(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 109977276 B
(45)授权公告日 2020.12.22

(21)申请号 201910221407.X
(22)申请日 2019.03.22
(65)同一申请的已公布的文献号
申请公布号 CN 109977276 A
(43)申请公布日 2019.07.05
(73)专利权人 华南理工大学
地址 510640 广东省广州市天河区五山路381号
(72)发明人 陆以勤 胡凡 覃健诚
(74)专利代理机构 广州市华学知识产权代理有限公司 44245
代理人 李斌
(51)Int.Cl.
G06F 16/903(2019.01)
G06F 16/33(2019.01)

审查员 陈飞

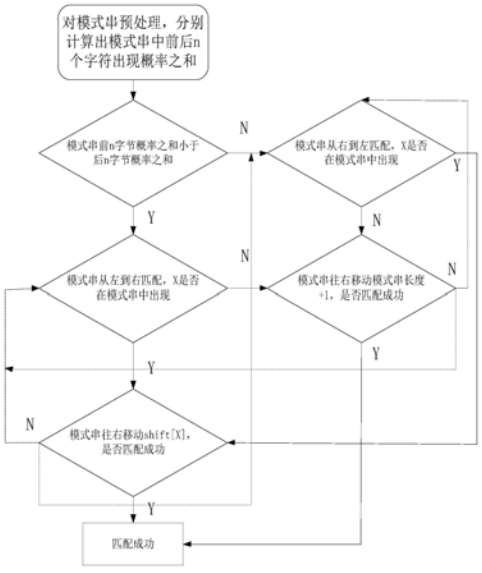
权利要求书1页 说明书5页 附图2页

(54)发明名称

一种基于Sunday算法改进的单模式匹配方法

(57)摘要

本发明公开了一种基于Sunday算法改进的单模式匹配方法,该单模式匹配方法通过判断文本字符串中参加匹配的最末位字符的下一位字符是否出现在模式串中,事先预处理模式串,根据模式串的特征来对模式串和文本字符串进行不同顺序匹配,若匹配成功,字符串匹配结束,若匹配不成功,则滑动模式串,继续利用上述的方法进行判断,直到模式串滑动到达文本字符串的末端或者匹配成功,字符串匹配才会结束。利用此单模式匹配方法,可有效的减小原算法的总匹配次数,提升文本的匹配效率。



CN 109977276 B

1.一种基于Sunday算法改进的单模式匹配方法,其特征在于,所述单模式匹配方法包括以下步骤:

S1、利用模式串的特征信息分别计算出模式串头部和尾部的字符概率,其中,所述模式串为待匹配的字符串;

S2、分别求出模式串中头部n个字符和尾部n个字符的概率之和,其中,n取值为1、2或者3,如果头部n个字符概率之和小于等于尾部n个字符概率之和,则跳转到步骤S3;反之,如果头部n个字符概率之和大于尾部n个字符概率之和,则跳转到步骤S4;

S3、将模式串与文本字符串左对齐,所述文本字符串为待搜索文本,将所述模式串在所述文本字符串上滑动,如果模式串滑动已超出文本字符串,则匹配失败;如未超出,则从左到右判断模式串与文本字符串对应位置上的字符是否匹配,若匹配,则匹配成功并结束字符串匹配;若不匹配,则跳转到步骤S5;

S4、将模式串与文本字符串右对齐,所述模式串在所述文本字符串上滑动,如果模式串滑动已超出文本字符串,则匹配失败;如未超出,则从右往左判断模式串与文本字符串对应位置上的字符是否匹配,若匹配,则匹配成功并结束字符串匹配;若不匹配,则跳转到步骤S5;

S5、如文本字符串中参加匹配的最末位字符的下一位字符没出现在模式串中,则模式串从左到右滑动距离步长=模式串长度+1;反之,如文本字符串中参加匹配的最末位字符的下一位字符出现在模式串中,则模式串从左到右滑动距离步长=匹配串中最右端的该字符到末尾的距离+1;同时,如是从步骤S3跳转到步骤S5的,则重复步骤S3;如是从步骤S4跳转到步骤S5的,则重复步骤S4。

2.根据权利要求1所述的一种基于Sunday算法改进的单模式匹配方法,其特征在于,所述步骤S1中根据模式匹配应用场景分别计算模式串头部和尾部的字符概率,当应用场景为查找一篇英文文章里面某字符串,则各字符根据已有文献给出的各字符在英文文章中出现概率的统计结果;当应用场景为查找随机混乱文本字符串中某模式串,则自定义用户采样比取出该随机混乱文本字符串,分别计算出各字符出现概率。

3.根据权利要求1所述的一种基于Sunday算法改进的单模式匹配方法,其特征在于,n的取值规则如下:

如模式串长度m>1且m<=3,则n=1;如模式串长度m>3且m<=5,则n=2;如模式串长度m>=6,则n=3。

一种基于Sunday算法改进的单模式匹配方法

技术领域

[0001] 本发明涉及单模式字符匹配技术领域，具体涉及一种基于Sunday算法改进的单模式匹配方法。

背景技术

[0002] 字符串模式匹配在很多方面都有越来越多的应用，更快的匹配速度一直是研究人员追求的目标。怎么才能使匹配算法的执行速度提高，也受到更多人的关注。在单模式匹配算法中，BM匹配算法和KMP匹配算法是最为著名的两种。在最不理想情况下，这两种算法均是线性的时间复杂度，但是在实际运用中，BM算法往往比KMP算法快上3～5倍。Daniel M.Sunday在20世纪90年代提出了比BM算法更快速、更容易理解的Sunday算法，使字符串的匹配效率有所提高。随着互联网的日渐庞大，信息也是越来越多，如何在海量的信息中快速查找自己所要的信息是网络搜索研究的热点所在。其中，字符串匹配算法起着非常重要的作用，一个高效的字符串匹配算法，可以极大的提高搜索的效率和质量。字符串匹配在网络领域有着广泛的应用。比如，拼写检查、语言翻译、数据压缩、搜索引擎、网络入侵检测等。Sunday算法核心思想是在匹配过程中，模式串并不被要求一定要按从左向右进行比较还是从右向左进行比较，这样如果匹配顺序方向选择不合适，将会增加很多无效匹配，Sunday算法效率将会显著降低。

发明内容

[0003] 本发明的目的是为了解决现有技术中原Sunday算法模式串总匹配无顺序性的技术问题，提供一种基于Sunday算法改进的单模式匹配方法。

[0004] 本发明的目的可以通过采取如下技术方案达到：

[0005] 一种基于Sunday算法改进的单模式匹配方法，该单模式匹配方法包括以下步骤：

[0006] S1、利用模式串的特征信息分别计算出模式串头部和尾部的字符概率，其中，所述模式串为待匹配的字符串。

[0007] S2、分别求出模式串中头部n个字符和尾部n个字符的概率之和，其中，n取值为1、2或者3，如果头部n个字符概率之和小于等于尾部n个字符概率之和，则跳转到步骤S3；反之，如果头部n个字符概率之和大于尾部n个字符概率之和，则跳转到步骤S4；

[0008] S3、将模式串与文本字符串左对齐，所述文本字符串为待搜索文本，将所述模式串在所述文本字符串上滑动，如果模式串滑动已超出文本字符串，则匹配失败；如未超出，则从左到右判断模式串与文本字符串对应位置上的字符是否匹配，若匹配，则匹配成功并结束字符串匹配；若不匹配，则跳转到步骤S5；

[0009] S4、将模式串与文本字符串右对齐，所述模式串在所述文本字符串上滑动，如果模式串滑动已超出文本字符串，则匹配失败；如未超出，则从右往左判断模式串与文本字符串对应位置上的字符是否匹配，若匹配，则匹配成功并结束字符串匹配；若不匹配，则跳转到步骤S5；

[0010] S5、如文本字符串中参加匹配的最末位字符的下一位字符没出现在模式串中，则模式串从左到右滑动距离步长＝模式串长度+1；反之，如文本字符串中参加匹配的最末位字符的下一位字符出现在模式串中，则模式串从左到右滑动距离步长＝匹配串中最右端的该字符到末尾的距离+1；同时，如是从步骤S3跳转到步骤S5的，则重复步骤S3；如是从步骤S4跳转到步骤S5的，则重复步骤S4。

[0011] 进一步地，步骤S1中利用模式串的特征信息分别计算模式串头部和尾部的字符概率，根据模式匹配具体应用场景，如查找一篇英文文章里面某字符串，则各字符根据已有文献给出的各字符在英文文章中出现概率的统计结果；如查找随机混乱文本字符串中某模式串，则自定义用户采样比取出该随机混乱文本字符串，分别计算出各字符出现概率。

[0012] 进一步地，n的取值规则如下：

[0013] 如模式串长度m>1且m<＝3，则n＝1；如模式串长度m>3且m<＝5，则n＝2；如模式串长度m>6，则n＝3。

[0014] 本发明相对于现有技术具有如下的优点及效果：

[0015] 1、对模式串中各字符针对不同业务场景计算概率，采用概率低的字符优先匹配原则，从而减少总匹配次数，提高了原Sunday算法效率。

[0016] 2、模式串一旦确定，通过预处理得出各字符出现概率再确定匹配方向(从左到右还是从右到左)，从而提高在实际应用场景中的匹配效率比如网络入侵中恶意代码的检测、论文检索中大段文本的搜索，病毒多特征扫描，由于这些应用的文本字符串一般都非常多，所以在本发明的应用上能有较大效率的提升。

附图说明

[0017] 图1是本发明中公开的基于Sunday算法改进的单模式匹配方法的流程步骤图；

[0018] 图2是本发明中Sunday算法流程图。

具体实施方式

[0019] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0020] 实施例

[0021] 本实施例公开了一种基于Sunday算法改进的单模式匹配方法，下面结合附图1对本实施例中单模式匹配方法作详细说明。

[0022] 一种基于Sunday算法改进的单模式匹配方法，包括以下步骤：

[0023] 步骤S1、利用模式串的特征信息分别计算出模式串头部和尾部的字符概率，所述模式串为待匹配的字符串。根据模式匹配具体应用场景，如查找一篇英文文章里面某字符串，则各字符根据已有文献给出的各字符在英文文章中出现概率的统计结果；如查找随机混乱文本字符串中某模式串，则自定义用户采样比取出该随机混乱文本字符串，分别计算出各字符出现概率。如各字符在英文文章中出现概率的统计结果：

[0024] 空格(Space)0.2；E 0.105；T 0.072；O 0.0654；A 0.063；N 0.059；I 0.055；R

0.054;S 0.052;H 0.047;D 0.035;L 0.029;C 0.023;F/U 0.0225;M 0.021;P 0.0175;Y 0.0120;W 0.012;G 0.011;B 0.0105;V 0.008;K 0.003;X 0.002;J 0.001;Q 0.001;Z 0.001;

[0025] 步骤S2、分别求出模式串中头部n个字符和尾部n个字符的概率之和,如果头部n个字符概率之和小于等于尾部n个字符概率之和,则跳转到步骤S3;反之,如果头部n个字符概率之和大于尾部n个字符概率之和,则跳转到步骤S4;如一条模式串中字符为:search,前三字节相加为:0.052+0.105+0.063＝0.22;后三字节相加为:0.054+0.023+0.047＝0.124;0.124小于0.22,所以跳转到步骤S3。

[0026] 步骤S3、模式串与文本字符串左对齐,所述文本字符串为待搜索文本,所述模式串在所述文本字符串上滑动,如果模式串滑动已超出文本字符串,则匹配失败;如未超出,则从左到右判断模式串与文本字符串对应位置上的字符是否匹配,若匹配,则匹配成功程序结束;若不匹配,则跳转到步骤S5;

[0027] 步骤S4、模式串与文本字符串左对齐,所述文本字符串为待搜索文本,所述模式串在所述文本字符串上滑动,如果模式串滑动已超出文本字符串,则匹配失败;如未超出,则从右往左判断模式串与文本字符串对应位置上的字符是否匹配,若匹配,则匹配成功程序结束;若不匹配,则跳转到步骤S5;

[0028] 步骤S5、如文本字符串中参加匹配的最末位字符的下一位字符没出现在模式串中,则模式串从左到右滑动距离步长＝模式串长度+1;反之,如文本字符串中参加匹配的最末位字符的下一位字符出现在模式串中,则模式串从左到右滑动距离步长＝匹配串中最右端的该字符到末尾的距离+1;如是从步骤S3跳转到步骤S5的,就重复步骤S3;如是从步骤S4跳转到步骤S5的,就重复步骤S4。

[0029] 附图1中虚线表示模式串往右滑动后匹配失败时,返回上一步骤。即如是从步骤S3跳转到步骤S5的,就重复步骤S3;如是从步骤S4跳转到步骤S5的,就重复步骤S4。

[0030] 下面,结合具体实施例来对本发明做进一步详细说明。要在英文文章文本字符串中查找模式串,例如:

[0031] 文本字符串为:s u b s t r i n g s e a r c h i n g

[0032] 模式串P为:s e a r c h

[0033] 设模式串为P,模式串长度为m,文本字符串中参加匹配的最末位字符的下一位字符为X,则移动位数公式如下:

[0034] $$shift[X] = \begin{cases} m - \max\{i < m | P[i] = X\} & if\ X\ is\ in\ P[0...m-1] \\ m + 1 & otherwise \end{cases}$$

[0035] 事先计算出移动位数表:

[0036] 这例子里模式串P＝“search”

[0037] 模式串长度m＝6

[0038] shift[s]＝6-max(s的位置)＝6-0＝6

[0039] shift[e]＝6-max(e的位置)＝6-1＝5

[0040] shift[a]＝6-max(a的位置)＝6-2＝4

[0041] shift[r]＝6-max(r的位置)＝6-3＝3

[0042] shift[c]＝6-max(c的位置)＝6-4＝2

[0043] shift[h]＝6-max(h的位置)＝6-5＝1

[0044] shift[其他]＝m+1＝6+1＝7

[0045] 步骤S1根据已有文献统计出了模式串中各字符出现概率,跳转到步骤S2。

[0046] 步骤S2中模式串长度为6,所以n＝3,即求模式串前3字节和后3字节概率之和;得出前3字节概率之和小于后3字节概率之和,所以跳转到步骤S3。

[0047] 第一次匹配:

主串: substring searching

[0048]

⇩

模式串: search

匹配次数: 1 次

[0049] 第二次匹配:

主串: substring searching

[0050]

⇩

模式串: search

匹配次数: 1 次

[0051] 第三次匹配成功:

[0052] 主串:s u b s t r i n g s e a r c h i n g

[0053] 模式串:s e a r c h

[0054] 匹配次数:6次

[0055] 总匹配次数为8次。

[0056] 附图2是原Sunday算法流程图。模式串并不被要求一定要按从左向右进行比较还是从右向左进行比较,这样如果匹配顺序方向选择不合适,将会增加很多无效匹配。如采用了从左往右匹配:

[0057] 第一次匹配:

主串: substring searching

[0058]

⇩

模式串: search

匹配次数: 2 次

[0059] 第二次匹配:

主串： substring searching

⇩

[0060]

模式串： search

匹配次数：1次

[0061] 第三次匹配成功：

[0062] 主串:substringsearching

[0063] 模式串:search

[0064] 匹配次数:6次

[0065] 总匹配次数为9次。

[0066] 可见,本发明改进的Sunday算法总匹配次数减少了1次,当具体业务场景数据量大的时候,将明显改善匹配效率。

[0067] 上述实施例为本发明较佳的实施方式,但本发明的实施方式并不受上述实施例的限制,其他的任何未背离本发明的精神实质与原理下所作的改变、修饰、替代、组合、简化,均应为等效的置换方式,都包含在本发明的保护范围之内。

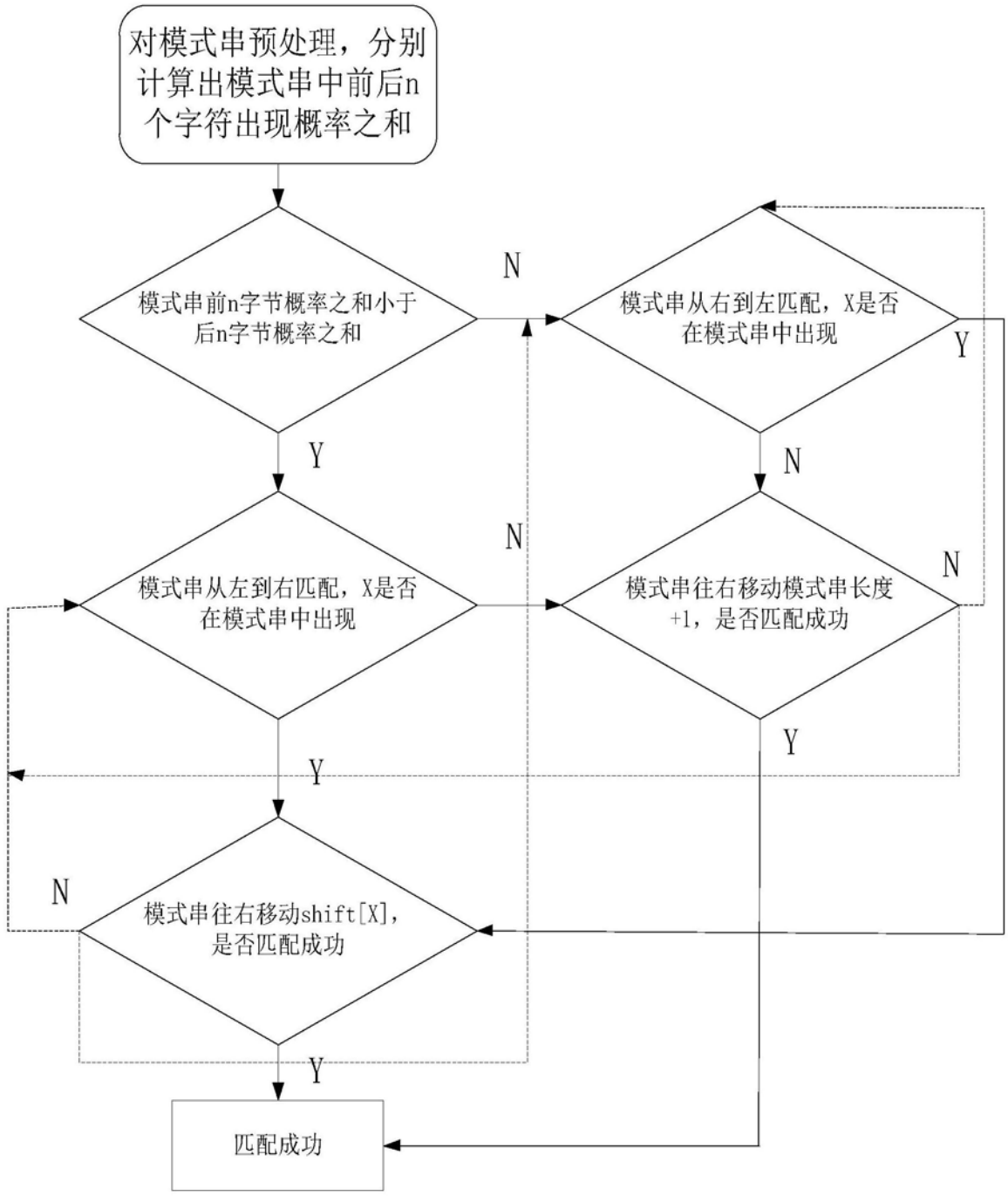

对模式串预处理，分别计算出模式串中前后n个字符出现概率之和
模式串前n字节概率之和小于后n字节概率之和
N
模式串从右到左匹配，X是否在模式串中出现
Y
Y
N
模式串从左到右匹配，X是否在模式串中出现
N
模式串往右移动模式串长度+1，是否匹配成功
N
Y
Y
N
模式串往右移动shift[X]，是否匹配成功
Y
匹配成功

图1

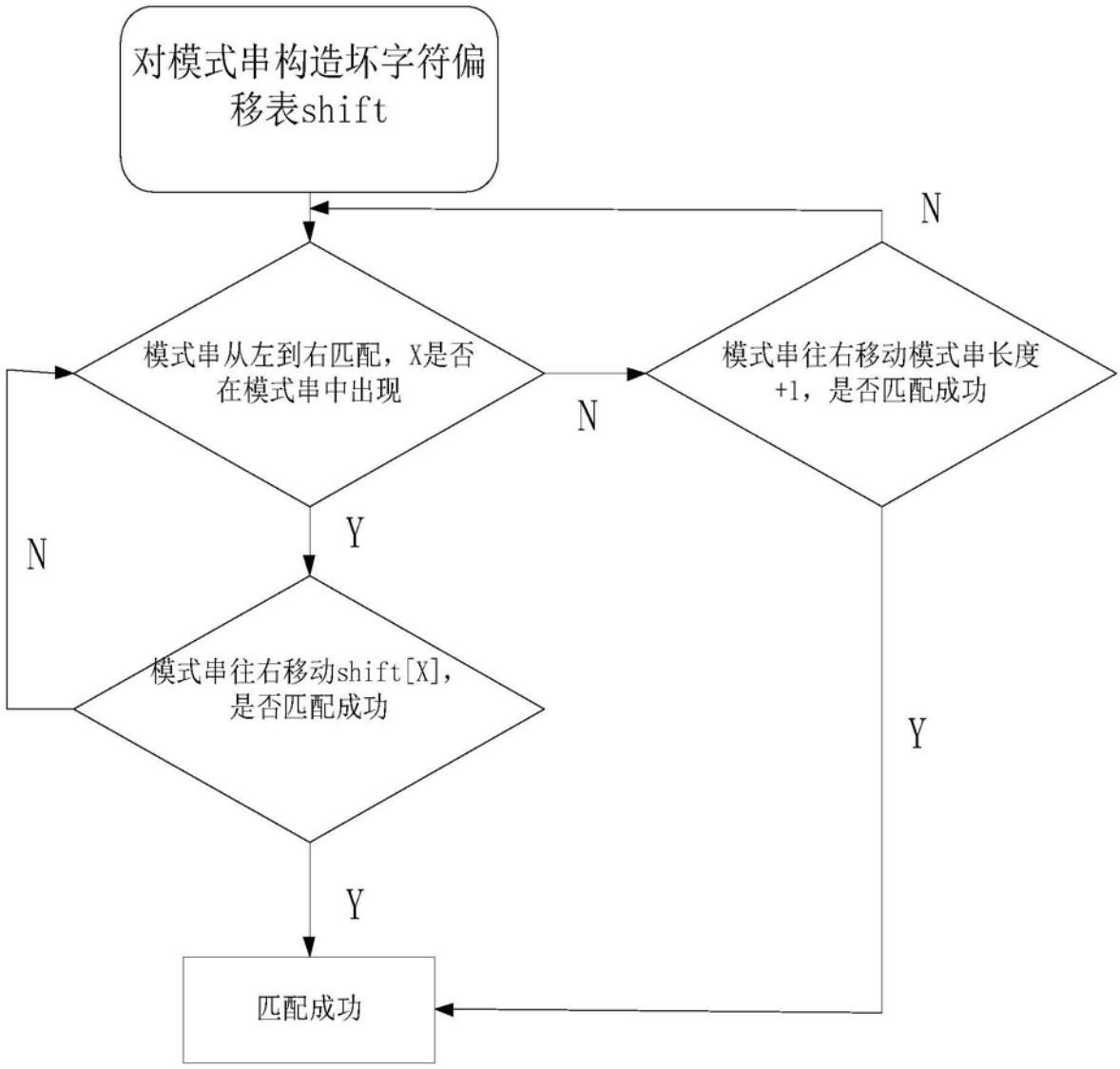

对模式串构造坏字符偏移表shift
模式串从左到右匹配，X是否在模式串中出现
N
模式串往右移动模式串长度+1，是否匹配成功
N
Y
N
模式串往右移动shift[X]，是否匹配成功
Y
Y
匹配成功

图2