

# Linguaggi e Traduttori

## a.a. 2006/2007

### Tema d'esame del 23 gennaio 2007

Si costruisca, utilizzando la coppia di programmi JFLEX e CUP o FLEX e BISON, un traduttore guidato dalla sintassi in grado di riconoscere un linguaggio che permette di calcolare delle espressioni booleane.

## Linguaggio di ingresso

Il file di ingresso al traduttore è composto da un'intestazione seguita da una lista di operazioni eseguite su variabili booleane.

L'intestazione è il seguente:

```
Nome_file: File1.dat;
Data: 30/maggio/2006;
Ora: 10:30 pm;
Email: pippo.pluto@polito.it;
```

Nell'intestazione sono contenuti un nome di file, una data, un'ora e uno o due indirizzi email, ogni elemento dell'intestazione è terminato dal simbolo “;” (punto e virgola). L'ordine in cui sono riportate le informazioni è quello dell'esempio, eccetto data e ora che possono apparire in ordine invertito.

In dettaglio le informazioni contenute sono le seguenti:

- **Nome file:** La parola “Nome\_file:” seguita da <nome\_file> dove <nome\_file> è una stringa di caratteri o numeri il cui simbolo iniziale è un carattere, seguita dal carattere “.” (punto) e ultimato da 3 caratteri.
- **Data:** La parola “Data: “ seguita da due cifre numeriche comprese tra 01 e 31 che rappresentano il giorno, seguiti da uno “/” (slash), seguiti da un mese (gennaio, febbraio, ...), seguiti da uno “/” (slash), seguiti da un anno rappresentato con 4 numeri.
- **Ora:** La parola “Ora:” seguita da un numero compreso tra 1 e 12 di una o due cifre, seguito da “.” (due punti), seguito da un numero di due cifre compreso tra 00 e 59, seguito dalle keywords “am” o “pm”.
- **Email:** La parola “Email:” seguita da una <email> dove <email> è una stringa di lettere, numeri e caratteri “.” (punto) e “\_” (underscore), seguita dal simbolo “@”, seguita da una stringa di lettere, un “.” (punto) e terminata da una stringa di lettere.

Le operazioni sulle variabili booleane sono una lista anche vuota di operazioni terminate dal simbolo “;” (punto e virgola).

In un'operazione booleana sono disponibili gli operatori AND, OR e NOT (lettere maiuscole) che eseguono rispettivamente le operazioni logiche di and, or e not su variabili booleane.

Le variabili booleane sono delle stringhe di lettere (sia maiuscole che minuscole), numeri e “\_” (underscore) in cui il primo carattere è una lettera o un “\_”. Le variabili booleane possono assumere i valori 0, 1, TRUE o FALSE. All'interno di un'espressione booleana si possono utilizzare sia le variabili che direttamente i valori 0, 1, TRUE e FALSE.

Il linguaggio permette di assegnare il risultato di un'operazione booleana ad una variabile o ad una lista di variabili booleane. Un esempio di un assegnamento è il seguente:

```
0: VAR A_0, VAR a_1;
TRUE : VAR B;
A_0 OR B OR FALSE : VAR _res1, VAR _res2, VAR _res3;
```

La prima operazione assegna a A\_0 e a\_1 il valore FALSE. La seconda assegna a B il valore 1. Mentre la terza assegna a \_res1, \_res2 e \_res3 il valore TRUE.

La sintassi dell'assegnamento è perciò la seguente: un'espressione booleana, seguita dal carattere “:” (due punti) e dalla lista di variabili alle quali si vuole assegnare il risultato dell'espressione booleana. Il nome di ogni variabile è preceduto dalla keyword “VAR”.

Si vuole che il nome delle variabili e il relativo valore booleano siano salvati all'interno di una symbol table, al fine di poter essere utilizzate nelle espressioni che le seguono.

**CONTINUA...**

Non si utilizzino variabili globali, eccetto quella per la gestione della symbol table. Per accedere al valore booleano al fine di inserire la variabile all'interno della symbol table, si richiede l'utilizzo di attributi ereditati. Soluzioni che non utilizzino gli attributi ereditati o che li utilizzino in modo scorretto (Es. Ricercare un simbolo in una posizione sbagliata all'interno dello stack) renderanno la parte di analisi semantica insufficiente. Per risolvere le espressioni si utilizzino, invece, gli attributi sintetizzati.

Si faccia attenzione a gestire in modo corretto le precedenze e le associatività.

Il programma dovrà, dato un file in ingresso, eseguire tutti i calcoli e fornire l'output riportato nell'esempio seguente.

## Esempio

A fronte del seguente file di ingresso:

```
Nome_file: File1.dat;
Data: 30/aprile/2006;
Ora: 10:30 pm;
Email: pippo.pluto@polito.it;
Email: idefix@polito.it;
0      : VAR a1, VAR B, VAR c;
TRUE   : VAR d_2;
NOT a1 AND B;
NOT ( (a AND c) OR (B AND d_2) ) : VAR F, VAR h ;
FALSE  : VAR m;
1 :VAR l, VAR i;
F OR 0 OR m OR l OR TRUE : VAR R ;
```

Il programma dovrà produrre il seguente output:

```
FILE : file1.dat
EMAIL: pippo.pluto@polito.it
EMAIL: idefix@polito.it
a=0; B=0; c=0;
d=1;
0;
F=1; h=1;
m=0;
l=1; i=1;
R=1;
```