PRÉ-PROCESSADOR DE LATEX

O segundo enunciado do projeto prático é muito semelhante ao enunciado 3. A diferença reside apenas no tipo de documento, que neste caso, em vez de html, é latex. Assim sendo, utilizamos as mesmas técnicas para a resolução do problema nomeadamente start conditions (inclusivas), e a mesma estrutura para a linguagem de anotação (#nome para declarar o início da secção de texto à qual se aplica uma propriedade, e ## para sinalizar o fim da secção).

Primeiramente, ativamos a stack do Flex e criamos a main com o menu e tudo necessário ao seu funcionamento (incluindo o FILE *output).

```
int main(int argc, char* argv[]){

    if (argc==1){
            printf("ERRO. Não introduziu ficheiro de input.\n");
            return 0;
        }

    printf("\n  Escolha umas das seguintes opções:\n");
    printf("      1  -> Imprimir output no terminal\n");
    printf("      2  -> Criar ficheiro com output\n");
    printf("      3  -> Sair\n");

    c = getchar(); printf("\n");

    yyin = fopen(argv[1], "r");

    if (c == '1')
        output = stdout;

    if (c == '2')
        output = fopen("output.tex","w");

    if (c == '3')
        return 0;

    yylex();

    fclose(output);

    return 0;
}
```

De seguida, ponderamos quais as start conditions seriam necessárias já que, ao contrário de html, em latex as marcas de fecho são quase sempre chavetas o que simplifica o processo.

Com isto em mente, criamos a start condition CLOSE que será utilizada em todos os casos relativos à formatação do texto (negrito, sublinhado, itálico e os vários níveis de título).

```
%%
    /* fechar chavetas */
<CLOSE>"##" {fprintf(output,"}"); yy_pop_state();}
```

start condition close

```
    /* negrito */
"#b"    {fprintf(output,"\\textbf{"); yy_push_state(CLOSE);}

    /* itálico */
"#i"    {fprintf(output,"\\textit{"); yy_push_state(CLOSE);}

    /* sublinhado */
"#u"    {fprintf(output,"\\underline{"); yy_push_state(CLOSE); }

    /* título nível 0 */
"#h0"   {fprintf(output,"\\chapter{"); yy_push_state(CLOSE); }

    /* título nível 1 */
"#h1"   {fprintf(output,"\\section{"); yy_push_state(CLOSE); }

    /* título nível 2 */
"#h2"   {fprintf(output,"\\subsection{"); yy_push_state(CLOSE); }

    /* título nível 3 */
"#h3"   {fprintf(output,"\\subsubsection{"); yy_push_state(CLOSE); }

    /* título nível 4 */
"#h4"   {fprintf(output,"\\paragraph{"); yy_push_state(CLOSE); }

    /* título nível 5 */
"#h5"   {fprintf(output,"\\subparagraph{"); yy_push_state(CLOSE); }
```

exemplos da sua utilização

Por outro lado, no que toca às listas, tivemos que utilizar uma abordagem diferente. A nossa linguagem de anotação cobre 3 tipos diferentes: unordered (listas não numeradas), ordered (listas numeradas) e description lists (dicionários).

Em cada um destes casos, a lista é considerada uma secção. Cada secção abre e fecha com comandos específicos, por isso definimos uma start condition por cada tipo de lista.

```
    /* unordered (unnumbered) list */
"#ul"   {fprintf(output,"\\begin{itemize}"); yy_push_state(UNORDLIST);}
<UNORDLIST>"##" {fprintf(output,"\\end{itemize}"); yy_pop_state();}

    /* ordered list */
"#ol"   {fprintf(output,"\\begin{enumerate}"); yy_push_state(ORDLIST);}
<ORDLIST>"##"   {fprintf(output,"\\end{enumerate}"); yy_pop_state();}

    /* descripion list */
"#dl"   {fprintf(output,"\\begin{description}"); yy_push_state(DESCLIST);}
<DESCLIST>"##"  {fprintf(output,"\\end{description}"); yy_pop_state();}
```

start conditions das listas

O segundo passo referente às listas, foi a criação das regras que definem cada ítem. As (un)ordered em latex usam \item … para sinalizar cada elemento, portanto utilizamos a mesma notação e ação para ambos os casos.

```
    /* list item */
"#li"   {fprintf(output,"\\item");}
```

As description lists, ou dicionários, possuem um comando diferent uma vez que é preciso identificar o termo a ser descrito. Deste modo, foi necessária a criação de uma start condition para este caso particular. Isto verifica-se uma

vez que o termo se encontra rodeado por parêntesis retos e não por chavetas, sendo impossível utilizar a start condition CLOSE previamente criada.

```
    /* descripion list item */
"#dli"  {fprintf(output,"\\item["); yy_push_state(DLITEM);}
<DLITEM>"##"    {fprintf(output,"]"); yy_pop_state();}
```