

# PRÁCTICAS DE PROCESADORES DEL LENGUAJE

CURSO 2008/2009

---

## TRABAJO DE EVALUACIÓN ORDINARIA

### HERRAMIENTAS PARA ANÁLISIS LR(1)

---

El objetivo de esta práctica es que diseñe y desarrolle en el lenguaje de programación C un conjunto de funciones que sirvan como herramientas LR(1). Las herramientas que debe implementar son las siguientes (esta descripción contiene un conjunto mínimo de características que el alumno puede modificar de manera justificada):

#### 1) Cierre de una configuración LR(1)

##### Entradas

- Gramática independiente del contexto (contenida en un fichero de texto plano)
- Configuración de partida.

##### Salidas

- Conjunto (contenido en un array unidimensional) de configuraciones resultado de la operación.

#### 2) Ir desde un estado a otro con un símbolo

##### Entradas

- Gramática independiente del contexto (contenida en un fichero de texto plano)
- Conjunto (contenido en un array unidimensional) de configuraciones.
- Símbolo (un string que contiene el símbolo de la transición)

##### Salidas

- Conjunto (contenido en un array unidimensional) de configuraciones que es el estado resultado de la operación.

Se sugiere al alumno que siga las recomendaciones que se detallan a continuación.

- 1) Los símbolos de la gramática deben ser strings C
- 2) El fichero de la gramática debe tener el formato que se muestra en la siguiente tabla.

Sección de símbolos terminales
// TERMINALES
Cada símbolo en una línea independiente
Sección de símbolos no terminales
//VARIABLES
Cada símbolo en una línea independiente
Se considerará que el axioma es el primero de los símbolos no terminales de esta lista
Sección de reglas
//REGLAS
El formato de cada reglas es:
<ul style="list-style-type: none"><li>• Comienza con la línea /*</li></ul>

- A continuación la siguiente línea mostrará el símbolo de la parte izquierda.
- Después se separará la parte derecha con la línea `||`
- La parte derecha tendrá tantas líneas como símbolos y en cada una de ellas sólo estará el correspondiente símbolo.
- La regla termina con la línea `*/`

Observe que una regla lambda tendrá como parte derecha las dos siguientes líneas

```
||
*/
```

A continuación se muestra cómo se representaría la gramática:

```
X::=SO
S::=aSb
S::=ab
O::=oO
O::=lambda
```

Sección de terminales	//TERMINALES a b o
Sección de no terminales	//VARIABLES X S O
Sección de reglas	//REGLAS /* X    S O */ /* S    a S b */ /* S    a b */ /* O    o O */ /* O    */

3) Las configuraciones (ya sea en la entrada estándar o en un fichero) seguirán el formato siguiente.

Comenzará con la línea

```
/*
```

A continuación la siguiente línea mostrará el símbolo de la parte izquierda

Después se separará la parte derecha con la línea

```
||
```

La parte derecha tendrá tantas líneas como símbolos y en cada una de ellas sólo estará el correspondiente símbolo

El símbolo utilizado para el puntero de análisis será el punto .

La parte de regla de la configuración terminará con la línea

```
>>
```

A continuación, en líneas separadas, se mostrarán los símbolos de adelanto

La configuración terminará con la línea

```
*/
```

La configuración  $S ::= a \cdot S b \{ \$, a \}$  se representaría como se indica en la tabla.

```
/*
S
||
a
.
S
b
>>
$
a
*/
```

Los conjuntos de configuraciones se representarán mediante la secuencia de representaciones de cada una de las configuraciones. Por ejemplo, el conjunto

$S ::= a \cdot S b \{ \$, a \}$

$O ::= \cdot \{ b, o \}$

Se representaría de la siguiente manera:

```
/*
S
||
a
.
S
b
>>
$
a
*/
/*
O
||
>>
b
o
*/
```

- 4) El fichero del programa se llamará lr1\_tools.c y se invocará con los siguientes argumentos y en el siguiente orden dependiendo del tipo de operación que se solicite. El tipo de operación se codifica mediante un número que es el primer argumento del programa y cuyo significado es 1=cierre, 2=ir a. La salida se mostrará siempre por la salida estándar.
- Para cierre (primer argumento 1)
  - Fichero que contiene la gramática
  - Fichero que contiene la configuración
  - Para ir a (primer argumento 2)
  - Fichero que contiene la gramática
  - Fichero que contiene el conjunto de configuraciones de partida
  - El símbolo mediante el que se transita que se escribirá literalmente en la entrada estándar

A continuación se muestran dos ejemplos de invocación del programa

```
>lr1_tools 1 gramatica.txt configuracion.txt  
>lr1_tools 2 gramatica.txt estado.txt simbolo1
```

Donde

gramatica.txt, configuracion.txt y estado.txt son los ficheros de texto plano que contienen respectivamente la gramática, configuración y conjunto de configuraciones necesarios para la operación solicitada

simbolo1 representa el símbolo "simbolo1"

### **Otras observaciones**

Debe proporcionar todos los ficheros necesarios para poder obtener el ejecutable de nombre lr1\_tools

Se recomienda que utilice la herramienta make para que el siguiente comando genere el fichero

ejecutable lr1\_tools

```
>make all
```

Las prácticas que no respeten las indicaciones anteriores podrán ser consideradas no entregadas.

Los aspectos que quedan abiertos en el enunciado deben ser completados por el alumno.

El alumno debe acompañar los ficheros fuentes con una breve memoria que describa esas decisiones.

Debe entregarse un fichero .zip que contenga los fuentes junto con la documentación junto con la entrega del compilador. El fichero debe llamarse como el del compilador pero añadiendo antes de la extensión la palabra tools\_lr1

Se recomienda que se ponga en contacto con cualquiera de los profesores de prácticas Alfonso Ortega o Marina de la Cruz para el seguimiento y resolución de dudas.