网树介绍

一、树与网树的区别

提起树，人们通常会想起如图 1 所示的自然界中的树。

图 1　自然界中的树

在数据结构中有一种极为重要的一种非线性数据结构，被称为树型数据结构（简称为树型结构），其如图 2 所示，这是因为这种结构看起来像一棵倒立的树，即树根朝上，树叶朝下的树。在树型结构中除了结点、树根、叶子、孩子和双亲等概念以外，还是存在许多与人类家庭关系相似的概念，如祖先、子孙、左孩子、右孩子、兄弟结点、堂兄弟结点、结点的层次和树的深度等概念。它具有以下的特点：

（1）每个结点有零个或多个孩子结点；

（2）若一个结点有多个孩子结点，则左边的孩子结点称为左孩子，而右边的孩子结点称为右孩子；

（2）没有父结点的结点称为树根；

（3）没有孩子结点的结点称为叶子；

（4）每一个非根结点有且只有一个父结点；

（5）除了根结点外，每个孩子结点可以分为多个不相交的子树；

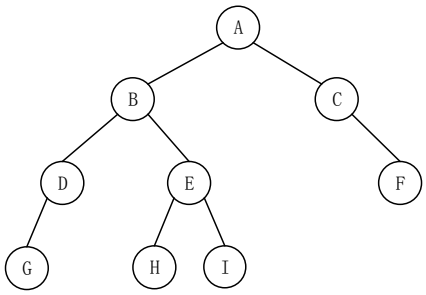

图 2　数据结构中的树型结构

例如在图 2 中，结点 A 可以称为树根；结点 A 的孩子结点是结点 B 和 C；

反过来说，结点 B 和 C 的双亲结点是结点 A；结点 G、H、I、F 可以称为叶子；结点 G 和 H 是结点 B 的子孙结点；我们可以说结点 B 是结点 G 和 H 的祖先结点；结点 D 和 E 是兄弟关系；而结点 D 与结点 F 则属于堂兄弟关系，这是因为这两结点都是以结点 A 为树根的第 3 层上；这棵树的深度为 4。显然树型结构可以用来表示数据之间一对多的关系，被广泛应用于求解多种问题，如文件的目录结构就是树型结构。不仅如此，其在现实生活中也有重要的应用，如可以用于描述人类的族谱。

然而树型数据结构也有缺点，例如尽管其在现实生活中所描述族谱，但是这种族谱是一种父系族谱，如果用其来描述父母双系族谱则无能为力，这是因为现实生活中，每个人都有父母两个双亲构成，用一对多的树型结构则难以表达实际这种二对多或多对多的情况。为此课题组提出了一种全新的数据结构名为网树型数据结构，简称网树结构或网树，这是因为网树结构是包含许多与树数据结构类似的概念，如：树根、叶子、层（级）、父亲、孩子等。尽管如此，与树结构相比，网树结构具有如下显著不同的特点。

- 一棵网树可能有 n 个根结点（$n\geq 1$）；
- 相同名称的网树结点可能出现在网树的不同层中多次出现，为了有效的描述一个结点，用n_j^i来表示第 j 层的结点 i；
- 除根结点外的任何一个结点可能有不止一个双亲结点，所有的双亲结点都必须在同一层上，即非根结点n_j^i ($j>1$)可能有多个双亲结点 $\{n_{j-1}^{i_1}, n_{j-1}^{i_2}, ..., n_{j-1}^{i_k}\}$ ($k\geq 1$)；
- 从一个网树结点到其祖先结点或根结点可能有很多路径。
- 在深度为 m 的网树中，第 m 层的叶子结点称为绝对叶子结点。
- 一个从树根到绝对叶子结点的路径被称为完全路径。

二、间隙约束模式匹配简介

间隙约束模式匹配其模式可以写作 $p_1[min_1,max_1]p_2...[min_{j-1}, max_{j-1}]p_j ... [min_{m-1}, max_{m-1}]p_m$ 的形式，这里 min_{j-1} 和 max_{j-1} 分别表示模式子串 p_{j-1} 与 p_j 之间可以通配的最小和最大字符数量，例如：模式 A[0, 1]T 可以表示字符 A 与 T 之间可以没有通配符或者 1 个“？”通配符，即：模式 A[0, 1]T 可以表示 AT 或 A?T 两种模式；若 A[0, 1]T[0, 1]C 则可以表示 ATC 或 AT?C 或 A?TC 或 A?T?C 四种模式，显然一个间隙约束模式可以表达的具有“?”通配符的模式数量是与模式串长度呈指数形式增长，因此，间隙约束模式比传统通配符“?”和“*”更为灵活。

间隙约束模式匹配不但更难于求解而且形式多样，目前存在三种形式：无特

殊条件、一次性条件和无重叠条件，下面举例说明这三种形式的区别。

例 1： 给定模式 $P=p_1[min_1,max_1]p_2[min_2,max_2]p_3$ =A[0,1]C[0,1]A，其在序列 $S=s_1s_2s_3s_4s_5s_6s_7s_8$=ACAACACA 中的出现个数为 5，如图 3 所示。

		1	2	3	4	5	6	7	8		
S	=	A	C	A	A	C	A	C	A		出现位置描述
		A	C	A						第一种出现	<1,2,3>
		A	C	·	A					第二种出现	<1,2,4>
				A	·	C	A			第三种出现	<3,5,6>
					A	C	A			第四种出现	<4,5,6>
							A	C	A	第五种出现	<6,7,8>

图 3　模式在序列中出现情况

无特殊条件是相对于一次性条件和无重叠条件而言的，其是指模式在序列中的所有出现均可以使用，即对出现没有任何约束的方式。因此例 1 中模式 P 在序列 S 的出现数为 5，即全部 5 个出现均是无特殊条件下的出现。

一次性条件是指序列中任何字符只能被任意模式子串最多使用一次。在例 1 中，模式 P 在序列 S 的出现数为 2，即两个出现：{<1,2,3>、<4,5,6>}或{<1,2,4>、<3,5,6>}（由于序列长度为 8，模式长度为 3，因此一次性条件下出现最多为 8/3=2.66，即理论上最多也只能有 2 个出现）。

无重叠条件序列模式挖掘是指序列中任何字符只能被同一个模式子串最多使用一次，与一次性条件不同之处在于，可以被不同模式子串多次使用。在例 1 中，出现<1,2,3>和出现<1,2,4>构成了重叠出现，因为 s_1 被模式子串 p_1 两次使用；然而出现<1,2,3>和出现<3,5,6>构成了无重叠出现，尽管 s_3 被使用两次，但是分别被模式子串 p_3 和 p_1 使用，构成了无重叠出现。因此例 1 中共有三个无重叠出现{<1,2,3>、<3,5,6>、<6,7,8>}。

三、网树与间隙约束模式匹配简介

例 2： 依然以例 1 为例进行说明，即给定序列 S=ACAACACA 及模式 P=A[0,1]C[0,1]A，所有的出现为 5 个，分别为<1,2,3>、<1,2,4>、<3,5,6>、<4,5,6>和<6,7,8>，这些出现可用如图 4 所示的一棵网树表示。

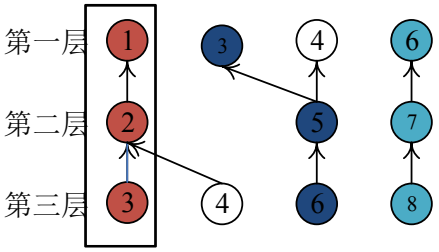

图 4　一棵网树

显然网树上存在 5 条不同树根-叶子路径，其中任意一条树根-叶子路径均对

应一个出现，例如从根结点 1 到达第三层叶子结点 4 的一条路径为<1,2,4>，这条路径与出现<1,2,4>是一致的。

由于网树中多个不同层上的结点可以具有相同的结点标签，如结点标签为 6 的结点既在第一层也在第三层，利用这一特点可以有效地甄别出哪些字符可以重复使用，哪些则不能，因此无重叠条件模式匹配问题就是网树上任意结点最多被使用一次。图 5 中三种不同颜色结点所构成的出现就构成了无重叠条件模式匹配问题的解。