

Apellidos, nombre:

DNI:

Instrucciones: Este enunciado y todos los folios usados deben entregarse al salir

Parte I: PREGUNTAS TIPO TEST. 30 %. Cada dos respuestas incorrectas anulan una correcta.

1. Las restricciones contextuales de los lenguajes de programación
 - a) suelen formar parte de la *semántica* de dichos lenguajes, en la que se usa la *tabla de símbolos*.
 - b) se ignoran hasta que no se lleva a cabo la fase *optimización dependiente de la máquina*.
 - c) suelen resolverse implementando *autómatas linealmente acotados* que reconocen *lenguajes sensibles al contexto*.
2. El uso de un intérprete es adecuado cuando...
 - a) se espera que el programa se ejecute frecuentemente.
 - b) se espera que cada instrucción se ejecute una vez.
 - c) las instrucciones tienen formatos complicados.
3. El arranque pleno
 - a) se basa en el uso de un compilador portable.
 - b) se basa en el uso de un compilador cruzado.
 - c) facilita el mantenimiento de un compilador, puesto que se consigue que el lenguaje de implementación coincida con el lenguaje fuente, aunque implica la escritura completa de dicho compilador.
4. El análisis de léxico
 - a) suele funcionar como un módulo independiente del compilador, devolviendo un fichero con la lista de tokens requeridos en el análisis sintáctico.
 - b) suele estar integrado en el compilador, mediante una rutina que devuelve un token cada vez que lo solicita el analizador sintáctico.
 - c) se resuelve usando *autómatas linealmente acotados* para reconocer las expresiones regulares y devolver los códigos de token.
5. Una *gramática ambigua*
 - a) no puede ser LL ni LR.
 - b) puede ser LL pero no LR.
 - c) puede ser LR pero no LL.
6. Elige la opción correcta:
 - a) Una *gramática LL(1)* es *no ambigua*, λ -libre y está *factorizada*.
 - b) Una *gramática LL(1)* es *no ambigua*, *no recursiva por la izquierda* y está *factorizada*.
 - c) Una *gramática no ambigua*, *no recursiva por la izquierda* y *factorizada* es una *gramática LL(1)*.
7. En la gestión de error en modo pánico de un analizador LL(1),
 - a) cuando ocurre un error en el cual $M[A, b] = \emptyset$, siendo A el símbolo de la cima de la pila y b el símbolo de la entrada, si $b \in SIGUIENTE(A)$, se desapila A.
 - b) cuando ocurre un error en el cual $M[A, b] = \emptyset$, siendo A el símbolo de la cima de la pila y b el símbolo de la entrada, se descarta el símbolo b de la entrada.
 - c) cuando ocurre un error en el cual $M[A, b] = \emptyset$, siendo A el símbolo de la cima de la pila y b el símbolo de la entrada, se continúa con el análisis.
8. Podemos afirmar que una gramática es LL(1)
 - a) si no es recursiva por la izquierda, ni ambigua ni está factorizada.
 - b) si es LR.
 - c) sii $\forall A \in V_N$, si existen producciones $A \rightarrow \alpha$ y $A \rightarrow \gamma$, entonces $predict(A \rightarrow \alpha) \cap predict(A \rightarrow \gamma) = \emptyset$.

9. Podemos afirmar que una gramática es LALR
- si no es recursiva por la izquierda, ni ambigua ni está factorizada.
 - si es SLR.
 - si no es ambigua.
10. Dada la siguiente gramática:
- $$\begin{array}{lcl}
 S & \rightarrow & S A \\
 & | & A \\
 A & \rightarrow & id = L ; \\
 L & \rightarrow & id \\
 & | & L = L
 \end{array}$$
- elegir, de entre los siguientes, el conjunto de *items válidos* para el *prefijo viable* $id =$
- $\{[A \rightarrow id = \cdot L ;], [L \rightarrow \cdot id], [L \rightarrow \cdot L = L]\}$
 - $\{[S \rightarrow A \cdot]\}$
 - $\{[A \rightarrow id = L \cdot;], [L \rightarrow L \cdot = L]\}$
11. Dada la misma gramática de la pregunta anterior, el pivote de la forma sentencial $id = L = L = id$; es
- $id = L = L = \underline{id}$;
 - $id = \underline{L} = \underline{L} = id$;
 - $id = L = \underline{L} = id$;
12. Si una gramática es S-Atribuida,
- se pueden evaluar sus atributos al mismo tiempo que se realiza el análisis sintáctico LR.
 - las acciones semánticas no se pueden llevar a cabo al mismo tiempo que se realiza un análisis sintáctico LR.
 - los atributos heredados se calcularán en función de los atributos sintentizados.
13. Si una gramática es L-Atribuida
- también es S-Atribuida.
 - siempre vamos a poder evaluar sus atributos al mismo tiempo que realizamos un análisis LL.
 - el cálculo de un atributo heredado de un símbolo X puede depender de los atributos de su hermano derecho.
14. La comprobación de tipos
- siempre se realiza de forma estática, puesto que la realiza el compilador.
 - siempre se realiza de manera dinámica, pues depende de la entrada del programa.
 - dependiendo del lenguaje se puede hacer de manera dinámica o estática.
15. La conversión de tipos
- siempre se realiza con pérdida de información.
 - puede realizarse junto con las reglas del comprobación de tipos.
 - nunca está motivada por el programador, sino que es una tarea exclusiva del compilador.

Parte II: PREGUNTAS CORTAS. 10 %.

1. Dada la gramática G con el siguiente conjunto de producciones:

$$\begin{aligned}T &\rightarrow B\ C \\B &\rightarrow int \mid float \\C &\rightarrow [\ num \]\ C \mid \lambda\end{aligned}$$

que genera el lenguaje para declarar arrays en C, enumerar los tokens de G, indicando cuales tendrían atributo asociado en caso de que usáramos la gramática para realizar un compilador. Finalmente, dada la entrada

int [45] [2]

indicar qué información proporcionaría el analizador léxico al sintáctico.

2. Supongamos que hemos calculado la colección LR(0) y la tabla SLR para la siguiente gramática:

$$E \rightarrow E \wedge E \mid E \vee E \mid id$$

de modo que los conjuntos I_5 e I_6 contienen los siguientes items:

$$\begin{aligned}I_5 &= \{E \rightarrow E \wedge E\bullet, E \rightarrow E \bullet \wedge E, E \rightarrow E \bullet \vee E\} \\I_6 &= \{E \rightarrow E \bullet \wedge E, E \rightarrow E \vee E\bullet, E \rightarrow E \bullet \vee E\}\end{aligned}$$

produciéndose en la tabla SLR los siguientes conflictos:

ESTADO	accion		...
	\wedge	\vee	...
...	
5	$r1/d3$	$r1/d4$...
6	$r2/d3$	$r2/d4$...

Resolverlos dando mayor prioridad al operador \wedge que al \vee y considerando que ambos son asociativos por la izquierda.

Parte III: PROBLEMAS. 60 %

B) Dada la gramática G , con $V_T = \{id, =, ;\}$, $V_N = \{S, A, L\}$, símbolo inicial S y el siguiente conjunto P de producciones:

$$\begin{array}{lcl} S & \rightarrow & S A \\ & | & A \\ A & \rightarrow & id = L ; \\ L & \rightarrow & id \\ & | & L = L \end{array}$$

responder a las siguientes cuestiones:

1. (1 punto) Decir, justificando las respuestas, y sin construir ninguna tabla de análisis, si G es:

- a) Propia.
- b) Ambigua.
- c) LR-Canónica(1).
- d) LALR(1).
- e) SLR(1).
- f) LL(1).

En caso de que G no cumpla alguna de estas propiedades, dar todas las razones que se conozcan para justificarlo.

2. (1.25 puntos) Construir los conjuntos *PRIMERO* y *SIGUIENTE* para cada no terminal, y *predict* para cada regla. Construir la tabla LL.
3. (1.5 puntos) Construir la colección LR(0) y la tabla de análisis SLR. Modificar esta tabla para conseguir que el operador de asignación ($=$) sea asociativo por la derecha. Si usáramos *Bison* para generar la tabla, ¿cómo podría conseguirse esa asociatividad por la derecha?
4. (0.75 puntos) Simular el algoritmo de análisis ascendente predictivo no recursivo para las cadenas de entrada $w_1 \equiv a = b = c = d$; y $w_2 \equiv a == b$;, realizando recuperación en modo pánico en caso de error. Comenzar la simulación en ambos casos sustituyendo cada lexema por el código de token correspondiente.
5. (1.5 puntos) Dar la definición dirigida por la sintaxis para que, suponiendo que disponemos de una tabla de símbolos con los identificadores ya declarados, cada uno con su tipo correspondiente, comprobar que la asignación se hace entre tipos compatibles. Para ello, suponemos que cada identificador dispone de un atributo entrada que apunta a su posición en dicha tabla de símbolos, y que podemos hacer uso de una función `buscaTipo(apTS)` que devuelve el tipo del identificador apuntado por `apTS`. Indicar los atributos necesarios para cada símbolo, el tipo de cada uno de ellos (sintetizado o heredado) y las acciones semánticas para cada regla. Finalmente, evaluar el árbol de derivación asociado a la cadena de entrada w_1 de la pregunta 4., suponiendo que todos los identificadores son de tipo entero ¿Es esta gramática S-atribuida? ¿Y L-Atribuida? Justificar las respuestas.