

UNIVERSITÀ DEGLI STUDI "ROMA TRE"
CORSO DI STUDI IN SCIENZE COMPUTAZIONALI
IN490 - LINGUAGGI DI PROGRAMMAZIONE – A.A. 2017-2018
M. PEDICINI

ESAME DEL 9/07/2018 – TEMPO 3H00

COGNOME _____ NOME _____ MATRICOLA _____

Esercizio 1. *One Instruction Set Computer (OISC), significa una macchina M con un linguaggio di una sola istruzione: un esempio di OISC è la macchina M_{bbj} (dove bbj sta per byte-byte-jump. La macchina ha una memoria che contiene byte e una sola istruzione bbj , l'istruzione ha tre argomenti interi, poichè non ci sono altre istruzioni è possibile omettere l'op-code nel compilato.*

L'esecuzione dell'istruzione bbj a b c copia il byte di indirizzo a nel byte di indirizzo b e poi l'esecuzione salta all'indirizzo c incondizionatamente.

Per maggiore chiarezza diamo qui lo pseudo-codice di un interprete per un programma contenuto nell'array di memoria $mem[]$ con indirizzi presi a 4 byte (32 bits) :

```
uint8_t mem[MEMSIZE];
uint32_t *pc = (uint32_t *)mem;
for(;;) {
    mem[pc[1]] = mem[pc[0]];
    pc = (uint32_t *) (mem + pc[2]);
}
```

Un programma in linguaggio macchina è dunque dato (come al solito) da una lista di interi da caricare in memoria che però non contiene opcodes. Essendo le operazioni orientate ai byte scriveremo il contenuto di una cella di memoria come esadecimale, in modo che due cifre esadecimali rappresentino il byte contenuto in memoria.

Ad esempio il programma che ha in memoria i valori 04 00 00 00 04 00 00 00 00 00 00 00 per cui la prima istruzione eseguita è bbj 4 4 0, rappresenta un loop infinito che copia il byte nella cella di memoria di indirizzo 00000004 (in se stessa) e ritorna ad eseguire la stessa tripla (loop infinito).

- (1) Tradurre l'istruzione JMP c (salto incondizionato) in bbj ;
- (2) Scrivere un programma in bbj corrispondente all'istruzione INC a che incrementa il byte contenuto all'indirizzo a [hint: creare una tabella per tutti i possibili valori dell'incremento e copiare il corrispondente valore].

Esercizio 2. (1) Quando una grammatica si dice ambigua? Fornire la definizione.
(2) Utilizzando la nozione di ambiguità appena fornita, mostrare se la grammatica

$$G = (\{*, b\}, \{S\}, S, R)$$

con

$$R = \begin{cases} S \rightarrow S * S \\ S \rightarrow b \end{cases}$$

sia ambigua oppure no (giustificare la risposta);

- (3) in caso G sia ambigua, fornire una grammatica G' non ambigua che genera lo stesso linguaggio, ovvero per cui $L(G) = L(G')$.

Esercizio 3. (1) Si descriva il costrutto dello scope statico.

- (2) *Si descriva il costrutto del passaggio per nome.*
- (3) *Cosa stampa il seguente frammento in un linguaggio con scope statico e passaggio dei parametri per nome? (La valutazione delle espressioni avviene da destra a sinistra).*

```
int x = 4;

int g(name int w){
    int x = 6;
    return x + w + w;
}

int f (name int w){
    return x + x + w;
}

write (x, f(g(x++)), x);
```