TABELLA DI FOLLOW PER G0

Id	FOLLOW0	FOLLOW1	FOLLOW2	FOLLOW3	FOLLOW4	FOLLOW5	FOLLOW6	FOLLOW7
Prog	$	$	$ end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	$ end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	$ end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	$ end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	$ end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	$ end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)
Bind		in	in	in	in	in	in	in
X			in	in	in	in	in	in
Exp		end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)	end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in)
ExpA		)	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in

Id	FOLLOW0	FOLLOW1	FOLLOW2	FOLLOW3	FOLLOW4	FOLLOW5	FOLLOW6	FOLLOW7
E1			)	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in
T		+ -	+ -)	+ - end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	+ - end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	+ - end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	+ - end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	+ - end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in
T1			+ -	+ -)	+ -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	+ -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	+ -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	+ -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in
F		* /	* / + -	* / + -)	* / + -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	* / + -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	* / + -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	* / + -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in

Id	FOLLOW0	FOLLOW1	FOLLOW2	FOLLOW3	FOLLOW4	FOLLOW5	FOLLOW6	FOLLOW7
Y			* /	* / + -	* / + -)	* / + -) end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (	* / + -) end and then else end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in	* / + -) end and then else end and then else lambda if let letrec cons car cdr eq leq atom var exp_const (in
OPA		var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (
OPM		var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (	var exp_const (
OPP		(	(	(	(	(	(	(
Seq_ Exp		)	)	)	)	)	)	)
Seq_ Var		)	)	)	)	)	)	)