并知识点的前提是两知识点或为根知识点，或是父子知识点，或是兄弟知识点。合并操作后，必须确保其余知识点与新知识点间的关系不发生变化，新知识点必须继承原知识点与其它知识点所有的知识点关系。依据两知识点间所处位置的不同，有三种合并形式。

1.1) 合并兄弟知识点 Merge_brothers(A,B)：表示将知识点 B 的属性合并到知识点 A 上。由于多个知识点的合并可以简化为若干次两个知识点的合并过程，因而这里仅讨论两知识点合并中知识关系的变化过程。合并兄弟知识点要求合并两知识点的属于关系；包含关系和针对其它知识点的关联关系，同时还要在知识点 A 中删除两知识点彼此间的关联关系。

前置条件：

$\exists_{X\in\Sigma}(\text{existence}(X.\beta, \text{Contain}(X,A)) \wedge \text{existence}(X.\beta, \text{Contain}(X,B)))$

后置条件：

$(copy(A.\beta, B.\beta) \wedge copy(A.\gamma, B.\gamma) \wedge copy(A.\zeta, B.\zeta) \wedge copy(A.\delta, B.\delta)) \wedge ((existence(A.\delta, Pre_relation(A,B)) \rightarrow deletion(A.\delta, Pre_relation(A,B))) \vee ((existence(A.\delta, Succ_relation(A,B)) \rightarrow deletion(A.\delta, Succ_relation(A,B))) \wedge (|B.\delta|=0) \wedge (|B.\beta|=0) \wedge Delete(B))$

1.2) 合并父子知识点 Merge_paternity(A,B)：仅在父知识点 A 有唯一的子知识点 B 且子知识点 B 有唯一父知识点 A 时方可进行合并。父子知识点合并后，A 拥有 B 所有的关联关系和父子关系且在 A 的包含属性集中需删除与 B 有关的包含属性值。

前置条件：

$existence(A.\beta, Contain(A,B)) \wedge existence(B.\gamma, Belonging(B,A)) \wedge (|A.\beta|=1) \wedge (|B.\gamma|=1)$

后置条件：

$copy(A.\beta, B.\beta) \wedge copy(A.\zeta, B.\zeta) \wedge copy(A.\delta, B.\delta) \wedge deletion(A.\beta, Contain(A,B)) \wedge (|B.\delta|=0) \wedge (|B.\beta|=0) \wedge Delete(B))$

1.3) 合并根知识点 Merge_root(A,B)：非兄弟知识点和父子知识点在仅为根知识点时方允许合并。设知识点 A 是知识点 B 的前导知识，知识点 B 是知识点 A 的后续知识。Merge_root(A,B)表示将 B 知识点合并到 A 知识点。

前置条件：

$|A.\zeta|=0 \wedge |B.\zeta|=0 \wedge B\neq\perp \wedge existence(A.\delta, Pre_relation(A,B)) \wedge existence(B.\delta, Succ_relation(B,A))$

后置条件：

$copy(A.\beta, B.\beta) \wedge copy(A.\zeta, B.\zeta) \wedge deletion(A.\delta, Pre_relation(A,B)) \wedge (|B.\delta|=0) \wedge (|B.\beta|=0) \wedge Delete(B))$

(2)分裂知识点 Split$(A, A_1, A_2\cdots, A_n)$：将知识点 A 分裂为 $A_1, A_2\cdots, A_n$。设$\Sigma_1=\{A_1, A_2\cdots, A_n\}$。知识点 A 的各种属性关系由 $A_1, A_2\cdots, A_n$ 的各属性关系承担，A 的父知识点与 $A_1, A_2\cdots, A_n$ 有包含关系。设 $B\in\Sigma$ 且 $B\neq A \wedge B\notin\Sigma_1$

前置条件：

$A\neq\perp \wedge A\neq\emptyset$

后置条件：

$\forall_{X\in\{\beta,\gamma,\delta,\zeta\}}\forall_i((0\leqslant i<|X|)\forall_{K\in\Sigma}((existence(A.X, f_i(A,B)) \rightarrow \exists_j(0\leqslant j<n existence(A_j.X, fi(A,B))) \wedge (existence(K.\beta, Contain(K,A)) \rightarrow \forall_j(0\leqslant j<n(addition(K.\beta', Contain(K,A_j)) \wedge invariability(K.\zeta', K.\zeta) \wedge invariability(K.\delta', K.\delta) \wedge invariability(K.\gamma', K.\gamma)) \wedge (existence(K.X, f(B,A)) \rightarrow \exists_j(0\leqslant j<n) existence(A_j.X, f(B,A_j))))$

3.3 基于知识点变动的课件自动生成描述

课件是由若干素材组成的，一份良好的课件，体现了素材与知识点间的一致性。当知识点发生变动时，课件中素材之间的关系(简称素材关系)也要发生相应变化以适应知识点变化。

以知识点挂接操作为例：设素材 a 是表述知识点 A 的素材，当将知识点 A 挂接到知识点 B 时，依据操作 $Hang(A,B)$ 的前置和后置条件，可知素材 a 也是用来表述知识点 B 的。若知识点 C 是知识点 B 的后续知识点，则在学习知识点 C 前，素材 a 也必须学习。在自动生成课件时，课件必须包含素材 a。依此关系可自动生成符合教学要求的课件。

结论与下一步工作 课件的教学可用性很大程度上依赖于对知识关系的准确把握，知识点操作的语义形式化为基于知识关系动态变化时的课件自动生成提供了理论依据。针对知识关系变化，素材经组合能形成反映知识关系的新课件。下一步工作是在此基础上研究知识关系变动情况下素材之间关系的一致性。

参 考 文 献

1 Hasegawa S, Ochimizu K. A Learning Management System Based on the Life Cycle Management Model of e-Learning Courseware [C]. In: the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05), 2005

2 Rodrigues E S, Ferreira J B, Pavani A M B. Designing Web Courseware: An Experince with Langage Exercises[C]. In: International Conference on Engineering Education. Oslo, Norway, August 2001

3 Deters J V R. Dynamic Courseware Generation on the WWW[J]. Journal of Educational Technologies, 1998, 29 (1): 5~14

4 Sun L, Williams S, Liu K. Knowledge Construction in e-Learning: Designing an e-Learning Environment[C]. ICEIS, 2003

5 谢深泉. 知识点及其网络的特性分析[J]. 软件学报, 1998, 9: 10