

	Let	Letrec	var	in	end	and	const	op	lambda	([])	::	nil	\$
Prog	Prog ::= let Bind in Exp end	Prog ::= letrec Bind in Exp end														
Bind			Bind ::= var = Exp X													
X				X ::= epsilon		X ::= and Bind										
Y	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= epsilon	Y ::= (Seq_Exp)	Y ::= epsilon		Y ::= epsilon		Y ::= epsilon	
Z	Z ::= Seq_Exp	Z ::= Seq_Exp	Z ::= Seq_Exp				Z ::= Seq_Exp	Z ::= Seq_Exp	Z ::= Seq_Exp				Z ::= Seq_Exp		Z ::= epsilon	Z ::= Seq_Exp
V													V ::= epsilon		V ::= ::Const list	
W			W ::= Seq Var											W ::= epsilon		
Exp	Exp ::= Prog	Exp ::= Prog	Exp ::= var Y				Exp ::= const	Exp ::= op (Seq_Exp)	Exp ::= lambda (Seq_Var) Exp				Exp ::= List			Exp ::= List
Seq_Exp	Seq_Exp ::= Exp Z	Seq_Exp ::= Exp Z	Seq_Exp ::= Exp Z				Seq_Exp ::= Exp Z	Seq_Exp ::= Exp Z	Seq_Exp ::= Exp Z				Seq_Exp ::= Exp Z			Seq_Exp ::= Exp Z
Seq_Var			Seq_Var ::= var W													
List													List ::= = [Const List]			List ::= = nil
Const_List							Const_List ::= const V						Const_List ::= List V			Const_List ::= List V