

# Compitino prima parte di Informatica 3

Angelo Gargantini  
Università di Bergamo

## Decidibilità

- 1) Cosa vuol dire che un problema non è decidibile? Dai un esempio di problema non decidibile. (5 righe)
- 2) *Facoltativo*: dimostra che tale problema non è decidibile

## Programmazione funzionale

- 3) Cosa fanno le seguenti istruzioni Lisp?  
quote - cons - car - cdr -  
lambda (5 righe)
- 4) Scrivi un piccolo esempio (chiamata e valutazione) per ognuna.
- 5) Quanto vale: (cond ((< 2 1) 2) ((< 1 2) 1))
- 6) Cosa serve il Garbage Collection ? (3 righe)

## Sintassi

- 7) Perché le seguenti regole per definire l'istruzione **if** sono ambigue? (supponi che assegnamento e expr siano ben definiti) Se c'è ambiguità, dai un esempio.

```
IstruzioneIf ::=  
if ( expr ) Istruzione |  
if ( expr ) Istruzione  
                else Istruzione  
Istruzione ::=  
    assegnamento | IstruzioneIf
```

## Algol: Pass-by-name

- 8) Considera la seguente procedura:  
procedure double(i, j);  
 integer i, j;  
 begin  
 i := i\*2;  
 j := j\*2  
 end;

Quale è l'effetto di double(k, A[k]); ?  
C'è qualche problema?

## Controllo dei tipi

- 9) Cosa vuol dire compile-time e run-time type checking? Quali sono i vantaggi/svantaggi di uno rispetto all'altro? Per ciascuno dei due type

checking, scrivi un'istruzione in un linguaggio di tua scelta e spiega brevemente in cosa consiste il type checking. (10 righe)

- 10) Posso scrivere un programma che trova tutti gli errori di tipo che si verificherebbero eseguendo un codice  $P$  (e solo quelli) analizzando staticamente  $P$ ? Perché?

- 11) *type inference*: considera la seguente procedura in ML

```
fun f(x) = 2 * x;
```

Di che tipo è  $f$ ? Quale è il ragionamento che fa l'interprete ML?

## C++ template

- 12) Scrivi una funzione parametrica in C++ (template) che prende due variabili e ne scambia il contenuto. Scrivi due esempi di chiamata di tale funzione.

## Semantica Assiomatica

- 13) prova  $\{P\}S\{Q\}$ , dove  $P = \{\text{true}\}$ ,  $Q = \{x \geq 0\}$  e  $S$ :

```
if x < 0 then x := 0 endif;  
x := x + 2;
```

- 14) dato il seguente programma

```
{y >= 0}  
i = 0;  
quad = 0;  
while ( i < y ) {  
    quad = quad + y;  
    i = i + 1;  
}  
{quad = y^2}
```

Proponi un invariante del ciclo while e dai una spiegazione (informale) perché secondo te esso è corretto.

- 15) *Facoltativo*: prova che il programma 13) è corretto.

## Soluzioni ad alcuni esercizi

11) ML valuta f come:

- `val it = fn : int → int`

Infatti \* potrebbe essere `int x int ->` oppure `float x float -> float`. Ma siccome 2 è int, l'interprete sceglie la prima possibilità. A questo punto x dovrà essere int e il risultato sarà int.

12)

```
template <typename T> void cambia(T& x, T& y){  
    T tmp = x; x=y; y=tmp;  
}
```

```
int i,j; scambia(i,j); // scambia due interi
```

```
float a,b; scambia(a,b); // scambia due float
```

13) Applico la regola per la sequenza e divido la dimostrazione in due parti:

```
{true} if x < 0 then x := 0 endif; {Q1}  
{Q1} x:= x+2 {x>=0}
```

Comincio con la seconda istruzione:

```
{Q1} x:= x+2 {x>=0}
```

Applico la regola per l'assegnamento e trovo

Q1 :  $x+2 \geq 0$  cioè  $x \geq -2$

Ora applico la regola per l'if e trovo

```
{}
```

14: L'invariante  $I = \{ \text{quad} = i \}$