Търсене в низове

Търсенето на низове във файл, отговарящи на някакво условие (pattern) е често срещана задача. Широко използвани методи: краен автомат и негови модификации -Knuth-Morris-Pratt (KMP) и Boyer-Moore (BM) (1977). Двата метода са съпоставими по бързодействие с линейна сложност зависеща от дължината на низа и дължината на файла в който се търси. BM е малко по-бърз но по-сложен. KMP е предмет на тази лекция.

Краен автомат (FSM) – математически изчислителен модел на абстрактна машина, която във всеки момент от времето се намира в едно от **краен брой състояния** (графично изобразявани с окръжности). Тя извършва **преход** (transition - графично изобразявано със стрелка) в друго състояние в резултат на някакво определено **входно въздействие** (маркер върху стрелката) и се дефинира със **списъка от състояния, начално състояние и условията за всеки преход**.

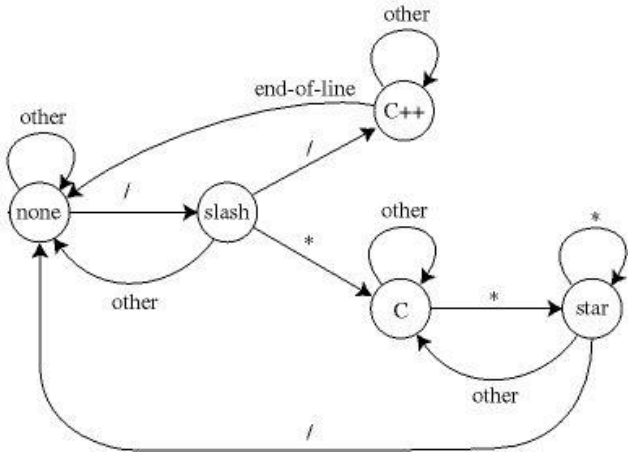
Едно приложенията ѝ е за представяне на език (множество от низове отговарящи на някакво условие). KMP използва условието за търсене за съставяне на FSM, след което тя се записва на език – например C и програмата се изпълнява. Основна трудност в метода е съставянето на FSM. Пример: преброяване на коментарите в един файл.

Условие:

1) C++: започва с // и завършва с EOL
2) C : започва с /* и завършва с */
3) Не се включват един в друг
4) EOF приключва алгоритъма

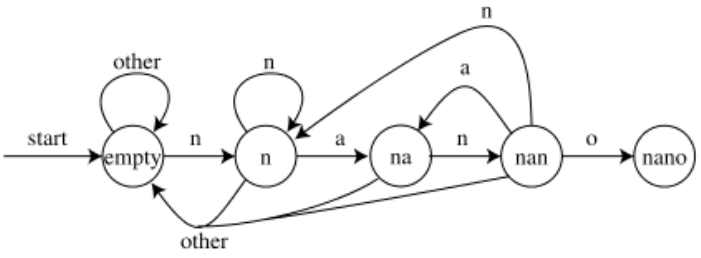
Съставяне на FSM. Ще използвам 2 форми – таблична и графична:

		/	*	EOL	EOF	other
1	none	slash	none	none	end	none
2	slash	C++	C	none	end	none
3	C++	C++	C++	none	end	C++
4	C	C	star	C	end	C
5	star	none	star	C	end	C



Втори пример: нека да се търси низ **nano:**

	empty	n	a	o	other
1	empty	n	empty	empty	empty
2	n	n	na	empty	empty
3	na	nan	empty	empty	empty
4	nan	n	na	**nano**	empty



FSM се променя, ако търсения низ трябва да е цяла дума – трябва да бъдат въведени **разделители**, които отделят думата. Те могат да бъдат от различен тип например празен интервал, препинателен знак ….
Въвеждам ново крайно състояние – nano_w и ново междинно – in_w (започната друга дума)

	empty	n	a	o	other	separator
1	empty	n	in_w	in_w	in_w	empty
2	in_w	in_w	in_w	in_w	in_w	empty
2	n	in_w	na	in_w	in_w	empty
3	na	nan	in_w	in_w	in_w	empty
4	nan	in_w	in_w	nano	in_w	empty
5	nano	in_w	in_w	in_w	in_w	nano_w

Запис на FSM от първия пример на C:

```
#include <stdio.h>
int main(){
   int state=1,c;
   int brC=0;
   int brCpp=0;
   FILE *f;
   f=fopen("automate.tst","rt");
   if(f==NULL){
      printf(“No such file\n”);
      return 2;
   }
   while((c = fgetc(f)) != EOF){
     switch(state){
       case 1: switch(c){
                 case '/': state=2;break;
               }
               break;
       case 2: switch(c){
                   case '/': state=3; brCpp++; break;
                   case '*': state=4; brC++; break;
                   default: state =1;
                 }
                 break;
       case 3: switch(c){
                   case '\n':state=1;
                 }
                 break;
       case 4: switch(c){
                   case '*': state=5;
                 }
                 break;
       case 5: switch (c){
                   case '/': state=1; break;
                   case '*': break;
                   default: state=4;
                  }
     }
   }
   fclose(f);
   printf( "found %d c  and %d c++ comments\n",
brC,brCpp);

}
```