ACLARACIONES:

Me surgió algunos problemas a la hora de hacer los regex, si lo probaba individualmente cumplía con los requisitos del ejercicio 1, el problema estaba cuando quería probarlos a todos en el flex. No podía entender por qué no funcionaba. El problema era que detectaba el mismo patrón en 2 regex como, por ejemplo, si tenemos las sentencia "Var1" esta sentencia es validad para id y idprod (REGEX INICIALES) que son los que utilizo en las reglas. Al final resolví este inconveniente haciendo regex que tenga algo distintivo que el otro regex no tenga, como por ejemplo el "@" para el caso de los nombres de valores o "_" para el caso de los nombres de propietarios.

REGEX INICIALES:

digito	[0-9]
letra	[a-zA-Z]
id	({letra})({letra}|{digito})*
idprod	({letra})({letra}|{digito}|_+{letra}|_+{digito})*

Ejercicio 1:

REGEX:

digito	[0-9]
letra	[a-zA-Z]
id	({letra})({letra}|{digito})*
idprod	({id})(_+({letra}|{digito})+)+
idvalor	(@)({id})

Ejercicio 2:

P → INSERT INTO idprod . id (LISTAC) VALUES (LISTAV)
P → INSERT INTO id . id (LISTAC) VALUES (LISTAV)
LISTAC → id
LISTAC → LISTA , id
LISTAV → id
LISTAV → LISTA , id
LISTAV → idvalor
LISTAV → LISTA , idvalor

Ejercicio 3.a:

EJERCICIO NRO 3.a

	id	(	)	asigna	take	+	*	cte	;	[	]	$	Asig	Lista	O
0	D2												1		
1												OK COMPILACION			
2				D3											
3					D4										
4		D5													
5						D6	D7								8
6								O --> +							
7								O --> *							
8									D9						
9								D10							
10									D11						
11										D12					
12	D13													14	
13	Lista --> id										Lista --> id				
14	D15										D16				
15	Lista --> Lista id										Lista --> Lista id				
16			D17												
17												Asig --> id asigna take (O; cte;[Lista])			

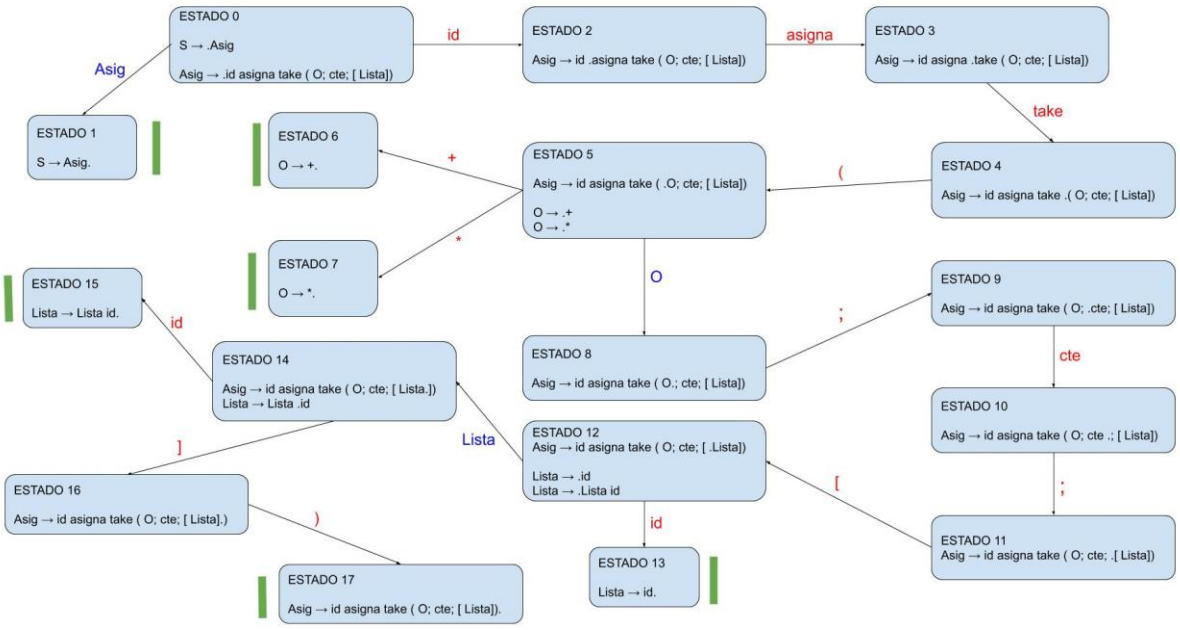
DESPLAZAMIENTO
ESTADO 2 (O, id)
ESTADO 3 (2, Asigna)
ESTADO 4 (3, take)
ESTADO 5 (4, ( )
ESTADO 6 (5, +)
ESTADO 7 (5, *)
ESTADO 9 (8, ;)
ESTADO 10 (9, cte)
ESTADO 11 (10, ;)
ESTADO 12 (11, [ )
ESTADO 13 (12, id)
ESTADO 15 (14, id)
ESTADO 16 (14,])
ESTADO 17 (16,))

GO TO
ESTADO 1 (0, Asig)
ESTADO 8 (5, O)
ESTADO 14 (12, Lista)

PRIMEROS
PRIM(S)=PRIM(Asig)={id}
PRIM(O)={id}
PRIM(Lista)={ * , + }

SIGUIENTES
sgt(S)=sgt(Asig)={$}
sgt(O)={ ; }
sgt(Lista)={] , id }

REDUCIR	
R0: ESTADO 1 S --> Asig.	(1; { $ })
R1: ESTADO 17 Asig --> id asigna take (O; cte; [Lista]).	(17; { $ })
R2: ESTADO 13 Lista --> id.	(13 ; {] , id})
R3: ESTADO 15 Lista --> Lista id.	(15; {] , ID})
R4: ESTADO 6 O --> +.	(6; { ; })
R5: ESTADO 7 O --> *.	(7; { ; })



Ejercicio 3.b:

EJERCICIO NRO 3.b CODIGO: a = TAKE (*; 3; [a b c])		
PILA	REGLA	LEXEMA
0		id asigna take (* ; cte ; [id id id]) $
0 id2		asigna take (* ; cte ; [id id id]) $
0 id2 asigna3		take (* ; cte ; [id id id]) $
0 id2 asigna3 take4		(* ; cte ; [id id id]) $
0 id2 asigna3 take4 (5		* ; cte ; [id id id]) $
0 id2 asigna3 take4 (5 *7	O --> *	; cte ; [id id id]) $
0 id2 asigna3 take4 (5 O8		; cte ; [id id id]) $
0 id2 asigna3 take4 (5 O8 ;9		cte ; [id id id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10		; [id id id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11		[id id id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12		id id id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 id13	Lista --> id	id id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14		id id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14 id15	Lista --> Lista id	id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14		id]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14 id15	Lista --> Lista id	]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14		]) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14]16		) $
0 id2 asigna3 take4 (5 O8 ;9 cte10 ;11 [12 Lista14]16)17	Asig --> id asigna take (O; cte;[Lista])	$
0 Asig1		$
COMPILACION OK		

Ejercicio 4:

EJERCICIO NRO 4 CODIGO: a = TAKE (*; *; [a b c])		
PILA	REGLA	LEXEMA
0		id asigna take (* ; * ; [id id id]) $
0 id2		asigna take (* ; * ; [id id id]) $
0 id2 asigna3		take (* ; * ; [id id id]) $
0 id2 asigna3 take4		(* ; * ; [id id id]) $
0 id2 asigna3 take4 (5		* ; * ; [id id id]) $
0 id2 asigna3 take4 (5 *7	O --> *	; * ; [id id id]) $
0 id2 asigna3 take4 (5 O8		; * ; [id id id]) $
0 id2 asigna3 take4 (5 O8 ;9		* ; [id id id]) $
ERROR DE COMPILACION: "Se esperaba una constante" en la tabla de parsing fila "9" columna "cte"		

CODIGO: a = TAKE (; 3; [a b c])		
PILA	REGLA	LEXEMA
0		id asigna take (; cte ; [id id id]) $
0 id2		asigna take (; cte ; [id id id]) $
0 id2 asigna3		take (; cte ; [id id id]) $
0 id2 asigna3 take4		(; cte ; [id id id]) $
0 id2 asigna3 take4 (5		; cte ; [id id id]) $
ERROR DE COMPILACION: "Se esperaba el operador '+' o '*' " en la tabla de parsing fila "5" columna "+" o fila "5" columna "*"		

Ejercicio 5:

Sí, es posible, esto lo podemos ver cuando armamos la tabla de parsing y en un campo de la tabla hay más de un desplazamiento (SHIFT) o reducir (REDUCE), esto lo que provoca es que para una misma sentencia haya más de un árbol de parsing produciéndose la ambigüedad.