基于概念的折半查找算法

时百胜

(上海大学计算机工程与科学学院 上海 200072) (苏州科技学院数理学院 苏州 215009)

摘 要 自顶向下或自底向上两种标准查找方式不适合于具体领域逻辑,且缺乏灵活性。模拟有序数组中的折半查找,提出逻辑空间上的折半查找方法,证明该查找方法在保持完备性和非冗余性的同时,还提供了更好的灵活性,最后给出了该查找的通用算法,并分析其复杂性。

关键词 逻辑信息系统,折半查找,算法,复杂性

Concept-based Binary Search Algorithm

SHI Bai-sheng

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

(School of Mathematics and Physics, University of Science and Technology of Suzhou, Suzhou 215009, China)

Abstract The standard strategies of top-down and bottom-up search are ill-suited to domain specfic logics, and lack flexibility. This paper therefore introduced a binary search strategy, that is analogous to a binary search in an ordered array. We proved this provides more flexibility in the search, while retaining completeness and non-redundancy. We presented a novel algorithm for learning using domain specific logics and binary search, and analysed its complexity.

Keywords Logic information system, Binary search, Algorithm, Complexity

1 引言

在逻辑信息系统中,所有对象用具体领域的逻辑描述,逻辑公式按照包含$\sqsubseteq$(蕴含)关系构成了逻辑表示空间,在逻辑表示空间下需要查找满足某些条件的逻辑公式,例如:数据挖掘、机器学习等。大多数现有查找算法采用自顶向下或自底向上的方式遍历搜索空间,在逻辑空间中,越往上描述对象的公式越一般(通用),而越往下描述对象的公式越具体。自顶向下查找是从描述所有对象的最通用公式开始,重复例化它,直到例化满足要求为止。底公式是搜索空间上最具体的公式,自底向上查找是从底公式开始,对它们进行泛化,或计算两公式之间的合一运算。该运算对一阶谓词逻辑是求最小一般泛化[1]、对描述逻辑是求最小公共包含[2]或对序列模式求最长公共子序列[3]。自底向上查找的困难之一是对象顺序的安排和噪声数据的敏感性,困难之二是底公式数量很大,甚至无穷大,以至求最小一般泛化(lggs)的数量可能无法处理。

目前存在例化和泛化两种运算相结合的方法。一种是在某些点应用泛化运算改变自顶向下的查找[5],另一种是从搜索空间的任意点开始轮流应用例化和泛化运算以描述更多的正例和较少的反例。但双向搜索的查找算法是贪婪的或随机的,不能保证找到最优的方法。

查找方法通常是在灵活性、完备性以及非冗余性之间找到一种折衷方案,查找方式越灵活,完备性和非冗余性就越困难。一方面,当仔细设计纯自顶向下的查找的细化运算时,它具有弱的完备性和非冗余性,但灵活性不够,采用启发式调整查找变得困难;另一方面,随机双向查找是非常灵活的,但它不能保证完备性,也不保证非冗余性。灵活性允许应用启发式搜索对每一个查找的位置进行评估,得到最好的位置,再从这个位置进行搜索直到目标。

本文提出一种新的基于概念的查找算法,它采用跳跃式双向遍历搜索空间,类似于有序数组上的折半查找。该算法的优点:(1)查找是数据驱动型,可以解决搜索空间中的无穷链问题,允许采用更广泛逻辑;(2)既能够保证算法的完备性、非冗余性,又具有更好的灵活性。

本文首先研究基于概念的查找和折半查找思想;然后研究逻辑空间上基于概念的折半查找的实现,最后给出折半查找的通用算法及其复杂性分析。

2 基于概念的查找

定义1 逻辑信息系统(背景)是一个三元组$K=(\mathcal{O},\mathcal{L},\delta)$,其中:

— $\mathcal{O}$是有限对象集;

— $\mathcal{L}$是一种逻辑;

— $\delta\in\mathcal{O}\to\mathcal{L}$是每个对象$o\in\mathcal{O}$到逻辑描述$\delta(o)\in\mathcal{L}$的映射。

对象用逻辑公式描述,不同语法表示的公式可以证明是等价的,公式之间的逻辑等价被认为是在查找算法中存在冗余的原因之一,在查找算法中这些等价应该被识别,只需考虑其中一个就可以了。

定义2 具体域逻辑定义为$\mathcal{L}=(L,\sqsubseteq)$,包含如下内容:

到稿日期:2008-06-26 返修日期:2009-03-17

时百胜(1965—),男,副教授,博士生,研究方向为数据挖掘、智能信息处理,E-mail:shi_baisheng@163.com。

(1)L 为公式语言；

(2)公式上的运算：合取($\sqcap$)、析取($\sqcup$)、否定($\neg$)、重言式($\top$)、矛盾式($\bot$)等；

(3)语义 S_L，所有公式的解释集 I。$i \models f$ 表示 i 是 f 的一个模型，任意公式 $f \in L$ 的模型集记为 $M(f)=\{i \in I \mid i \models f\}$；

(4)包含(蕴涵)关系$\sqsubseteq$：公式之间的泛化顺序。$f \sqsubseteq g$ 当且仅当 f 的所有模型也是 g 的模型。当 $f \sqsubseteq g$ 和 $g \sqsubseteq f$，f 和 g 逻辑等价，记为 $f \equiv g$。设 $F \subseteq L, g \in L$，$F \sqsubseteq g$ 等价于 $\exists f \in F: f \sqsubseteq g$；$g \sqsubseteq F$ 等价于 $\forall f \in F: g \sqsubseteq f$。

除了逻辑等价外，在具体数据集中还存在另一种弱逻辑等价形式。如果两个公式覆盖相同的对象集则它们概念上等价，该等价类称之为形式概念，这一思想可用形式概念分析[6]理论进行形式化，对数据集进行有意义的分组。

定义 3 设 $K=(\mathcal{O}, \mathcal{L}, \delta)$ 为背景。对象 o 称为公式 f 的覆盖，当且仅当 $\delta(o) \sqsubseteq f$；公式 $f \in L$ 的外延(或覆盖)定义为 f 所覆盖的所有对象集：$ext_K(f)=\{o \in \mathcal{O} \mid \delta(o) \sqsubseteq f\}$。

定义 4 设 $(\mathcal{O}, \mathcal{L}, \delta)$ 为背景，概念是一对 (Ext, Int)，其中 $Int \subseteq \mathcal{L}$ 为内涵，$Ext \subseteq \mathcal{O}$ 为外延且满足 $Ext \neq \emptyset$，Int 是覆盖 Ext 中所有对象的所有公式的非空类：

$$Int=\{f \in \mathcal{L} \mid ext_K(f)=Ext\}, \quad Int \neq \emptyset$$

并非所有对象集一定是外延，这取决于逻辑的表达力和背景。例如：表 1 所列的逻辑描述背景中，对象集 $\{o_3\}$ 不是外延，因为 o_3 描述覆盖 o_2，覆盖 o_3 的所有公式也覆盖 o_2，然而，$\{o_2, o_3\}$ 和 $\{o_2\}$ 两对象集为外延，图 1 为至少覆盖一个对象的所有公式图，这些公式被形式概念划分成 4 部分(概念)。

表 1 逻辑描述背景

对象	描述
o1	a=2 & FR
o2	a=3 & FR
o3	a=3 & (FR∨JR)

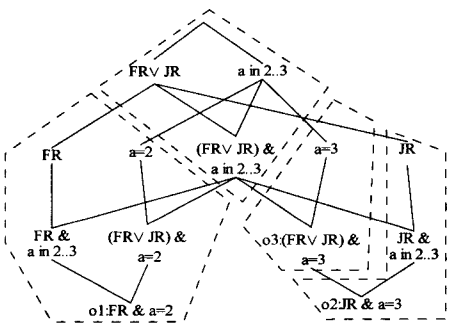

图 1 逻辑搜索空间被概念划分

这样，逻辑查找空间被划分成若干形式概念，目的是要在该区间上查找满足要求的形式概念逻辑表示，因此，理想的做法是查找在概念之间进行而不是在公式之间。尽管公式的数量可以无穷大，但概念数量肯定是有限的不会超过 2^n(n 为对象的数量)。然而，概念不可以直接查找，因为还没有容易的方法在概念内涵中选择一个概念表示。一种大胆选择是最通用表示，一种谨慎选择是最具体表示，这两种选择方式都存在不足。本文设计一种在逻辑空间上的查找算法，它近似概念之间的查找，下一节引入这种查找的基本思想。

3 折半查找

概念不可以直接探测，只能通过逻辑公式，与生成公式的数量相比访问概念的数量最大化。而概念是由背景决定的，这种查找探测具有强烈的数据驱动性质。

为了更好地理解概念内涵的结构，首先证明概念内涵由互不影响的包含链构成。

定理 1 设 x, y 是属于同一概念内涵的两个公式(即 $ext(x)=ext(y)$)。那么所有满足 $x \sqsubseteq z \sqsubseteq y$ 的公式 z 同属一个概念内涵。

证明：从定义 3 直接得到，如果 $x \sqsubseteq z \sqsubseteq y$，那么 $ext(x) \subseteq ext(z) \subseteq ext(y)$。$x$ 和 y 属于同一内涵，它们具有相同外延，那么 z 也属于相同外延，因此属于同一概念内涵。

这意味概念内涵集中于查找空间的一个狭窄的区域，整个查找空间被划分成若干个这样的区域。当查找过程沿着一个或几个链一步步到达概念时，显然存在冗余，如图 1 所示，每个概念含有长度 2 或 3 的这种链。

为了近似概念之间的查找，应该考虑较大的跳跃而不是最小的步距，许多随机查找完成这种跳跃，但它们通常是语法驱动型而非数据驱动型，不能保证这种查找的完备性和非冗余性。

在数据结构中讲述的折半查找方法是将数组有序(递增或递减)排列，查找过程中采用跳跃式方式查找，即先以有序数组的中点元素为比较对象，如果要找的元素值小于该中点元素，则将待查序列缩小为左半部分，否则为右半部分。通过一次比较，将查找区间缩小一半。折半查找是一种高效的查找方法。它可以明显减少比较次数，提高查找效率。但是，折半查找的先决条件是查找表中的数据元素必须有序。

本文是要在逻辑空间上实现折半查找，在逻辑空间上实现跳跃以到达不同的概念，问题在于通过这种跳跃可能丢失一些有意义的概念，使查找不完备，而有序数组的折半查找，从向上和向下两个方向实现跳跃，这要求逻辑空间上的折半查找应设法避免循环、不完备和冗余。下一节我们重点研究如何实现这种查找，并证明这种实现具有良好性质。

4 逻辑空间上折半查找的实现

在逻辑空间中实现折半查找，有两个重要概念：折半位置和折半运算。对于有序数组的折半查找，两个概念分别指有序数组中位置的区间以及定位该区间的中间元素。在逻辑搜索空间中，这需要重新定义。

逻辑空间中的折半位置定义为逻辑公式区间$[y, Z]$，其中 y 是下界，公式集 Z 为上界，该区间上的折半运算定义为产生新的公式集 X(等价于中间元素)，每个生成的公式 x 满足：(1) x 比 y 更一般，即 $y \sqsubseteq x$；(2) x 不比 Z 中任意公式更不一般，即 $Z \not\sqsubseteq x$，增加这些到目前已生成的所有公式集 F 中。在折半运算中增加对象 o 用来保证每个生成的公式 x 比 y 覆盖更多的对象，以到达不同的概念，这意味 o 不属于 y 的外延。为了避免产生两次相同的公式，位置(y, o, Z)中的公式集 Z 定义为比公式 y 和对象描述 $\delta(o)$ 更一般的、且最具体的提取公式，即 y 和 $\delta(o)$ 的最小上界，记为 $lub_F(y, \delta(o))$。

定义 5 设 $K=(\mathcal{O}, (L, \sqsubseteq), \delta)$ 为背景，$F \subseteq L$ 为目前生成的公式集(开始为 $\delta(\mathcal{O})$)，关于 F，折半位置是一个三元组(y, o, Z)，其中 $y \in F, o \in \mathcal{O} \backslash ext_K(y)$，且 $Z \subseteq F$ 定义为 F 中 y 和 $\delta(o)$ 的最小上界，即 $lub_F(y, \delta(o))$，简写为(y, o)，因为 Z 可从它和 F 中推出。

逻辑区间定义为折半位置的一种抽象，折半运算定义为应用于逻辑区间上的逻辑运算。

定义 6 设$(L,\sqsubseteq)$是一种逻辑，逻辑区间是三元组(y,d,Z)，其中$y,d\in L,Z\subseteq L,y\sqsubseteq Z$和$d\sqsubseteq Z$。折半运算是从任意逻辑区间$(y,d,Z)$到公式集$X\subseteq L$的函数，所有$x\in X$满足：(1) $y\sqsubseteq x$，(2)$d\sqsubseteq x$，(3)$Z\not\sqsubseteq x$。

按照上述定义，生成的新公式x是已生成的公式y和y没有覆盖的对象o的泛化。从这里可以看出，折半查找类似于自底向上策略，主要区别在于生成的公式不需是最小一般泛化，因此生成的公式集不需要是最小一般泛化，通常，在查找中使用表达力强的逻辑会有许多非常具体的$lggs$，而使用简单逻辑则可能会丢失有意义公式的风险，这两者之间存在折衷。在折半查找中，这种折衷消失了，因为可以先产生想要泛化的有限集，如果需要，再应用折半运算细化它们。

向同一位置多次应用折半运算会出现产生公式的冗余吗？下面证明折半查找在冗余性方面的性质。

定理 2 所有生成的公式至少覆盖背景中的两个对象。

证明：按照定义 5，从位置(y,o,Z)生成的所有公式x比y和$\delta(o)$两者更一般，这要求如果y至少覆盖一个对象，那么x至少覆盖 2 个对象。因为初始时，公式y是对象描述(定义 4)，至少覆盖一个对象。证毕。

这意味着折半查找具有强烈的数据驱动型，避免考虑与背景无关的公式。而通常自顶向下的查找要考虑覆盖空对象的公式。底公式的使用可以保证生成的公式至少覆盖 1 个对象，而不是 2 个，即使自底向上策略应用泛化运算不能保证这个结果。

定理 3 设F为目前生成的公式集，关于F，所有生成的特征x都是新的，即$x\notin F$。

证明：反证法。设x是从某个折半位置(y,o,Z)生成的特征，且属于F。由定义 6 知：$y\sqsubseteq x,\delta(o)\sqsubseteq x,Z\not\sqsubseteq x$，且还有$x\in F$，这要求$Z=lubs_F(y,\delta(o))\sqsubseteq x$，这是矛盾的。

逻辑空间上的折半查找是非常灵活的，因为它对折半位置的顺序无要求；查找也是双向的，既可以用逻辑上界也可用逻辑下界产生公式。此外，在查找中没有冗余，尤其消除了循环的危险。因此，当向同一折半位置几次应用折半运算将会发生什么呢？首先，折半位置由一对(y,o)描述，逻辑区间的上界 Z 为$lubs_F(y,\delta(o))$，当从这个位置生成的公式x插入到F中，得到 Z 的更新为$lubs_{F\cup\{x\}}(y,\delta(o))$。由定义 6 知，这要求$x$属于 Z 的新定义，避免从$(y,o)$中再产生它。这样，Z 作为从位置$(y,o)$或其它位置已产生公式的存储器，生成的公式限定在$(y,o)$和$lubs_{F\cup\{x\}}(y,\delta(o))$或$(x,o')$和$lubs_{F\cup\{x\}}(x,\delta(o'))$之间，其中$o'$是$x$没覆盖的对象，因此是折半查找思想。与有序数组中的折半查找不同，该查找必须在两个半逻辑区间中进行。

定理 4 设F为目前生成的公式集，对于从位置(y,o,Z)生成的所有公式x，x的概念区别于y或$\delta(o)$的概念。

证明：因为x是y和$\delta(o)$的泛化(定义 6)，得到$ext(y)\cup ext(\delta(o))\subseteq ext(x)$，而且$ext(y)\neq\emptyset$(定理 2)和$o\notin ext(y)$，这要求$ext(x)$严格大于$ext(y)$和$ext(\delta(o))$，因此属于不同的概念(定义 3)。

上述定理证明通过要求新生成的公式x到达不同于下界y的概念的概念来探测尽可能多的概念思想，这可通过使用附加对象做到。使这个概念区别任何上界$z\in Z$的概念也是可以的，方法是使用Z中的所有上界都覆盖而下界y不覆盖的另一对象o'，生成的公式将会覆盖o而不覆盖o'，不过，在折半运算的定义中处理这个负约束是困难的。更重要的是这将会使折半查找不完备，因为即使生成的公式属于上界的概念，下界产生更多的泛化公式和到达新概念有一定意义。在假设折半运算自身完备的情况下，下面证明折半查找的完备性。

首先，设折半运算为 binary，逻辑区间上的递归应用定义为连续应用的结果。

定义 7 设(y,d,Z)为逻辑区间，$binary_n(y,d,Z)$表示该区间上n次应用折半运算，定义如下：

$binary_0(y,d,Z)=\emptyset$

$binary_{n+1}(y,d,Z)=binary_{n+1}(y,d,Z)\cup binary(y,d,Z\cup binary_n(y,d,Z))$

该定义有助于定义折半运算的完备性，因为它能近似所有包含一个逻辑区间 2 个下界的公式。

定义 8 折半运算 binary 是完备的，当且仅当对于所有逻辑区间(y,d,Z)和满足$y\sqsubseteq x,d\sqsubseteq x,Z\not\sqsubseteq x$所有公式$x$。存在一个有限数$n$和公式$x'\in binary_n(y,d,Z)$使$x'\sqsubseteq x$。

使用较弱条件$x'\sqsubseteq x$代替$x'\equiv x$原因是会破坏折半的思想，因为这要假定逐步探测。折半运算的作用的确是在下界y和上界Z之间某处找到公式，如果x'太具体，可以通过向另一个位置(x',y',Z')应用折半运算来泛化它。同时，如果它们表示同一概念，可使用x'代替x作为区分概念的目标。

关于区分概念，下面证明折半查找是完备的这个重要结论。

定理 5 设 binary 是完备的折半运算，F是已生成的公式集，对于满足$ext(x)\neq\emptyset$的所有公式$x\notin F$，且在F中不存在更具体的概念表示(即$\nexists x'\in F:x'\sqsubseteq x,ext(x')=ext(x)$)，通过有限次的折半运算，可以产生比$x$更具体的公式$x'$且它们同属一个概念。

证明：首先，辨识从折半位置(y,o,Z)生成的公式x，设$y\in F$是满足$y\sqsubseteq x$的公式，因为$ext(x)$非空，至少包含一个对象描述，初始F含有所有对象的描述。现在，如果$ext(y)=ext(x)$，这与假设x没有更具体的概念表示矛盾，否则，肯定存在由x覆盖而y不覆盖的对象o，那么(y,o,Z)是一个折半位置，其中$Z=lubs_F(y,\delta(o))$。如果存在上界$z\in Z$满足$z\sqsubseteq x$，那么通过用z取代y作为下界重新开始，这个取代下界的过程将结束，因为z的外延严格大于y的外延(增加对象o)，且x的外延是有限的。否则，x是折半位置(y,o,Z)可接受的产生特征。

其次，由定义 7，我们推导出通过向逻辑区间$(y,\delta(o),Z)$应用有限次数的折半运算可产生公式$x'\sqsubseteq x$。如果$ext(x')=ext(x)$，那么结论被证明。否则，在这个证明的开始，用新的公式x'取代y重新开始，这一取代过程将结束，因为由定义 5 我们还知道，x'具有比y更大的外延，因为它覆盖$ext_K(y)$和o。

从区分概念的角度，这个定理可以给出更简单的形式。

推论 1 设K为背景，通过从$F=\delta(\emptyset)$开始的折半查找，K的所有概念被区分，即，通过有限次数的运算找到一个逻辑表示。

证明：设 c 是概念，在 F 中没有它的表示，且 $x \notin F$ 是一个未知表示。由定义 4 知，$ext(x) \neq \emptyset$，因为概念是具有非空外延，按照定理 5，概念 c 的表示 $x' \sqsubseteq x$ 可在有限次数的运算中产生。

定理 5 的证明揭示了 3 个迭代。第一是经由折半位置的迭代，为了找到可产生公式 x 的折半位置；第二迭代是折半运算的递归应用，直到找到 x 的实例 x' 为止；第三个迭代是进一步泛化 x' 直到到达 x 的概念；这意味从新生成的公式构建新的折半位置。

定理 5 及其推论证明在有限时间内可以区分概念，即使逻辑查找空间是无限的，即使 lggs 是不确定的或无限的，在不约束应用域逻辑的情况下这个结论通过自顶向下和自底向上查找不可能获得。

5 通用算法及其复杂性

下面给出算法的伪码，设背景 $K=(\mathcal{O},(L,\sqsubseteq),\delta)$，初始公式集 F，从 F 中得到折半位置的初始集 P，事实上，折半位置仅有 2 个部分 (y,o) 存储在 P 中，第三个部分 $Z=lubs_F(y,\delta(o))$ 取决于当前特征集 F，这些位置用于通过折半运算产生新的公式集，进而又产生新位置。当没有新的公式集产生时，位置删除，当没有位置余下时，整个过程结束。

```
1  F←δ(∅);
2  P←{(y,o)∈F×∅|o∉ext_K(y)};
3  while P≠∅ do
4     (y,o)←选取合适的 P;
5     Z←lubs_F(y,δ(o));
6     X←binary(y,δ(o),Z);
7     if X=∅ then
8        P←P\{(y,o)};
9     else
10       F←F∪X;
11       P←P∪{(x,o)∈X×∅|o∉ext_K(x)};
12 done
```

上述算法的输出是表示背景 K 的所有可能概念的公式集。在我们的实现中，生成的公式集 F 内部表示为包含序的 Hasse 图，该表示与表(list)相比具有不少优点：(1)代价高的包含测试仅用于 F 中新元素的插入，在算法其它地方不用；式(2)的外延可以通过该图简单的向下遍历来计算，事实上，在公式插入时完成并储存；(3) 2 元素的 lubs 也能通过图的遍历来计算，节省了许多包含测试。

算法伪码的第 4 行，可通过启发式算法产生最想要的位置，这要求位置是有序的。P 内部用二叉树表示，这种表示能够选择最佳位置，增加新位置用 $O(\log|P|)$ 时间，而采用表结构的时间为 $O(|P|)$。

现在讨论算法的复杂性，首先，定义背景中的对象数量为 $n=|\mathcal{O}|$，$N=|F|$ 为生成的公式的数量，其次，位置的数量 $|P|$ 不超过 Nn，再者，每次折半查找运算只产生一个公式。算法中重要几行的复杂性如下：(1)第 4 行为 $O(\log(Nn))$：最佳位置的检索时间；(2) 第 5 行为 $O(N)$：*lubs* 的计算时间；(3) 第 6 行为 $O(|dicho|)$：折半运算的时间；(4) 第 10 行为 $O(N|\sqsubseteq|)$：在 Hasse 图 $(F,\sqsubseteq)$ 中新公式的插入；(5)第 11 行为 $O(n(|h|+\log(Nn)))$：将计算新位置以及插入它们的时间所产生 N 个公式的局部复杂性综合起来，整个算法的最坏复杂性为：$O(N^2|\sqsubseteq|+N|dicho|+Nn(|h|+\log(N)))$。第一项是非常重要的，在包含图中反映新公式的插入，第二项对应折半运算的应用，第三项表示评价和排序位置的代价。各种实验已经证明两个主要的开销是在图中新公式的插入和折半运算上，这也与理论上的复杂性一致，因为当生成的公式越来越多，第一项所花的运行时间也在增加。

结束语 本文提出一种新的基于概念的折半查找算法。它的优点：(1)该算法是通用算法，所用的逻辑是抽象的，可用简单的基本逻辑，或具体领域使用的复杂逻辑来实例化它。(2) 因为折半位置的顺序是任意，所以它提供了更为灵活的查找，并在查找的同时保持完备性和非冗余性。同时，折半运算也存在一些灵活性，因为它的定义并不像细化运算和泛化运算那样受约束。

参 考 文 献

[1] Califf M E, Mooney R J. Bottom-up relational learning of pattern matching rules for information extraction[J]. Journal of Machine Learning Research, 2003, 4: 177-210

[2] Baader F, Usters R K, Molitor R. Computing least common subsumers in description logics with existential restrictions[C]// Proc. of the 16th Int Joint Conf. on Artificial Intelligence (IJCAI-99). 1999: 96-101

[3] 郑宗汉. 算法设计与分析[M]. 北京：清华大学出版社，2005

[4] Badea L. A refinement Operator for Theories [C] // Inductive Logic Programming, LNCS 2157. 2001

[5] Lita L V. Instance - Based Question Answering [D]. Carnegie Mellon University, 2006

[6] Ganter B, Wille R. Formal Concept Analysis — Mathematical Foundations[M]. Springer, 1999

(上接第 158 页)

[8] 高颖，曹存根，眭跃飞. 音乐领域本体的建立和分析[J]. 计算机科学，2004，31(1)：103-107

[9] 胡艳丽，白亮，张维明，等. 知识网格中基于领域本体的智能检索[J]. 计算机科学，2007，34(8)：202-207

[10] 顾慧翔，俞勇. 基于领域本体和知识推理的语义互联网应用[J]. 上海交通大学学报，2004，38(4)：583-585

[11] Dong Z Q, Dong Q. HowNet[EB/OL]. http://www.keenage.com/, 2008

[12] 赵天忠，苗壮，张亚非，等. 基于 WordNet 重用的领域本体构件方法[J]. 系统仿真学报，2007，19(19)：4583-4586

[13] 李晓博，缪淮扣，刘静. 基于形式规格说明的构件匹配[J]. 计算机应用与软件，2006，23(10)：10-12

[14] Wang H, Feng Y, David C. Verifying the Reusability of Software Component Specification Framwork and Algorithms [J]. Information Science, 1998, 112(12): 169-197

[15] 袁柳，李战怀，陈世亮. 基于本体的 Deep Web 数据标注[J]. 软件学报，2008，19(2)：237-245