

# Linguaggi e Traduttori

## a.a. 2005/2006

### Tema d'esame del 7 Febbraio 2006

Si costruisca, utilizzando la coppia di programmi JFLEX e CUP, un traduttore guidato dalla sintassi in grado di riconoscere un linguaggio che permette di calcolare la dimensione di totale di file mp3 scaricati da un utente.

## Linguaggio di ingresso

Il file di ingresso al traduttore è diviso in due sezioni:

- la prima identificata dalla parola chiave "mp3\_list:" nella quale vengono descritte una serie di caratteristiche che identificano un file audio (Bit/rate e durata)
- la seconda è identificata dalla parola chiave "server:", in questa sezione vengono riportati i file mp3 scaricati da ogni utente

### Prima sezione: mp3\_list:

In questa sezione sono presenti una serie di file mp3 suddivisi per bit/rate.

Ogni riga di questa sezione è perciò iniziata da un bit/rate, seguita dal carattere ":" (Due punti) da una lista di file mp3 con associata una durata in secondi e terminata dal carattere ";" (Punto e virgola).

Un bit/rate è un numero intero seguito dalla stringa "Kb/s" (Kbits/secondo).

Ogni elemento della lista di file mp3 e relativa durata è composto da un nome di file con estensione mp3 e da un intero che rappresenta i secondi di durata della canzone. (Es. WeAreTheChampion1.mp3 120). Il nome di un file mp3 è una stringa di lettere, numeri e caratteri "\_" (Underscore), il cui primo carattere è una lettera. L'estensione è sempre "mp3".

Gli elementi della lista sono separati dal carattere ";" (Virgola).

Un esempio di sezione mp3\_list:

```
mp3_list:
256 Kb/s : TheShowMustGoOn_a12.mp3 150 , WeAreTheChampion1.mp3 120 , Time.mp3 30 ;
128 Kb/s : TheUnforgiven.mp3 200 ;
96 Kb/s : TheEnd.mp3 400 , NottePrimaDegliEsami.mp3 250 ;
```

Dopo aver analizzato questa prima parte del file di ingresso il parser dovrà aver creato in memoria una tabella di hash in cui la chiave sarà il nome del file e l'informazione contenuta la dimensione del file in bits.

Per calcolare la dimensione di ogni file bisogna moltiplicare il bit/rate per la durata. Nel caso del file *TheShowMustGoOn.mp3*, esso avrà una dimensione di  $150 \cdot 256 = 38400$  bits.

Per ottenere il valore di bit/rate, al fine di moltiplicarlo per la durata e inserire una entry nella tabella di hash, si richiede l'utilizzo di attributi ereditati.

Soluzioni che non utilizzino gli attributi ereditati o che li utilizzino in modo scorretto (Es. RESULT=(Integer)stack[top-2] anziché RESULT=(Integer)stack[top-1]) renderanno la parte di analisi semantica insufficiente.

Esempio della tabella di hash generata:

CHIAVE	VALORE
<i>TheShowMustGoOn_a12.mp3</i>	38400
<i>WeAreTheChampion1.mp3</i>	30720
<i>Time.mp3</i>	7680
<i>TheUnforgiven.mp3</i>	25600
<i>TheEnd.mp3</i>	38400
<i>NottePrimaDegliEsami.mp3</i>	24000

### Seconda sezione: "server:"

La seconda sezione inizia con la parola chiave "server:"

Appena dopo la keyword di inizio sezione ("server:") sono presenti la keyword "data:" seguita da una data in formato GG/MM/AAAA e la keyword "time:" seguita da un'ora nel formato "HH:MM". L'ordine in cui le due keyword si presentano nel file di ingresso è casuale (sempre dopo "server": ma possono essere invertite) e devono essere entrambe presenti e uniche (Lo si gestisca con una grammatica opportuna).

**Continua...**

GG nella data è un numero tra 01 e 31, MM nella data è un numero compreso tra 01 e 12, HH nell'ora è un numero tra 00 e 23, MM nell'ora è un numero tra 00 e 59. (Lo si gestisca con le espressioni regolari nello scanner)

Il file di ingresso finisce con una lista anche vuota di utenti e dei rispettivi file scaricati.

Ogni elemento della lista è composto da un indirizzo IP, seguito dal carattere ":" (Due Punti), da un elenco di file con estensione mp3, separati dal carattere ";" (virgola) e terminato dal carattere ";" (Punto e Virgola).

Un indirizzo IP è una serie di quattro numeri interi separati da un "." (Punto). I numeri interi sono compresi nel range 0-255; non sono corretti numeri come 000 o 012, lo sono i numeri 0 e 12. (Lo si gestisca con le espressioni regolari nello scanner)

Un esempio di sezione server:

Lessicalmente corretto	Lessicalmente scorretto (errori sottolineati)
<pre>server: time: 00:59 data:20/02/2006 130.0.12.255 : TheEnd.mp3, Time.mp3, TheUnforgiven.mp3 ; 130.0.12.254 : TheEnd.mp3 ;</pre>	<pre>server: data: 20/13/2006 time: 0:59 130.0.12.280 : TheEnd.mp3, Time.mp3, TheUnforgiven.mp3 ; 130.0.012.254 : TheEnd.mp3 ;</pre>

Il programma, per ogni indirizzo IP, dovrà calcolare la dimensione totale dei file elencati espressa in bits. Non verranno accettate soluzioni che utilizzino variabili globali per memorizzare la dimensione totale, si usino gli attributi sintetizzati e l'oggetto predefinito RESULT per propagare la somma nell'albero di derivazione.

L'output del programma dovrà essere il seguente:

```
OUTPUT:
130.0.12.255
TheEnd.mp3 38400
Time.mp3 7680
TheUnforgiven.mp3 25600
TOTALE: 71680
130.0.12.254
TheEnd.mp3 38400
TOTALE: 38400
```

## Esempio

A fronte del seguente file di ingresso:

```
mp3_list:
256 Kb/s : TheShowMustGoOn_a12.mp3 150 , WeAreTheChampion1.mp3 120 , Time.mp3 30 ;
128 Kb/s : TheUnforgiven.mp3 200 ;
96 Kb/s: TheEnd.mp3 400 , NottePrimaDegliEsami.mp3 250 ;
server:
time: 00:59
data:20/02/2006
130.0.12.255 : TheEnd.mp3, Time.mp3, TheUnforgiven.mp3 ;
130.0.12.254 : TheEnd.mp3 ;
```

Il programma dovrà produrre il seguente output:

```
OUTPUT:
130.0.12.255
TheEnd.mp3 38400
Time.mp3 7680
TheUnforgiven.mp3 25600
TOTALE: 71680
130.0.12.254
TheEnd.mp3 38400
TOTALE: 38400
```