



4º Ingeniería Informática

II26 Procesadores de lenguaje

Preguntas de examen (julio 2001–septiembre 2010)



UNIVERSITAT
JAUME·I

Índice

Introducción	5
I Análisis léxico	7
II Análisis sintáctico	15
III Análisis semántico	21
IV Análisis ascendente	33
V Preguntas sobre análisis ascendente y descendente	39

Introducción

Esta recopilación recoge preguntas de examen desde julio de 2001 hasta septiembre de 2010. En concreto, se recogen las preguntas “de teoría”, es decir aquellas que no corresponden a modificaciones del compilador. Hemos clasificado las preguntas en cinco secciones:

Análisis léxico: agrupa preguntas sobre la fase de análisis léxico. Principalmente sobre expresiones regulares y autómatas de estados finitos.

Análisis sintáctico: agrupa preguntas sobre el análisis sintáctico sobre gramáticas LL(1).

Análisis semántico: agrupa preguntas sobre esquemas de traducción dirigidos por la sintaxis.

Análisis ascendente: agrupa preguntas sobre análisis LR.

Preguntas sobre análisis ascendente y descendente: agrupa preguntas sobre gramáticas LL y LR.

|

Análisis léxico

PREGUNTA I-1

(EXAMEN: 12 DE JULIO DE 2001)

¿Cuáles de las siguientes restricciones a los números de un lenguaje de programación pueden expresarse en el analizador léxico mediante expresiones regulares y cuáles deben expresarse mediante las acciones asociadas? Justifica las respuestas.

- El número tiene que ser primo.
- El número tiene que ser múltiplo de siete.
- El número tiene que ser capicúa.
- El número tiene que ocupar menos de 16bits.

PREGUNTA I-2

(EXAMEN: 18 DE SEPTIEMBRE DE 2001)

¿Qué lenguajes representan las siguientes expresiones regulares?

- $0(0|1)^*0$
- $(0|1)^*0(0|1)(0|1)$
- $0^*10^*10^*10^*$
- $(00|11)^*((01|10)(00|11)^*(01|10)(00|11)^*)^*$

PREGUNTA I-3

(EXAMEN: 19 DE DICIEMBRE DE 2001)

Escribe expresiones regulares, o demuestra que no existen, para los siguientes lenguajes:

- Cadenas sobre el alfabeto $\{a, b, c\}$ tales que si aparece al menos una b , debe haber al menos una a .
- Cadenas sobre el alfabeto $\{a, b, c\}$ tales que no contienen la subcadena ba .
- Números binarios mayores que 101001 .
- Números binarios de seis bits palíndromos.

PREGUNTA I-4

(EXAMEN: 17 DE DICIEMBRE DE 2002)

Escribe expresiones regulares para los siguientes lenguajes:

- Comentarios de un lenguaje de programación que comienzan por una secuencia de tres asteriscos y terminan por otra secuencia de tres asteriscos sin ninguna de tales secuencias entre ellas. Por ejemplo:

`*** Esto es un comentario ***`

- Identificadores formados por letras minúsculas que no coincidan ni con la palabra reservada `fi` ni con `fins`.
- Números tales que en la secuencia de sus dígitos no aparece el número doce (`13124` no valdría y `54213` sí).

PREGUNTA I-5

(EXAMEN: 31 DE MAYO DE 2003)

Rellena el siguiente cuadro, indicando, con SI o NO, si la cadena es aceptada por la correspondiente expresión regular:

	λ	$z?a$	az^*	$??$	$b-c?$	(a)	(b)	$)$	\backslash	$\backslash\backslash$
$[\wedge az]^*$?										
$([\wedge a]^*[\wedge z]^*)^*$										
$([\wedge az]^*)^*$										
$\backslash([\wedge az]^*\backslash)^*$										
$[\wedge (az)^*]^*$?										

PREGUNTA I-6

(EXAMEN: 30 DE JUNIO DE 2003)

Escribe expresiones regulares o autómatas finitos deterministas; o bien demuestra que no existen, para los siguientes lenguajes^a:

- Cadenas de ceros y unos que representan caracteres ASCII imprimibles (7 bits con valor entre 32 y 127).
- Comentarios de un lenguaje de programación que empiezan por la secuencia `{*}`, terminan con `*` y no contienen ni saltos de línea, ni ninguna secuencia `*`.
- Cadenas formadas únicamente por los tokens **id** (identificador), **entrada** (palabra clave `entrada`), **corcheteAb** (corchete abierto), **corcheteCer** (corchete cerrado), **flecha** (la secuencia `->`), **pyc** (punto y coma) y que son sentencias válidas en MM3.
- Literales de cadena de MM3 con, al menos, una secuencia de escape.

^aMM3 fue el lenguaje empleado en el curso 2002-2003 para construir un compilador en prácticas.

PREGUNTA I-7

(EXAMEN: 9 DE SEPTIEMBRE DE 2003)

Las siguientes expresiones regulares son intentos de modelar los comentarios de un determinado lenguaje de programación. En este lenguaje, los comentarios comienzan con la secuencia `*->` y terminan en `<-*`, sin ninguna aparición de `<-*` en su interior. Por cada expresión, di si es o no correcta y, si no lo es, escribe una cadena para la que falle.

1. `*->[^<-*]*<-*`
2. `*->([<|[^-]|[*])*<-*`
3. `*->([<|<[^-]|<-[*])*<-*`
4. `*->([<|<+[^-]|<+[-<+[^-*<])*<-*`

PREGUNTA I-8

(EXAMEN: 13 DE DICIEMBRE DE 2003)

Describe brevemente cómo organizarías la parte del analizador léxico que controla la indentación si tuvieras que escribir un intérprete Python. Observa en el siguiente ejemplo los componentes emitidos para cada una de las líneas:

```

for i in l:           for id in id ptos nl
    for j in l:       dentro for id in id ptos nl
        for k in l:  dentro for id in id ptos nl
            print k  dentro print id nl
        print i     fuera fuera print id nl
    fuera eof

```

Como ves, cuando se añade un nivel de indentación (con independencia del número de espacios empleados para ello), se emite un componente **dentro**; cuando se reduce la indentación, se emiten tantos componentes **fuera** como niveles de indentación se retrocede. Al llegar al fin de fichero, se emiten los componentes **fuera** correspondientes a los niveles de indentación no cerrados en ese momento.

Hay un error léxico si, al reducir la indentación, la posición alcanzada no coincide con ninguna de las anteriores. Por ejemplo:

```

for i in l:           for id in id ptos nl
    for j in l:       dentro for id in id ptos nl
    for k in l:       Error léxico

```

Puedes asumir que el espacio en blanco de antes de la línea no incluye ningún tabulador y que ningún componente ocupa más de una línea.

Pista: puede ser útil tener un atributo que contenga el número de **fuera** pendientes de emitir y que lo primero que haga el léxico ante una llamada a **siguiente** sea comprobar el valor de ese atributo.

PREGUNTA I-9

(EXAMEN: 21 DE JUNIO DE 2004)

El método explicado en el tema de análisis léxico para generar autómatas a partir de expresiones regulares utilizando ítems, ¿genera autómatas mínimos? Justifica la respuesta.

PREGUNTA I-10

(EXAMEN: 1 DE SEPTIEMBRE DE 2004)

Sean r y s dos expresiones regulares. Definimos el operador *lista* (\uparrow) de modo que la expresión regular $(r\uparrow s)$ representa el lenguaje $L_{(r\uparrow s)} = L_r(L_s L_r)^*$. Este operador nos permite especificar cómodamente listas de elementos con un separador. Por ejemplo, las listas de enteros separados por barras las podemos especificar como $[0-9]^+\uparrow\backslash|$.

Explica cómo se debe modificar el algoritmo de cálculo de construcción de autómatas a partir de expresiones regulares para tener en cuenta este nuevo operador. Para ello, basta con que expliques cómo modificar el cuadro de los apuntes que explica las transformaciones que se deben aplicar a los ítems no básicos para calcular la clausura de los estados (este es el cuadro 1 de la página 13 de los apuntes).

Utiliza el algoritmo para construir el autómata correspondiente a la expresión regular $[0-9]^+\uparrow\backslash|$.

PREGUNTA I-11

(EXAMEN: 18 DE DICIEMBRE DE 2004)

Sea la r_n la expresión regular $\overbrace{(a|ab) \dots (a|ab)}^n$. Se pide:

- Lenguaje representado por r_n .
- Número de estados del AFD construido a partir de r_n utilizando ítems; o bien, número de estados del AFN construido mediante el algoritmo de bloques.

PREGUNTA I-12

(EXAMEN: 20 DE JUNIO DE 2005)

Queremos incluir en nuestro lenguaje de expresiones regulares el operador de repetición. Este operador es *postfijo* y se

escribe como un exponente que indica el número de repeticiones de la expresión a la que afecta. Así $L_{r^n} = \overbrace{L_r \dots L_r}^{n \text{ veces}}$.

Escribe, utilizando el operador de repetición, una expresión regular para secuencias de entre uno y cien enteros separados por comas.

Explica cómo deberíamos modificar el algoritmo de construcción de autómatas a partir de expresiones regulares, su tabla o ambos para incluir este operador *sin transformar la expresión en su equivalente mediante concatenaciones*. Aplica tu versión modificada a la expresión regular $((ab)^2a)^*$.

PREGUNTA I-13

(EXAMEN: 8 DE SEPTIEMBRE DE 2005)

Escribe expresiones regulares o demuestra que no existen para los siguientes lenguajes:

- Cadenas de cero o más dígitos distintos ordenados de menor a mayor.
- Cadenas de cero o más dígitos, posiblemente repetidos, ordenados de menor a mayor.
- Secuencias de uno o más números naturales separados por comas y ordenados de menor a mayor.
- Números naturales múltiplos de cuatro.

Nota: Para los números naturales, se considera que son válidas secuencias iniciales de ceros.

PREGUNTA I-14

(EXAMEN: 16 DE SEPTIEMBRE DE 2005)

Escribe autómatas finitos deterministas o demuestra que no existen para los siguientes lenguajes:

- Cadenas de cero o más vocales distintas ordenadas alfabéticamente.
- Cadenas de cero o más vocales, posiblemente repetidas, ordenadas alfabéticamente.
- Secuencias de uno o más identificadores separados por comas y ordenados alfabéticamente.
- Números naturales múltiplos de cuatro.

Nota: Para los números naturales, se considera que son válidas secuencias iniciales de ceros.

PREGUNTA I-15

(EXAMEN: 16 DE DICIEMBRE DE 2005)

En un lenguaje de programación se escriben los comentarios comenzando por el carácter almohadilla (#) y terminando con otro carácter almohadilla. Entre ambas almohadillas se admite cualquier número de caracteres que no sean almohadillas y otras almohadillas si están inmediatamente precedidas de, al menos, una almohadilla y no tocan a la que cierra el comentario. Por ejemplo, tanto “# incremento i #” como “### i##j #” son comentarios válidos. No lo son, sin embargo “#i####” o “#i#j#”.

Dadas las siguientes expresiones regulares, indica para cada una de ellas si corresponde o no a los comentarios de este lenguaje. En caso negativo, proporciona un contraejemplo, en caso positivo, demuéstralo construyendo el AFD correspondiente a la expresión.

- $\#^+([\^{\#}]\#\#)^*\#$
- $\#([\^{\#}]|\#\#^+[\^{\#}])^*\#$
- $\#([\#\^{\#}])^*\#$
- $\#^+[\^{\#}]^+(\#\#^+[\^{\#}]^+)^*\#$

PREGUNTA I-16

(EXAMEN: 24 DE JUNIO DE 2006)

Sea r_n la expresión regular $ab|aabb|\dots| \overbrace{a\dots a}^n \overbrace{b\dots b}^n$. ¿Cuántos estados tiene el autómata construido para r_n mediante el algoritmo explicado en clase?

Si el autómata resultante es mínimo, demuéstralo; en caso contrario, di cuántos estados tiene el autómata mínimo.

PREGUNTA I-17

(EXAMEN: 15 DE SEPTIEMBRE DE 2006)

Dada una expresión regular r definimos $ne(r)$ como el número de estados del AFD construido mediante el método de los ítems. Dadas las expresiones regulares r y s , demuestra la verdad o falsedad de las siguientes proposiciones:

- Si $ne(r) \leq ne(s)$, entonces $L(r) \subseteq L(s)$.
- Si $L(r) \subseteq L(s)$, entonces $ne(r) \leq ne(s)$.
- $ne(r) \leq ne(r|s)$.

PREGUNTA I-18

(EXAMEN: 12 DE DICIEMBRE DE 2006)

Sea r_n la expresión regular $(a|ba|baa|\dots| \overbrace{ba\dots a}^n)^*$. Escribe el autómata finito determinista obtenido para r_3 mediante el algoritmo de construcción usando ítems. Para un n cualquiera mayor que cero, ¿cuántos estados tiene el autómata finito determinista obtenido para r_n mediante el algoritmo de construcción usando ítems?

PREGUNTA I-19

(EXAMEN: 26 DE JUNIO DE 2007)

Sea r_n la expresión regular $(\overbrace{ab\dots b}^n | \overbrace{aab\dots b}^{n-1} | \dots | \overbrace{a\dots abb}^{n-1} | \overbrace{a\dots ab}^n)$. Escribe el autómata finito determinista obtenido para r_3 mediante el algoritmo de construcción usando ítems. Para un n cualquiera mayor que cero, ¿cuántos estados tiene el autómata finito determinista obtenido para r_n mediante el algoritmo de construcción usando ítems?

PREGUNTA I-20

(EXAMEN: 14 DE SEPTIEMBRE DE 2007)

Sea r_n la expresión regular $b^*(ab)^*(aab)^*\dots(\overbrace{a\dots ab}^n)^*$. Escribe el autómata finito determinista obtenido para r_3 mediante el algoritmo de construcción usando ítems. Para un n cualquiera mayor que cero, ¿cuántos estados tiene el autómata finito determinista obtenido para r_n mediante el algoritmo de construcción usando ítems?

PREGUNTA I-21

(EXAMEN: 11 DE DICIEMBRE DE 2007)

En un lenguaje de programación ficticio, los comentarios comienzan con la secuencia de tres caracteres <<< y terminan al llegar a un salto de línea o al aparecer la secuencia de tres caracteres >>>, lo que suceda primero. Modela, mediante una expresión regular, el conjunto de estos comentarios. Debes tener en cuenta que el delimitador >>> formaría parte del comentario, pero el salto de línea no formaría parte del comentario.

PREGUNTA I-22

(EXAMEN: 24 DE JUNIO DE 2008)

Escribe tanto una expresión regular como un AFD para el lenguaje de los comentarios con las siguientes características:

1. Empiezan por << y terminan por >>.
2. En su interior admiten fragmentos especiales que empiezan y terminan por comillas simples '.
3. En el interior de un comentario las comillas ' sólo pueden aparecer en una cantidad par, delimitando los fragmentos especiales.
4. En el interior de un comentario no se admite la secuencia >>, excepto dentro de los fragmentos especiales.

Esto son ejemplos de comentarios válidos:

```
<<>>
<< abc >>
<< abc 'de' fg 'hijk' lm >>
<< '' abc 'de' fg 'hi>>jk' '>>>>>' l>m >>
```

Y esto no son comentarios válidos:

```
<< abc 'def >>
<< abc >> def >>
```

PREGUNTA I-23

(EXAMEN: 12 DE SEPTIEMBRE DE 2008)

Escribe tanto una expresión regular como un AFD para los comentarios de un lenguaje de programación tales que: comienzan por una cadena del lenguaje definido por la expresión regular <-⁺\|; terminan con la primera aparición de una cadena del lenguaje definido por la expresión regular \|⁺>; y no es necesario que coincidan las longitudes de la cadenas de inicio y fin del comentario.

Esto son ejemplos de comentarios válidos:

```
<--| |-->
<---| calculamos a-b |-->
<----| ejecutar si |x-y|>6 |-->
<--| |-|-->
```

Y estos no son válidos:

```
<---|-->
<---| a|----->b |-->
```

PREGUNTA I-24

(EXAMEN: 19 DE DICIEMBRE DE 2008)

Escribe tanto un AFD como una expresión regular para modelar el lenguaje de las cadenas de dígitos que empiezan por 111, por 12 o por 21, y terminan con la primera de esas tres subsecuencias que aparezca tras la inicial. Supón que el alfabeto está formado sólo por dígitos.

PREGUNTA I-25

(EXAMEN: 30 DE JUNIO DE 2009)

Escribe un AFD para el lenguaje de las cadenas de dígitos que no empiezan ni terminan por 22. Por ejemplo, serían válidas las cadenas 2, 212, 314, 122223, pero no las cadenas 22, 223, 12222.

Asume que el alfabeto es el de los dígitos.

PREGUNTA I-26

(EXAMEN: 11 DE SEPTIEMBRE DE 2009)

Suponiendo que el alfabeto es el de los dígitos 0 a 9:

1. Escribe una expresión regular para reconocer las cadenas de dos o más dígitos impares tales que el último dígito es mayor que todos los anteriores. Por ejemplo, serían válidas las cadenas 135, 1317 o 357 pero no las cadenas 235 (2 es par), 735 (5 no es mayor que 7), 5 (sólo tiene un dígito) ni 11 (1 no es mayor que 1).
2. Dibuja un AFD para reconocer el sublenguaje formado por cadenas que cumplen las mismas restricciones del apartado anterior y en las que además no aparecen los dígitos 7 ni 9.

PREGUNTA I-27

(EXAMEN: 11 DE DICIEMBRE DE 2009)

Escribe tanto un AFD como una expresión regular para reconocer las cadenas de dígitos, posiblemente vacías, que **no contienen** la subcadena 1112 (formada por esos 4 caracteres seguidos, sin ninguna separación entre ellos). Puedes suponer que el alfabeto es el de los dígitos.

Por ejemplo, 11132112, 1121 y 2111 pertenecen al lenguaje que debes modelar, mientras que 1112, 1111122222 y 3211123 no pertenecen a él.

PREGUNTA I-28

(EXAMEN: 28 DE JUNIO DE 2010)

Los analizadores de lenguaje XML ignoran las denominadas secciones **CDATA**. Cada una de dichas secciones se abre exactamente con la secuencia de caracteres `<![CDATA[` y se cierra con la primera secuencia de tres caracteres `]]>` que siga a la secuencia de apertura.

Escribe tanto un AFD como una expresión regular para modelar dichas secciones.

PREGUNTA I-29

(EXAMEN: 10 DE SEPTIEMBRE DE 2010)

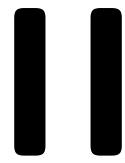
Escribe una expresión regular y un AFD para las cadenas largas de Python. Estas se abren con una secuencia de tres comillas simples ('''') o de tres comillas dobles (""") y se cierran con la primera aparición de una secuencia de tres caracteres no escapados que coincida con la secuencia de apertura. Entre la apertura y el cierre pueden aparecer cero o más repeticiones de:

- Cualquier carácter ASCII (incluyendo nueva línea y tabulador) salvo una barra invertida aislada.
- Secuencias de escape formadas por el carácter barra invertida seguida de cualquier carácter ASCII (incluyendo nueva línea, tabulador y la propia barra).

Un ejemplo de este tipo de cadenas es:

```
'''Esto es una 'cadena larga' que tiene varias lineas,
que tiene secuencias de escape como \a, \7 y \t
y no se cierra con tres comillas dobles "" sino con
tres simples.'''
```

Importante: En tu solución, debes sustituir ' por 1 y " por 2 para evitar errores de interpretación durante la corrección.



Análisis sintáctico

PREGUNTA II-1

(EXAMEN: 2 DE JUNIO DE 2001)

Estás preparando un nuevo método de análisis sintáctico para gramáticas incontextuales y has llegado a la conclusión de que necesitas saber, para una forma sentencial, cuál es el conjunto de terminales que pueden aparecer en segunda posición en las sentencias que se deriven de ella. Decides formalizar tu función segundos de la siguiente manera:

$$\text{segundos}(\alpha) = \{a \in \Sigma \mid \exists b \in \Sigma, \beta \in \Sigma^* : \alpha \xrightarrow{*} ba\beta\}$$

Escribe un algoritmo análogo al que se emplea para el cálculo de primeros. Puedes asumir que tu gramática no tiene recursividad por la izquierda y que dispones de las funciones:

- $\text{primeros}(\alpha)$: devuelve los primeros de una forma sentencial α , incluyendo la cadena vacía si $\alpha \xrightarrow{*} \lambda$.
- $\text{unitario}(\langle A \rangle)$ devuelve cierto si $\exists a \in \Sigma : \langle A \rangle \xrightarrow{*} a$, falso en caso contrario.

PREGUNTA II-2

(EXAMEN: 19 DE DICIEMBRE DE 2001)

Nos han propuesto la siguiente gramática para resolver el problema de la ambigüedad del if-then:

$$\begin{aligned} \langle S \rangle &\rightarrow \text{if } \langle E \rangle \text{ then } \langle S \rangle \\ \langle S \rangle &\rightarrow \langle M \rangle \\ \langle M \rangle &\rightarrow \text{if } \langle E \rangle \text{ then } \langle M \rangle \text{ else } \langle S \rangle \\ \langle M \rangle &\rightarrow \text{otro} \\ \langle E \rangle &\rightarrow \text{id} \end{aligned}$$

Demuestra que esta gramática es ambigua.

PREGUNTA II-3

(EXAMEN: 3 DE JULIO DE 2002)

Queremos enriquecer las GPDR con un nuevo operador: el operador lista (\wedge). Este operador se define de modo que $(a \wedge b)$ es igual a $a(ba)^*$. Así, la regla para describir una lista de identificadores separados por comas se escribe como:

$$\langle ID \rangle \rightarrow \text{id } \wedge ,$$

Reescribe los algoritmos del tema 3 para el cálculo de primeros, anulables y siguientes de modo que tengan en cuenta este operador. Utilízalos para comprobar si la siguiente gramática es RLL(1):

$$\begin{aligned} \langle E \rangle &\rightarrow \langle T \rangle \wedge (+|-) \\ \langle T \rangle &\rightarrow \langle F \rangle \wedge (*|/) \\ \langle F \rangle &\rightarrow \text{id|cte|“(”}\langle E \rangle\text{“)”} \end{aligned}$$

PREGUNTA II-4

(EXAMEN: 17 DE DICIEMBRE DE 2002)

Sea $G = (N, \Sigma, P, \langle S \rangle)$ una gramática incontextual tal que si $\langle A \rangle \rightarrow \alpha \in P$ y $\langle A \rangle \rightarrow \beta \in P$, entonces $\alpha = \beta$. Se pide:

- Demostrar que G es LL(1).
- Decir qué podemos afirmar acerca de si G es RLL(1) en caso de que G sea una GPDR.

PREGUNTA II-5

(EXAMEN: 13 DE DICIEMBRE DE 2003)

¿Cuáles de las siguientes afirmaciones acerca de las tablas de análisis LL(1) son ciertas y cuáles falsas? Justifica brevemente las respuestas.

1. No puede haber dos filas no vacías idénticas.
2. No puede haber dos filas vacías.
3. No puede haber dos columnas no vacías idénticas.
4. No puede haber dos columnas vacías.

PREGUNTA II-6

(EXAMEN: 24 DE JUNIO DE 2006)

Diseña un método para calcular los últimos de un no terminal $\langle A \rangle$, definidos como

$$\text{últimos}(\langle A \rangle) = \{a \in \Sigma \mid \exists \alpha \in \Sigma^* : \langle A \rangle \xRightarrow{*} \alpha a\}.$$

Es decir, $\text{últimos}(\langle A \rangle)$ es el conjunto de terminales que pueden ser el último símbolo de las cadenas generadas por $\langle A \rangle$. Escribe la traza de la aplicación de tu método en el cálculo de los últimos de $\langle A \rangle$ en la gramática siguiente:

$$\begin{aligned} \langle A \rangle &\rightarrow \langle B \rangle \langle A \rangle | a | \lambda \\ \langle B \rangle &\rightarrow \langle A \rangle \langle C \rangle | b \\ \langle C \rangle &\rightarrow \langle C \rangle c \langle C \rangle | d \langle A \rangle \end{aligned}$$

PREGUNTA II-7

(EXAMEN: 11 DE DICIEMBRE DE 2007)

Demuestra que la siguiente gramática es ambigua:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle A \rangle \langle B \rangle | \langle C \rangle \\ \langle A \rangle &\rightarrow a \langle A \rangle b | ab \\ \langle B \rangle &\rightarrow c \langle B \rangle d | cd \\ \langle C \rangle &\rightarrow a \langle C \rangle d | a \langle D \rangle d \\ \langle D \rangle &\rightarrow b \langle D \rangle c | bc \end{aligned}$$

PREGUNTA II-8

(EXAMEN: 24 DE JUNIO DE 2008)

Escribe las producciones de una gramática $G = (\{\langle A \rangle, \langle B \rangle, \langle C \rangle\}, \{a, b, c\}, P, \langle A \rangle)$ sabiendo que es LL(1), que todos los no terminales son anulables y que las únicas celdas no vacías de su tabla de análisis son las sombreadas en el esquema siguiente:

	a	b	c	\$
$\langle A \rangle$				
$\langle B \rangle$				
$\langle C \rangle$				

PREGUNTA II-9

(EXAMEN: 12 DE SEPTIEMBRE DE 2008)

Obtén una gramática incontextual, sin partes derechas regulares, que sea LL(1) y que genere el mismo lenguaje que la siguiente:

$$\langle S \rangle \rightarrow (a|c|ba|bc)^*b?$$

siendo $\{a, b, c\}$ el conjunto de terminales. Demuestra que tu gramática es LL(1).

PREGUNTA II-10

(EXAMEN: 19 DE DICIEMBRE DE 2008)

Sea G una gramática LL(1) y $\langle A \rangle$ un no terminal para el que existe una única regla $\langle A \rangle \rightarrow \alpha$. Llamemos $G_{\langle A \rangle}$ a la gramática en la que todas las apariciones de $\langle A \rangle$ en partes derechas de las reglas de G han sido sustituidas por α . Demuestra que $G_{\langle A \rangle}$ es LL(1).

PREGUNTA II-11

(EXAMEN: 30 DE JUNIO DE 2009)

Para cada una de las siguientes tablas de análisis LL(1):

	a	b	\$
(A)			
(B)			
(C)			

Tabla a)

	a	b	\$
(A)			
(B)			
(C)			

Tabla b)

	a	b	\$
(A)			
(B)			
(C)			

Tabla c)

escribe, o demuestra que no es posible, una gramática incontextual que sea LL(1), cuyo símbolo inicial sea $\langle A \rangle$, que no tenga símbolos inútiles, cuyos no terminales sean todos anulables y tal que las únicas celdas no vacías de su tabla de análisis sean las sombreadas.

PREGUNTA II-12

(EXAMEN: 11 DE SEPTIEMBRE DE 2009)

Sea G una gramática LL(1) en la que todos los símbolos no terminales son no anulables. Supongamos que elegimos una producción cualquiera, $\langle A \rangle \rightarrow \alpha$, y añadimos a la parte derecha una secuencia arbitraria de símbolos terminales y no terminales, β , obteniendo la regla $\langle A \rangle \rightarrow \alpha\beta$. ¿Podemos asegurar que la gramática resultante sigue siendo LL(1)? Justifica tu respuesta.

PREGUNTA II-13

(EXAMEN: 11 DE DICIEMBRE DE 2009)

Para cada una de las dos gramáticas siguientes, escribe una gramática incontextual, sin partes derechas regulares, que sea LL(1) y que genere el mismo lenguaje:

(a) $\langle \text{CódigoMorse1} \rangle \rightarrow (\text{punto punto} \mid \text{punto punto punto})^*$

(b) $\langle \text{CódigoMorse2} \rangle \rightarrow (\text{punto punto} \mid \text{punto punto punto})^* (\text{punto raya})^*$

Demuestra en cada caso que tu gramática es LL(1).

PREGUNTA II-14

(EXAMEN: 28 DE JUNIO DE 2010)

Escribe una gramática incontextual, sin partes derechas regulares, que sea LL(1) y genere el mismo lenguaje que la siguiente:

$$\langle \text{Agenda} \rangle \rightarrow (\text{nombre teléfono})^* (\text{nombre email})^*$$

Demuestra que tu solución es LL(1).

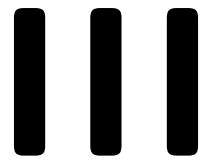
PREGUNTA II-15

(EXAMEN: 10 DE SEPTIEMBRE DE 2010)

Escribe una gramática con partes derechas regulares que sea RLL(1) y que genere el mismo lenguaje que la siguiente:

$$\begin{aligned} \langle \text{Número} \rangle &\rightarrow \langle \text{Octal} \rangle \mid \langle \text{Decimal} \rangle \mid \langle \text{Hexadecimal} \rangle \\ \langle \text{Octal} \rangle &\rightarrow (\langle \text{DígitoOctal} \rangle)^+ \text{o} \\ \langle \text{Decimal} \rangle &\rightarrow (\langle \text{DígitoDecimal} \rangle)^+ \text{d} \\ \langle \text{Hexadecimal} \rangle &\rightarrow (\langle \text{DígitoHexadecimal} \rangle)^+ \text{h} \\ \langle \text{DígitoOctal} \rangle &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \\ \langle \text{DígitoDecimal} \rangle &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\ \langle \text{DígitoHexadecimal} \rangle &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid \text{A} \mid \text{B} \mid \text{C} \mid \text{D} \mid \text{E} \mid \text{F} \end{aligned}$$

Algunos ejemplos de números generados por la gramática son: 123d, 123o, 123h, 789d, 9d, 9h, 28d, 28h, A2h, 1Ah, FFh.



Análisis semántico

PREGUNTA III-1

(EXAMEN: 22 DE MAYO DE 2004)

Sea G la siguiente gramática para modelar polinomios:

$$\begin{aligned}\langle P \rangle &\rightarrow \langle P \rangle + \langle M \rangle \mid \langle M \rangle \\ \langle M \rangle &\rightarrow \text{ent } \langle F \rangle \mid \langle F \rangle \text{ent} \\ \langle F \rangle &\rightarrow \langle F \rangle \langle I \rangle \mid \langle I \rangle \\ \langle I \rangle &\rightarrow \text{id} \mid \text{id} \text{ent}\end{aligned}$$

donde **ent** representa literales enteros e **id**, identificadores. Se pide:

- Por 1,5 puntos:
 - Construye el autómata de prefijos viables para G utilizando ítems LR(0).
 - A partir del autómata, escribe la tabla de acciones LR(0).
 - Si la tabla anterior tiene conflictos, escribe la tabla de acciones SLR.
 - Si la tabla anterior tiene conflictos, indica cómo se podrían resolver modificando la tabla o la gramática.
- Por 1,5 puntos:
 - Añade a G^a las acciones semánticas necesarias para que en el atributo *derivada* de $\langle P \rangle$ se obtenga la derivada del polinomio respecto a la variable x . Puedes elegir la representación que consideres apropiada (árbol, lista, cadena, ...). Utiliza los atributos que consideres necesarios, pero ninguna variable global. Ten en cuenta que la variable puede aparecer más de una vez en el monomio (ej, $2x^3yx$).
- Por 0,5 puntos:
 - Reescribe la gramática utilizando partes derechas regulares de modo que se obtenga una sola regla.
 - ¿Es regular el lenguaje generado por la gramática? Justifica la respuesta.

^aAunque hayas tenido que modificar G para resolver los conflictos, las acciones debes escribirlas sobre G , no sobre la versión modificada.

PREGUNTA III-2

(EXAMEN: 21 DE JUNIO DE 2004)

Sea G la siguiente gramática:

$$\begin{aligned}\langle A \rangle &\rightarrow \langle B \rangle \langle C \rangle \mid a \\ \langle B \rangle &\rightarrow \langle A \rangle a \langle B \rangle \mid \langle C \rangle a \\ \langle C \rangle &\rightarrow a \langle C \rangle \langle C \rangle \mid aba \langle C \rangle \mid a\end{aligned}$$

Añade a G las reglas semánticas necesarias para que el atributo *ia* de $\langle A \rangle$ contenga el número de *aes* al inicio de la cadena generada. Por ejemplo, dadas las cadenas **aaaaba**, **abaaaa** y **aaa**, los valores de *ia* serían 4, 1 y 3, respectivamente. Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

PREGUNTA III-3

(EXAMEN: 1 DE SEPTIEMBRE DE 2004)

Diremos que un carácter es un *extremo derecho* si es el último de la cadena o es distinto del carácter siguiente. Por ejemplo, en la cadena **aabcccaa**, hay cuatro extremos derechos: la segunda **a**, la **b**, la tercera **c** y la última **a**.

Dada la gramática G con las reglas

$$\begin{aligned}\langle A \rangle &\rightarrow \langle B \rangle \langle C \rangle | a \\ \langle B \rangle &\rightarrow \langle B \rangle a \langle B \rangle | b | \langle C \rangle \\ \langle C \rangle &\rightarrow a \langle C \rangle \langle B \rangle | b\end{aligned}$$

añade a G las reglas semánticas necesarias para que el atributo sintetizado nae de $\langle A \rangle$ contenga el número de aes que son extremos derechos. Por ejemplo, para las cadenas **babb** y **aabbbabb** los valores de nae serían 1 y 2, respectivamente. Puedes utilizar los atributos *sintetizados* adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

PREGUNTA III-4

(EXAMEN: 18 DE DICIEMBRE DE 2004)

Sea G la siguiente gramática:

$$\begin{aligned}\langle A \rangle &\rightarrow \langle B \rangle \langle C \rangle | a \\ \langle B \rangle &\rightarrow \langle A \rangle a \langle B \rangle | \langle C \rangle a \\ \langle C \rangle &\rightarrow a \langle C \rangle \langle C \rangle | a b a \langle C \rangle | a\end{aligned}$$

Añade a G las reglas semánticas necesarias para que el atributo fa de $\langle A \rangle$ contenga el número de aes al final de la cadena generada. Por ejemplo, dadas las cadenas **aaaaba**, **abaaaa** y **aaa**, los valores de fa serían 1, 4 y 3, respectivamente. Puedes utilizar los atributos *sintetizados* adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

PREGUNTA III-5

(EXAMEN: 20 DE JUNIO DE 2005)

Sea G la siguiente gramática:

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle \\ \langle A \rangle &\rightarrow \langle A \rangle \langle B \rangle \langle A \rangle | a \langle A \rangle | a b \\ \langle B \rangle &\rightarrow \langle B \rangle a | \langle B \rangle b | \lambda\end{aligned}$$

Supongamos que la primera regla tiene asociada la acción

$$\langle S \rangle \rightarrow \langle A \rangle \{ \text{escribe}(\text{"La subcadena de aes más larga tiene ", } \langle A \rangle . \text{ma}, \text{" aes."}); \}$$

donde ma tiene el número de aes contenido en la subcadena más larga formada únicamente por aes. Por ejemplo, para la cadena **abaaaab**, el valor de ma será 4. Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor de este atributo.

Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

PREGUNTA III-6

(EXAMEN: 8 DE SEPTIEMBRE DE 2005)

Sea G la siguiente gramática:

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle \\ \langle A \rangle &\rightarrow \langle A \rangle \langle B \rangle \langle A \rangle \mid \langle B \rangle \langle A \rangle \mid \langle B \rangle \\ \langle B \rangle &\rightarrow \mathbf{num} \mid \langle B \rangle \mathbf{num} \mid \lambda\end{aligned}$$

Supongamos que la primera regla tiene asociada la acción

$$\langle S \rangle \rightarrow \langle A \rangle \{\mathbf{si} \langle A \rangle.\mathbf{ord} \mathbf{entonces} \mathbf{escribe} \text{ (“Está ordenada”) } \mathbf{si} \mathbf{no} \mathbf{escribe} \text{ (“Está desordenada”)}\}$$

donde *ord* es un atributo de tipo lógico que es cierto si la secuencia de números enteros, posiblemente repetidos, es vacía o está ordenada de menor a mayor y falso en caso contrario. Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor de este atributo.

Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico. Explica el significado de cada atributo que añadas. Asume que los componentes de la categoría **num** tienen su valor en el atributo v .

PREGUNTA III-7

(EXAMEN: 16 DE SEPTIEMBRE DE 2005)

Sea G la siguiente gramática:

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle \\ \langle A \rangle &\rightarrow \langle A \rangle \langle B \rangle \langle A \rangle \mid \langle B \rangle \langle A \rangle \mid \langle B \rangle \\ \langle B \rangle &\rightarrow \mathbf{num} \mid \langle B \rangle \mathbf{num} \mid \lambda\end{aligned}$$

Supongamos que la primera regla tiene asociada la acción

$$\langle S \rangle \rightarrow \langle A \rangle \{\mathbf{si} \langle A \rangle.\mathbf{nrc} \mathbf{entonces} \mathbf{escribe} \text{ (“No hay problemas”) } \mathbf{si} \mathbf{no} \mathbf{escribe} \text{ (“Hay repetidos consecutivos”)}\}$$

donde *nrc* (“no repetidos consecutivos”) es un atributo de tipo lógico que es cierto si la secuencia de números enteros es vacía o no hay dos números *consecutivos* iguales y falso en caso contrario. Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor de este atributo.

Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico. Explica el significado de cada atributo que añadas. Asume que los componentes de la categoría **num** tienen su valor en el atributo v .

PREGUNTA III-8

(EXAMEN: 16 DE DICIEMBRE DE 2005)

Sea G la siguiente gramática:

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle \\ \langle A \rangle &\rightarrow \langle B \rangle \langle A \rangle \langle B \rangle \mid \mathbf{a} \langle A \rangle \mid \mathbf{a} \mathbf{b} \\ \langle B \rangle &\rightarrow \langle B \rangle \mathbf{a} \mid \langle B \rangle \mathbf{b} \mid \lambda\end{aligned}$$

Supongamos que la primera regla tiene asociada la acción

$$\langle S \rangle \rightarrow \langle A \rangle \{\mathbf{si} \langle A \rangle.\mathbf{alt} \mathbf{entonces} \mathbf{escribe} \text{ (“Es alternante”) } \mathbf{si} \mathbf{no} \mathbf{escribe} \text{ (“No es alternante”)}\}$$

donde *alt* es cierto si la cadena generada no tiene dos símbolos consecutivos iguales. Por ejemplo, si la cadena generada fuera **abab**, el atributo sería cierto mientras que si fuera **aab**, el atributo sería falso. Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor de este atributo.

Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

PREGUNTA III-9

(EXAMEN: 15 DE SEPTIEMBRE DE 2006)

Supongamos que tenemos la siguiente gramática para las expresiones de un lenguaje de programación:

$$\begin{aligned}\langle E \rangle &\rightarrow \langle E \rangle + \langle T \rangle | \langle T \rangle \\ \langle T \rangle &\rightarrow \langle T \rangle * \langle F \rangle | \langle F \rangle \\ \langle F \rangle &\rightarrow \text{entero} | \text{cadena} | \text{"("} \langle E \rangle \text{"}"\end{aligned}$$

Como ves, son expresiones sobre enteros y cadenas. Las reglas semánticas permiten la suma y el producto de enteros, la concatenación de cadenas (mediante $+$) y la repetición de cadenas un número de veces (mediante $*$ aplicado a una cadena y un entero, en cualquier orden).

Añade *al final de cada regla*, las acciones necesarias para sintetizar en el símbolo inicial $\langle E \rangle$ los siguientes atributos:

- *tipo*, que contiene el tipo de la expresión (entero, cadena o error),
- *mse*, que contiene la longitud del producto de l30 de junio de 2009 iterales enteros más largo que multiplica *directamente* (sin paréntesis) un literal de cadena. Algunos ejemplos de expresiones con los correspondientes valores del atributo son:

Expresión	<i>mse</i>
entero*cadena*entero*entero	2
entero*cadena*(entero*entero)	1
(entero*cadena)*entero*entero	1
cadena*(entero+entero)*entero	0

PREGUNTA III-10

(EXAMEN: 12 DE DICIEMBRE DE 2006)

Queremos reconocer cadenas de paréntesis y corchetes de modo que las subcadenas obtenidas considerando sólo los paréntesis o sólo los corchetes estén bien parentizadas, sin importar las relaciones entre paréntesis y corchetes. Así, la cadena $(([])([]))$ es correcta porque tanto $(([]))$ como $[[[]]]$ están bien parentizadas. Sin embargo, no lo es la cadena $(([]))$ porque $]$ no está bien parentizada.

Modela estas cadenas de una de las dos formas siguientes:

- Mediante una GPDR.
- Mediante un esquema de traducción de modo que el atributo sintetizado *bien* del símbolo inicial sea cierto si la cadena generada está bien parentizada y falso en caso contrario. Si optas por esta vía, sólo puedes emplear atributos sintetizados de tipo lógico, carácter o entero y las acciones deben estar al final de las reglas.

En ambos casos, no debes preocuparte de si la gramática resultante es o no de las familias LL, RLL o LR.

PREGUNTA III-11

(EXAMEN: 26 DE JUNIO DE 2007)

Sea G la siguiente gramática, que genera listas de enteros separados por comas:

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle \\ \langle A \rangle &\rightarrow \text{díg coma díg} \langle A \rangle | \text{díg} \langle B \rangle \text{díg} \langle B \rangle \text{coma} \langle B \rangle \\ \langle B \rangle &\rightarrow \text{díg} \langle B \rangle \text{díg} | \langle B \rangle \text{coma} \langle A \rangle \text{coma} \langle B \rangle | \text{díg}\end{aligned}$$

donde **díg** es un terminal que representa un dígito con valor **díg.v** y **coma** representa una coma. Supongamos que la regla inicial tiene la acción semántica siguiente:

$$\langle S \rangle \rightarrow \langle A \rangle \{ \text{escribe}(\text{"El máximo es "}, \langle A \rangle.\text{máx}); \}$$

con el objetivo de escribir cuál es el mayor entero de la lista. Por ejemplo, para la lista $3,45,8$ el máximo es 45.

Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor del atributo **máx** de $\langle A \rangle$. Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

Nota: puedes serte útil la función *nd* que te da el número de dígitos de un número entero.

PREGUNTA III-12

(EXAMEN: 14 DE SEPTIEMBRE DE 2007)

Sea G la siguiente gramática, que genera listas de identificadores separados por comas:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle A \rangle \\ \langle A \rangle &\rightarrow \text{letra coma letra } \langle A \rangle | \text{letra } \langle B \rangle \text{ letra } | \langle B \rangle \text{ coma } \langle B \rangle \\ \langle B \rangle &\rightarrow \text{letra } \langle B \rangle \text{ coma } \langle A \rangle \text{ letra } | \langle B \rangle \text{ coma } \langle A \rangle | \text{letra} \end{aligned}$$

donde **letra** es un terminal que representa una letra con valor **letra.v** y **coma** representa una coma. Supongamos que la regla inicial tiene la acción semántica siguiente:

$$\langle S \rangle \rightarrow \langle A \rangle \{ \text{escribe}(\text{"Hay "}, \langle A \rangle.\text{nm}, \text{" identificadores que comienzan por mayúscula"}); \}$$

que escribe el número de identificadores de la lista que comienzan por mayúscula.

Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor del atributo nm de $\langle A \rangle$. Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

Nota: puede serte útil la función *may* que devuelve cierto si el carácter que se le pasa es una letra mayúscula.

PREGUNTA III-13

(EXAMEN: 11 DE DICIEMBRE DE 2007)

Sea G la siguiente gramática, que genera expresiones formadas por enteros y operadores de suma:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle \text{Expr} \rangle \\ \langle \text{Expr} \rangle &\rightarrow \langle \text{Expr} \rangle \langle \text{Expr} \rangle | \text{díg} | \langle \text{Suma} \rangle \\ \langle \text{Suma} \rangle &\rightarrow \text{díg} + \text{díg} | \text{díg} \text{ díg} \end{aligned}$$

donde **díg** es un terminal que representa un dígito del uno al nueve con valor **díg.v**. Supongamos que la regla inicial tiene la acción semántica siguiente:

$$\langle S \rangle \rightarrow \langle \text{Expr} \rangle \{ \text{escribe}(\text{"La suma es "}, \langle \text{Expr} \rangle.\text{suma}); \}$$

con el objetivo de escribir cuál es la suma de la expresión. Por ejemplo, para la expresión $15+27$ la suma es 42.

Añade, *al final de cada una de las restantes reglas*, las acciones semánticas necesarias para que se calcule el valor del atributo suma de $\langle \text{Expr} \rangle$. Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

Nota: puede serte útil el operador \oplus que, dados dos enteros, los concatena como si fueran cadenas. Por ejemplo: $12 \oplus 84 = 1284$.

PREGUNTA III-14

(EXAMEN: 24 DE JUNIO DE 2008)

Supongamos que `escribeMayores` es una sentencia que escribe los elementos de una lista que son mayores que su primer parámetro. Por ejemplo, la sentencia

```
escribeMayores (2+3, (4; 5+6+7; 8; 2));
```

escribiría los valores 18 y 8.

La siguiente gramática modela la sentencia `escribeMayores`:

$\langle \text{Sentencia} \rangle \rightarrow \text{escribeMayores abreParéntesis} \langle \text{Expresión} \rangle \text{ coma} \langle \text{Lista} \rangle \text{ cierraParéntesis puntoYComa}$
 $\langle \text{Lista} \rangle \rightarrow \text{abreParéntesis} \langle \text{Expresión} \rangle \langle \text{MásExpresiones} \rangle \text{ cierraParéntesis}$
 $\langle \text{MásExpresiones} \rangle \rightarrow \text{puntoYComa} \langle \text{Expresión} \rangle \langle \text{MásExpresiones} \rangle | \lambda$
 $\langle \text{Expresión} \rangle \rightarrow \text{número} \langle \text{MásNúmeros} \rangle$
 $\langle \text{MásNúmeros} \rangle \rightarrow \text{suma número} \langle \text{MásNúmeros} \rangle | \lambda$

Añade a esta gramática (sin modificarla de ninguna forma) las acciones semánticas que ejecuten `escribeMayores`. Asume que los componentes de la categoría `número` son enteros y que su valor está en el atributo `v`. Sólo puedes utilizar atributos de tipo entero o lógico (no está permitido usar, por ejemplo, listas o variables globales). Además, no puedes utilizar como heredados atributos sintetizados y viceversa.

Indica para cada atributo si es heredado o sintetizado y qué representa.

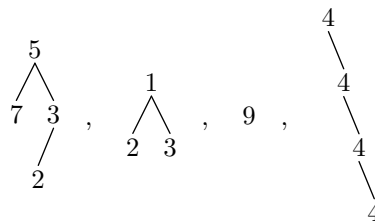
PREGUNTA III-15

(EXAMEN: 12 DE SEPTIEMBRE DE 2008)

Considera la siguiente gramática, que modela un bosque formado por una secuencia de árboles binarios separados por comas:

$\langle \text{Bosque} \rangle \rightarrow \langle \text{Arbin} \rangle \langle \text{MásArbin} \rangle$
 $\langle \text{MásArbin} \rangle \rightarrow \text{coma} \langle \text{Arbin} \rangle \langle \text{MásArbin} \rangle | \lambda$
 $\langle \text{Arbin} \rangle \rightarrow \text{abreParéntesis} (\text{entero} (\langle \text{Arbin} \rangle \langle \text{Arbin} \rangle))^? \text{ cierraParéntesis}$

Por ejemplo, el bosque



se puede representar mediante la cadena

```
(5(7()())(3(2)())), (1(2)(3)), (9), (4()(4()(4()(4))))
```

Añade acciones semánticas a la gramática anterior, sin modificarla, para que el esquema de traducción resultante escriba, mientras se lleva a cabo un análisis RLL(1) de la entrada, la profundidad de aquellos árboles del bosque que superen la profundidad de todos los anteriores.

Sólo puedes utilizar atributos de tipo lógico, entero o cadena (y no, por ejemplo, de tipo lista). Además, no puedes utilizar ningún objeto global ni tampoco puedes utilizar como heredados atributos sintetizados y viceversa.

Indica para cada atributo si es heredado o sintetizado y qué representa.

Nota: la profundidad de un árbol binario es la cantidad de nodos en el camino más largo desde la raíz hasta una hoja. En el ejemplo anterior, las profundidades son 3, 2, 1 y 4, respectivamente. Por lo tanto, la salida es 3 4, correspondiente a las profundidades de los árboles primero y último.

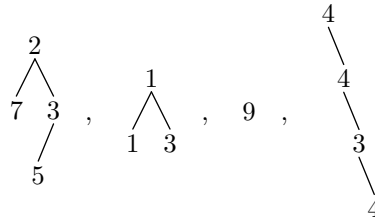
PREGUNTA III-16

(EXAMEN: 19 DE DICIEMBRE DE 2008)

Considera la siguiente gramática, que modela un bosque formado por una secuencia de árboles binarios separados por comas:

$$\begin{aligned} \langle \text{Bosque} \rangle &\rightarrow \langle \text{Arbin} \rangle \langle \text{MásArbin} \rangle \\ \langle \text{MásArbin} \rangle &\rightarrow \text{coma} \langle \text{Arbin} \rangle \langle \text{MásArbin} \rangle | \lambda \\ \langle \text{Arbin} \rangle &\rightarrow \text{abreParéntesis} (\text{entero} (\langle \text{Arbin} \rangle \langle \text{Arbin} \rangle)?)? \text{cierraParéntesis} \end{aligned}$$

Por ejemplo, el bosque



se puede representar mediante la cadena

$$(2(7()())(3(5)())), (1(1)(3)), (9), (4()(4()(3()(4))))$$

Añade acciones semánticas a la gramática anterior, sin modificarla, para que el esquema de traducción resultante escriba, al finalizar un análisis RLL(1) de la entrada, el mensaje “todos sí” si todos los árboles del bosque tienen orden de montículo y “alguno no” si alguno no lo tiene.

Sólo puedes utilizar atributos de tipo lógico y entero (y no, por ejemplo, de tipo lista). Además, no puedes utilizar ningún objeto global ni tampoco puedes utilizar como heredados atributos sintetizados y viceversa. Asume que los componentes de tipo **entero** tienen un atributo, *v*, que contiene el valor del entero (y que puede ser negativo).

Indica para cada atributo si es heredado o sintetizado y qué representa.

Nota: consideramos que un árbol binario tiene orden de montículo si todos los nodos tienen un valor menor o igual que cualquiera de sus hijos, si estos existen. En nuestro ejemplo, los tres primeros árboles lo cumplen y el último no. Por lo tanto, se emitiría el mensaje “alguno no”.

PREGUNTA III-17

(EXAMEN: 30 DE JUNIO DE 2009)

Resuelve los dos apartados siguientes teniendo en cuenta que no puedes modificar las gramáticas, que no está permitido utilizar variables globales, que todos los atributos que utilices han de ser de tipo entero o lógico y que no puedes utilizar como heredados atributos sintetizados y viceversa. Indica para cada atributo si es heredado o sintetizado y qué representa.

Apartado a)

Considera la siguiente gramática para expresiones formadas por sumas de productos de números:

$$\langle \text{Expresión} \rangle \rightarrow \text{núm (por núm)}^* (\text{ más núm (por núm)}^*)^*$$

Añade acciones semánticas a la gramática anterior para sintetizar, mientras se lleva a cabo un análisis RLL(1) de la entrada, el atributo *valor* de $\langle \text{Expresión} \rangle$ teniendo en cuenta las prioridades habituales de la suma y el producto (por ejemplo, $3+4*5$ tiene como resultado 23 y no 35). Asume que los componentes de la categoría **núm** son enteros positivos y que su valor está en el atributo *v*.

Apartado b)

Considera ahora la siguiente gramática, que modela secuencias formadas por una cantidad impar de expresiones separadas por dos puntos,

$$\begin{aligned} \langle \text{Sentencia} \rangle &\rightarrow \langle \text{Expresión} \rangle \langle \text{MásExpresiones} \rangle \\ \langle \text{MásExpresiones} \rangle &\rightarrow \mathbf{dp} \langle \text{Expresión} \rangle \mathbf{dp} \langle \text{Expresión} \rangle \langle \text{MásExpresiones} \rangle | \lambda \end{aligned}$$

donde el no terminal $\langle \text{Expresión} \rangle$ obtiene siempre un atributo sintetizado *valor* con el resultado de evaluar la expresión, según se ha visto en el apartado anterior.

Añade a la gramática las acciones semánticas necesarias para sintetizar, mientras se lleva a cabo un análisis RLL(1) de la entrada, el atributo *nésimo* de $\langle \text{Sentencia} \rangle$ con el resultado de evaluar la *n*-ésima expresión, contando desde 1 y siendo *n* el resultado de evaluar la primera. Si el valor de la primera expresión es mayor que el número de expresiones, debe devolverse el valor de la última.

Ten en cuenta los siguientes ejemplos:

Entrada	nésimo
3:3+4:5*8:3*2:4+5	40
12	12
5:3+4:1+2*5	11

PREGUNTA III-18

(EXAMEN: 11 DE SEPTIEMBRE DE 2009)

En un determinado lenguaje se trabaja con órdenes que permiten imprimir páginas individuales y/o rangos de páginas, con la restricción de que no se aceptan páginas individuales después de la aparición del primer rango.

La siguiente gramática incontextual modela este tipo de órdenes:

$$\begin{aligned} \langle \text{Imprimir} \rangle &\rightarrow \text{imprime número } \langle \text{MásPáginas} \rangle \\ \langle \text{MásPáginas} \rangle &\rightarrow \text{número } \langle \text{MásPáginas} \rangle | \text{rango número } \langle \text{MásRangos} \rangle | \lambda \\ \langle \text{MásRangos} \rangle &\rightarrow \text{número rango número } \langle \text{MásRangos} \rangle | \lambda \end{aligned}$$

Añade a la gramática las acciones semánticas necesarias para sintetizar, mientras se lleva a cabo un análisis RLL(1) de la entrada, el atributo *cp* de $\langle \text{Imprimir} \rangle$ con el resultado de contar la cantidad total de páginas a imprimir. Si en algún rango el número de la primera página es mayor que el de la última, ese rango aporta 0 páginas al total. Estos son algunos ejemplos:

Entrada	<i>cp</i>
imprime 5	1
imprime 5 7 5	3
imprime 12..15	4
imprime 12..15 3..10 7..5	12
imprime 5 7 5 12..15 3..10	15

Ten en cuenta que no puedes modificar la gramática, que no está permitido utilizar variables globales, que todos los atributos que utilices han de ser de tipo entero o lógico y que no puedes utilizar como heredados atributos sintetizados y viceversa. Indica para cada atributo si es heredado o sintetizado y qué representa.

Asume que los componentes de la categoría **número** son enteros positivos y que su valor está en el atributo *v*.

PREGUNTA III-19

(EXAMEN: 11 DE DICIEMBRE DE 2009)

La siguiente gramática

$$\begin{aligned} \langle A \rangle &\rightarrow \langle B \rangle \\ \langle B \rangle &\rightarrow \langle C \rangle (\text{op1 } \langle B \rangle)? \\ \langle C \rangle &\rightarrow \text{entero } \langle D \rangle \\ \langle D \rangle &\rightarrow \text{op2 } \langle C \rangle \\ \langle D \rangle &\rightarrow \lambda \end{aligned}$$

modela expresiones construibles utilizando números y dos operadores, uno asociado al componente léxico **op1** y otro, más prioritario, asociado al componente léxico **op2**. Ambos operadores son binarios, infijos, asociativos por la izquierda y con resultado y operandos de tipo entero. Posiblemente tú escribirías una gramática diferente para modelar eso; no obstante, este ejercicio consiste en trabajar con esa versión.

Añade a la gramática las acciones semánticas necesarias para sintetizar, mientras se lleva a cabo un análisis LL(1) de la entrada, el atributo *resultado* de $\langle A \rangle$ con el resultado de evaluar la expresión.

Asume que los componentes de la categoría **entero** son enteros positivos cuyo valor está en el atributo *valor*, y que dispones, en el lenguaje de las acciones semánticas, de los mismos operadores con esta representación: \oplus para **op1** y \otimes para **op2**.

Ten en cuenta que no puedes modificar la gramática, que no está permitido utilizar variables globales, que todos los atributos que utilices han de ser de tipo entero o lógico y que no puedes utilizar como heredados atributos sintetizados y viceversa. Indica para cada atributo si es heredado o sintetizado y qué representa.

PREGUNTA III-20

(EXAMEN: 28 DE JUNIO DE 2010)

Considera la siguiente gramática para expresiones formadas por sumas de productos de números:

$$\begin{aligned} \langle \text{Expresión} \rangle &\rightarrow \text{número } \langle \text{RestoExpresión} \rangle \\ \langle \text{RestoExpresión} \rangle &\rightarrow (\text{ más } | \text{ por }) \text{número } \langle \text{RestoExpresión} \rangle | \lambda \end{aligned}$$

Añade a la gramática las acciones semánticas necesarias para sintetizar, mientras se lleva a cabo un análisis RLL(1) de la entrada, el atributo v de $\langle \text{Expresión} \rangle$ con el resultado de evaluar la expresión, siguiendo las reglas habituales de prioridad. Por ejemplo, para $0*1*2+3*4+5*6$ el valor de v es 42. Asume que los componentes de la categoría **número** son enteros y que su valor está en el atributo v .

Ten en cuenta que no puedes modificar la gramática, que no está permitido utilizar variables globales, que todos los atributos que utilices han de ser de tipo simple (entero, lógico o carácter) y que no puedes utilizar como heredados atributos sintetizados y viceversa. Indica para cada atributo si es heredado o sintetizado y qué representa.

PREGUNTA III-21

(EXAMEN: 10 DE SEPTIEMBRE DE 2010)

Considera la siguiente gramática de un sencillo lenguaje ficticio:

$$\begin{aligned} \langle \text{Sentencia} \rangle &\rightarrow \text{escribe } \langle \text{Expresión} \rangle \\ \langle \text{Expresión} \rangle &\rightarrow \langle \text{Término} \rangle \langle \text{RestoExpresión} \rangle \\ \langle \text{RestoExpresión} \rangle &\rightarrow \text{arroba } \langle \text{Término} \rangle \langle \text{RestoExpresión} \rangle | \lambda \\ \langle \text{Término} \rangle &\rightarrow (\langle \text{Prefijo} \rangle)^* \text{número} \\ \langle \text{Prefijo} \rangle &\rightarrow \text{almohadilla } | \text{circunflejo} \end{aligned}$$

El terminal **arroba** corresponde a un operador @ que es binario, infijo y asociativo *por la izquierda*. Los terminales **almohadilla** y **circunflejo** corresponden, respectivamente, a dos operadores # y ^, que son unarios, prefijos y más prioritarios que el operador @. Todos ellos tienen operandos y resultado de tipo entero.

Añade las acciones semánticas necesarias para que, al mismo tiempo que se realiza un análisis RLL(1) de la entrada, se calcule y escriba el resultado de evaluar la expresión. Por supuesto, debes tener en cuenta la asociatividad y prioridad de los operadores para obtener el resultado correcto.

Asume que los componentes de la categoría **número** son enteros cuyo valor está en el atributo v , y que dispones también de esos mismos operadores @, # y ^ en el lenguaje de las acciones semánticas.

Ten en cuenta que no puedes modificar la gramática, que no puedes utilizar variables globales, que todos los atributos que utilices han de ser de tipo entero o lógico (este requisito no afecta a las posibles variables locales) y que ningún atributo puede ser heredado en un sitio y sintetizado en otro. Indica para cada atributo si es heredado o sintetizado y qué representa.

IV

Análisis ascendente

PREGUNTA IV-1

(EXAMEN: 2 DE JUNIO DE 2001)

Indica, para cada uno de los siguientes pares de estados, si la gramática correspondiente tiene un conflicto SLR, si no podemos saberlo o si los estados son incorrectos. Justifica cada respuesta:

1)

$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$
$\langle B \rangle \rightarrow \langle A \rangle \langle B \rangle c \cdot$

$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$
$\langle A \rangle \rightarrow \cdot ab$
$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$
$\langle B \rangle \rightarrow \cdot \langle C \rangle$
$\langle B \rangle \rightarrow \cdot d$
$\langle C \rangle \rightarrow \cdot a$
$\langle C \rangle \rightarrow \cdot b$

2)

$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$
$\langle A \rangle \rightarrow ab \cdot$

$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$
$\langle A \rangle \rightarrow \cdot ab$
$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$
$\langle B \rangle \rightarrow \cdot \langle C \rangle$
$\langle B \rangle \rightarrow \cdot d$
$\langle C \rangle \rightarrow \cdot a$
$\langle C \rangle \rightarrow \cdot \langle D \rangle$

3)

$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$
$\langle A \rangle \rightarrow ab \cdot$

$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$
$\langle A \rangle \rightarrow \cdot ab$
$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$
$\langle B \rangle \rightarrow \cdot \langle C \rangle$
$\langle B \rangle \rightarrow \cdot d$
$\langle C \rangle \rightarrow \cdot a$
$\langle C \rangle \rightarrow \cdot b$

4)

$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$
$\langle A \rangle \rightarrow ab \cdot$

$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$
$\langle A \rangle \rightarrow \cdot ab$
$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$
$\langle B \rangle \rightarrow \cdot \langle C \rangle$
$\langle B \rangle \rightarrow \cdot d$
$\langle C \rangle \rightarrow \cdot a$
$\langle C \rangle \rightarrow \cdot$

PREGUNTA IV-2

(EXAMEN: 12 DE JULIO DE 2001)

Sea G una gramática con dos no terminales $\langle A \rangle$ y $\langle B \rangle$ tales que hay un terminal a que es primero de ambos. ¿Cuáles de las siguientes afirmaciones son ciertas? Justifica las respuestas.

- Si existen las reglas $\langle C \rangle \rightarrow \langle A \rangle \alpha \langle B \rangle \beta$, la gramática no puede ser LL(1).
- Si existe la regla $\langle C \rangle \rightarrow \langle A \rangle \alpha \langle B \rangle \beta$, la gramática no puede ser LL(1).
- Si existen las reglas $\langle C \rangle \rightarrow \langle A \rangle \alpha \langle B \rangle \beta$, la gramática no puede ser LR(1).
- Si existe la regla $\langle C \rangle \rightarrow \langle A \rangle \alpha \langle B \rangle \beta$, la gramática no puede ser LR(1).

PREGUNTA IV-3

(EXAMEN: 18 DE SEPTIEMBRE DE 2001)

Construye la tabla de análisis SLR para la gramática siguiente:

$$\begin{aligned} \langle E \rangle &\rightarrow \langle E \rangle + \langle T \rangle \langle T \rangle \\ \langle T \rangle &\rightarrow \langle T \rangle \langle F \rangle \langle F \rangle \\ \langle F \rangle &\rightarrow \langle F \rangle * \langle a \rangle \langle b \rangle \end{aligned}$$

Utilízala para analizar la cadena $ab+ab*$.

PREGUNTA IV-4

(EXAMEN: 1 DE JUNIO DE 2002)

Dada la siguiente gramática:

$$\begin{aligned} \langle E \rangle &\rightarrow \langle E \rangle + \langle T \rangle \langle T \rangle \\ \langle T \rangle &\rightarrow \langle T \rangle * \langle F \rangle \langle F \rangle \\ \langle F \rangle &\rightarrow \text{id} \langle \text{cte} \mid \langle \text{cte} \rangle \langle \text{cte} \rangle \mid \langle \text{cte} \rangle \langle E \rangle \end{aligned}$$

escribe las tablas de acciones y sucesores correspondientes a un analizador SLR. Resuelve los conflictos que se plantean de modo que una expresión con paréntesis sin cerrar se interprete como si esos paréntesis estuvieran cerrados al final. Es decir, de modo que $\text{id}*(\text{cte}+\text{id}*\text{cte}$ se interprete como $\text{id}*(\text{cte}+\text{id}*\text{cte})$.

PREGUNTA IV-5

(EXAMEN: 2 DE SEPTIEMBRE DE 2002)

Sea la gramática G_n con las siguientes reglas:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle A_1 \rangle \\ \langle A_1 \rangle &\rightarrow \langle A_2 \rangle \langle A_2 \rangle \\ \langle A_2 \rangle &\rightarrow \langle A_3 \rangle \langle A_3 \rangle \langle A_3 \rangle \\ &\dots \\ \langle A_{n-1} \rangle &\rightarrow \overbrace{\langle A_n \rangle \dots \langle A_n \rangle}^n \\ \langle A_n \rangle &\rightarrow \mathbf{a} \end{aligned}$$

Se pide:

- Número de estados del automata de prefijos viables de la gramática G_n .
- Decidir si la gramática G_n es o no SLR.
- Lenguaje generado por G_n .

Justifica las respuestas.

PREGUNTA IV-6

(EXAMEN: 17 DE DICIEMBRE DE 2002)

Sea G una gramática en la que la sentencia x tiene la siguiente derivación:

$$\langle S \rangle \xRightarrow{*} \alpha\beta_1\gamma \Rightarrow \alpha\beta_2\gamma \Rightarrow \dots \Rightarrow \alpha\beta_n\gamma \xRightarrow{*} x$$

¿Puede ser G una gramática LR(1) en los siguientes casos?

- Si existen i y j diferentes tales que $\beta_i = \beta_j$.
- Si la derivación anterior es una derivación canónica por la izquierda.

Justifica las respuestas.

PREGUNTA IV-7

(EXAMEN: 31 DE MAYO DE 2003)

Sea $G_n = (N, \Sigma, P, \langle A_1 \rangle)$ una gramática incontextual con:

- $N = \{\langle A_1 \rangle, \dots, \langle A_n \rangle\}$,
- $\Sigma = \{a_1, \dots, a_n\}$

y las reglas:

$$\begin{aligned} \langle A_1 \rangle &\rightarrow \langle A_2 \rangle \langle A_2 \rangle | a_1 \\ \langle A_2 \rangle &\rightarrow \langle A_3 \rangle \langle A_3 \rangle | a_2 \\ &\dots \\ \langle A_n \rangle &\rightarrow a_n \end{aligned}$$

Se pide:

- Calcular el número de estados del autómata de prefijos viables de G_n y determinar si la gramática es SLR.
- Ídem en el caso de que existan exactamente un i y un j tales $a_i = a_j$ e $i < j$.

PREGUNTA IV-8

(EXAMEN: 30 DE JUNIO DE 2003)

Sea G una gramática, ya aumentada, con n no terminales, tal que la parte derecha más larga de una regla no tiene más de m símbolos. Demuestra:

- Si G es LR(0), el autómata de prefijos viables tiene al menos $m + n$ estados.
- Si G es SLR, el autómata de prefijos viables tiene al menos $\max(m, n)$ estados.

Nota: el estado generado por el ítem $\langle S \rangle \rightarrow \langle S \rangle \cdot$ se cuenta en el caso del autómata LR(0) pero no para el caso SLR. ($\langle S \rangle$ es el no terminal añadido para aumentar la gramática y $\langle S \rangle$ el no terminal inicial de la gramática original).

PREGUNTA IV-9

(EXAMEN: 9 DE SEPTIEMBRE DE 2003)

Escribe una gramática SLR con símbolo inicial $\langle A \rangle$ y tal que uno de los estados de su autómata de prefijos viables contenga, entre otros, los ítems: $\langle A \rangle \rightarrow \langle B \rangle a \cdot \langle C \rangle$, $\langle B \rangle \rightarrow a \cdot \langle E \rangle$ y $\langle C \rangle \rightarrow a \cdot \langle F \rangle$. Escribe el autómata completo de tu gramática y demuestra que no tiene conflictos SLR.

PREGUNTA IV-10

(EXAMEN: 22 DE MAYO DE 2004)

Dadas las siguientes gramáticas para modelar el lenguaje formado por las cadenas de aes y bes alternadas que comienzan y terminan por a:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle A \rangle (\langle B \rangle \langle S \rangle)^* \\ \langle A \rangle &\rightarrow a \\ \langle B \rangle &\rightarrow b \end{aligned}$$

$$\begin{aligned} \langle S \rangle &\rightarrow ((\langle A \rangle \langle B \rangle)^* \langle S \rangle) | \lambda \\ \langle A \rangle &\rightarrow a \\ \langle B \rangle &\rightarrow b \end{aligned}$$

$$\begin{aligned} \langle S \rangle &\rightarrow ((\langle A \rangle \langle B \rangle \langle S \rangle)^* \\ \langle A \rangle &\rightarrow a \\ \langle B \rangle &\rightarrow b \end{aligned}$$

se pide, por cada gramática:

- Justificar si el lenguaje generado se corresponde o no con el esperado.
- Escribir la correspondiente tabla de análisis RLL(1).
- En caso de conflictos, comentar si se pueden resolver o no manipulando la tabla.

PREGUNTA IV-11

(EXAMEN: 21 DE JUNIO DE 2004)

Sea G la siguiente gramática:

$$\begin{aligned} \langle S \rangle &\rightarrow a \langle B \rangle \langle S \rangle | b \langle A \rangle \langle S \rangle | \lambda \\ \langle A \rangle &\rightarrow a | b \langle A \rangle \langle A \rangle \\ \langle B \rangle &\rightarrow b | a \langle B \rangle \langle B \rangle \end{aligned}$$

Se pide:

- Construir el autómata de prefijos viables y la tabla de acciones SLR.
- Si la tabla no tiene conflictos, mostrar cómo se analizaría y qué árbol de derivación tendría la cadena **aabbba**.
- Si la tabla tiene conflictos, explicar si se deben a que la gramática es ambigua o a que necesitaríamos un método de análisis más potente.

PREGUNTA IV-12

(EXAMEN: 20 DE JUNIO DE 2005)

Sea G una gramática SLR y P el conjunto de todos los caminos en su autómata de prefijos viables que partan del estado inicial y lleguen hasta un estado que tiene asociada la acción de reducir con, al menos, un símbolo de entrada.

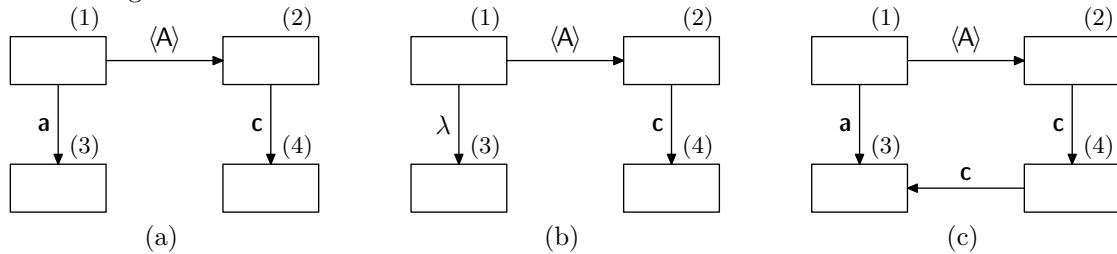
Demuestra la verdad o falsedad de las siguientes afirmaciones:

- Para todo $p \in P$, existe un prefijo de p que es parte derecha de alguna regla.
- Para todo $p \in P$, existe un sufijo de p que es parte derecha de alguna regla.
- Si existe un número n tal que todos los $p \in P$ tienen longitud inferior a n , entonces el lenguaje generado por G es finito.

PREGUNTA IV-13

(EXAMEN: 8 DE SEPTIEMBRE DE 2005)

Para cada uno de los siguientes autómatas

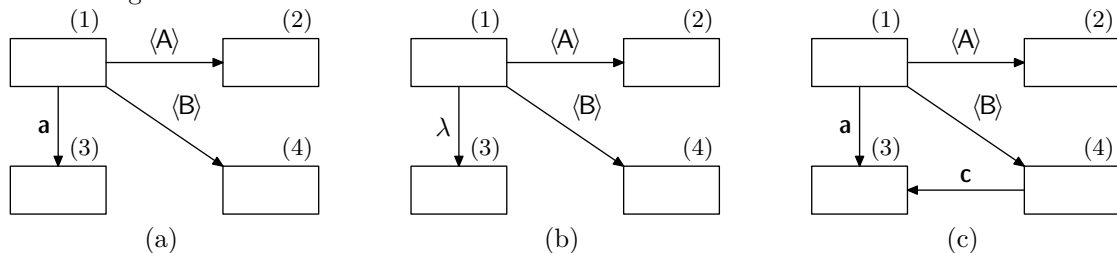


escribe una gramática SLR cuyo autómata de prefijos viables se corresponda con él o demuestra que no es posible.

PREGUNTA IV-14

(EXAMEN: 16 DE SEPTIEMBRE DE 2005)

Para cada uno de los siguientes autómatas



escribe una gramática SLR cuyo autómata de prefijos viables se corresponda con él o demuestra que no es posible.

PREGUNTA IV-15

(EXAMEN: 24 DE JUNIO DE 2006)

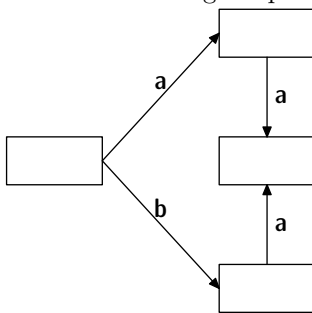
¿Puede un estado del autómata de prefijos viables de una gramática SLR contener simultáneamente los ítems $\langle A \rangle \rightarrow \cdot \langle A \rangle \langle A \rangle$, $\langle A \rangle \rightarrow \langle A \rangle \cdot \langle A \rangle$ y $\langle A \rangle \rightarrow \langle A \rangle \langle A \rangle \cdot$? ¿Y los ítems $\langle B \rangle \rightarrow \cdot \langle A \rangle \langle A \rangle$, $\langle B \rangle \rightarrow \langle A \rangle \cdot \langle A \rangle$ y $\langle B \rangle \rightarrow \langle A \rangle \langle A \rangle \cdot$?

Para cada pregunta, si es imposible, demuéstalo; si es posible, escribe una gramática en la que se cumpla.

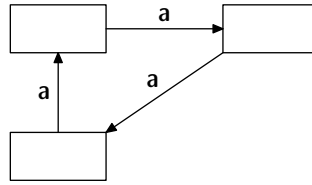
PREGUNTA IV-16

(EXAMEN: 15 DE SEPTIEMBRE DE 2006)

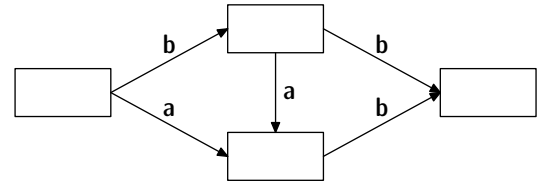
A continuación hay tres porciones de otros tantos autómatas de prefijos viables. Por cada uno, escribe una gramática SLR cuyo autómata contenga la parte mostrada o demuestra que ello no es posible.



(a)



(b)



(c)

V

**Preguntas sobre
análisis ascendente y
descendente**

PREGUNTA V-1

(EXAMEN: 1 DE JUNIO DE 2002)

Di si las siguientes afirmaciones son o no ciertas. Justifica tus respuestas.

- Si una gramática RLL(1) contiene, entre otras, las reglas $\langle A \rangle \rightarrow a(\langle B \rangle)^*c$ y $\langle B \rangle \rightarrow b$, entonces $\langle B \rangle$ no es anulable.
- Si una gramática SLR contiene la regla $\langle A \rangle \rightarrow \langle A \rangle \langle A \rangle a$, entonces a no es primero de $\langle A \rangle$.
- Si una gramática contiene las reglas $\langle A \rangle \rightarrow ab$ y $\langle B \rangle \rightarrow ab$, no puede ser LL(1).
- Si un estado del autómata de prefijos viables de una gramática contiene los ítems $\langle A \rangle \rightarrow \langle B \rangle \cdot \langle C \rangle$ y $\langle C \rangle \rightarrow \cdot \langle A \rangle \langle D \rangle$ y la gramática tiene las reglas $\langle A \rangle \rightarrow a$ y $\langle C \rangle \rightarrow a$, entonces la gramática no es SLR.

PREGUNTA V-2

(EXAMEN: 1 DE SEPTIEMBRE DE 2004)

Responde, justificadamente, a las siguientes preguntas:

- ¿Puede ser SLR una gramática que tenga las reglas $\langle A \rangle \rightarrow a\langle B \rangle$ y $\langle A \rangle \rightarrow a$?
- Sea G una gramática que tiene las reglas $\langle A \rangle \rightarrow \langle B \rangle a$ y $\langle A \rangle \rightarrow a$:
 - ¿Qué podemos decir acerca de $\langle B \rangle$ si sabemos que G es LL(1)?
 - Si $\langle B \rangle$ es anulable, ¿puede G ser SLR?, ¿y LR(1)?

PREGUNTA V-3

(EXAMEN: 18 DE DICIEMBRE DE 2004)

Responde, justificadamente, a las siguientes preguntas:

- Sean G_1 y G_2 dos gramáticas LL(1) que tienen los mismos no terminales. Sean M_1 , M_2 sus respectivas tablas de análisis. Decimos que las tablas son similares si para todo no terminal $\langle A \rangle$ y todo terminal a (o marca de fin de entrada, $\$$), la entrada $M_1[\langle A \rangle, a]$ no está vacía si y sólo si no lo está la entrada $M_2[\langle A \rangle, a]$. ¿Pueden ser distintos $L(G_1)$ y $L(G_2)$ si M_1 y M_2 son similares?
- Sea G la gramática siguiente:

$$\begin{aligned}\langle A \rangle &\rightarrow a\langle A \rangle a \\ \langle A \rangle &\rightarrow \lambda\end{aligned}$$

- ¿Cuál es el lenguaje generado por G ?
- ¿Puedes demostrar que G no es ambigua?
- ¿Es G SLR?, ¿y LR(1)?

PREGUNTA V-4

(EXAMEN: 12 DE DICIEMBRE DE 2006)

Sean $G_1 = (N_1, \Sigma, P_1, \langle S_1 \rangle)$ y $G_2 = (N_2, \Sigma, P_2, \langle S_2 \rangle)$ dos gramáticas con $N_1 \cap N_2 = \emptyset$. Definimos las gramáticas unión, G_U , y concatenación, G_C , de la siguiente manera:

- $G_U = (N_1 \cup N_2 \cup \{\langle S_U \rangle\}, \Sigma, P_1 \cup P_2 \cup \{\langle S_U \rangle \rightarrow \langle S_1 \rangle \langle S_2 \rangle\}, \langle S_U \rangle)$.
- $G_C = (N_1 \cup N_2 \cup \{\langle S_C \rangle\}, \Sigma, P_1 \cup P_2 \cup \{\langle S_C \rangle \rightarrow \langle S_1 \rangle \langle S_2 \rangle\}, \langle S_C \rangle)$.

Donde $\langle S_C \rangle$ y $\langle S_U \rangle$ son dos nuevos terminales no incluidos en N_1 ni N_2 .

Supón que $L(G_1) \cap L(G_2) = \emptyset$. Para cada una de las afirmaciones siguientes, da un contraejemplo si es falsa o justifica su veracidad.

- Si G_1 y G_2 son LL(1), también lo es G_U .
- Si G_1 y G_2 son LL(1), también lo es G_C .
- Si G_1 y G_2 son SLR, también lo es G_U .
- Si G_1 y G_2 son SLR, también lo es G_C .

PREGUNTA V-5

(EXAMEN: 26 DE JUNIO DE 2007)

Sean $G = (N, \Sigma, P, \langle S \rangle)$ una gramática. Definimos la gramática con permutación de los no terminales $\langle A \rangle$ y $\langle B \rangle$ como la gramática $G_{\langle A \rangle | \langle B \rangle} = (N, \Sigma, P_{\langle A \rangle | \langle B \rangle}, \langle S \rangle)$ donde $P_{\langle A \rangle | \langle B \rangle}$ tiene las mismas reglas que P pero sustituyendo en la parte derecha cada $\langle A \rangle$ por una $\langle B \rangle$ y viceversa. Por ejemplo, la regla $\langle A \rangle \rightarrow \mathbf{a} \langle A \rangle \mathbf{b} \langle B \rangle$ se cambiaría por $\langle A \rangle \rightarrow \mathbf{a} \langle B \rangle \mathbf{b} \langle A \rangle$.

Por otro lado, definimos la gramática con reduplicación del terminal \mathbf{a} como la gramática $G_{\mathbf{a}^2} = (N, \Sigma, P_{\mathbf{a}^2}, \langle S \rangle)$ donde $P_{\mathbf{a}^2}$ tiene las mismas reglas que P pero sustituyendo en la parte derecha cada \mathbf{a} por \mathbf{aa} . Por ejemplo, la regla $\langle A \rangle \rightarrow \mathbf{a} \langle A \rangle \mathbf{b} \langle B \rangle$ se cambiaría por $\langle A \rangle \rightarrow \mathbf{aa} \langle A \rangle \mathbf{b} \langle B \rangle$.

Para cada una de las afirmaciones siguientes, da un contraejemplo si es falsa o justifica su veracidad:

- Si G es LL(1), entonces $G_{\langle A \rangle | \langle B \rangle}$ es LL(1).
- Si G es SLR, entonces $G_{\langle A \rangle | \langle B \rangle}$ es SLR.
- Si G es LL(1), entonces $G_{\mathbf{a}^2}$ es LL(1).
- Si G es SLR, entonces $G_{\mathbf{a}^2}$ es SLR.

PREGUNTA V-6

(EXAMEN: 14 DE SEPTIEMBRE DE 2007)

Sean $G = (N, \Sigma, P, \langle S \rangle)$ una gramática y δ una derivación $\langle S \rangle \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n$. Llamaremos *firma izquierda* de δ a la secuencia $X_1 X_2 \dots X_n$ formada por los primeros símbolos (terminales o no terminales) de cada una de las cadenas $\alpha_1, \alpha_2, \dots, \alpha_n$. Análogamente, llamaremos *firma derecha* de δ a la secuencia formada por los últimos símbolos de $\alpha_1, \alpha_2, \dots, \alpha_n$. Por ejemplo, para la gramática

$$\begin{aligned} \langle A \rangle &\rightarrow \langle B \rangle \langle C \rangle | \langle B \rangle \\ \langle B \rangle &\rightarrow \mathbf{a} | \mathbf{b} \\ \langle C \rangle &\rightarrow \mathbf{c} \langle B \rangle \end{aligned}$$

y la derivación $\langle A \rangle \Rightarrow \langle B \rangle \langle C \rangle \Rightarrow \mathbf{a} \langle C \rangle \Rightarrow \mathbf{a} \mathbf{c} \langle B \rangle \Rightarrow \mathbf{a} \mathbf{c} \mathbf{b}$, la firma izquierda es $\langle B \rangle \mathbf{a} \mathbf{a} \mathbf{a}$ y la firma derecha es $\langle C \rangle \langle C \rangle \langle B \rangle \mathbf{b}$.

Para cada una de las afirmaciones siguientes, da un contraejemplo si es falsa o justifica su veracidad:

1. Si G es LL(1), entonces existe un número n tal que la firma izquierda de cualquier derivación canónica por la izquierda tiene menos de n no terminales.
2. Si G es SLR, entonces existe un número n tal que la firma izquierda de cualquier derivación canónica por la izquierda tiene menos de n no terminales.
3. Si G es LL(1), entonces existe un número n tal que la firma derecha de cualquier derivación canónica por la derecha tiene menos de n no terminales.
4. Si G es SLR, entonces existe un número n tal que la firma derecha de cualquier derivación canónica por la derecha tiene menos de n no terminales.