Trabajo Práctico
Teoría de Lenguaje, Autómatas y Compiladores

Pablo Muiña, Augusto Nizzo, Marcos Pianelli

24 de noviembre de 2009

1. Índice

2. Introducción

3. Consideraciones realizadas

Se tuvo en cuenta que los tokens que el programa aceptaría solo serían las referenciadas en el *cheetsheet* provisto en el material de referencia.

4. Wikitext : Tokens aceptados, sus respectivas expresiones regulares y autómatas

Los tokens a detectar según el *cheetsheet* son los siguientes :

1. `[[`
2. `|`
3. `]]`
4. `[`
5. `]`

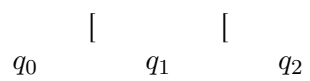
6. `http://`
7. `''`
8. `'''`
9. `'''''`
10. `#REDIRECT`
11. `==`
12. `===`
13. `====`
14. `=====`
15. `======`
16. `REFLIST`
17. `<ref name="`
18. `">`
19. `</ref>`
20. `" />`
21. `,`
22. `BULLET_LIST_ITEM`
23. `NUMBERED_LIST_ITEM`
24. `INDENTING_ITEM`
25. `SIGNATURE`
26. `[[File:`
27. `thumb`
28. `alt=`

Las expresiones regulares y los autómatas correspondientes a cada token son los siguientes:

16. Expresión regular = (<references/>)|({{Reflist}})

22. Expresión regular = `\*+`
23. Expresión regular = `#+`
24. Expresión regular = `:+`
25. Expresión regular = `~{3,5}`

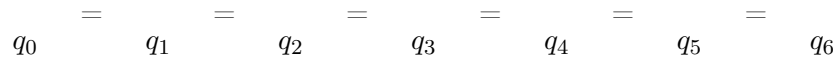
1.



2.

16.

17.



5. Funcionamiento

El analizador léxico funciona con estados, en los cuales, mantiene un control de el estado en el que está, por ejemplo, cuando se matchea la cadena de entrada con la expresión regular

`^:`

Una de las instrucciones es entrar en el estado INDENT, luego, si estando en ese estado, se matchea la expresión regular

```
:{1, 20}
```

se tendrá como salida el código asociado a los distintos nieves de indentación con los elementos de html cómo $< dl >$, $< dd >$, $< /dd >$, $< /dl >$. Esto nos trae cómo problema, que no sabemos si la construcción del elementos en *Wikicode* es correcta, y por lo tanto puede llevar a resultados inesperados, inclusive dificil de detectar los errores para el usuario. Ya que los tags de links, imágines, etc. puede que no se cierren bien. Si el string de entrada está bien formado, cada vez que se matchea un elemento html se vuelve al estado inicial.

6. Modo de uso de los programas

Para poder utilizar el programa se tiene que ejecutar el MakeFile, posiciónandose en la carpeta *tla2009* y ejecutando el comando *make* en el shell de preferencia.

```
Make
```

Para ejecutar el programa de conversión de WikiText a HTML se tipea :

```
./pwiki inputFile outputFile
```

Donde inputFile es el archivo de entrada el cual alberga el texto de formato WikiText y outputFile es el archivo de salida con formato HTML.

Para ejecutar el programa que cuenta la cantidad de lexemas aparecios se tipea:

```
./plexemes inputFile
```

Donde inputFile es el archivo de entrada el cual alberga el texto de formato WikiTexts.

7. Comentarios

Debido a un error de cómo encaramos el trabajo, buscamos una manera de que el analisis de los token sea reentrante. Es decir, al encontrar un token que matchea .^{ex}ternamenteçon una expresión regular, se poda el comienzo y el final según corresponda con los caracteres que no hacen falta y luego, el token restante se analiza nuevamente. Por ejemplo, soponer la expresión regular

```
'{2}([^']|[^']'|[^']'{3})+'{2}
```

Que es la que corresponde a matchear texto en itálica, simpre que no haya exactamente dos comillas contenidas en pseudocódigo, las instrucciones que realiza son:

IMPRIMIR()
TOKEN[LONGITUD-2] ⟵ NULL CHARACTER
TOKEN ⟵ TOKEN + 2
ANALIZAR(TOKEN)
IMPRIMIR()

La referencia de este modo de uso se puede ver en

`http://flex.sourceforge.net/manual/Reentrant-Uses.html#Reentrant-Uses`

No pudimos integrarlo a todos elementos del *Wikicode*, por ejemplo, las listas no son reentrantes, y por lo tanto, si las mismas contienen links, texto en negrita, itálica etc, no van a ser parseadas. Luego de entender nuestro error, aprendimos que debíamos usar la instrucción de *lex* "/", para matchear una expresión, pero no consumirla, es decir, que nuestra expresión regular debería haber sido:

```
'{2}/([^']|[^']'|[^']'{3})+'{2}([^']|$)
```

y en las instrucciones, llevar un flag de control, si las itálicas estaban cerradas abrirlas y si estaban abiertas, rechazar para que matchee con la próxima regla

```
'{2}/([^']|$)
```

y cerrar las itálicas y cambiar el estado de itálicas a cerrado.

Este cambio de cómo encarar las cosas, cambiaría significativamente los autómatas, ya que, en lugar de matchear un texto, y entrar a un estado especial y luego a partir de ese estado tomar deciciones de que elementos dar como salida, sería matchear toda la cadena que forma un elemento html. El único que realiza este, son los elementos en itálicas, negritas y negritas e itálicas juntas.