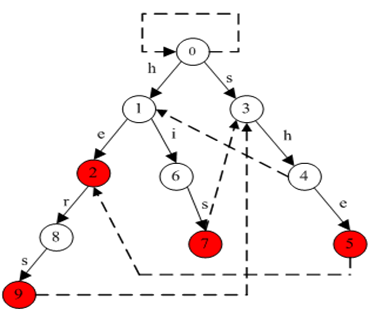
Aho-Corasick算法又叫AC自动机算法，是一种多模式匹配算法。Aho-Corasick算法可以在目标串查找多个模式串，出现次数以及出现的位置。

Aho-Corasick算法原理

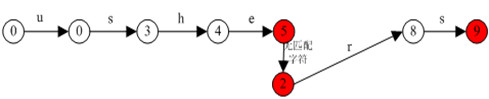
Aho-Corasick算法主要是应用有限自动机的状态转移来模拟字符的比较，下面对有限状态机做几点说明：



上图是由多模式串{he,she,his,hers}构成的一个有限状态机：

1.该状态当字符匹配是按实线标注的状态进行转换，当所有实线路径都不满足（即下一个字符都不匹配时）按虚线状态进行转换。

2.对ushers匹配过程如下图所示：



当转移到红色结点时表示已经匹配并且获得模式串

Aho-Corasick算法步骤

Aho-Corasick算法和前面的算法一样都要对模式串进行预处理，预处理主要包括字典树Tire的构造，构建状态转移表（goto），失效函数（failure function），输出表（Output）。

Aho-Corasick算法包括以下3个步骤

1.构建字典树Tire

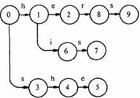
2.构建状态转移表，失效函数（failure function），输出表（Output）

3.搜索路径（进行匹配）

下面3个步骤分别进行介绍

构建字典树Tire

Tire是哈希树的变种，Tire树的边是模式串的字符，结点就是Tire的状态表，下图是多模式串{he,she,his,hers}的Tire树结构：



构建goto函数、failure function和Output函数

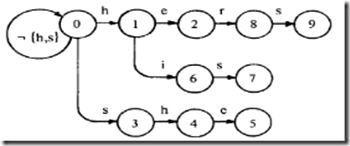
       goto函数（状态转移函数）：goto(pre,v)=next,完成这样的任务：在当前状态pre，输入字符v，得到下一个状态next，如果没有下个状态则next=failure。

       failure function：失效函数是处理当前状态是failure时的处理。

      output函数：当完成匹配是根据状态输出匹配的模式串。

下面是多模式串{he,she,his,hers}的goto函数，failure函数，output函数

goto函数：

  
failure函数

http://dl.iteye.com/upload/attachment/0075/2650/3d3450ca-7947-343e-a84f-7f39f0f4d2b9.png

output函数

