

EXERCICES

Analyse Lexicale

1 Prise en main Lex

Que font les deux programmes suivants :

Programme 1

```
%option nomain
%option noyywrap
%{
    int num_lignes = 0, num_cars = 0;
}%
%%
\n    ++num_lignes; ++num_cars;
.    ++num_cars;
%%
main() {
    yylex();
    printf("nombre de lignes = %d, nombre de cars = %d\n", num_lignes, num_cars);
}
```

Programme 2

```
%option main
%option noyywrap
%%
[ \t]+$ ;
[ \t]    printf(" ");
```

1. Testez les deux programmes.
2. Modifiez le programme 1 pour qu'il compte aussi les nombres et les identificateurs qui sont dans le texte source puis testez le.
3. Modifiez le programme 2 pour qu'il supprime les sauts de ligne.

2 Analyse du HTML

Le langage HTML permet d'annoter un texte de balises qui spécifient comment certains éléments du texte sont affichés par un browser.

Ainsi, un texte qui se trouve entre les balises ** et ** est affiché en gras (bold), un texte entre **<i> et </i>** en italique. De la même manière, une liste numérotée (ordered list) se trouve entre ** et **, une liste sans numérotation (unordered list) entre ** et **, les éléments de la liste étant entre ** et **. Un nouveau paragraphe est introduit par **<p>**.

1. Écrivez un fichier Lex numlist.l qui transforme toute liste non numérotée en liste numérotée. Le fichier aura la forme :

```
%option main
```

Compilation

Page 1/2

```
%option noyywrap
%%
```

Dans la ligne suivant le `%%`, on écrit les expressions régulières et les actions correspondantes.

2. Exécutez la commande `lex numlist.l -o numlist`
Lex génère un fichier `lex.yy.c` avec une fonction main standard.
Après compilation, vous obtenez un programme `numlist` qui lit depuis l'entrée standard `stdin` et écrit sur la sortie standard `stdout`. Si vous souhaitez tester votre programme sur un fichier `html ex1.html` invoquez la commande suivante :
`numlist < ex1.html > ex1out.html`
3. Insérez les mots `bold` et `italics` dans le fichier `html`, puis étendez votre programme de façon à ce qu'il mette le mot `bold` en gras et le mot `italics` en italique.
4. Comment peut-on éviter un balisage redondant (de la forme ` ... `) ?

3 Transformation du HTML en Latex

Dans le même style que l'exercice précédent, transformez un fichier HTML en LATEX. Un document LATEX est inclus entre

```
\documentclass{article}
\begin{document}
...
\end{document}
```

Le titre d'une section resp. sous-section est inclus entre `\section{...}` resp. `\subsection{...}`. Pour créer une liste numérotée, on utilise l'environnement

```
\begin{enumerate}
\item ....
\end{enumerate}
```

Pour une liste non numérotée, on utilise le mot-clé `itemize`.

Un nouveau paragraphe commence tout simplement avec une ligne vide.

1. Créez un fichier `htmltex.l` et exécutez l'analyseur comme précédemment :
`htmltex < ex2.html > ex2out.tex`
2. Testez votre programme en invoquant `latex` avec le fichier généré :
`Latex ex2out.tex`

4 Redirection des entrées/sorties

Lex gère deux variables, `yyin` et `yyout`, qui pointent sur le fichier de lecture resp. écriture de Lex (par défaut : `stdin` et `stdout`). Pour rediriger les entrées/sorties, il vous faut donc initialiser ces deux variables correctement. Pour cela, effacez `%option main` et écrivez votre propre fonction `main`.

1. Écrivez un traducteur qui ne lit pas de l'entrée standard, comme précédemment, mais d'un fichier, et écrit sur un autre fichier.

Compilation

Page 2/2