

几种叙词表复杂逻辑错误检查算法研究*

□ 徐硕 乔晓东 朱礼军 / 中国科学技术信息研究所 北京 100038

摘要: 目前叙词表的编制仍以手工为主, 计算机为辅, 因此难免会出现各种逻辑错误。为了保证叙词表的一致性, 经常需要首先检查其中的逻辑错误, 然后对其进行纠正。但复杂逻辑错误的检查通常比较困难, 尚没有有效算法可以利用。针对这种现状, 文章借助图论的相关知识, 设计了几种检查复杂逻辑错误的算法, 并分析了相应算法的时间复杂度。而且这些算法也可可为其他词汇知识组织系统的逻辑错误检查提供一定的参考。

关键词: 叙词表, 一致性, 逻辑错误, 算法, 图论

DOI: 10.3772/j.issn.1673-2286.2010.08.008

叙词表是指将文献作者、标引者和检索者使用的自然语言转换成规范化的叙词型主题检索语言的术语控制工具, 亦称主题词表、检索词典; 它是一种概括某一学科领域, 以规范化的、受控的、动态性的叙词(主题词)为基本成分和以参照系统显示词间关系, 用于标引、存储和检索文献的词典^[1]。按照自动化程度, 可将叙词表编制方式分为三种^[2]: 手工编制法、计算机辅助编制法和自动构建法。由于计算机相关技术发展水平的限制, 辅助编制法目前处于主流地位, 并且我国现有的叙词表(主题词表)大部分是手工编纂的, 因此难免会出现各种逻辑错误。

为了保证叙词表的一致性, 经常需要对其中的逻辑错误进行纠正, 从而提高叙词表的科学性。目前常用的检查方法^[3]是首先将叙词表转化为本体, 然后借助本体语言的推理能力进行严格的一致性检测。但该方法只能检测一些比较简单的逻辑错误, 比如款目词重复、关系不闭合、款目词与其参照项相等等。本文将借助图论的相关知识, 设计几种有效的算法, 以便解决一些复杂逻辑错误的检查问题。

1 预备知识

图是一种数据结构^[4], 通常表示为 $G=(V, E)$, 其

中 V 为节点的有穷非空集合, E 是两个节点间的关系集合。若 $\langle v, v' \rangle \in E$ 表示从节点 v 到节点 v' 的一条弧, 且称 v 为初始点, 称 v' 为终端点, 此时的图称为有向图。若 $\langle v, v' \rangle \in E$ 必有 $\langle v', v \rangle \in E$, 即 E 是对称的, 则以无序对 (v, v') 代替这两个有序对, 表示 v 和 v' 之间存在一条边, 此时的图称为无向图。叙词表中的用/代关系或属/分关系以及相关的款目词构成了一个有向图, 而参考关系以及相关的款目词构成了一个无向图。

对于无向图 $G=(V, E)$, 顶点 v 的度定义为和 v 相邻接的边的数目, 记为 $D(v)$ 。对于有向图 $G=(V, E)$, 如果 $\langle v, v' \rangle \in E$, 则称节点 v 是节点 v' 的父节点, 称节点 v' 为节点 v 的孩子节点。以顶点 v 为终端点的弧的数目称为 v 的入度, 记为 $ID(v)$; 以 v 为初始点的弧的数目称为 v 的出度, 记为 $OD(v)$; 顶点 v 的度定义为入度与出度之和, 即 $D(v) = ID(v) + OD(v)$ 。

在无向图 $G=(V, E)$ 中, 如果 $(v_{i,j-1}, v_{i,j}) \in E$ ($1 \leq j \leq m$), 其中 $v_{i,0} = v, v_{i,m} = v'$, 则节点序列 $(v_{i,0}, v_{i,1}, \dots, v_{i,m})$ 称为节点 v 与 v' 的一条长为 m 的路径。当然, 在有向图 $G=(V, E)$ 中两个节点的路径也可类似定义。第一个节点和最后一个节点相同的路径称为回路或环。序列中节点不重复出现的路径称为简单路径。

对于一部叙词表来说, 无论根据用/代关系还是根据属/分关系构建的有向图, 通常不只一个节点的入度

* 本研究受“十一五”国家科技支撑计划“知识组织系统的集成及服务研究与实现”(2006BAH03B03)和中国科学技术信息研究所重点项目“汉语科技词系统建设与应用工程——新能源汽车领域完善及领域扩展”(2009KP01-3-2)资助。

为0,也就是说很可能会形成许多子图。同样,根据参考关系构建的无向图也会形成许多子图,但每个子图都是一个完全图。本文主要考虑属/分以及参考关系,对于属/分关系,我们将叙词 w 的属分关系图定义为包含叙词 w 的那个有向子图,记为 $G^w=(V^w, E^w)$,并定义一个有向图 G 的深度为入度为0的节点到出度为0的节点的路径的最大长度,记为 $h(G)$ 。

2 属/分关系中的逻辑错误检查算法

在描述各种逻辑错误检查算法之前,需要定义一个函数 $ascent(w_1, w_2, G^w, G^w)$,用于判断两个叙词 w_1 在有向图 G^w 或 G^w 中是否是 w_2 的祖先节点。具体算法如下:

Algorithm	
1. 判断一个叙词是否为另一个叙词的祖先节点。	
输入: 两个叙词 w_1, w_2 , 以及对应的属/分关系图 $G^w=(V^w, E^w)$, $G^w=(V^w, E^w)$ 。	
输出: 如果 w_1 是 w_2 的一个祖先节点, 返回TRUE, 否则返回FALSE。	
1	IF $w_1 \in parent(w_2, G^w)$ OR $w_2 \in parent(w_1, G^w)$ THEN
2	RETURN TRUE;
3	END IF
4	FOR EACH w IN $parent(w_2, G^w)$
5	RETURN $ascent(w_1, w, G^w, G^w)$;
6	END FOR
7	RETURN FALSE;

需要注意的是函数 $ascent(\cdot, \cdot, \cdot, \cdot)$ 是采用递归方式定义的,它的基本思想是自底向上遍历属/分关系图。之所以采用自底向上遍历的方式,主要原因是属/分关系图通常是一个有向无环图(DAG)^[5],出度大于2的节点数通常要远远大于入度大于2的节点数,因此这种遍历方式的时间代价比较小。该算法的时间复杂度为 $O(\min\{h(G^w), h(G^w)\})$ 。

2.1 有两个以上的上位词,即多属

当然,某个叙词有两个以上(含两个)的上位词(如图1中(a)所示)不一定是逻辑错误,但根据我们的经验,多属项通常隐含某种错误,因此需要对此类情形仔细检查。对这类叙词的检查比较容易,只需判断每个叙词对应节点的入度是否大于2即可。

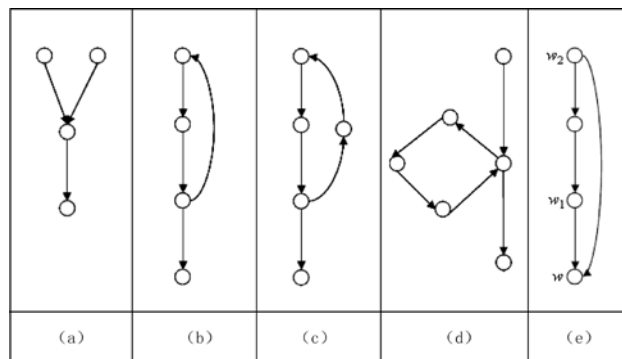


图1 属分关系中的逻辑错误示意图

2.2 多个属项出现在同一条(单线)属分链上

用图论描述语言,这种逻辑错误是指对应的属/分关系图 $G=(V, E)$ 中存在环路,如图1中(b)~(d)所示。这样就将检查这种逻辑错误的问题转换为检查 G 中是否存在环路的问题。检查环路通常采用深度优先遍历(DFS)有向图 $G^{[4,6]}$,时间复杂度为 $O(|V|+|E|)$ 。

2.3 属分关系出现“循环”

假设叙词 w 有两个上位词 w_1 与 w_2 ,“属分关系出现‘循环’”是指 w_1/w_2 是 w_2/w_1 的直接或间接上位词,如图1中(e)所示,这是2.1节中多属项的一个特例。检查该类逻辑错误的具体算法如下:

Algorithm	
2. 检查“属分关系出现‘循环’”的逻辑错误。	
输入: 一个表示属/分关系的有向图 $G=(V, E) (V \geq 3)$ 。	
输出: 所有符合要求的叙词三元组。	
1	FOR EACH v IN V
2	IF $ID(v) \geq 2$ THEN
3	FOR $parent(v, G)$ 中任意无序词对 (w_1, w_2)
4	IF $ascent(w_1, w_2, G, G)$ OR $ascent(w_2, w_1, G, G)$ THEN
5	输出 v, w_1, w_2 ;
6	END IF
7	END FOR
8	END IF
9	END FOR

3 参考关系中的逻辑错误检查算法

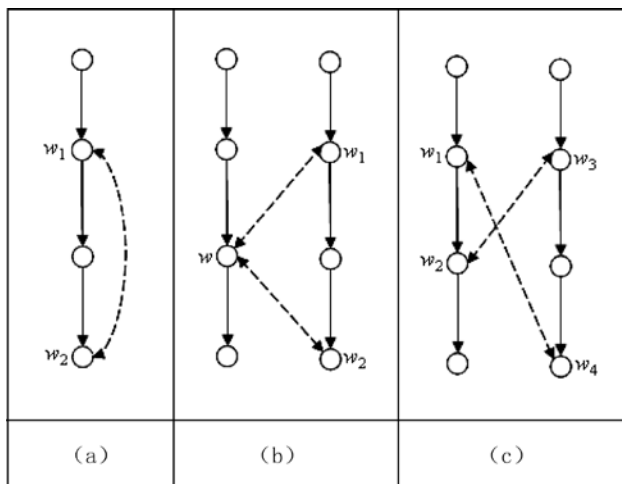


图2 相关关系中的逻辑错误示意图

3.1 参上、下级词

也就是两个叙词 w_1 与 w_2 间不仅具有参考关系,而且具有属/分关系,如图2中(a)所示。具体算法如下:

Algorithm	
3. 检查“参上、下级词”的逻辑错误。	
输入: 一个具有参考关系的词集 $S = \{w_1, w_2, \dots, w_n\} (n \geq 2)$, 以及词集 S 中所有词的属/分关系图 $G^w = (V^w, E^w), w \in S$ 。	
输出: 所有符合要求的叙词二元组。	
1	FOR 词集 S 中任意无序词对 $(w_i, w_j) (i < j)$
2	IF ascent(w_i, w_j, G^w, G^w) OR ascent(w_j, w_i, G^w, G^w) THEN
3	输出 w_i, w_j ;
4	END IF
5	END FOR

该算法的时间复杂度为 $O(|S|^2 \times \max\{h(G^w): w \in S\})$ 。需要注意的是, 该算法将一个具有参考关系的词集作为一个整体来考虑, 输出该集合中所有满足要求的逻辑错误。

3.2 多个参项词在一条属分链上

也就是对于某个叙词 w , 存在两个参项词 w_1, w_2 , 使得 w_1 与 w_2 间存在属/分关系, 如图2中(b)所示。具体算法如下:

Algorithm

4. 检查“多个参项词在一条属分链上”的逻辑错误。

输入: 一个具有参考关系的词集 $S = \{w_1, w_2, \dots, w_n\} (n \geq 3)$, 以及词集 S 中所有词的属/分关系图 $G^w = (V^w, E^w), w \in S$ 。

输出: 所有符合要求的叙词三元组。

1	FOR EACH w IN S
2	FOR 词集 S 中任意无序词对 $(w_i, w_j) (i < j)$
3	IF ascent(w_i, w_j, G^w, G^w) OR ascent(w_j, w_i, G^w, G^w) THEN
4	输出 w, w_i, w_j ;
5	END IF
6	END FOR

该算法的时间复杂度为 $O(|S|^3 \times \max\{h(G^w): w \in S\})$ 。需要注意的是, 该算法将一个具有参考关系的词集作为一个整体来考虑, 输出该集合中所有满足要求的逻辑错误。

3.3 上参下, 下参上, 相关关系出现交叉

假设叙词 w_1 是 w_2 的直接/间接上位词, w_1 的参项为 w_4 , w_2 的参项为 w_3 , “上参下, 下参上, 相关关系出现交叉”的逻辑错误是指 w_4 作为 w_3 的直接/间接下位词的情形, 如图2中(c)所示。具体算法如下页中的算法5所示。

该算法的基本思想为: 对于两个具有参考关系的词集 S_1, S_2 , 如果 S_1 中的某个叙词是 S_2 中的某个叙词的上位词, 并且 S_1 中的另外一个叙词却是 S_2 中的某个叙词的下位词, 那么将这两个集合间具有属分关系的所有叙词对输出, 以便人工纠正相应的错误。该算法的时间复杂度为 $O(|S_1| \times |S_2| \times \max\{h(G^w): w \in S_1 \cup S_2\})$ 。

4 结束语

高效的叙词表逻辑错误检查算法不仅可大大节省叙词表编制人员的工作量, 同时可有力保证所编制叙词表的质量, 为叙词表的推广应用提供保障。本文将属/分关系及相应的叙词表示成有向图, 将参考关系及相应的叙词表示成无向图, 借助图论的相关知识, 为几种复杂的逻辑错误检查设计了有效的算法, 并给出了相应的时间复杂度分析。虽然这些算法是针对叙词

Algorithm

5. 检查“上参下，下参上，相关关系出现交叉”的逻辑错误。

输入：两个具有参考关系的词集 $S_1 = \{w_{1,1}, w_{1,2}, \dots, w_{1,n}\} (n \geq 2)$ ，以及 $S_2 = \{w_{2,1}, w_{2,2}, \dots, w_{2,m}\} (m \geq 2)$ ，以及词集 $S_1 \cup S_2$ 中所有词的属/分关系图 $G^w = (V^w, E^w), w \in S_1 \cup S_2$ 。输出：所有符合要求的叙词二元组。

```

1  SF1=SF2=∅
2  FOR EACH w1 IN S1
3    FOR EACH w2 IN S2
4      IF ascent(w1,w2, Gw, Gw1) THEN
5        SF1=SF1 ∪ {<w1, w2>};
6      ELSE ascent(w2,w1, Gw, Gw1) THEN
7        SF2=SF2 ∪ {<w2, w1>};
8      END IF
9    END FOR
10  END FOR
11  IF SF1≠∅ AND SF2≠∅ THEN
12    输出SF1和SF2;
13  END IF
    
```

表设计的，但是对其他词汇知识组织系统的逻辑错误 检查也具有一定的参考价值。

参考文献

- [1] 叙词表[EB/OL]. [2010-04-12]. <http://baike.baidu.com/view/554473.htm>.
- [2] 杜慧平. 国外计算机辅助编制叙词表软件评价[J]. 现代图书情报技术, 2009(9): 17-21.
- [3] 曾新红, 林伟明. 中文叙词表本体一致性检测机制研究与实现[J]. 现代图书情报技术, 2008(5): 1-9.
- [4] 严蔚敏, 吴伟民. 数据结构 (C语言版) [M]. 北京: 清华大学出版社, 2007: 156-192.
- [5] THULASIRAMAN K, SWAMY M N S. Graphs: Theory and Algorithms [M]. John Wiley and Son, 1992: 118-119.
- [6] CORMEN T H, LEISERSON C E, RIVEST R L, STEIN C. Introduction to Algorithms [M], 2nd ed. The MIT Press, 2001: 525-700.

作者简介

徐硕 (1979-), 男, 博士, 目前于中国科学技术信息研究所从事博士后科研工作。研究方向: 数据挖掘、信息抽取、生物信息等。通讯地址: 北京市海淀区复兴路15号, 中国科学技术信息研究所信息技术支持中心 100038。E-mail: xush@istic.ac.cn
 乔晓东 (1965-), 男, 研究员, 硕士生导师, 中国科学技术信息研究所/信息技术支持中心主任, 中国互联网协会理事, 中国情报学会计算机应用分会副主任委员, CILIP会员, 研究方向为信息资源管理工作和信息服务, 发表论文50余篇, 为本文通讯作者。通讯地址: 同上。E-mail: qiaox@istic.ac.cn
 朱礼军 (1973-), 男, 副研究员, 硕士生导师。2005年起, 在中国科学技术信息研究所/信息技术支持中心从事一线科研工作。目前主要研究Semantic Web、Web Service和知识技术在科技信息服务、电子政务/商务中的应用以及知识组织系统的集成与服务体系。通讯地址: 同上。E-mail: zhulj@istic.ac.cn

Research on Algorithms for Checking Several Complex Logic Errors in Thesauri

Shuo Xu, Xiao-Dong Qiao, Li-Jun Zhu / Institute of Scientific and Technical Information of China, Beijing, 100038

Abstract: Currently, the thesauri are constructed mainly by manual, which is aided by computers. So there are various logical errors within the thesauri. In order to make sure the thesauri consistent, logical errors often need to be checked, and then are corrected. However, it is very difficult to check complex logical errors, and there aren't effective algorithms that can be utilized. For present status, with the help of related knowledge in graph theory, this study designs several algorithms that are able to check complex logical errors, and analyzes the resulting algorithms' time complexity. What's more, these algorithms can provide reference to some extent for checking logical errors in other lexical knowledge organization system (KOS).

Keywords: Thesaurus, Consistency, Logical error, Algorithm, Graph theory

(收稿日期: 2010-06-22)