- Lispkit Compiler -

Grammatica LL(1) per l'analizzatore sintattico

Stefano Munari - 1130908

October 28, 2015

Contents

1 Calcolo di First e Follow per G0

X	FIRST(X)
Prog	{ *let in end, letrec in end* }
Bind	{ *var* = }
X	{ *and, epsilon* }
Exp	{ *let in end, letrec in end, lambda, var, exp_const,* (, *cons, car, cdr, eq, leq, atom, if then else* }
ExpA	{ *var, exp_const,* (}
E1	{ +, −, *epsilon* }
T	{ *var, exp_const,* (}
T1	{ ∗, /, *epsilon* }
F	{ *var, exp_const,* (}
Y	{ (, *epsilon* }
OPA	{ +, − }
OPM	{ ∗, / }
OPP	{ *cons, car, cdr, eq, leq, atom* }
Seq_Exp	{ *let in end, letrec in end, lambda, var, exp_const,* (, *cons, car, cdr, eq, leq, atom, if then else, epsilon* }
Seq_Var	{ *var, epsilon* }

Table 1: Calcolo di FIRST per G0

X	FOLLOW(X)
Prog	{ $, FOLLOW(Exp) }
Bind	{ $, FOLLOW(X), *in* }
X	{ $, FOLLOW(Bind) }
Exp	{ $, FIRST(Seq_Exp) - epsilon, FOLLOW(Seq_Exp), *then, else*, FIRST(X)-epsilon, FOLLOW(Bind), end }
ExpA	{ $,), FOLLOW(Exp) }
E1	{ $, FOLLOW(ExpA) }
T	{ $, +, −, FOLLOW(ExpA), FOLLOW(E1) }
T1	{ $, FOLLOW(T) }
F	{ $, ∗, /, FOLLOW(T1) }
Y	{ $, FOLLOW(F) }
OPA	{ $, *var, exp_const,* (}
OPM	{ $, *var, exp_const,* (, FOLLOW(T1) }
OPP	{ $, (}
Seq_Exp	{ $,) }
Seq_Var	{ $,) }

Table 2: Calcolo di FOLLOW per G0

2 Costruzione della tabella di parsing per G0

ASD

3 Correzione della grammatica G0 in G1

ASD

4 Calcolo di First e Follow per G1

ASD

5 Costruzione della tabella di parsing per G1

ASD