Contents

1 Esercizio 3.2

1.1 Calcolo di NULL, FIRST e FOLLOW

	NULL	FIRST	FOLLOW
⟨prog⟩		{ `assign`, `print`, `read`, `for`, `if`, `{` }	{ `$` }
⟨statlist⟩		{ `assign`, `print`, `read`, `for`, `if`, `{` }	{ `EOF`, `}` }
⟨statlistp⟩	x	{ `;` }	{ `EOF`, `}` }
⟨stat⟩		{ `assign`, `print`, `read`, `for`, `if`, `{` }	{ `;`, `else`, `end`, `EOF`, `}` }
⟨assignlist⟩		{ `[` }	{ `;`, `else`, `end`, `EOF`, `}` }
⟨assignlistp⟩	x	{ `[` }	{ `;`, `else`, `end`, `EOF`, `}` }
⟨idlist⟩		{ `ID` }	{ `)`, `]` }
⟨idlistp⟩	x	{ `,` }	{ `)`, `]` }
⟨bexpr⟩		{ `<`, `>`, `<=`, `>=`, `==`, `<>` }	{ `)` }
⟨expr⟩		{ `+`, `-`, `*`, `/`, `NUM`, `ID` }	{ `,`, `;`, `to`, `+`, `-`, `*`, `/`, `)`, `NUM`, `ID` }
⟨exprlist⟩		{ `+`, `-`, `*`, `/`, `NUM`, `ID` }	{ `)` }
⟨exprlistp⟩	x	{ `,` }	{ `)` }

1.2 Calcolo degli insiemi GUIDA

GUIDA($\langle prog \rangle \rightarrow \langle statlist \rangle$ `EOF`)	{ `assign`, `print`, `read`, `for`, `if`, `{` }
GUIDA($\langle statlist \rangle \rightarrow \langle stat \rangle \langle statlistp \rangle$)	{ `assign`, `print`, `read`, `for`, `if`, `{` }
GUIDA($\langle statlistp \rangle \rightarrow$ `;` $\langle stat \rangle \langle statlistp \rangle$)	{ `;` }
GUIDA($\langle statlistp \rangle \rightarrow \varepsilon$)	{ `EOF`, `}` }
GUIDA($\langle stat \rangle \rightarrow$ `assign` $\langle assignlist \rangle$)	{ `assign` }
GUIDA($\langle stat \rangle \rightarrow$ `print (` $\langle exprlist \rangle$ `)`)	{ `print` }
GUIDA($\langle stat \rangle \rightarrow$ `read (` $\langle idlist \rangle$ `)`)	{ `read` }
GUIDA($\langle stat \rangle \rightarrow$ `for (` `ID :=` $\langle expr \rangle$ `;` $\langle bexpr \rangle$ `) do` $\langle stat \rangle$)	{ `for` }
GUIDA($\langle stat \rangle \rightarrow$ `for (` $\langle bexpr \rangle$ `) do` $\langle stat \rangle$)	{ `for` }
GUIDA($\langle stat \rangle \rightarrow$ `if (` $\langle bexpr \rangle$ `)` $\langle stat \rangle$ `else` $\langle stat \rangle$ `end`)	{ `if` }
GUIDA($\langle stat \rangle \rightarrow$ `if (` $\langle bexpr \rangle$ `)` $\langle stat \rangle$ `end`)	{ `if` }
GUIDA($\langle stat \rangle \rightarrow$ `{` $\langle statlist \rangle$ `}`)	{ `{` }
GUIDA($\langle assignlist \rangle \rightarrow$ `[` $\langle expr \rangle$ `to` $\langle idlist \rangle$ `]` $\langle assignlistp \rangle$)	{ `[` }
GUIDA($\langle assignlistp \rangle \rightarrow$ `[` $\langle expr \rangle$ `to` $\langle idlist \rangle$ `]` $\langle assignlistp \rangle$)	{ `[` }
GUIDA($\langle assignlistp \rangle \rightarrow \varepsilon$)	{ `;`, `else`, `end`, `EOF`, `}` }
GUIDA($\langle idlist \rangle \rightarrow$ `ID` $\langle idlistp \rangle$)	{ `ID` }
GUIDA($\langle idlistp \rangle \rightarrow$ `,` `ID` $\langle idlistp \rangle$)	{ `,` }
GUIDA($\langle idlistp \rangle \rightarrow \varepsilon$)	{ `)`, `]` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<` $\langle expr \rangle \langle expr \rangle$)	{ `<` }
GUIDA($\langle bexpr \rangle \rightarrow$ `>` $\langle expr \rangle \langle expr \rangle$)	{ `>` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<=` $\langle expr \rangle \langle expr \rangle$)	{ `<=` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<=` $\langle expr \rangle \langle expr \rangle$)	{ `<=` }
GUIDA($\langle bexpr \rangle \rightarrow$ `==` $\langle expr \rangle \langle expr \rangle$)	{ `==` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<>` $\langle expr \rangle \langle expr \rangle$)	{ `<>` }
GUIDA($\langle expr \rangle \rightarrow$ `+ (` $\langle exprlist \rangle$ `)`)	{ `+` }
GUIDA($\langle expr \rangle \rightarrow$ `-` $\langle expr \rangle \langle expr \rangle$)	{ `-` }
GUIDA($\langle expr \rangle \rightarrow$ `* (` $\langle exprlist \rangle$ `)`)	{ `*` }
GUIDA($\langle expr \rangle \rightarrow$ `/` $\langle expr \rangle \langle expr \rangle$)	{ `/` }
GUIDA($\langle expr \rangle \rightarrow$ `NUM`)	{ `NUM` }
GUIDA($\langle expr \rangle \rightarrow$ `ID`)	{ `ID` }
GUIDA($\langle exprlist \rangle \rightarrow \langle expr \rangle \langle exprlistp \rangle$)	{ `+`, `-`, `*`, `/`, `NUM`, `ID` }
GUIDA($\langle exprlistp \rangle \rightarrow$ `,` $\langle expr \rangle \rightarrow \langle exprlistp \rangle$)	{ `,` }
GUIDA($\langle exprlistp \rangle \rightarrow \varepsilon$)	{ `)` }

1.3 Trasformazione in una grammatica LL(1) equivalente

La grammatica data non è LL(1) per via dei seguenti insiemi guida per la variabile ⟨*stat*⟩.

1.3.1 Produzione per for

Dati gli insiemi guida non LL(1) della grammatica per `for`:

GUIDA(⟨*stat*⟩ → `for ( ID :=` ⟨*expr*⟩ `;` ⟨*bexpr*⟩ `) do` ⟨*stat*⟩)	{ `for` }
GUIDA(⟨*stat*⟩ → `for (` ⟨*bexpr*⟩ `) do` ⟨*stat*⟩)	{ `for` }

Fattorizzando la parte non comune ad ambo le produzioni introducendo una variabile ⟨*statc*⟩, otteniamo un nuovo insieme guida per ⟨*stat*⟩ che risulta essere LL(1):

GUIDA(⟨*stat*⟩ → `for (` ⟨*statc*⟩ ⟨*bexpr*⟩ `) do` ⟨*stat*⟩)	{ `for` }
GUIDA(⟨*statc*⟩ → `ID :=` ⟨*expr*⟩ `;`)	{ `ID` }
GUIDA(⟨*statc*⟩ → ε	{ `<`, `>`, `<=`, `>=`, `==`, `<>` }

1.3.2 Produzione per if

Dati gli insiemi guida non LL(1) della grammatica per `if`:

GUIDA(⟨*stat*⟩ → `if (` ⟨*bexpr*⟩ `)` ⟨*stat*⟩ `else` ⟨*stat*⟩ `end`)	{ `if` }
GUIDA(⟨*stat*⟩ → `if (` ⟨*bexpr*⟩ `)` ⟨*stat*⟩ `end`)	{ `if` }

Fattorizzando la parte non comune ad ambo le produzioni introducendo una variabile ⟨*statp*⟩, otteniamo un nuovo insieme guida per ⟨*stat*⟩ che risulta essere LL(1):

GUIDA(⟨*stat*⟩ → `if (` ⟨*bexpr*⟩ `)` ⟨*stat*⟩⟨*statp*⟩ `end`)	{ `if` }
GUIDA(⟨*statp*⟩ → `else` ⟨*stat*⟩)	{ `else` }
GUIDA(⟨*statp*⟩ → ε)	{ `end` }

1.4 Grammatica LL(1) equivalente

1.4.1 Calcolo di NULL, FIRST e FOLLOW

	NULL	FIRST	FOLLOW
⟨prog⟩		{ `assign`, `print`, `read`, `for`, `if`, `{` }	{ `$` }
⟨statlist⟩		{ `assign`, `print`, `read`, `for`, `if`, `{` }	{ `EOF`, `}` }
⟨statlistp⟩	x	{ `;` }	{ `EOF`, `}` }
⟨stat⟩		{ `assign`, `print`, `read`, `for`, `if`, `{` }	{ `;`, `else`, `end`, `EOF`, `}` }
⟨statc⟩	x	{ `ID` }	{ `<`, `>`, `<=`, `>=`, `==`, `<>` }
⟨statp⟩	x	{ `else` }	{ `end` }
⟨assignlist⟩		{ `[` }	{ `;`, `else`, `end`, `EOF`, `}` }
⟨assignlistp⟩	x	{ `[` }	{ `;`, `else`, `end`, `EOF`, `}` }
⟨idlist⟩		{ `ID` }	{ `)`, `]` }
⟨idlistp⟩	x	{ `,` }	{ `)`, `]` }
⟨bexpr⟩		{ `<`, `>`, `<=`, `>=`, `==`, `<>` }	{ `)` }
⟨expr⟩		{ `+`, `-`, `*`, `/`, `NUM`, `ID` }	{ `,`, `;`, `to`, `+`, `-`, `*`, `/`, `)`, `NUM`, `ID` }
⟨exprlist⟩		{ `+`, `-`, `*`, `/`, `NUM`, `ID` }	{ `)` }
⟨exprlistp⟩	x	{ `,` }	{ `)` }

1.4.2 Calcolo degli insiemi GUIDA

GUIDA($\langle prog \rangle \rightarrow \langle statlist \rangle$ `EOF`)	{ `assign`, `print`, `read`, `for`, `if`, `{` }
GUIDA($\langle statlist \rangle \rightarrow \langle stat \rangle \langle statlistp \rangle$)	{ `assign`, `print`, `read`, `for`, `if`, `{` }
GUIDA($\langle statlistp \rangle \rightarrow$ `;` $\langle stat \rangle \langle statlistp \rangle$)	{ `;` }
GUIDA($\langle statlistp \rangle \rightarrow \varepsilon$)	{ `EOF`, `}` }
GUIDA($\langle stat \rangle \rightarrow$ `assign` $\langle assignlist \rangle$)	{ `assign` }
GUIDA($\langle stat \rangle \rightarrow$ `print (` $\langle exprlist \rangle$ `)`)	{ `print` }
GUIDA($\langle stat \rangle \rightarrow$ `read (` $\langle idlist \rangle$ `)`)	{ `read` }
GUIDA($\langle stat \rangle \rightarrow$ `for (` $\langle statc \rangle$ $\langle bexpr \rangle$ `) do` $\langle stat \rangle$)	{ `for` }
GUIDA($\langle stat \rangle \rightarrow$ `if (` $\langle bexpr \rangle$ `)` $\langle stat \rangle \langle statp \rangle$ `end`)	{ `if` }
GUIDA($\langle stat \rangle \rightarrow$ `{` $\langle statlist \rangle$ `}`)	{ `{` }
GUIDA($\langle statc \rangle \rightarrow$ `ID :=` $\langle expr \rangle$ `;`)	{ `ID` }
GUIDA($\langle statc \rangle \rightarrow \varepsilon$	{ `<`, `>`, `<=`, `>=`, `==`, `<>` }
GUIDA($\langle statp \rangle \rightarrow$ `else` $\langle stat \rangle$)	{ `else` }
GUIDA($\langle statp \rangle \rightarrow \varepsilon$)	{ `end` }
GUIDA($\langle assignlist \rangle \rightarrow$ `[` $\langle expr \rangle$ `to` $\langle idlist \rangle$ `]` $\langle assignlistp \rangle$)	{ `[` }
GUIDA($\langle assignlistp \rangle \rightarrow$ `[` $\langle expr \rangle$ `to` $\langle idlist \rangle$ `]` $\langle assignlistp \rangle$)	{ `[` }
GUIDA($\langle assignlistp \rangle \rightarrow \varepsilon$)	{ `;`, `else`, `end`, `EOF`, `}` }
GUIDA($\langle idlist \rangle \rightarrow$ `ID` $\langle idlistp \rangle$)	{ `ID` }
GUIDA($\langle idlistp \rangle \rightarrow$ `,` `ID` $\langle idlistp \rangle$)	{ `,` }
GUIDA($\langle idlistp \rangle \rightarrow \varepsilon$)	{ `)`, `]` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<` $\langle expr \rangle \langle expr \rangle$)	{ `<` }
GUIDA($\langle bexpr \rangle \rightarrow$ `>` $\langle expr \rangle \langle expr \rangle$)	{ `>` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<=` $\langle expr \rangle \langle expr \rangle$)	{ `<=` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<=` $\langle expr \rangle \langle expr \rangle$)	{ `<=` }
GUIDA($\langle bexpr \rangle \rightarrow$ `==` $\langle expr \rangle \langle expr \rangle$)	{ `==` }
GUIDA($\langle bexpr \rangle \rightarrow$ `<>` $\langle expr \rangle \langle expr \rangle$)	{ `<>` }
GUIDA($\langle expr \rangle \rightarrow$ `+ (` $\langle exprlist \rangle$ `)`)	{ `+` }
GUIDA($\langle expr \rangle \rightarrow$ `-` $\langle expr \rangle \langle expr \rangle$)	{ `-` }
GUIDA($\langle expr \rangle \rightarrow$ `* (` $\langle exprlist \rangle$ `)`)	{ `*` }
GUIDA($\langle expr \rangle \rightarrow$ `/` $\langle expr \rangle \langle expr \rangle$)	{ `/` }
GUIDA($\langle expr \rangle \rightarrow$ `NUM`)	{ `NUM` }
GUIDA($\langle expr \rangle \rightarrow$ `ID`)	{ `ID` }
GUIDA($\langle exprlist \rangle \rightarrow \langle expr \rangle \langle exprlistp \rangle$)	{ `+`, `-`, `*`, `/`, `NUM`, `ID` }
GUIDA($\langle exprlistp \rangle \rightarrow$ `,` $\langle expr \rangle \langle exprlistp \rangle$)	{ `,` }
GUIDA($\langle exprlistp \rangle \rightarrow \varepsilon$)	{ `)` }