目 录

张庆国

第三章　文献情报检索的数据结构和检索技术

文献情报检索系统的主要处理对象是大量的数据，能否合理地组织数据，能否在计算机上更有效地存贮和处理数据，直接影响系统运行的效率。本章首先介绍几种情报检索中常用的数据结构，然后讨论计算机软件技术中与情报检索关系最为密切的一个方面——查找技术。

3.1　情报检索中的数据结构

3.1.1　逻辑结构与物理结构

1.　逻辑结构

我们已经涉及到了检索文档和文档中记录的关系，在这里，我们可以把记录看成为"基本元素"，把文档看成是由这些基本元素构成的整体。在这个文档中，记录与记录之间一般存在某种关系，这种关系是逻辑上的，由记录内容所决定的，于是，这些记录和它们之间的逻辑关系构成一个逻辑结构。

我们在其他方面也遇见过一个整体中的基本元素以及这些基本元素之间的关系，例如，把主题词和文献看成是一个集合中的元素，则在主题词和文献之间存在着逻辑关系——标引关系，在主题词与主题词之间存在着用、代、属、分、参等关系；在一个检索提问式中，各检索词之间也存在关系——逻辑组配关系，等等。

对数据间关系的描述称为数据的逻辑结构，形式地可以用一个

二元组

$$B=(K,\ R)$$

来表示，其中K是基本元素的有穷集合，R是K上的关系的有穷集合。

在对数据结构的讨论中，通常把基本元素称为“结点”，注意结点的概念是相对的。例如在文档和记录的关系中，记录是结点，而在记录和其属性的关系中，属性是结点，记录是由属性组成的整体。

设$B=(K,\ R)$是一个逻辑结构，$r\in R$，$k,\ k'\in K$，如果$<k,\ k'>\in r$，则称k'是k的后继，k是k'的前驱。称k和k'是相邻的结点（都是相对r而言）。如果不存在一个k'使$<k,\ k'>\in r$，则称k为关于r的终端结点。如果不存在一个k'使$<k',\ k>\in r$，则称k为关于r的开始结点。如果k不是终端结点，也不是开始结点，则称k为内部结点。

我们只讨论包含一个关系的R，即$R=\{r\}$。

数据的逻辑结构常常简称为数据结构。数据结构分为线性结构和非线性结构。在线性结构里只有一个终端结点和一个开始结点，并且所有的结点都最多只有一个前驱和一个后继。线性表是典型的线性结构。非线性结构中最重要的是树结构。树结构中只有一个称为根的结点没有前驱。其他结点有且仅有一个前驱。最一般的情况是对结点的前驱和后继都不作限制。这种结构称作图。对于线性表、树和图，我们将分别在 3.1.2 、3.1.3 、3.1.4 中简单介绍。

2 物理结构

数据的逻辑结构是从逻辑的观点来观察数据，它与数据的存贮无关，是独立于计算机的。数据的物理结构是逻辑结构在计算机存

贮器里的实现。它又称为存贮结构。数据的物理结构是依赖于具体的计算机的。

计算机的存贮器是由有限多个存贮单元组成的，每个存贮单元有唯一的整数地址。各存贮单元的地址是连续编码的，每个存贮单元Z都有唯一的后继单元suc(z)。如果z'是z的后继单元，则称z和z'是相邻的。一片相邻的存贮单元的整体叫作一个存贮区域，记作M。

设有逻辑结构B=(K, R)。要把B存贮在计算机中，首先必须建立一个从K的结点到M的单元的映象S：K->M，即对于每一个k∈K，都有唯一的z∈M，使S(k)=z，同时这个映象还必须明显地或隐含地体现关系R。

我们用ωk表示结点k的值，用LOC(k)表示结点k对应的存贮单元的地址。用CON(z)表示地址为z的存贮单元里存贮的内容，于是有

$$\omega k = CON(LOC(k))$$

有四种基本的存贮映象方法，即有四种基本的物理结构，它们是：

① 顺序的方法

这种方法主要用于线性的数据结构，它把相邻的结点存贮在相邻的存贮单元里，结点之间的关系由存贮器的单元的邻接关系来体现。如果结点k所占存贮的第一个单元为z，而最后一个单元为z_1，那么z_1的后继单元$z' = suc(z_1)$就是k的后继k'的第一个存贮单元。

② 链接的方法

这种方法是给结点附加上指针字段，即是将对应于结点的存贮

单元分为两部分。一部分存放结点本身的信息，称数据项，另一部分存放此结点的后继结点即对应的存贮单元的地址，称指针项。指针项可以包括一个或多个指针，以指向结点的一个或多个后继，或记录其他信息。

③ 索引的方法

在线性的结构里，结点可以排成一个序列，k_1，k_2，…，k_n，每个结点k_i在序列里都有对应的位置数i，这个位置数就是结点的索引。索引的存贮结构就是用结点的索引号i（或关键码A_i）和结点的存贮地址来建立附加的索引表，使得索引表里第i项的值就是第i个结点（即结点k_i）的存贮地址。

④ 散列的方法（Hash法）

主要思想是根据结点的值来确定它的存贮地址。在结点k的各个字段里取一个或几个字段的值$\omega_i k$作为关键码，结点k_j对应的存贮地址由函数H确定

$$LOC(k_j) = H(\omega_i k$$

这种方法又称KAT法（Key-to-Address Trans lation）用散列法进行存贮映象，K的结点k随机地、分散地存贮在M的存贮单元里。

在讨论数据的物理结构即存贮结构时，有两点需要注意。

首先，数据的逻辑结构与物理结构是无关的，一种逻辑结构可以由多种存贮结构实现。例如，一棵逻辑上的树可以由链接的方法或散列的方法实现，在经过适当的处理后，它也可以由顺序的方法或索引的方法实现。一种存贮结构也可以实现多种逻辑结构，例如，散列的方法可分别实现线性表、树和图。

其次，数据的存贮结构与数据的查找紧密相关。在设计存贮结构

构时，我们给出了一个映象 S：S(k) = z，于是把结点 k 存贮到 z 中；在查找数据时，我们同样是利用这个映象。设我们要查找结点 k，S(k) = z，于是到单元 z 中取出结点 k。

查找技术在 3.2 中讨论。

3.1.2 线性表

线性表的逻辑结构是 B = (K, R)，其中 K 是包含 n 个结点的集合，R 仅包含一个关系 N

$$N = \{<k_{i-1}, k_i> \mid 2 \leq i \leq n\}$$

即是说，K 里所有的结点都可以按关系 N 排成一个序列 k_1，k_2，…，k_n。在这个序列里，每个结点都有相对位置 i，i 叫作 k_i 的索引。表中的结点又称作表目，或元素。

线性表的每个表目可以包括一个或多个字段。能够唯一地标识表目的一个字段或一组字段叫作关键码。例如，按文献号顺序排列的主文档是一个表，每一篇文献记录是一个表目，每一个表目中有文献号、篇名、作者、出版年等字段，其中“文献号”字段可以唯一地标识一篇文献，文献号就可以作为关键码。非关键码字段的信息又称为属性。

用不同的方法（顺序、链接、索引、散列）存贮一个线性表，可以有不同的用途，下面介绍情报检索中常用的一些。

/. 顺序表和目录表

用顺序的方法来存贮一个线性表，即把表目一个接一个地放在顺序的一片单元里，这样的表称为顺序表。

例如，情报检索中的主题词，如果考虑其字顺关系，则它们是一个线性结构。在目前的大多数检索系统中，也是按字顺把词典存

贮在存贮器上的，因此，词典是一个顺序表。

再例如，在SDI中，新到文献也是排成一个顺序表的。

顺序表的逻辑结构和物理结构都很简单，并且存贮密度大，节省空间，由于结点与结点之间在物理上的关系反映了原有的逻辑关系，所以可以在顺序表上再加索引后成为索引顺序文件，这是目前词典常采取的一种形式。

但是，顺序表的结点增删比较麻烦。增加一个结点到与其相应的位置后，要向后移动其后面的所有结点；删去一个结点后，为了填补空闲区域，要往前移动其后面的所有结点。由于这个原因，大型词典的词汇更新和删除工作往往是成批处理的。

在一个表中，表目的大小往往是不一致的，如倒排档中的主题词记录，主文档中的文献记录等，可变长的记录处理起来很不方便。为此，可以把表分成两部分，一部分叫目录表，它的表目包括两个字段，一个字段是关键码，另一个字段是地址。另一部分叫属性数据，它包括表里关键码以外的其他信息。属性数据与目录表分别存在不同的存贮区域里，目录表里结点的地址字段指向属性数据存贮区域里对应于这个关键码的信息。目录表的结点是固定长的。

例如，主文档中的文献记录是可变长的，检索不便，我们便为其建一个目录表。这个目录表就是在第二章中提到的主文档索引文档。

2 栈

栈是情报检索中常用的一种线性表。我们在2.5.1中曾给出了它的非形式化定义，在这个定义中包含了它的存贮结构特点。

栈可以用顺序表的方式实现。分配一块连续的存贮区域存放栈里的表目，并用一个变量T来指明当前的栈顶（图3.1）

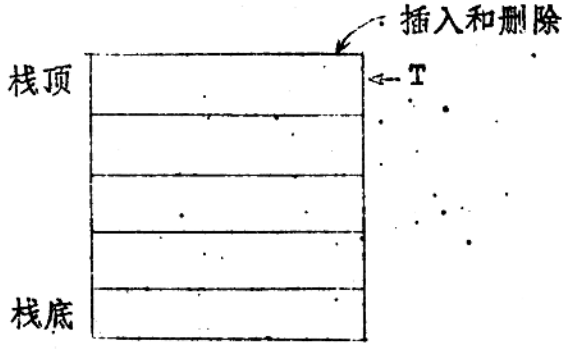


图3.1　栈

栈的基本运算有：

① Push：插入（或叫推入）一个元素到栈顶。

② Pop：从栈顶删除一个元素，也就是托出栈顶元素。

③ Top：看一下当前的栈顶元素。

我们在2.5.1中曾给出一种检索逻辑式的运算方法：首先将检索式转换成逆波兰表示法，然后对逆波兰检索式进行运算。运算规则是：每个运算符的两个运算项目是逆波兰形式中紧挨在这个运算符之前的两个运算项目。我们下面给出一个算法，这个算法把上述的两个步骤并作一步完成，即：它不形成独立的逆波兰形式，而是在转换过程中就正确地选择运算项目及它们的运算符并对其进行运算。

我们用两个栈。一个是我们曾用过的运算符栈，在这里称它为sop栈；逆波兰表示区在这里也用一个栈，我们称它为svar栈，是表示运算项目无条件进栈的意思。三种逻辑运算符的优先数仍如2.5.1中所规定。在下面的算法中，C是逻辑式中的字符，可能是是运算项目，也可能是运算符；OP是sop栈中的元素，即运算符；

V_1、V_2、R都是svar栈中的元素，即运算项目，其中R是运算的中间结果。

算法3.1　检索逻辑式的运算

① 从左向右扫描逻辑表达式。若表达式已取完则转步骤⑦，否则取来一个字符C。

② 若C是运算项目，则Push(svar，C)，转去执行步骤①。

③ 若C是运算符且运算符栈为空，则

Push(sop，C)，转步骤1

④ 若C是运算符且运算符栈非空，则

Top(sop，op)，比较栈顶运算符op和新取来的运算符C的优先数，若C的优先数不小，则Push(sop，C)，转步骤1。

⑤ Top(sop，op)，Pop(sop)，

Top(svar，V_1)，Pop(svar)，

Top(svar，V_2)，Pop(svar)，

执行运算V_1opV_2，运算结果为R

Push(svar，R)

⑥ 转步骤③

⑦ 反复执行步骤⑤，直到sop为空则结束。svar的栈顶元素值就是逻辑式的计算结果。

除了顺序表、目录表和栈之外，线性表还包括链表、队列等形式，介绍从略。

3.1.3 树

树形结构是结点之间具有分支关系的结构，它很类似于自然界中的树。在非线性结构中树形结构是最重要，应用最广泛的一种，尤其是在大量的数据处理中更为突出。

树的逻辑结构定义为 B＝(K，R)，其中K是包含n个结点的有穷集合。R是K上关系的集合，它只包含一个元素N，并满足

a．有且仅有一个结点 $w \in K$，w被称为树的根，它相对关系N没有前驱。

b．除根结点w外，K中每个结点相对关系N都有且仅有一个前驱。

c．对于K里除w外的任何结点k，都存在一个结点序列 k_0，k_1，…，k_s，使得

$$k_0 = w,\ k_s = k,\ <k_{i-1}, k_i> \in N,\ 1 \leq i \leq s$$

这样的结点序列称为从根到结点k的一条路径。

树结构在客观世界和情报检索领域中大量存在。

例如，分类法是一种树形结构

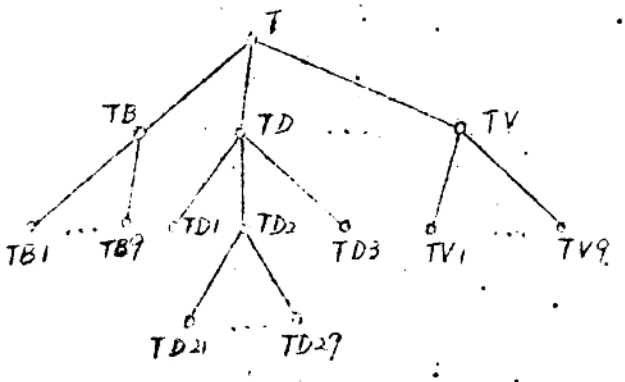


图3.2 分类树

再例如，主题词若按其字母的等级顺序，则也是一棵树。

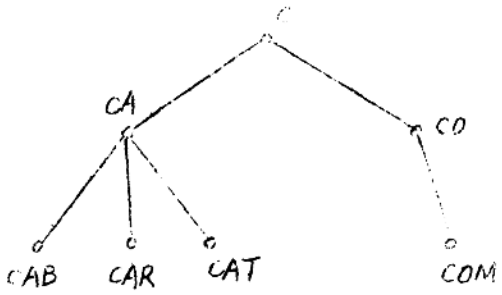


图 3. 3　主题词树

再例如，逻辑表达式其实是树形结构而不是线形结构。表达式（A＋B＊（C＋D））＋E＊F可表示为

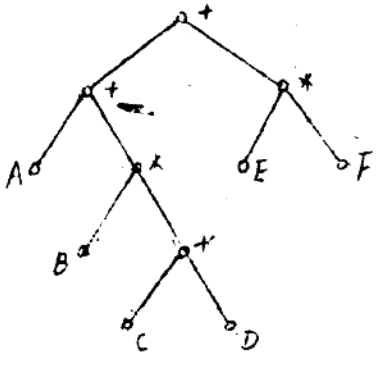


图 3. 4　表达式的树形结构

在一棵树中，终端结点即没有后继的结点称为树叶。在实际应用中，树叶相对于其他结点往往有特殊的意义。如在图 3．2 中，树叶是最低级的类目；在图 3．3 中，树叶是实际的主题词；在图 3．4 中，树叶是运算项目。

在图 3．4 中，我们给出了一个每个内部结点只有两个分支的树形结构，这称之为二叉树。二叉树与树不同，它的严格定义是

定义：二叉树是结点的有限集合，这个有限集合或者为空，或

者由一个根及两个不相交的分别称作这个根的左子树和右子树的二叉树组成。

这是个递归的定义。二叉树可以是空集合。因此根可以有空的左子树或右子树，或左右子树皆为空。图3.5是二叉树的五种基本形态。

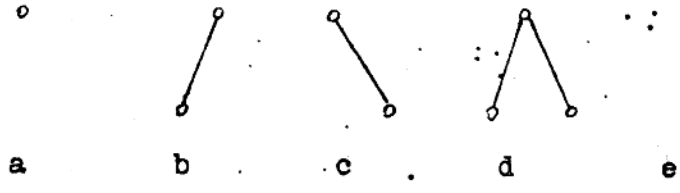


图3.5 二叉树的五种基本形态

对于二叉树，有两点需要特别注意。①二叉树的结点的子树个数只能是0，1，2；②结点的子树要区分为左子树和右子树。

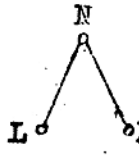
树和二叉树都可以用多种存贮方法实现。

1. 顺序的方法

树和二叉树本身都是非线性的数据结构，要用顺序的方法实现存贮，首先要把它们线性化。线性化的主要方法是对树或二叉树进行周游。

二叉树的周游方法

考虑到二叉树的基本组成部分是



所以，周游二叉树有三种主要的方式

(1) 前序法（NLR次序），其递归定义是