Ejercicios de examen: primer parcial

1. (1 pto) Calcular un autómata finito determinista que reconozca el lenguaje generado por la expresión regular $(a + d)c^*$. Utilizar el algoritmo de Berry & Sethi. Indicar, explícitamente todos los cálculos necesarios.

2. (1 pto) Calcular un autómata finito determinista que reconozca el lenguaje generado por la expresión regular $(a + c^*)d$. Utilizar el algoritmo de Berry & Sethi. Indicar, explícitamente todos los cálculos necesarios.

3. (1.25 ptos) Calcular un autómata finito determinista que reconozca el lenguaje generado por la expresión regular $(a^* + c^*)d^*$. Utilizar el algoritmo de Berry & Sethi. Indicar, explícitamente todos los cálculos necesarios.

4. (1.25 ptos) Diseñar una gramática $\mathcal{G}$ que verifique:
   (a) (0.5 ptos) Que sea fuertemente $LL(1)$
   (b) (0.75 ptos) Que no sea $LL(1)$

5. (1 pto) Razonar la verdad o falsedad de la afirmación siguiente, dada $\mathcal{G}$ una gramática independiente del contexto.

   "Si $\mathcal{G}$ es LL(2), entonces $\mathcal{G}$ es fuertemente LL(2)"

6. (1 pto) Razonar la verdad o falsedad de las afirmaciones siguientes. Sea $\mathcal{G}$ una gramática cíclica[1] y $\mathcal{A}$ su autómata LR(1) asociado:
   (a) (0.5 ptos) $\mathcal{A}$ posee un número infinito de estados.
   (b) (0.5 ptos) $\mathcal{A}$ posee un número infinito de transiciones.

7. (1.25 ptos) Razonar la verdad o falsedad de las afirmaciones siguientes. Sea $\mathcal{G}$ una gramática:
   (a) (0.5 ptos) $\mathcal{G}$ puede ser ambigua y, sin embargo, su autómata SLR(1) no presentar conflictos.
   (b) (0.75 ptos) $\mathcal{G}$ puede presentar conflictos en su autómata SLR(1) y, sin embargo, no ser ambigua.

8. (1 pto) Razonar la verdad o falsedad de las afirmaciones siguientes. Sea $\mathcal{G}$ una gramática independiente del contexto:
   (a) (0.5 ptos) El autómata LALR(1) de $\mathcal{G}$ puede presentar conflictos y, sin embargo, $\mathcal{L}(\mathcal{G})$ ser un lenguaje no ambiguo.
   (b) (0.5 ptos) $\mathcal{G}$ puede presentar conflictos en su autómata LR(1) y, sin embargo, no ser ambigua.

9. (1 pto) Diseñar una gramática $\mathcal{G}$ que verifique:
   (a) (0.5 ptos) Que sea $LALR(2)$
   (b) (0.5 ptos) Que no sea $LALR(1)$

[1] esto es, existen entradas para las que el número de análisis sintácticos posibles es infinito.

10. (1 pto) Dada la gramática:

$$\begin{array}{llllll} 1) & A \to BC\,] & 3) & B \to [& 4) & C \to aA \\ 2) & A \to [\,] & & & 5) & C \to b \\ & & & & 6) & C \to A \end{array}$$

(a) (0.6 ptos) Calcular su tabla de precedencias, indicando los cálculos realizados.

	A	B	C	[	]	a	b	$
A								
B								
C								
[								
]								
a								
b								
$								

(b) (0.4 ptos) Construír el analizador de precedencia simple correspondiente a la tabla anterior y analizar la cadena $w = [a[a[b]]]$.

11. (1 pto) Dada la gramática:

$$\begin{array}{lll} E \to E + T & T \to T * F & F \to (E) \\ E \to T & T \to F & F \to a \end{array}$$

(a) (0.5 ptos) Construír su tabla de precedencias, indicando los cálculos realizados.

	E	T	F	+	*	(	)	a	$
E									
T									
F									
+									
*									
(									
)									
a									
$									

(b) (0.25 ptos) **Razonar** si es una gramática de precedencia simple o de precedencia débil.

(c) (0.25 ptos) Analizar la cadena "$a * (a + a)$", usando el algoritmo de precedencia simple o débil (según sea la gramática). No es necesario definir formalmente el analizador.

12. (1 pto) Dada una gramática $\mathcal{G} = \{N,\ \Sigma,\ P,\ S\}$ con $N = \{S, A\}$, $\Sigma = \{a, b, c, d, e, f\}$, y P $=$ $\{S \to UVWXYZ,\ S \to Y,\ A \to Y,\ A \to XYZ\}$, donde $U, V, W, X, Y, Z \in N \cup \Sigma$, y la siguiente tabla de precedencias:

	S	A	a	b	c	d	e	f	\$
S					$\gtrdot$ $\doteq$				$\gtrdot$
A				$\doteq$			$\doteq$		
a		$\lessdot$ $\doteq$						$\lessdot$	
b	$\doteq$		$\lessdot$			$\lessdot$			
c	$\doteq$		$\lessdot$			$\lessdot$			
d					$\gtrdot$				$\gtrdot$
e								$\doteq$	
f				$\gtrdot$			$\gtrdot$		
\$			$\lessdot$			$\lessdot$			

(a) (0.5 puntos) Construír las reglas de la gramática, razonando la respuesta.

(b) (0.5 puntos) Razonar si es una gramática de precedencia simple o de precedencia débil.
En caso de que no sea una gramática de precedencia simple, reescribirla para obtener una gramática de precedencia simple que genere el mismo lenguaje.