TABELLA DI FIRST PER G1

FIRST di ogni simbolo terminale è il simbolo terminale stesso: i.e. FIRST(atom) → { atom }

Id	FIRST0	FIRST1	FIRST2	FIRST3	FIRST4	FIRST5	FIRST6
Prog		let letrec	let letrec	let letrec	let letrec	let letrec	let letrec
Bind		var	var	var	var	var	var
X		and epsilon	and epsilon	and epsilon	and epsilon	and epsilon	and epsilon
Exp		lambda if	lambda if let letrec cons car cdr eq leq atom	lambda if let letrec cons car cdr eq leq atom	lambda if let letrec cons car cdr eq leq atom var exp_const (	lambda if let letrec cons car cdr eq leq atom var exp_const (	lambda if let letrec cons car cdr eq leq atom var exp_const (
ExpA				var exp_const (	var exp_const (	var exp_const (	var exp_const (
E1		epsilon	epsilon + -	epsilon + -	epsilon + -	epsilon + -	epsilon + -
T			var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (
T1		epsilon	epsilon * /	epsilon * /	epsilon * /	epsilon * /	epsilon * /
F		var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (
Y		(epsilon	(epsilon	(epsilon	(epsilon	(epsilon	(epsilon
OPA		+ -	+ -	+ -	+ -	+ -	+ -
OPM		* /	* /	* /	* /	* /	* /
OPP		cons car cdr eq leq atom	cons car cdr eq leq atom	cons car cdr eq leq atom	cons car cdr eq leq atom	cons car cdr eq leq atom	cons car cdr eq leq atom
Seq_Exp		epsilon	epsilon lambda if	epsilon lambda if let letrec cons car cdr eq leq atom	epsilon lambda if let letrec cons car cdr eq leq atom	epsilon lambda if let letrec cons car cdr eq leq atom var exp_const (	epsilon lambda if let letrec cons car cdr eq leq atom var exp_const (

Id	FIRST0	FIRST1	FIRST2	FIRST3	FIRST4	FIRST5	FIRST6
Seq_Var		var epsilon	var epsilon	var epsilon	var epsilon	var epsilon	var epsilon
Sep_Exp		, epsilon	, epsilon	, epsilon	, epsilon	, epsilon	, epsilon