

## 地图数据库中基本图形对象的拓扑关系及其编辑\*

王 峰 杨树强 陈火旺

(国防科技大学计算机系 长沙 410073)

**摘 要** 地图数据库是地理信息系统(GIS)的核心。本文提出面向对象地图数据库的逻辑数据模型和数据结构,分析了地图的基本图形对象(结点和弧段)间的拓扑关系,围绕地图数据库的拓扑一致性,讨论了各种编辑操作的实现步骤。

**关键词** 地理信息系统, 地图数据库, 拓扑关系。

**分类号** TP391.41

## Topologic Relationships and Editing of Elementary Graphic Objects in Object-oriented Geographical Database

Wang Feng Yang Shuqiang Chen Huowang

(Department of Computer Science, NUDT, Changsha, 410073)

**Abstract** This article proposes the logic data model of object-oriented geographic database. The topologic relationships between the elementary graphic object are analyzed in detail, and the editing operations to be performed on them are identified and analyzed. These editing operations ensure that topologic consistency in the database is preserved.

**Key words** GIS, geographic database systems, topologic relationships

地图是地理信息的主要载体。空间地理对象可以抽象为点状地物类、线状地物类和面状地物类。特定地理区域具有相同专题属性的点状地物对象、线状地物对象和面状地物对象的集合就是该区的一个地图层,如某县的交通网层、水系层、森林覆盖层等。特定地理区域具有相同比例尺的地图层便组成该区域的一张地图。

### 1 面向对象地图数据库的逻辑数据模型(LDM)

图1描述了面向对象地图数据库的逻辑数据模型(LDM)。LDM是一个混合数据模

\* 1996年3月11日收稿

型,它涉及两类数据:(1)地理对象的空间位置数据。空间位置数据分为三种:(a)地理对象的位置,以X,Y坐标对表示;(b)地理对象之间的拓扑关系;(c)地理对象的形状和大小。空间位置数据在地图上通常以结点(node object)、弧段(arc object)、链(link object)、多边形(polygon object)等图形对象表示。(2)地理对象的属性数据。属性数据在地图上通常用图符、颜色、网纹、地名、注记等表示。

图1所示的LDM满足以下四个约束条件:(1)对象类是互斥的,即每个地理对象只能属于一个类;(2)弧段用直线段表示;(3)每对结点最多有一条弧段直接相连,每个结点可以在一条或多条链上,每条链包含一条或多条弧段;(4)两条弧段不能相交;假如相交,用连结于同一结点的四条弧段来表示。

## 2 结点和弧段之间的拓扑关系

在LDM中,点状地物、线状地物、面状地物等地理对象分别以结点、链、多边形等图形对象表示,编辑地理对象就是编辑图形对象。图形对象最终是通过结点和弧段这两个基本图形对象表示的,因此,地图的图形编辑最终是结点和弧段的编辑。每次编辑操作完成后,结点和弧段必须满足上文所列约束条件的(3)和(4),即任意两条弧段不相交,不覆盖,任意两个结点不重合,在结点和弧段间不存在不一致的拓扑关系。由此可见,在编辑操作中必须提供自动检测和维护数据库拓扑关系一致的机制。我们的策略是根据结点和弧段间的拓扑关系来设计能消除不一致拓扑关系的编辑操作,这些操作是系统的基本编辑操作,任何复杂的编辑操作都可分解成它们完成。

在结点和弧段这两类图形对象间存在三组拓扑关系:

node/node、node/arc、arc/arc。我们用 Pullar 和 Egenhofer 提出的 Four-Intersection 点集方法来定义这三组拓扑关系<sup>[1,2]</sup>。该方法将图形对象当作一个由两部分组成的点集:边界(以@表示)和内部(以#表示)。两对象间的关系可以通过同时计算4个子集

的交获得,这4个交是: $@ \cap @$ ,  $# \cap #$ ,  $@ \cap #$ ,  $# \cap @$ 。当交集为空时,以 $\Phi$ 表示,否则以1表示。这4种求交运算,产生16种结果,剔去拓扑上不可能的关系,得到12种拓扑关系。

定义1 弧段arc的边界是该弧段的两个端点,表示为 $@arc$ 。弧段中的点组成弧段的内部(不包括两端点),表示为 $#arc$ 。弧段的内部( $#arc$ )可由插值得到。

定义2 结点node的边界是其自身,表示为 $@node$ ;结点的内部为空,即 $#node = \Phi$ 。

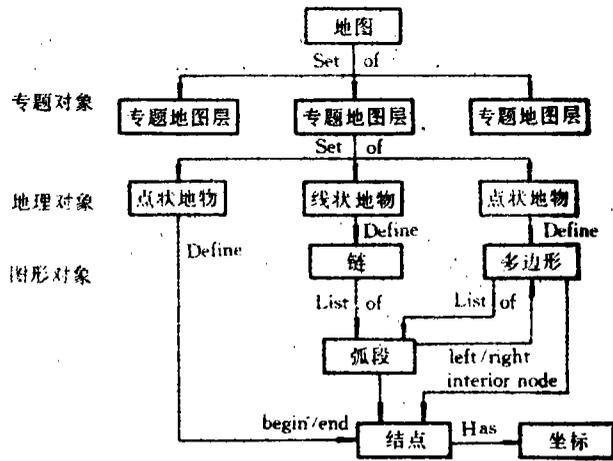


图1 面向对象地图数据库的逻辑数据模型(LDM)

$\Phi$ 。

表 1 描述了基于 Four-Intersection 点集方法得到的弧段和结点间可能存在的拓扑关系，相离关系被省略了，因为它对数据库中拓扑一致性无影响。

表 1 结点和弧段之间的拓扑关系

拓扑关系	例子	@∩@	#∩#	#∩@	@∩#
On(node/node)	$n_1 : n_2$	1	$\Phi$	$\Phi$	$\Phi$
On(node/arc)	$n \text{---} a$	1	$\Phi$	$\Phi$	$\Phi$
Cross(arc/node)	$\begin{array}{c} n \\ \text{---} a \\ a_1 \end{array}$	$\Phi$	$\Phi$	1	$\Phi$
Intersect(arc/arc)	$\begin{array}{c} \text{---} a_1 \\   \\ \text{---} a_2 \end{array}$	$\Phi$	1	$\Phi$	$\Phi$
Touch(arc/arc)	$a_1 \text{---} a_2$	$\Phi$	$\Phi$	1	$\Phi$
	$\begin{array}{c} a_1 \text{---}   \\   \\ a_2 \end{array}$	$\Phi$	$\Phi$	$\Phi$	1
Meet(arc/arc)	$a_1 \text{---} a_2$	1	$\Phi$	$\Phi$	$\Phi$
Equal(arc/arc)	$a_1 \text{---} a_2$	1	1	$\Phi$	$\Phi$
Contain(arc/arc)	$\begin{array}{c} a_1 \text{---} \\ \text{---} \\ a_2 \end{array}$	$\Phi$	1	1	$\Phi$
	$\begin{array}{c} a_1 \text{---} \\ \text{---} \\ a_2 \end{array}$	$\Phi$	1	$\Phi$	1
Overlap(arc/arc)	$\begin{array}{c} a_1 \text{---} \\ \text{---} \\ a_2 \end{array}$	$\Phi$	1	1	1
	$\begin{array}{c} \text{---} \\ \text{---} \\ a_2 \end{array}$	$\Phi$	1	1	1
ComBounds(arc/arc)	$\begin{array}{c} \text{---} a_1 \\ \text{---} a_2 \end{array}$	1	1	1	$\Phi$
	$\begin{array}{c} \text{---} a_1 \\ \text{---} a_2 \end{array}$	1	1	$\Phi$	1

### 3 编辑基本图形对象

地图的图形编辑涉及地理对象的位置、大小、形状和对象间的拓扑关系。同时，编辑操作还必须维持地图数据库中拓扑关系的一致性。当一图形对象被插入库中或从库中删除时，地图数据库管理系统完成两方面的工作：(1)计算该对象同库中现存图形对象的拓扑关系；(2)根据计算结果进行相应的编辑操作<sup>[2]</sup>。我们以编辑多边形对象为例，说明如何将复杂对象的编辑转化为基本图形对象的编辑。

(1)插入一个多边形对象：1)插入构成该多边形对象边界的各弧段；2)修改与该多边形对象相邻的各多边形边界；3)修改包含在该多边形对象区域内所有结点对象的数据成员 polygon；4)修改包含在该多边形对象区域内的所有弧段的数据成员 left\_polygon、right\_polygon；5)删除冗余的弧段。

图 2 给出一个插入多边形对象的例子。左图是一幅已存入数据库的小地图， $A_5$  是待插入的多边形对象，右图是插入  $A_5$  后的地图。从图 2 可以看出，新对象  $A_5$  引起了多边形对象  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$  的变化，同时又改变了链 L 的数据。当插入多边形对象  $A_5$  时，调用插入弧段算法产生新的结点 11、12、13、14、15、16。同时将弧段(9, 5)、(5, 4)分别分割成

(9, 16)和(16, 5)、(5, 13)和(13, 4); 调用修改弧段算法, 修改弧段(16, 5)的数据, 调用删除弧段算法, 删除弧段(12, 5)、(11, 5)、(14, 5)、(15, 5)。

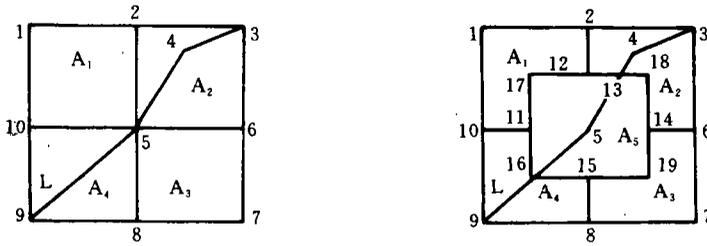


图2 单个多边形对象的编辑

(2)删除多边形对象: 1)选择和修改相邻多边形对象, 如插入新弧段, 修改它们的边界; 2)修改包含在原多边形对象区域内所有结点的数据; 3)修改所涉及弧段的数据; 4)删除冗余的弧段。

如图2, 考虑从右图中删除多边形对象  $A_5$ : 1)修改多边形对象  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$  的边界(插入弧段(11, 5)、(12, 5)、(14, 5)和(15, 5)); 2)修改弧段(17, 12)、(12, 13)、(13, 18)、(18, 14)、(14, 19)、(19, 15)、(15, 16)、(16, 11)、(11, 17)、(16, 5)和(5, 13)的数据; 3)删除弧段(17, 12)、(12, 13)、(13, 18)、(18, 14)、(14, 19)、(19, 15)、(15, 16)、(16, 11)、(11, 17)。

#### 4 结束语

我们用图论中的欧拉恒等式(Euler Constant)来检验数据库的拓扑一致性:  $v - e + r = k + 1$ 。  $v$  代表结点对象(node object)个数,  $e$  代表弧段对象(arc object)个数,  $r$  代表多边形对象(polygon object)个数(包括地图的外围),  $k$  是子图个数。

在本文中, 我们提出了编辑面向对象地图数据库中基本图形对象的操作, 并且维持数据库的拓扑一致性。对链、多边形等图形对象, 以及点状地物、线状地物、面状地物等地理对象的编辑, 都可由这些操作实现。目前, 我们在研制的地理信息系统原型  $YH-GIS$  中以交互式图形的方式实现这些操作。

#### 参考文献

- 1 Pullar D V, Egenhofer M J. Toward formal definitions of topological relations among spatial objects. In proceedings of the 3rd international symposium on spatial data handling, sydney, Australia, August, 1988: 225~241
- 2 Kufoniyi O, Moloenaar M, Bouloucos T. Topologic editing of relationally structured single-valued vector maps. ITC Journal, 1993(4): 332~339

(责任编辑 张 静)