规则描述中的统配符

黄玉霞　（中国科学院软件研究所　100080）

摘要：本文在规则描述中引入了统配符，并给出了相应的处理方案，可应用于信息管理与专家系统领域。

关键词：规则　统配符

一、统配符的引入

众所周知，DOS 文件系统中引入的统配符‘*’是可以与任何文件名匹配的符号，它给查找文件带来了极大的方便。例如要找一个批处理文件 ucdos.bat，而又不确切记得前面的 ucdos 时，可先用 dir *.bat 列出所有批处理文件如下：

a.bat

b.bat

ucoos.bat

在信息管理或专家系统中，经常需要描述这样一些规则，其形式为

表 1

		规则 1	规则 2	…
条件	c_1	c_{11}	c_{12}	…
	c_2	c_{21}	c_{22}	…
	c_3	c_{31}	c_{22}	…
	⋮	⋮	⋮	⋮
	c_m	c_{m1}	c_{m2}	…
动作集合（或序列）	a_1	a_{11}	a_{12}	…
	a_2	a_{21}	a_{22}	…
	a_3	a_{31}	a_{32}	…
	⋮	⋮	⋮	⋮
	a_n	a_{n1}	a_{n2}	…

规则 1 表示当条件 $c_1 = c_{11}$，$c_2 = c_{21}$，…，$c_m = c_{m1}$ 时发生动作 a_{11}，a_{21}，a_{31}，…，a_{n1} 或顺序发生动作 a_{11}，a_{21}，a_{31}，…，a_{n1}。

这里的条件是通过枚举的方式给出的，每个条件 c_i 的值域是有限的（设为 l_i），因而规则的最大个数是 $l_1 * l_2 * \cdots * l_m$。随着条件数 m 的增大，这个排列数可能是相当大的。事实上，大量存在着这样的情况：某种条件的取值可以按其对后面的动作影响作新的划分。例如，若条件 c_1 取值为 $c_{14}, c_{15}, \cdots, c_{1l_1}$ 时后面的动作相同，c_1 可以进一步划分为 $c_{11}, c_{12}, c_{13}, c_1 *$。这个 $c_1 *$ 我们就用统配符 * 号来表示，它代表 c_{11}, c_{12}, c_{13} 以外 c_1 的所有可选值。当枚举集 $\{c_{11}, c_{12}, c_{13}\}$ 变为空集时，它表示可选值集合中的所有元素，与文件系统中的统配符意义相近。

二、规则中的列明优先原则

在引用规则时，条件的匹配采取列明优先的原则，即当条件的列明值和统配符值同时存在时，优先选用列明值相应的规则，施加其相应的动作。

三、统配符的范例

我们最近在开发国际集装箱运输运价系统时，发现集装箱运输的海运费费率是和集装箱的规格、箱型、货类、卸港、客户、运输条款等诸多条件相关的。

其中：集装箱规格 size 的值域为：{20#，40#}

箱型 type 的值域为：{GP（普通箱），HC（高箱），FR（冷藏箱），OT（开顶箱）}

货类 cargotype 的值域为：{GC（干货），HZ（危品），*（其他）}

卸港 pod（port of disload）的值域为：$\{NY（纽约），HMB（汉堡），\ldots\}_{300}$

运输条款 term 的值域为：{YY（场到场），Y D（场到门），DY（门到 场），DD（门到门）}

客户 customer 的值域为：{客户 1，客户 2，*（其他客户）}

部分海运费费率规则表 OFTRATE 可表述为：

表 2

size	type	cargotype	pod	term	customer	oftrate(usd)
20	GP	GC	NY	YY	客户 1	800
20	GP	GC	NY	YY	客户 2	810
20	GP	GC	NY	YY	*	850
20	GP	HZ	NY	YY	客户 1	1000
20	GP	HZ	NY	YY	客户 2	1100
20	GP	HZ	NY	YY	*	1200
20	GP	*	NY	YY	客户 1	900
20	GP	*	NY	YY	客户 2	920
20	GP	*	NY	YY	*	960

表 2 中，前面几列是条件，oftrate 是动作，表示费率 = 800 等值，表体列出的是到纽约的 20 # 普通箱用 YY 运输条款对不同客户和货类的费率值。在表达客户值时用‘ * ’表示客户 1 和客户 2 以外的其他用户，而客户 1，客户 2 是货运公司的长期用户，享受各自的优惠费率。在表达货类时，用‘ * ’表示 GC 和 HZ 以外的其他货类，其运价介于干货与危品之间。

当没有枚举值出现时，‘ * ’也可以代表全部值的集合。例如：当运价与运输条款无关时，则用‘ * ’表示运输条款的全部值。

表 3

sige	type	cargotype	pod	term	customer	oftrate
20	GP	GC	HMB	*	客户 1	700

表 3 表示的是为客户 1 运 20 # 普通箱到汉堡，干货类货不管用什么运输条款费率都是 700。

如果还存在另一条费率规则：

表 4

size	type	cargotgpe	pod	term	customer	oftrate
20	GP	GC	HMB	*	*	750

表 4 则表示在上例中除客户 1 以外的其他客户(包括客户 2)，其海运费费率均为 750。

列明优先的规则在这里体现为：对于客户 1，因为在表 3 中有列明的客户值 customer = 客户 1，因而海运费率 = 700，而不是 750。对于客户 2 和其他客户，因为没有列明匹配的客户值，只能用表 4 的规则，即 customer = * 的费率 750。

四、规则引用中‘ * ’的处理
——列明优先原则的实现

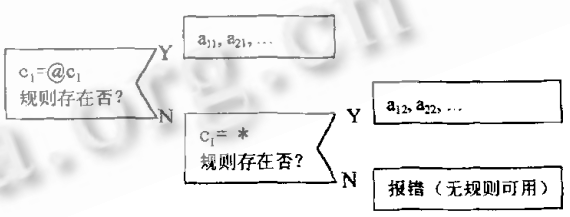
我们先从最简单的一个条件的规则来分析。设 c_1 的值域为 $\{c_{11}, c_{12}, \ldots, c_{1l_1}, *\}$，对于 $c_1 = c_{11}$，一次找到的匹配规则可能有两条：

表 5

	规则 1	规则 2
c_1	c_{11}	*
动作集	a_{11}	a_{12}
	a_{21}	a_{22}
	⋮	⋮

对于 c_1 的任意指定值@c_1(假定为 c_{11})，列明优先在程序中的实现方案有两个。

方案 1：



方案 1 对应的选择语句有两个。

方案 2：

用一个选择语句一次把匹配的规则都找到，然后通过排序找出列明优先者。大家知道，在 ASC11 字符集中，‘ * ’号位于字母、数字符之前，更位于汉字之前(注：选‘ * ’作为统配符出于两方面的考虑：(1)用户易于接受，(2)可选取值集合为字母、数字、汉字串)。对于给定的值 c_{11}，我们只要用‘$c_1 = c_{11}$，or c_1 = * ’一次查找，而将可能找到的两条规则用降序排列，取满足条件的第一个规则即可。用的 SQL 语言如下：

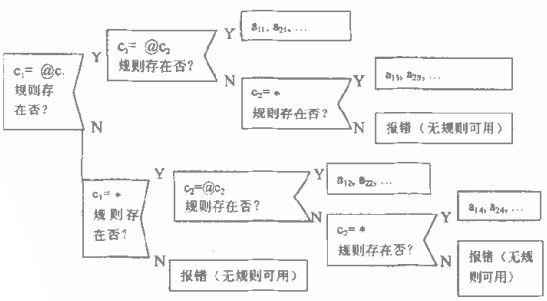
SELECT * INTO 临时表 FROM 原表
WHERE $c_1 = @c_1$ OR $c_1 = *$
ORDER BY c_1 DESC

对于两个以上的匹配条件，设条件 c_2 的值域为$\{c_{21}, c_{22}, \ldots, c_{2l_2}, *\}$，$c_1 = @c_1$(假定为 c_{11})，$c_2 = @c_2$(假定为 c_{21})，则匹配的规则至多有四条：

表 6

		规则 1	规则 2	规则 3	规则 4
条件	c_1	c_{11}	*	c_{11}	*
	c_2	c_{21}	c_{21}	*	*
动作集		a_{11}	a_{12}	a_{13}	a_{14}
		a_{21}	a_{22}	a_{23}	a_{24}
		⋮	⋮	⋮	⋮

列明优先在程序中的实现也有两个方案。
方案 1：



方案 1 对应的选择语句有 6 个。
方案 2：
实现列明优先的一个选择语句如下：

SELECT * INTO 临时表 FROM 原表
WHERE $c_1 = @c_1$ OR $c_1 = *$
AND $c_2 = @c_2$ OR $c_2 = *$
ORDER BY c_1 DESC,
c_2 DESC

然后取临时表中第一条记录的费率即可。两个方案比较不难看出，方案 1 每一字段的匹配都需要一个 SELECT 语句，而方案 2 总共只需要一个 SELECT 语句。当允许统配符出现的字段从一个增加到两个时，方案 1 的复杂度增加了许多，它不止表现在 SELECT 语句数的增加，而且还表现在程序的逻辑结构要复杂得多。相比之下，方案 2 在逻辑上的复杂度几乎没有增加，这一点当允许匹配的字段增加为多个时更为突出，我们在查找特定条件下的海运费的费率值时用了以下的 SQL 语句：

SELECT * INTO OFTTMP FROM OFTRATE
WHERE size = @ size
AND type = @ type
AND cargotype = @ cargotype or cargotype = '*'
AND pod = @ pod
AND term = @ term
AND customer = @ customer or customer = '*'
ORDER BY customer DESC,
cargotype DESC

方案 2 不仅程序结构清晰，查询速度也相当理想，若允许匹配的字段不止一个(设为两个)时，

规则 1　c_{11}　c_{21}　有两个列明匹配
规则 2　*　c_{21}
规则 3　c_{11}　*
规则 4　*　*

当规则 1 存在时，毫无疑问，应优先选它；规则 1 不存在时，规则 2 和规则 3 都只有一个列明匹配；若它们同时存在，应该如何选择呢？这时，应考虑在实际问题中条件 1 和条件 2 究竟哪个更重要，将重要的条件紧随 ORDER BY 子句后首先列出。例如，在客户和货类的条件中，客户更为重要，则可写成：

ORDER BY customer DESC,
cargotype DESC

五、结束语

在描述规则时，引入统配符'*'，可以增强表现能力，使规则数得到可观的减少，这种减少会给规则引用时的匹配算法带来复杂性，用(三)中介绍的第二种做法，可使匹配算法结构清晰，易于扩充，并且有较快的运算速度。在信息管理和专家系统中，规则被大量地描述、引用，希望本文的介绍能对设计者有所补益。

(来稿时间：1998 年 6 月)