



# UNIVERSITÀ DI PISA

Dipartimento di Informatica  
Corso di Laurea Triennale in Informatica

Corso 2° anno - 9 CFU

## Paradigmi di programmazione

**Professore:**  
Prof. Paolo Milazzo

**Autore:**  
Filippo Ghirardini

---

Anno Accademico 2023/2024

## Contents

<b>1</b>	<b>Lambda calcolo</b>	<b>2</b>
<b>2</b>	<b>Sistemi di tipi</b>	<b>3</b>
2.1	Perché? . . . . .	3
2.2	Cosa sono i tipi? . . . . .	3
2.3	Come funziona? . . . . .	3

# 1   Lambda calcolo

## 2 Sistemi di tipi

### 2.1 Perché?

Dato che nel lambda calcolo i programmi e i valori sono funzioni possiamo facilmente scrivere programmi che non sono corretti rispetto all'uso inteso dei valori. Ad esempio:

**Esempio 2.1** (Errore tipi). Nella seguente espressione si può applicare 0 a *False*, ottenendo quindi un risultato che però non ha alcun senso:

$$False\ 0 = (\lambda t.\lambda f.f)(\lambda z.\lambda s.z) \rightarrow \lambda f.f \quad (1)$$

Analogamente per una macchina tutto è un bit: istruzioni, dati e operazioni. Un esempio più pratico è il seguente:

**Esempio 2.2.** L'istruzione nel corpo dell'*if* contiene un errore di tipo (stringa divisa per un intero). Se non avessimo il controllo dei tipi l'unico modo per scoprire l'errore sarebbe eseguire numerosi test per riuscire a coprire tutte le possibilità, fino ad entrare nel corpo dell'*if*. Richiederebbe tempo e risorse e non ci garantisce neanche la certezza di aver provato tutti i casi possibili.

---

```
if (condizione_complicata) {
  return "hello"/10;
}
```

---

Se in certi linguaggi di programmazione ci troveremmo davanti ad errori di esecuzione, in altri (come ad esempio JavaScript) otterremmo un errore nel risultato in quanto l'interprete proverebbe a fare un cast manuale.

Concludendo, la mancanza di **type safety** aumenta il numero di bug, rendendo così un software meno funzionale e più vulnerabile.

### 2.2 Cosa sono i tipi?

I **sistemi di tipo** sono meccanismi che permettono di rilevare in anticipo errori di programmazione.

**Definizione 2.1** (Tipo). *Il tipo è un **attributo** di un dato che descrive come il linguaggio di programmazione permetta di usare quel particolare dato.*

Un tipo serve quindi a limitare i valori che un'espressione può assumere, che operazioni possono essere effettuate sui dati e in che modo questi ultimi possono essere salvati.

**Definizione 2.2** (Sistema dei tipi). *Un sistema dei tipi è un metodo **sintattico**, **effettivo** per dimostrare l'assenza di comportamenti anomali del programma **strutturando** le operazioni del programma in base ai tipi di valori che calcolano.*

Analizziamo i tre aspetti:

- *Sintattico*: l'analisi viene effettuata dal punto di vista sintattico
- *Effettivo*: si può definire un algoritmo che effettui questa analisi
- *Strutturale*: i tipi assegnati si ottengono in maniera **composizionale** dalle sottoespressioni.

### 2.3 Come funziona?

Un sistema dei tipi associa dei tipi ai valori calcolati. Esaminando il flusso dei valori calcolati prova a dimostrare che non avvengano errori (di tipo, non in generale)facendo un controllo, che può avvenire in due modi:

- *Statico*: avviene in fase di compilazione, non degradando le prestazioni
- *Dinamico*: avviene in fase di esecuzione e aumenta il tempo di esecuzione

## **2.4 Come si progetta?**

### **2.4.1 Specifiche**

### **2.4.2 Dimostrazione**