AC Automata

AC自动机

回顾

- Trie的结构
- KMP中的next数组 —— fail指针

AC Automata

- 数据结构

```
int chd[MAX_NODE][MAX_CHILD];
int fail[MAX_NODE];//类比next数组
int val[MAX_NODE];
int ID[128];
int sz;
```

AC Automata

- Next数组的意义

当P[i+1]不匹配(P[0]..P[i]已经匹配成功)时,

那么P[0]..P[next[i]]一定是匹配成功的,

那么只需要沿着next一直找到P[next[i]+1]匹配的时候,

继续进行匹配，否则i=next[i](沿着next继续找)

AC Automata

- Kmp中的匹配

```
int kmp(char *A, char *B, int *next) {
    int j = -1, ret = 0;
    for (int i = 0; A[i]; i++) {
        while (j != -1 && A[i] != B[j + 1]) j = next[j];
        if (A[i] == B[j + 1]) j++;
        if (!B[j + 1]) {
            ret++;
            j = next[j];
        }
    }
    return ret;
}
```

AC Automata

- 用fail匹配

```
int Query(const char *s) {
    int ret = 0;
    int cur = 1, tmp;//Trie的根节点为1号节点
    for (int i = 0; s[i]; i++) {
        if (chd[cur][ID[s[i]]]) cur = chd[cur][ID[s[i]]];
        else {
            while (cur != 1 && chd[cur][ID[s[i]]] == 0)
                cur = fail[cur];
            if (chd[cur][ID[s[i]]]) cur = chd[cur][ID[s[i]]];
        }
```

AC Automata

- 用fail匹配

```
        tmp = cur;
        while (tmp != 1 && val[tmp] != -1) {
            ret += val[tmp];
            val[tmp] = -1;//单次匹配
            tmp = fail[tmp];
        }
    }
    return ret;
}
```

AC Automata

- KMP中的next数组 —— fail指针

```
void prekmp(char *B, int *next) {
    next[0] = -1;
    int j = -1;
    for (int i = 1; B[i]; i++) {
        while (j != -1 && B[i] != B[j + 1])
            j = next[j];
        if (B[i] == B[j + 1]) j++;
        next[i] = j;
    }
}
```

AC Automata

- fail指针

```
void Build_AC() {
    while (!q.empty()) q.pop();
    q.push(1);
    fail[1] = 1;
    while (!q.empty()) {
        int cur = q.front();
        q.pop();
        for (int i = 0; i < MAX_CHILD; i++)
            if (chd[cur][i]) {
```

AC Automata

- fail指针

```
                if (cur == 1) fail[chd[cur][i]] = 1;
                else {
                    int tmp = fail[cur];
                    while (tmp != 1 && chd[tmp][i] == 0) tmp = fail[tmp];
                    if (chd[tmp][i]) fail[chd[cur][i]] = chd[tmp][i];
                    else fail[chd[cur][i]] = 1;
                }
                q.push(chd[cur][i]);
            }
    }
}
```

AC Automata

- fail指针

```
                if (cur == 1) fail[chd[cur][i]] = 1;
                else {
                    int tmp = fail[cur];
                    while (tmp != 1 && chd[tmp][i] == 0) tmp = fail[tmp];
                    if (chd[tmp][i]) fail[chd[cur][i]] = chd[tmp][i];
                    else fail[chd[cur][i]] = 1;
                }
                q.push(chd[cur][i]);
            }
    }
}
```

AC Automata

- 大牛博客

http://www.notonlysuccess.com/index.php/aho-corasick-automaton/

- 论文