

⑥
483-487
一个概念图编辑工具的设计与实现

张 蕾

(西北大学计算机科学系, 710069, 西安; 31岁, 女, 讲师)

A 摘要 分析了一个基于 prolog 语言的概念图编辑工具的设计思想与实现方法。具体研究了显示图与线性图相互转换的问题, 为概念结构知识表示方法的应用提供了方便。

关键词 概念图; 线性分析; 线性化

分类号 TP18

显示图
概念图编辑工具, 设计, prolog 语言

概念图是一种新的知识表示方法, 最早是 Sowa 在 1984 年提出的^[1]。纵观这一理论的发展, 有以下两个趋势: 一是理论本身的完善与发展; 二是将这一理论应用到计算机科学的各个领域。在这一理论的应用中较多的是用于自然语言处理、知识获取、规划及推理等方面。国内对概念图的应用尚处于研究阶段, 尚未开发出实用的系统。本文提供了一个通用的概念图编辑工具——CGET, 可推动概念结构理论在我国的应用与研究。

CGET 的功能: ①实现概念图在计算机内部的存贮; ②实现概念图的显示图和线性图的相互转换; ③实现概念图的编辑。CGET 的设计目标: ①提供一个友好的用户界面, 方便用户使用该工具。②用户能够以图的方式浏览、生成、删除和修改一个概念图, 能够以文件的方式方便地存取一个概念图, 也能够选择一个概念图的显示图到线性图的相互转换。

1 概念图模式及其内部表示

一个概念图是具有概念和概念关系两种节点的有向连通图^[2]。概念节点表示实体、属性、状态及事件, 概念关系节点表明概念之间的关系如何。概念和概念关系涉及到下列 3 个术语。

- 所指对象: 个体、值或集合。

- 概念类型: 用方框(或简写成方括号)内的概念类型标记表示。如: [PERSON], [COLOR] 和 [EAT] 都是概念。

- 概念关系: 用圆圈(或简写成圆括号)内的关系标记表示。如(LOC), (AGT) 和 (INST) 分别表示位置关系、动因关系和手段工具等关系。

一个概念是用两个域来描述的, 即类型标记域和所指对象域, 这两个域之间用冒号分开。每个概念的类型标记必须是非空的。所指对象域可以是空的, 这时省略冒号。所指对象域有以下几种类型:

1.1 所指对象域为空

概念的所指对象是隐含的, 省略不写, 表示一个一般概念。例如, 概念图:

$$[\text{PERSON}] \leftarrow (\text{POSS}) \leftarrow [\text{BALL}]$$

可以理解成下面几个句子:

A person possesses a ball.

A person possesses balls.

Persons possess a ball.

Persons possess balls.

要想表示一个确切的含义还得添加信息。

1.2 所指对象域为确定的个体

如果要表示“The person John possesses ball(s)”.就在冒号后跟一个唯一确定的个体名:

[PERSON:John]←(POSS)←[BALL]

1.3 所指对象域为集合

所指对象为空或者为某个确定的个体时,表示一个一般概念或者个体概念。但是,仅有这样的表达能力是不能超越一阶逻辑的。在一阶逻辑中,如果没有集合的方法,有些语义就表达不清。在概念图中也引入集合的方法。所指对象为集合时,分下列 4 种情形^[3]:

• $\{i_1, \dots, i_n\}$:表示集合中所有元素一起满足某种关系,相当于英语中的“together”。例如:

[PERSON:{Mary, John}]←(AGNT)←[DANCE]

表示“Mary and John are dancing, together.”

• $\text{Dist}\{i_1, \dots, i_n\}$:集合前加了前缀 Dist,表示集合中的每一个元素分别满足某种关系,相当于英语中的“each”。例如:

[PERSON:Dist{Mary, John}]←(AGNT)←[DANCE]

表示“Mary is dancing and John is dancing.”

• $\text{Resp}(i_1, \dots, i_n)$:表示序列中的每个元素与另一序列中的相应元素分别有某种关系。例如:

[PERSON:Resp<Dick, Jerry>]→(MARRY)→[PERSON:Resp<Pat, Betty>]

表示“Dick and Jerry are married to Pat and Betty, respectively.”

• $\{i_1 | \dots | i_n\}$:表示集合中的一个元素满足某种性质,但具体是哪一个元素是不确定的。例如:

[ELEPHANT:'Clyde']→(LIVE)→(LOC)→[CONTINENT:{Afric|Asia}]

表示“The elephant clyde lives in either Africa or Asia.”

在计算机存储器中,概念节点、概念关系节点和概念图是用 prolog 谓词来存放的,其结构分别如图 1、图 2 和图 3 所示。这种表示概念图的结构^[4]是一种很稳定的结构,因为,每个概念要指出与之相联的关系,每个关系要指出与之相联的概念,这样连成一个网状结构,无论从哪个节点出发,无论沿哪个方向,都能访问到图中其他各点。

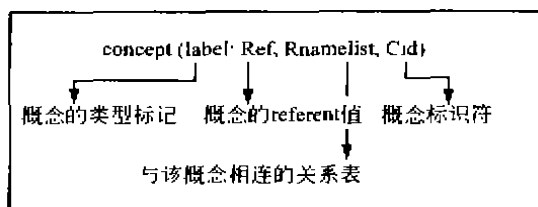


图 1 概念的谓词表示

Fig. 1 The Concept Predicate in Prolog

2 线性分析与线性化

一个概念图有显示图和线性图两种模式。显式图是二维的,是图的形式,直观且便于理解,往往用于和用户的交互。线性图是一维的,虽不直观,但能以正文文件的方式存取且便于计算机打印输出。同一概念图的这两种模式表示形式不同,但表示的含义是完全相同的。CGET 提供了这两种形式的相互转换功能。如图 4 所示。我们把从线性图到内部表示的过程称为线性分析;把从内部表示到线性图的过程称为线性化。这两个过程的实现,实际上就完成了显示图和线性图的相互转换。

2.1 线性分析

线性分析是讨论如何分析线性图,产生相应的内部表示,这种内部表示直接对应着一个显示图。线性图的语法用下列产生式规则来描述:

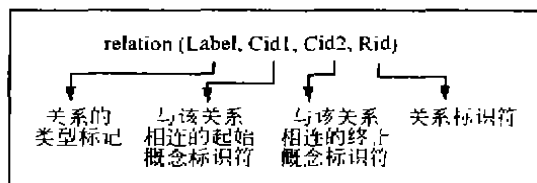


图 2 关系的谓词表示

Fig. 2 The Relation Predicate in Prolog

- 1A) CGRAPH → NODE " . "
- 1B) NODE → C-NODE | R-NODE
- 1C) C-NODE → CONCEPT[RLINK]
- 1D) R-NODE → RELATION[CONLINK]
- 2) CONCEPT → "[TYPEFIELD

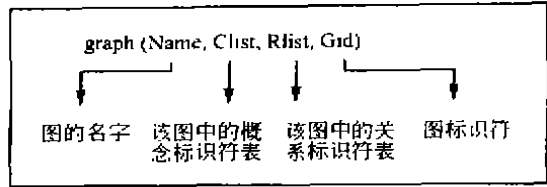


图 3 概念图的谓词表示

Fig. 3 The Graph Predicate in Prolog

[" : " REFFIELD] " "

- 3) RELATION → "(" TYPELABEL ") "
- 4) BLINK → nil | ARC R-NODE | " - " RLIST " , "
- 5) CONLINK → nil | ARC C-NODE | " - " CONLIST " , "

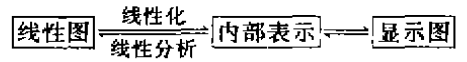


图 4 显示图与线性图相互转换示意图

Fig. 4 The Mutual Change of Display form and Linear form for Conceptual Graphs

- 6) RLIST → {NEWLINE R-NODE} ...
- 7) CONLIST → {NEWLINE ARC C-NODE} ...
- 8) ARC → [NUMBER] { " ← " | " → " } .

实现线性分析的主要思想：对产生式规则中的每一个非终结点，都有一个“分析例程”。例如，C-NODE 分析例程调用 CONCEPT 分析例程，并确定是否调用 RLINK 分析例程；RLINK 分析例程调用 R-NODE 分析例程，而 R-NODE 分析例程又可能调用 CONLINK 分析例程；从 CONLINK 分析例程有可能又回到 C-NODE 分析例程，这样就产生了递归。随着非终结点分析例程的调用，递归一层层地深入，直到遇到一个终结点递归才停止，然后再从调用点逐步返回。

2.2 线性化

线性化是讨论如何从概念图的内部表示产生相应的线性图。如果显示图中不含闭合回路，则问题相对简单。如果显示图中含有闭合回路，则线性化时，首先要考虑如何断开回路。例如，英语“a monkey eating a walnut with a spoon made out of the walnuts shell.”用概念图表示时必然出现闭合回路，如图 5 所示。CGET 表示概念图的内部数据结构，完全能够表示回路，但是，线性图中却无法直接表示回路，需从某个节点断开回路，这个节点就称为断点。因此，线性化要解决的两个基本问题就是：①头节点怎么选，②如何判断某个节点是否为断点。

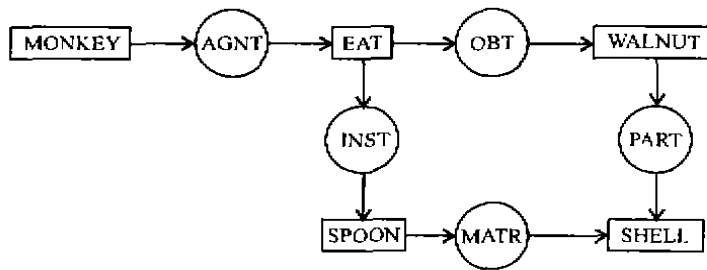


图 5 “a monkey eating a walnut with a spoon made out of the walnut's shell” 的显示图

Fig. 5 The Display Graph for a Monkey Eating a Walnut with a Spoon Made Out of the Walnut's Shell

选哪个节点打头，关系的顺序如何排列，原则上是任意的。不过，选择不同的节点作为线性图的头节点会产生不同的线性图，这些线性图意义上是相同的，选好了很简单。实际上，选择弧最多的节点（无论是概念节点，还是关系节点）作为头节点，选好了这样产生的线性图行数最少，最简单易读。例如，对图 6 中的概念图，若选[EAT]做头节点，则线性图如下：

```
[EAT]—
  (AGNT)→[MONKEY]
  (OBJ)→[WALNUT; * x]
  (INST)→[SPOON]→(MATR)→[SHELL]←(PART)←[WALNUT; * x].
```

若选[SPOON]作头节点，则相应线性图如下：

```

[SPOON]—
  (INST)←[EAT]—
    (OBJ)→[WALNUT: * x]
    (AGNT)→[MONKEY],
  (MATR)→[SHELL]←(PART)→[WALNUT: * x]。

```

显然,后一个图比前一个行数多且易读性差。

如何判断某个节点是否为断点,这个问题相对复杂。断开回路时是从某个概念断开,而不允许从某个关系断开。一旦某个关系被线性化,就一定要连续线性化其后的弧和概念。如果其后的概念是断点,而又尚未量化,那么,这时已经不可能再退回去量化这个概念了。所以,必须在处理完关系之前,先量化断点。这就需要采用向前“看”技术,来预测断点。这时会遇到下列 3 种情况:

- 如果是从一个概念进入回路,回路中的概念 C,对它向前“看”的第二步,就检测出一个已访问过的概念,此时,C 就是断点。
- 如果是从一个关系进入回路,回路中的概念 C,对它向前“看”的第一步就检测出一个已访问过的关系,此时,C 就是断点。
- 对回路中的概念 C,C 和它的关系之间有多条路径,则 C 就是断点。

因此,CGET 采用一个检测断点的算子,对每个概念节点,判断其是否为断点。检测的项目及结果如附表。只当情况 a 成立时,概念 C 不是断点,若当遇到情况 b,c,d 中的任意一种时,C 都是断点。

附表 断点检测情况分析

Tab. The Conditions of Testing a Cut Node

检测项目 情 况	C 的每个关系 R(C 的 前趋除外)是否被访问过	R 的每个概念(C 除外)是否被访问过	C 和 R 之间是否存 在多条路径
a	否	否	否
b	否	否	是
c	否	是	—
d	是	—	—

线性化过程也要遵循线性图的语法。对这组产生式规则中的每个非终结点,都有一个“生成算子”,每个算子用来产生相应的线性表示。线性化算法实际上就是按产生式规则指导什么时候调用哪个算子。

3 用户界面

3.1 基本用户界面

CGET 基本的用户界面由条形主菜单和用户工作窗口两部分组成。用户工作窗口中,概念用旁边带标记的矩形框表示;关系用旁边带标记的圆圈表示;它们之间用带箭头的弧联结。条形主菜单提供了用户窗口可进行的操作:

(1)通过选择 Load 菜单项,输入一个线性图的文件名。当线性图装入后,在工作窗口就显示出同一个概念图的显示图。

(2)通过选择 Select 菜单项,输入一个目标名,则在相应目标(关系或概念)下弹出一个 ACTION 菜单,选择 ACTION 菜单中的各项对该目标进行各种操作。

(3)通过选择 Relation(或 Concept)插入一个关系(或概念),输入关系的类型(或概念的类型标记及所指对象)后,生成一个带标记的圆圈(或带标记的方框),将其移动到适当位置上。

(4)通过选择 Connect 菜单项,输入弧的起点和终点,产生一条由起点指向终点的带箭头的弧。

(5)通过选择 Save 菜单项,编辑后的显示图以线性图显示出来并按用户输入的文件名存入文件中。

(6)通过选择 Delete 菜单项,能够清除整个工作窗口。对每个概念或关系的删除,选择 ACTION 菜

单中的 delete 操作。

(7)通过选择 HELP 菜单项,帮助用户使用该工具。

(8)通过选择 EXIT 退出 CGET。

3.2 用户界面的实现

实现用户界面时,将一个概念图分成方框、圆圈、标记和有向弧 4 个基本元素。提供了画方框、圆、标记、有向弧的函数。

一个工作窗口是将窗口、菜单、图的基本元素集于一个实际的界面上。为了响应用户在工作窗口中的各种操作,还需要一个控制模块,控制对图的基本元素的生成、删除、修改、移动等编辑功能和其他操作。对每个图的基本元素都由两部分构成:①工作窗口中的可见形式(方框、圆、标记或带箭头的弧),这称它为像;②每个像的内部支撑结构。这种支撑结构包含下列信息:

- 存放所有像及其位置的表。
- 描述像之间关系的树。
- 相应的函数集合。

控制模块利用这些信息,实现用户操作:

(1)生成一个像(概念或关系)。调用画框(或圆)、画标识函数;生成相应的内部表示,内部结构如第二节中所述;修改像的支撑结构中的结点位置表和继承树。

(2)删除一个像(概念或关系)。将像及其相关的弧从窗口中抹掉;删除相应的内部表示,修改结点位置表和继承树。

(3)移动一个像。仅仅像的可见元素被替换。当移动时,图的元素及相关弧被抹掉,在相应位置上重画;相应的内部表示不变。

4 结束语

CGET 是实现概念图的最基本的工具,有关概念图的类型定义、语义网,图式等知识表达方法都可以进一步扩充,在此基础上还可以实现概念图的运算和推理。

参 考 文 献

- 1 Jhon F Sowa. Conceptual Structures. UK, Addison-Wesley, 1984
- 2 白振兴. 一种新知识表示方法:概念结构. 计算机科学, 1992, 19(6): 21~26
- 3 Jhon F Sowa. Implementing a semantic interpreter using conceptual graphs. IBM J. RES. DEVELOP., 1986, 30(1): 57~62
- 4 张蕾. 用概念图表示规则及其推理方法. 西北大学学报(自然科学版), 1994, 24(4): 319~322

责任编辑 张素敏

A Conceptual Graph Editing Tool

Zhang Lei

(Department of Computer Science, Northwest University, 710069, Xi'an)

Abstract The designing and implementing of a conceptual graph editing tool in prolog are described, which supports graphic editing and manipulation of conceptual graphs.

Key words conceptual graph; parsing linear form for conceptual graph; linearizing