

1. Dada la gramática:

- (r1) $S \rightarrow B A$ (r4) $A \rightarrow \% B A$ (r7) $A \rightarrow \epsilon$
 (r2) $B \rightarrow D C$ (r5) $C \rightarrow \& D C$ (r8) $C \rightarrow \epsilon$
 (r3) $D \rightarrow (S)$ (r6) $D \rightarrow b$

se pide:

a) Demostred que la gramática es LL(1) y construid la tabla de análisis LL(1).

Cálculo de siguientes: $\text{sig}(S) = \{ \$,) \}$
 $\text{sig}(B) = \{ \%, \$,) \}$
 $\text{sig}(D) = \{ \&, \%, \$,) \}$
 $\text{sig}(C) = \{ \%, \$,) \}$
 $\text{sig}(A) = \{ \$,) \}$

Símbolos de anticipación para cada regla:

$\text{Primeros}(B A \oplus \text{sig}(S)) = \{ (, b \}$
 $\text{Primeros}(D C \oplus \text{sig}(B)) = \{ (, b \}$
 $\text{Primeros}((S) \oplus \text{sig}(D)) = \{ (\}$
 $\text{Primeros}(b \oplus \text{sig}(D)) = \{ b \}$
 $\text{Primeros}(\% B A \oplus \text{sig}(A)) = \{ \% \}$
 $\text{Primeros}(\epsilon \oplus \text{sig}(A)) = \text{sig}(A) = \{ \$,) \}$
 $\text{Primeros}(\& D C \oplus \text{sig}(C)) = \{ \& \}$
 $\text{Primeros}(\epsilon \oplus \text{sig}(C)) = \text{sig}(C) = \{ \%, \$,) \}$

de lo que se deduce que la gramática es LL(1).

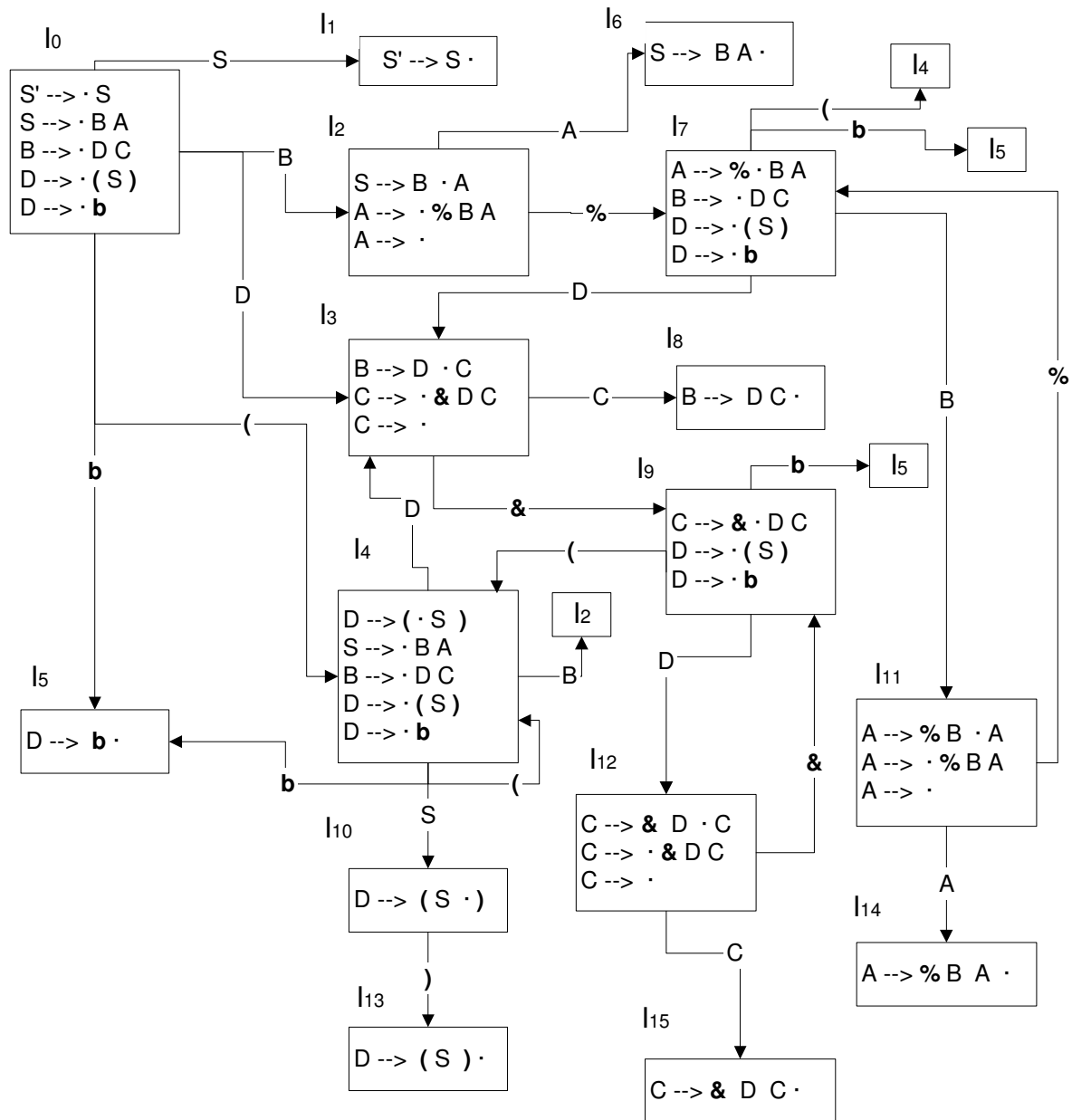
Tabla de análisis LL(1)

	()	%	&	b	\$
S	$S \rightarrow B A$				$S \rightarrow B A$	
B	$B \rightarrow D C$				$B \rightarrow D C$	
D	$D \rightarrow (S)$				$D \rightarrow b$	
C		$C \rightarrow \epsilon$	$C \rightarrow \epsilon$	$C \rightarrow \& D C$		$C \rightarrow \epsilon$
A		$A \rightarrow \epsilon$	$A \rightarrow \% B A$			$A \rightarrow \epsilon$
(pop					
)		pop				
%			pop			
&				pop		
b					pop	
\$						Aceptar

b) Obtened la traza de análisis sintáctico LL(1) para la cadena: ((b))

<i>(Pila</i>	<i>, Cadena de entrada,</i>	<i>Regla aplicada</i>	
S \$,	((b)) \$,	r1	-
- B A \$,	((b)) \$,	r2	-
- D C A \$,	((b)) \$,	r3	-
- (S) C A \$,	((b)) \$,	pop	-
- S) C A \$,	(b)) \$,	r1	-
- B A) C A \$,	(b)) \$,	r2	-
- D C A) C A \$,	(b)) \$,	r3	-
- (S) C A) C A \$,	(b)) \$,	pop	-
- S) C A) C A \$,	b)) \$,	r1	-
- B A) C A) C A \$,	b)) \$,	r2	-
- D C A) C A) C A \$,	b)) \$,	r3	-
- b C A) C A) C A \$,	b)) \$,	pop	-
- C A) C A) C A \$,)) \$,	r8	-
- A) C A) C A \$,)) \$,	r7	-
-) C A) C A \$,)) \$,	pop	-
- C A) C A \$,) \$,	r8	-
- A) C A \$,) \$,	r7	-
-) C A \$,) \$,	pop	-
- C A \$,	\$,	r8	-
- A \$,	\$,	r7	-
- \$,	\$,	Aceptar	-

c) Construid la colección de conjuntos de ítems LR(0) para esta gramática.



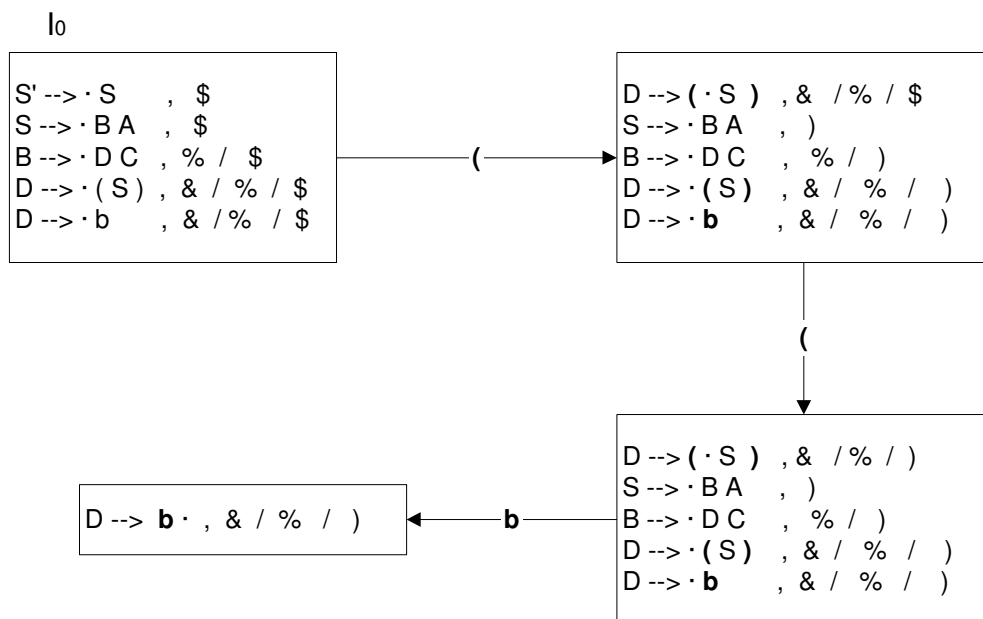
d) Obtened la derivación a derechas para la cadena del apartado b) y:

$S \rightarrow BA \rightarrow B \rightarrow DC \rightarrow D \rightarrow (S) \rightarrow (BA) \rightarrow (B) \rightarrow (DC) \rightarrow (D) \rightarrow ((S)) \rightarrow ((BA)) \rightarrow ((B)) \rightarrow ((DC)) \rightarrow ((D)) \rightarrow ((b))$

1) Indicad los pivotes de cada paso de la derivación.

$(S \rightarrow BA, 2), (A \rightarrow \epsilon, 1), (B \rightarrow DC, 2), (C \rightarrow \epsilon, 1), (D \rightarrow (S), 3), \dots$

2) Obtened (sólo) los estados necesarios de la colección de conjuntos de ítems LR(1) que permitan reconocer al prefijo viable $((b$.



3) Obtened la relación de los ítems válidos LR(1) para este prefijo viable.

$D \rightarrow b \cdot , \&$
 $D \rightarrow b \cdot , \%$
 $D \rightarrow b \cdot ,)$

Examen de Procesadores de Lenguajes

20 de junio de 2007

Solución a la 2ª pregunta

(3.5 ptos.) Diseñad un ETDS que genere código intermedio para el siguiente fragmento de una gramática independiente del contexto.

```
I  →  yacase E of L default I end
I  →  I ; I
I  →  exit
L  →  L ; cte : I
L  →  cte : I
L  →  else I
```

La instrucción **yacase** (*yet another case*) es similar a la del PASCAL: si la **cte** coincide con la expresión **E** debe ejecutar la secuencia de instrucciones **I** asociada y terminar la búsqueda en la lista de ítems. La instrucción **exit** supone la salida inmediata de la instrucción **yacase**.

I ⇒ yacase E of L default I end	L.h:=E.pos; Completalans(L.fin, Ω); CompletaLans(Fusionalans(L.exit, I ¹ .exit), Ω); I.exit:=nil;
I ⇒ I ; I	I.exit:=Fusionalans(I ¹ .exit, I ² .exit);
I ⇒ exit	I.exit:=Crealans(Ω); Emite(goto ⊗);
L ⇒ cte : I	L.false:=Crealans(Ω); Emite(if L.h≠cte.val goto ⊗); L.fin:=Crealans(Ω); Emite(goto ⊗); Completalans(L.false, Ω); L.exit:=I.exit;
L ⇒ else I	L.fin:=Crealans(Ω); Emite(goto ⊗); L.exit:=I.exit
L ⇒ L ; cte : I	L ¹ .h:=L.h; L.false:=Crealans(Ω); Emite(if L.h≠cte.val goto ⊗); L.fin:=Fusionalans(L ¹ .fin, Crealans(Ω)); Emite(goto ⊗); Completalans(L.false, Ω); L.exit:=Fusionalans(L ¹ .exit, I.exit);