

Utilisation des conditions de départ

Exemple.

- On désire interpréter l'expression régulière `[a-zA-Z]+` (un mot d'au moins une lettre) de deux manières, selon le contexte :
 - ✓ **Contexte 1.** Si mot entre guillemet, alors il représente une constante alphanumérique (chaîne de caractères)
 - ✓ **Contexte 2.** Si mot non entre guillemet, alors il est un identificateur
- On définit deux états `const_alpha` et `normal`, respectivement pour les deux contextes. Ces états sont utilisés comme suit :
 - ✓ Initialement, état égal à 0
 - ✓ Tout au début, état est mis à `normal` par l'instruction `BEGIN normal`
 - ✓ chaque fois qu'on rencontre des guillemets, on commute d'un état à l'autre (entre `normal` et `const_alpha`)
 - ✓ lorsqu'on reconnaît une expression régulière `[a-zA-Z]+`, alors :
 - ✓ si l'état courant est `normal` : alors un identificateur est reconnu
 - ✓ si l'état courant est `const_alpha` : alors une chaîne de caractères est reconnue
- Exemple : Commentez le programme suivant

```
%start normal const_alpha
%%
<normal>[a-zA-Z]+ {fprintf(yyout,`identificateur : %s\n`,yytext);}
<normal>` {BEGIN const_alpha;}
<const_alpha>[a-zA-Z]+ {fprintf(yyout,`chaîne : %s\n`,yytext);}
<const_alpha>` {BEGIN normal;}
<normal,const_alpha>. { /* aucune action */ }
<normal,const_alpha>\n { /* aucune action */ }
%%
main()
{
main()
{ errno_t Err1=fopen_s(&yyin,"H:\\ESI-COMPILATION\\ex.in","r");
  errno_t Err2=fopen_s(&yyout,"H:\\ESI-COMPILATION\\ex.out","w");
  if (Err1 == 0)
  {
      BEGIN normal;
      yylex();
  }
  else
      printf( "The file was not opened\n" );
  fclose(yyin);
  fclose(yyout);
}
```

- Question : Comment peut-on résoudre le même exercice sans utiliser les conditions de départ ?

