I compitino 2005 - Informatica 3 – prof. Gargantini

Computabilità (sintetico) ~3

Dimostra che il problema dell'Halt non è computabile.

Macchine di Turing ~7

Scrivi una macchina di Turing che ha come alfabeto {0,1,#}, elabora la sequenza di input, che inizia nella posizione iniziale della testina, continua a destra e finisce con #, modificando in 0 il primo 1 di ogni **sottosequenza di soli 1**.

Lisp ~1+

Definisci una fuzione **terzo** in lisp che restituisce il terzo elemento di una lista

Sintassi ~2

Data la grammatica

```
e::= b | eb | e + b  | e * b
b::= 0 | 1
```

Quali frasi posso costruire? Posso costruire la frase: 0010010 + 0111 ? La grammatica permette alberi di parser ambigui?

Algol passaggio by name ~2

Considera la seguente procedura

```
integer procedure incSum(j,k);
  integer j,k;
  begin
    j:=j+1; incSum:= k + j
  end;
```

Il seguente codice cosa scrive e perchè?

```
integer a,b; a := 4; b:= 10;
print(incSum(a,b))
```

Type checking ~2

Quali sono i vantaggi e gli svantaggi del controllo statico (compile-time) della correttezza dei tipi?

Polimorfismo parametrico in C++ ~2

Scrivi una funzione polimorfica **choose** in C++ che prende due parametri dello stesso tipo e restituisce uno dei due a caso (supponi di poter usare una fuzione float rnd() che restituisce un numero casuale compreso tra 0 e 1).

Record di attivazione ~7

Dato il seguente codice:

```
int a, x;

int foo(int a, int b){
     int c = 5;
     if (a == b) return x;
     else return gun(c);}

int gun(int a) {
     return foo(0,0) + 4 ;}

void main(){
  a = 0;  x = 8;
  { int x, y;
    x = 0; y = 7;
    a = foo(x,y);
  }
}
```

Disegna lo stato della pila con tutti i record di attivazione alla sua massima estensione. Scrivi a parte i valori ritornati e dove verranno copiati.

Passaggio parametri (sintetico!!!) ~5

Dato il seguente pseudo-codice, che vorrebbe calcolare il prodotto di due numeri mediante somme ripetute:

```
prod (integer a, b, c) {
 c := 0;
 while (a != 0) do
    c := c + b;
    a: = a -1;
 endwhile
 return c
}
```

Per quale passaggio dei parametri funziona effettivamente? Considera come esempio quando la chiamata **prod(x,x,y)** calcola effettivamente $y = x^2$? (considera solo per riferimento e per valore).

Nuovi tipi in C (max 5 righe) ~1

Posso introdurre nuovi tipi in C? Come?

Soluzioni

Computabilità: 10 min (Come libro).

Turing : 20 min (30)
Esempi:
00000000111110000110 0 =>
00000000111110000010 0

Prendo tre stati:
s1: sto analizzando 0 , s2 scorri gli 1 e f stato finale
s1 -> 0,-,>,s1 : trovo 0, non faccio nulla, sposo a dex
s1 -> 1,0,>,s2 : trovo 1, lo sostituisco in 0, sposto a dex
s2 -> 1,-,>,s2 : un altro 1, scorro
s2 -> 0,-,>,s1 : fine sotto sequenza di 1, torno a s1
s1 -> #,-,>, f
s2 -> #,-,>, f

Sintassi 10 minuti (40)
Posso costruire tutti le espressioni con * e + tra numeri binari. Infatti grazie alla regola e::=eb posso costruire un numero binario sia quando è operando di sin sia quando è operando di dex (in questo caso devo prima costruire l'operaione e o poi l'operando. Per esempio 10 + 11 posso scriverlo = b => 1 = eb => 10 = e+b => 10+1 = eb => 10 +11. La frase d'esempio può essere costruita. Non ho mai ambiguità.

Algol: 10 min (50)
Diventa:
inc1Sum(a,b); a:=a+1; inc1Sum:= b + a; end;

Il seguente codice cosa scrive e perchè?
integer a,b; a = 4; b= 10; print(inc1Sum(a,b))
15.

Record di attivazione 20 min (70 min)

```
----RA main ----
a (12)
-----------------
x 8
------------------------
-------RA main ----
CL  -----> ad a
---------
x  0
---------
y  7
---------
foo(0,7)  (12)
-----------
------- RA foo(0,7) -----
CL ---> al CL precedente
```

```
-------
RRA --> a foo(0,7)
-------
a  0
-------
b  7
-------
c  5
-------
gun(5) (12)
-------
------- RA gun(5) ------
CL ---> al CL precedente
-------
RRA in gun(5)
-------
```

```
a  5
-------
foo(0,0)
-------
-------
------- R
CL --->
-------
RRA -->
-------
a  0
-------
b  0
-------
c 5
```

Passaggio parametri (15 min – 85)
Il metodo sia restituisce c che lo modifica.
Se c è passata per valore, prod non modificherà c.
In genere a non deve essere passata per riferimento, altrimenti viene modificata.
Comunque, le combinazioni corrette tale che dopo **prod(x,x,y)** $y = x^2$ sono:
VVR: la modifica che faccio su a è locale, quindi quando faccio c:=c+b va bene
VRR: come sopra
RVR: modifico a, ma ho copiato b OK
Altre combinazioni sono chiaramente sbagliate:
RR-: se passo a e b come riferimento, quando modifico a modifico anche b
Le combinazioni
- -V (tranne RRV) restituiscono il valore corretto ma non modificano in modo atteso c.

Polimorfismo parametrico(10 min – 90)
template <typename T> T choose (T x, T y){

```
   if (rnd()<=0.5) return x;
   else return y;
}
```

Lisp (10 min -100)
Definizione di tipi in C
Mediante typedef, esempio ...