I61- Compilation et théorie des langages

Licence 3 - 2016/2017

1 Expresions régulières: `regex.h`

L'objet de cette section et d'utiliser la bibliothèque `C regex` afin de programmer un analyseur lexical rudimentaire.

Le programme suivant est un exemple de départ de la manière dont s'utilise `regex`:

```c
/*
 Fichier exemple match.c
*/
#include <regex.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  regex_t regex;
  if (regcomp(&regex, argv[1], REG_EXTENDED)!=0)
    return 1;
  if (regexec(&regex, arg[2], 0, NULL, 0)==0)
    printf("motif trouve !\n");
  else
    printf("motif non trouve !\n");
  regfree(regex);
  return 0;
}
```

```
$ gcc -Wall match.c -o match.exe
$ ./match.exe "(a|b)*b" abbaba
motif trouve !
```

1. Modifier le programme pour que celui-ci affiche les indices de début et de fin des motifs reconnus. Il faut utiliser pour cela la structure prédéfinie `regmatch_t` (voir `man regex` pour les détails).

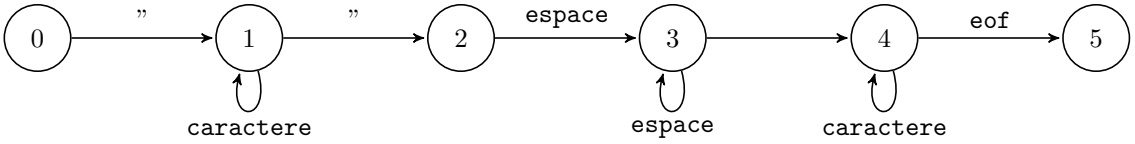
   ```c
   typedef struct {
       regoff_t rm_so;
       regoff_t rm_eo;
       } regmatch_t;
   ```

2. Écrire une commande `mygrep.exe` en `C` qui met en évidence un motif dans un fichier texte le soulignant.

   ```
   $ cat /tmp/file.txt
   lo        Link encap:Local Loopback
             inet addr:127.0.0.1  Mask:255.0.0.0
             inet6 addr: ::1/128 Scope:Host
   $ ./mygrep.exe "([0-9]{1,3}\.){3}[0-9]{1,3}" /tmp/file.txt
             inet addr:127.0.0.1  Mask:255.0.0.0
                       ^^^^^^^^^       ^^^^^^^^^
   ```

3. Écrire un programme `C` permettant d'afficher les différentes unités lexicales reconnues dans un fichier. Les noms d'unités lexicales et leur motif seront données dans un fichier à part. Pour cela, on utilisera la structure `unilex_t`. Il faut programmer dans un premier temps une fonction `creer_unliex_table` permettant de parcourir un fichier de spécification comme le fichier `spec.txt` de l'exemple et de générer un tableau d'objet de type `unilex_t`. Une manière de faire peut être de s'inspirer de la machine à états suivante:

```
typedef struct {
  regex_t regex;
  char name[32];
} unilex_t;
```




```
$ cat /tmp/file.txt
Nous sommes le 15 fevrier.

$ cat /tmp/spec.txt
"[ \.]" SEP
"[0-1]+" NBR
"[a-zA-Z]+" MOT
$ ./unilex.exe /tmp/spec.txt  /tmp/file.txt
<MOT,Nous> <SEP, > < MOT, sommes> <SEP, > <MOT, le> <SEP, >
<NBR, 15> <SEP, > <MOT, fevrier> <SEP, .>
```