



## Capitolo 4.4 Haskell Tuple types

Corso di Laurea Magistrale in Ingegneria Informatica e  
dell'Automazione

Anno accademico 2019/2020

Prof. MARCO GAVANELLI

QUESTO MATERIALE DIDATTICO È PER USO  
PERSONALE DELLO STUDENTE ED È COPERTO  
DA COPYRIGHT. NE È SEVERAMENTE VIETATA  
LA RIPRODUZIONE O IL RIUTILIZZO ANCHE  
PARZIALE, AI SENSI E PER GLI EFFETTI DELLA  
LEGGE SUL DIRITTO D'AUTORE.

Part of these slides were adapted  
from the material of the book  
Graham Hutton, Programming in  
Haskell, Cambridge University  
Press, 2<sup>nd</sup> edition, 2016



### Note:

- The type of a tuple encodes its size:

```
(False, True)      :: (Bool, Bool)
```

```
(False, True, False) :: (Bool, Bool, Bool)
```

- The type of the components is unrestricted:

```
('a', (False, 'b')) :: (Char, (Bool, Char))
```

```
(True, ['a', 'b'])  :: (Bool, [Char])
```

18

## Tuple Types

A tuple is a sequence of values of different types:

```
(False, True)      :: (Bool, Bool)
```

```
(False, 'a', True) :: (Bool, Char, Bool)
```

In general:

$(t_1, t_2, \dots, t_n)$  is the type of  $n$ -tuples whose  $i$ -th  
components have type  $t_i$  for any  $i$  in  $1 \dots n$ .

17

## Accessing Tuple Elements

`fst` retrieves the first element

```
Prelude> fst (1,2)  
1
```

`snd` retrieves the second element

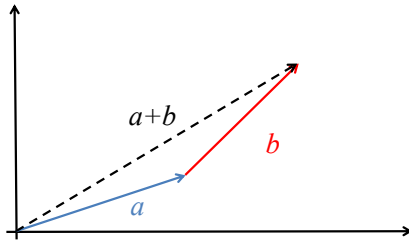
```
Prelude> snd (1,2)  
2
```

Only for  
2-tuples!  
Work only  
for tuples of  
exactly 2  
elements

19

## Esercizio

- Scrivere una funzione `addVectors` che ha come parametri due vettori nel piano e ne calcola la somma



20

## Pattern matching on tuples

- Sum of two vectors:

```
addVectors (x1, y1) (x2, y2) =  
  (x1 + x2, y1 + y2)
```

- extract the elements from a 3-tuple

```
first (x, _, _) = x  
second (_, y, _) = y  
third (_, _, z) = z
```

21

## zip'

- Si scriva una funzione `zip'` che, date due liste, fornisce una lista di coppie. La coppia in posizione *i*-esima della lista è costituita da
  - l'elemento *i*-esimo della prima lista
  - l'elemento *i*-esimo della seconda lista
- Se le due liste hanno lunghezza diversa, `zip'` restituisce una lista con la lunghezza minima delle due
- Es:  

```
> zip' [1,2,3] ['a','b']  
[(1, 'a'), (2, 'b')]
```

22