

Relazione

Compilatori

Compilatori

A.A. 2019/2020

Prof.ssa Sabrina Mantaci

Studenti:

Tommaso Scalici (Informatica - 0665134)

Salvatore Apicella (Informatica - 0670120)

- Descrizione della realizzazione del progetto

L'obiettivo del progetto è la realizzazione di un traduttore guidato dalla sintassi in grado di riconoscere un linguaggio per gestire la suddivisione delle spese mensili in un condominio.

Per fare ciò, abbiamo utilizzato **Flex** e **Bison**, generatori automatici di analizzatori lessicali e sintattici (di tipo LALR) rispettivamente; il traduttore ottenuto accetta quindi un file in input, e dopo le diverse fasi di analisi, restituisce in output un file di testo contenente la suddivisione delle spese condominiali per ciascun condomino

- Analisi del file di input

Il file di input è diviso in tre sezioni, separate da due sequenze di simboli che fungono da separatori, la prima contenente una data in formato MM/YYYY, la seconda contenente le spese condominiali (nome seguito dall'importo) suddivise su più righe, la terza contenente i dati dei condomini, anch'essi suddivisi su più righe.

- Realizzazione dell'analizzatore lessicale (**Flex**)

Abbiamo innanzitutto dichiarato nella sezione delle *dichiarazioni* i pattern di stringhe che ci interessava riconoscere all'interno del file di input.

Successivamente abbiamo scritto le regole di traduzione mediante i pattern espressi come espressioni regolari e le azioni da eseguire al riconoscimento di ciascun token.

Infine abbiamo definito la funzione **main**, nella quale associamo lo stream yyin al file da leggere in input e richiamiamo le funzioni di inizializzazione della **symbol table** e le funzioni per avviare l'analisi lessicale (**yylex**) e sintattica (**yyparse**).

Definizione regolare	Pattern
digit	[0-9]
number	{digit}+
letter	[A-Za-z]
word	{letter}+
name	[A-Z]{letter}+
date	(0[1-9] 10 11 12)\/[0-9]{4}
email	[a-z0-9._%+]+@[a-z0-9.]+\.[a-z]{2,4}
fiscalcode	[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]
telephone	[1-9][0-9]{9}
millesimal	([1-9][0-9]{0,2} 1000)
whitespace	[]

Token	Lessema
SEPARATOR_1	"&&&&"
SEPARATOR_2	"\$\$\$\$\$"
DASH	"_"
DATE	05/2018
EXPENSE	Luce, Pulizia, ecc.
NUM_REAL	102.40, 53,20, ecc.
UNIT	3, 8, 5, ecc.
FISCALCODE	RSSMRA77L07A546F, ecc.
TELEPHONE	3358586345, ecc.
EMAIL	m.rosselli@gmail.com
MILLESIMAL	Numeri compresi tra 1 e 1000

- Realizzazione dell'analizzatore sintattico (**Bison**)

Per la realizzazione del parser, nella sezione dichiarativa abbiamo utilizzato il costrutto union per definire la collezione dei tipi da associare ai token e dichiarato tutti i token da riconoscere nel linguaggio.

Nella sezione delle regole della grammatica, abbiamo scritto tutte le regole di produzione della grammatica CF per riconoscere il linguaggio.

Token	Tipo
SEPARATOR_1	-
SEPARATOR_2	-
DASH	-
DATE	<string>
EXPENSE	<string>
NUM_REAL	<float_t>
UNIT	<int_t>
FISCALCODE	<string>
TELEPHONE	<string>
EMAIL	<string>
MILLESIMAL	<int_t>

```

Start          : section1 SEPARATOR_1 section2 SEPARATOR_2 section3
                ;

section1       : DATE
                ;

section2       : section2 EXPENSE NUM_REAL { addExpense($3); }
                |
                ;

section3       : section3 NAME DASH FISCALCODE DASH EMAIL DASH
                TELEPHONE DASH MILLESIMAL DASH UNIT
                {
                    addSymbol($2, $4, $6, $8, $10, $12);
                }
                |
                ;

```

- Analisi semantica e symbol table

Per la costruzione della symbol table abbiamo deciso di implementare una *lista concatenata* per avere una struttura dinamica indipendente dal numero dei simboli in input e una gestione più semplice della struttura stessa. Abbiamo utilizzato la funzione **addExpense** come regola semantica associata alla regola di produzione *section2* per calcolarci il totale delle spese condominiali.

La funzione **addSymbol** viene utilizzata per aggiungere un nuovo simbolo alla tabella, dove ciascun simbolo contiene tutte le informazioni di ciascun condomino individuate nella *section3* e per calcolare il valore di spesa per quel condomino utilizzando il valore di millesimale corrispondente e la somma delle spese precedentemente calcolata.

