	&&	(	)	*	++	,	--	.	;	;	boolean	cadena	for	function	id	if	input	int	numero	output	return	string	var	void	{	}	E(final de cadena)
A	A → lambda	A → (A')	A → lambda	A → lambda	·	A → lambda	A → lambda	A → lambda	·	A → lambda	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
A1	·	A1 → A2	A1 → lambda	·	·	·	·	·	·	·	·	A1 → A2	·	·	A1 → A2	·	·	·	A1 → A2	·	·	·	·	·	·	·	·
A2	·	A2 → R&A	A2 → lambda	·	·	·	·	·	·	·	·	A2 → R&A	·	·	A2 → R&A	·	·	·	A2 → R&A	·	·	·	·	·	·	·	·
A3	·	·	A3 → lambda	·	·	A3 → A2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
B	·	·	·	·	·	·	·	·	·	·	·	B → U	B → for (F1 ; R1 ; R2) { U }	·	B → U	B → if (R1) U	B → U	·	B → U	B → U	B → U	·	B → var T id C	·	·	·	·
C	·	·	·	·	C → (R)	·	·	C → ;	C → (R)	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
D	·	·	·	·	·	·	·	·	·	·	·	·	·	D → function void (S) { U }	·	·	·	·	·	·	·	·	·	·	·	·	·
D1	·	·	·	·	·	·	·	·	·	·	D1 → D2	·	·	·	·	·	·	D1 → D2	·	·	·	D1 → D2	·	D1 → void	·	·	·
D2	·	·	·	·	·	·	·	·	·	·	D2 → D2a	·	·	·	·	·	·	D2 → D2a	·	·	·	D2 → D2a	·	·	·	·	·
D2a	·	·	·	·	·	·	·	·	·	·	D2a → T id	·	·	·	·	·	·	D2a → T id	·	·	·	D2a → T id	·	·	·	·	·
D3	·	·	D3 → lambda	·	·	D3 → D3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
F1	·	·	·	·	·	·	·	F1 → lambda	·	·	·	·	·	·	F1 → int R	·	·	·	·	·	·	·	F1 → var T id L R	·	·	·	·
F2	·	·	F2 → lambda	·	·	·	·	·	·	·	·	·	·	·	F2 → int R	·	·	·	·	·	·	·	·	·	·	·	·
F	·	·	·	·	·	·	·	·	·	·	·	F → T	·	·	·	·	·	·	·	·	·	·	F → T	F → void	·	·	·
L	·	·	·	·	L → ++	·	·	·	L → --	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
G	G → &&	·	·	G → *	·	G → /	·	·	·	G → mo	·	·	·	·	·	·	·	·	G → R G	G → R G	·	·	G → R G	·	·	G → lambda	·
G1	·	·	·	·	·	·	·	·	·	·	·	G1 → R G	G → R G	·	G1 → R G	·	G → R G	·	·	·	G → R G	·	·	·	·	·	·
H	·	R → R1 R2	·	·	·	·	·	·	·	·	·	R → R1 R2	·	·	R → R1 R2	·	·	·	R → R1 R2	·	·	·	·	·	·	·	·
H1	·	R1 → (R)	·	·	·	·	·	·	·	·	·	R1 → numero	·	·	R1 → nulo	·	·	·	R1 → numero	·	·	·	·	·	·	·	·
R2	R2 → G R	·	R2 → lambda	R2 → G R	·	R2 → lambda	R2 → G R	R2 → lambda	·	R2 → G R	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
S	·	·	·	·	·	·	·	·	·	·	·	S → B S	S → B S	S → D S	S → B S	S → B S	S → B S	·	S → B S	S → B S	S → B S	·	S → B S	·	·	·	S → lambda