

Понятие за машина на състоянието

Машината на състоянието представлява метод за обработка на последователност от данни. Употребата на машината на състоянието се среща в разнообразни решения за разпознаване на текст, комуникация по специфичен протокол.

Идейно автоматът се представя като съвкупност от състояние. Привсяко едно състояние се могат да се изпълняват различни дейност, както и да се взима решение за преход в друго състояние.

Преходите в различни състояния могат да са условни и задължителни. Условните състояния се използват, когато са изпълнение дадени критерии, тогава се преминава към следваща стъпка. Задължителните са, такива преходи, които ще се извършат задължително при следващата итерация в автомата.

Описание на автомат

За описанието на автомат най-лесно първо се съставя блокова схема. В нея за разлика на блоковите схеми за алгоритми има само два вида елементи – състояния и преходи. Състоянията се изобразяват като окръжности, в които е написан номера (или името) на съответното състояние (по-лесно е в програмирането на C на автомати да се използва или номериране или изброим тип или дефинирани константни стойности). Преходите се изобразяват като стрелки, около които се записва условието за преход от едно състояние в друго.

Пример:

Да се състави блокова схема на преходите на алгоритъм за пътуване, който е описан по следния начин: Пътешествието започва от град София и се осъществява, чрез автобус. Ако броят на пътешествениците е по-голям от 10 се посещава град Плевен, ако е по-малък се посещава град Пловдив. Когато са в Плевен в зависимост от това дали чуждестранните туристи са повече следващата дестинация е Велико Търново, ако повечето са българи следва посещение на Крушунските водопади. Велико Търново и Крушунските водопади са крайни дестинации на пътуването.

Ако посетят Пловдив в зависимост от това дали е слънчево могат да посетят Карлово (при слънчево време) или Бачковския манастир, които отново са крайните дестинации на пътуването.

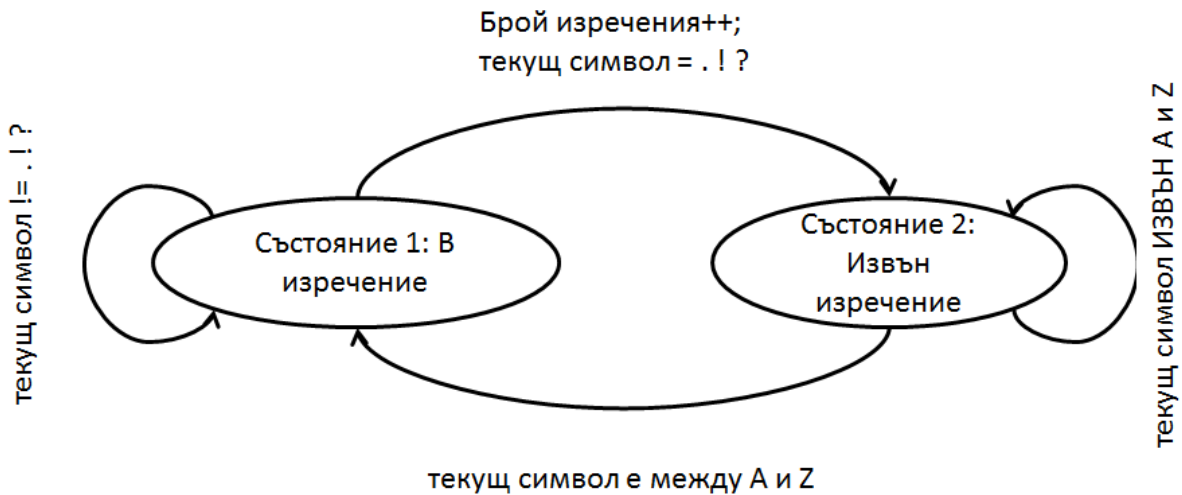
Този описателен пример се свежда до следната блок схема:



По този начин чрез прехотите (условията за пътуване) и състоянията (градовете) може да се конструират различни варианти на пътешествието.

В компютърните системи, машините на състоянията се използват за разпознаване на различни процеси и последователности, чрез които могат да бъдат взимани решения. Пример за това е задачата за преброяването на изреченията в един текстов файл. Задачата може да звучи елементарна и да не налага използването на автомати, но реализирана чрез автомати изглежда по подобен начин:

- 1) **Трябва да се определят състоянията, в които ще се намира автомата:** две състояния „Състояние 1: В изречение“ и „Състояние 2: Извън изречение“
- 2) **Трябва да се определят условията за преход:**
 - a. От „**Извън изречение**“ в състояние „**В изречение**“ се преминава при текущ символ, който е главна буква. В противен случай се остава в състояние „**Извън изречение**“;
 - b. От състояние „**В изречение**“ се преминава в състояние „**Извън изречение**“, когато текущия символ препинателен знак и броят на разпознатите изречения се увеличава с единица. В противен случай се остава в състояние „**В изречение**“;
- 3) **Избира се начално състояние:** „Извън изречение“
- 4) **Начертава се блок-схема, която онагледява преходите между различните състояния.**



5) Реализация на програмнен код.

Реализацията на програмния код не представлява сложност, ако се спазват няколко условия за реализация:

1. Програмният код, описващ машината на състояния се реализира чрез цикъл;
2. Допустимо е само обновяване на входните данни за една итерация на цикъл;
3. За всяка итерация на цикъла се взема под внимание текущото състояние на автомата;

Използвайки следните правила и съставената в т. 4 блокова схема, програмната реализация на дадената задача изглежда по този начин:

```

int end_of_sentence(char current_char)
{
    if(current_char == '.' || current_char == '!' || current_char == "?")
        return true;
    else
        return false;
}

/* ... Функция, където е реализирана машината на състояние ... */
FILE* fp;
int status = 2; //Дефиниране на управляваща променлива (съхранява текущото състояние на автомата); задава се и началното състояние на автомата
char current_char = 0; //Променлива за съхраняване на текущия символ, прочетен от файла
int result = 0; //Променлива за съхраняване на откритите изречения
/*... Програмен код ...*/
fp = fopen("file", "r"); //Отваряне на входен файл за обработка
/* ... Проверка на отворения файл ...*/
while(current_char != EOF) //Цикъл, в който ще бъде реализирана машината на състоянието
{
    current_char = fgetc(fp); //Прочитане на символ от файла
    switch(status)
    {
        case 1: //Описание на състоянието "В изречение"
            if(end_of_sentence(current_char)) //Условие за преход
            {

```

```
        result ++;
        status = 2; //Преход, който ще се осъществи на следващата
итерация
    }
    break;
    case 2: //Описание на състоянието "Извън изречение"
        if(current_char >= 'A' && current_char <= 'Z') //Условие за преход
            status = 1; //Преход, който ще се осъществи на следващата
итерация
        break;
    }
}

//В result се пази крайният резултат от изпълнението на машината на състоянието
//Затваряне на файла и край на функцията
```

Задачи за изпълнение

Като използвате машините на състоянията, напишете следните програми:

- Програма за преброяване на имената в един текстови файл.
- Програма за преброяване на числата в един текстови файл.
- Програма за откриване и преброяване на стринговете и константните символи в един сорс файл.
- Програма, която изтрива всички коментари от даден входен сорс файл, а резултатът записва в друг изходен файл.