# ammare nel $\lambda$-calcolo

uca Padovani
Linguaggi e Paradigmi di Programmazione

# valori booleani

## Definizione

- $\mathtt{TRUE} \stackrel{\text{def}}{=} \lambda x.\lambda y.x$ *valore booleano "vero"*
- $\mathtt{FALSE} \stackrel{\text{def}}{=} \lambda x.\lambda y.y$ *valore booleano "falso"*
- $\mathtt{IF} \stackrel{\text{def}}{=} \lambda z.z$ *espressione condizionale if-then-else*

## Proposizione

1. $\mathtt{IF}\ \mathtt{TRUE}\ M\ N \Leftrightarrow M$
2. $\mathtt{IF}\ \mathtt{FALSE}\ M\ N \Leftrightarrow N$

## Dimostrazione.

1. $\mathtt{IF}\ \mathtt{TRUE}\ M\ N \to \mathtt{TRUE}\ M\ N \to (\lambda y.M)\ N \to M$
2. $\mathtt{IF}\ \mathtt{FALSE}\ M\ N \to \mathtt{FALSE}\ M\ N \to (\lambda y.y)\ N \to N$ $\square$

### Esercizio

- qual è il problema di usare questo `IF` in un linguaggio zelante?

# operatori logici

## Definizione

- $\mathtt{AND} \stackrel{\text{def}}{=} \lambda x.\lambda y.\mathtt{IF}\ x\ y\ \mathtt{FALSE}$ *congiunzione logica*
- $\mathtt{OR} \stackrel{\text{def}}{=} \lambda x.\lambda y.\mathtt{IF}\ x\ \mathtt{TRUE}\ y$ *disgiunzione logica*
- $\mathtt{NOT} \stackrel{\text{def}}{=} \lambda x.\mathtt{IF}\ x\ \mathtt{FALSE}\ \mathtt{TRUE}$ *negazione logica*

## Proposizione

1. $\mathtt{AND}\ \mathtt{TRUE}\ \mathtt{TRUE} \Leftrightarrow \mathtt{TRUE}$
2. $\mathtt{AND}\ \mathtt{TRUE}\ \mathtt{FALSE} \Leftrightarrow \mathtt{FALSE}$
3. $\mathtt{AND}\ \mathtt{FALSE}\ \mathtt{TRUE} \Leftrightarrow \mathtt{FALSE}$
4. $\mathtt{AND}\ \mathtt{FALSE}\ \mathtt{FALSE} \Leftrightarrow \mathtt{FALSE}$

## Dimostrazione di 2.

$$\begin{aligned}
\mathtt{AND}\ \mathtt{TRUE}\ \mathtt{FALSE} &\to (\lambda y.\mathtt{IF}\ \mathtt{TRUE}\ y\ \mathtt{FALSE})\ \mathtt{FALSE} \\
&\to \mathtt{IF}\ \mathtt{TRUE}\ \mathtt{FALSE}\ \mathtt{FALSE} \Rightarrow \mathtt{FALSE} \qquad \square
\end{aligned}$$

# coppie

## Definizione

- $\mathtt{PAIR} \stackrel{\text{def}}{=} \lambda x.\lambda y.\lambda z.z\ x\ y$ *costruttore delle coppie*
- $\mathtt{FST} \stackrel{\text{def}}{=} \lambda p.p\ \mathtt{TRUE}$ *prima componente di una coppia*
- $\mathtt{SND} \stackrel{\text{def}}{=} \lambda p.p\ \mathtt{FALSE}$ *seconda componente di una coppia*

## Proposizione

1. $\mathtt{FST}\ (\mathtt{PAIR}\ M\ N) \Leftrightarrow M$
2. $\mathtt{SND}\ (\mathtt{PAIR}\ M\ N) \Leftrightarrow N$

## Dimostrazione di 1.

$$\begin{array}{rcl} \mathtt{FST}\ (\mathtt{PAIR}\ M\ N) & \to & \mathtt{PAIR}\ M\ N\ \mathtt{TRUE} \\ & \to & (\lambda y.\lambda z.z\ M\ y)\ N\ \mathtt{TRUE} \\ & \to & (\lambda z.z\ M\ N)\ \mathtt{TRUE} \\ & \to & \mathtt{TRUE}\ M\ N \\ & \Rightarrow & M \end{array} \qquad \square$$

# numeri naturali (codifica di Church)

## Definizione

*Dato $k \in \mathbb{N}$ scriviamo $M^k\ N$ per $\underbrace{M\ (M\ (\cdots (M}_{k\ volte}\ N)))$. In particolare*

$M^0\ N = N$.

- $\underline{n} \stackrel{\text{def}}{=} \lambda f.\lambda x.f^n\ x$ *codifica del numero naturale n*
- $\mathtt{SUCC} \stackrel{\text{def}}{=} \lambda a.\lambda f.\lambda x.a\ f\ (f\ x)$ *funzione successore*

## Proposizione

$\mathtt{SUCC}\ \underline{n} \Leftrightarrow \underline{n+1}$

## Dimostrazione per induzione su *n*.

| | |
|---|---|
| $\mathtt{SUCC}\ \underline{n} \rightarrow \lambda f.\lambda x.\underline{n}\ f\ (f\ x)$ | definizione di $\mathtt{SUCC}$ |
| $\Rightarrow \lambda f.\lambda x.f^n\ (f\ x)$ | definizione di $\underline{n}$ e $\beta$-riduzioni |
| $= \lambda f.\lambda x.f^{n+1}\ x$ | definizione di $f^{n+1}\ x$ |
| $= \underline{n+1}$ | definizione di $\underline{n+1}$ $\square$ |

# operazioni sui numeri naturali (1/3)

## Definizione

- $\mathtt{ADD} \stackrel{\text{def}}{=} \lambda a.\lambda b.b\ \mathtt{SUCC}\ a$ — *somma*
- $\mathtt{MUL} \stackrel{\text{def}}{=} \lambda a.\lambda b.b\ (\mathtt{ADD}\ a)\ \underline{0}$ — *moltiplicazione*
- $\mathtt{EXP} \stackrel{\text{def}}{=} \lambda a.\lambda b.b\ (\mathtt{MUL}\ a)\ \underline{1}$ — *elevamento a potenza*

## Proposizione

1. $\mathtt{ADD}\ \underline{m}\ \underline{n} \Leftrightarrow \underline{m+n}$
2. $\mathtt{MUL}\ \underline{m}\ \underline{n} \Leftrightarrow \underline{m \times n}$
3. $\mathtt{EXP}\ \underline{m}\ \underline{n} \Leftrightarrow \underline{m^n}$

## Dimostrazione di 1.

$$\begin{array}{llll} \mathtt{ADD}\ \underline{m}\ \underline{n} & \Rightarrow & \underline{n}\ \mathtt{SUCC}\ \underline{m} & \text{definizione di } \mathtt{ADD} \\ & \Rightarrow & \mathtt{SUCC}^n\ \underline{m} & \text{definizione di } \underline{n} \\ & \Leftrightarrow & \underline{m+n} & \text{proprietà di } \mathtt{SUCC} \quad \square \end{array}$$

# operazioni sui numeri naturali (2/3)

Calcolare il **predecessore** di un numero è difficile perché occorre "rimuovere" un'applicazione. L'idea è di calcolare una sequenza di $n$ coppie

$$(0,0); (0,1); (1,2); \dots; (n-1, n)$$

e poi estrarre $n-1$ dall'$n$-esima coppia

## Definizione (predecessore)

- $\mathtt{NEXT} \stackrel{\text{def}}{=} \lambda p.\mathtt{PAIR}\,(\mathtt{SND}\,p)\,(\mathtt{SUCC}\,(\mathtt{SND}\,p))$
- $\mathtt{PRED} \stackrel{\text{def}}{=} \lambda a.\mathtt{FST}\,(a\;\mathtt{NEXT}\,(\mathtt{PAIR}\,\underline{0}\;\underline{0}))$

## Proposizione

1. $\mathtt{NEXT}\,(\mathtt{PAIR}\,\underline{m}\;\underline{n}) \Leftrightarrow \mathtt{PAIR}\,\underline{n}\;\underline{n+1}$
2. $\mathtt{PRED}\,\underline{n} \Leftrightarrow \begin{cases} \underline{0} & se\ n = 0 \\ \underline{n-1} & se\ n > 0 \end{cases}$

### Esercizio

- dimostrare la proposizione

# operazioni sui numeri naturali (3/3)

## Definizione

$\mathtt{ISZERO} \stackrel{\text{def}}{=} \lambda a.a\,(\lambda x.\mathtt{FALSE})\ \mathtt{TRUE}$ *test per zero*

## Proposizione

$$\mathtt{ISZERO}\ \underline{n} \Leftrightarrow \begin{cases} \mathtt{TRUE} & se\ n = 0 \\ \mathtt{FALSE} & se\ n \neq 0 \end{cases}$$

## Dimostrazione.

$$\begin{array}{rcll} \mathtt{ISZERO}\ \underline{0} & \to & \underline{0}\,(\lambda x.\mathtt{FALSE})\ \mathtt{TRUE} & \text{def. di } \mathtt{ISZERO} \\ & \Leftrightarrow & \mathtt{TRUE} & \text{definizione di } \underline{0} \\ \\ \mathtt{ISZERO}\ \underline{n+1} & \to & \underline{n+1}\,(\lambda x.\mathtt{FALSE})\ \mathtt{TRUE} & \text{def. di } \mathtt{ISZERO} \\ & \Leftrightarrow & (\lambda x.\mathtt{FALSE})^{n+1}\ \mathtt{TRUE} & \text{def. di } \underline{n+1} \\ & = & (\lambda x.\mathtt{FALSE})\ ((\lambda x.\mathtt{FALSE})^{n}\ \mathtt{TRUE}) & \text{def. di } M^{n+1} \\ & \to & \mathtt{FALSE} & \beta\text{-riduzione} \quad \square \end{array}$$

# ricorsione, per tentativi

$$\texttt{FACT} \stackrel{\text{def}}{=} \lambda a.\texttt{IF}\ (\texttt{ISZERO}\ a)\ \underline{1}\ (\texttt{MUL}\ a\ (\texttt{FACT}\ (\texttt{PRED}\ a)))$$

- non è una definizione (**FACT** compare anche a destra)

$$\texttt{FACT} \stackrel{\text{def}}{=} (\lambda f.\lambda a.\texttt{IF}\ (\texttt{ISZERO}\ a)\ \underline{1}\ (\texttt{MUL}\ a\ (f\ (\texttt{PRED}\ a))))\ \texttt{FACT}$$

- non è ancora una definizione, ma ricorda qualcosa…

### Definizione (punto fisso)

*$x$ è punto fisso di $F$ se $x = F(x)$*

### Idea

1. $\texttt{AUX} \stackrel{\text{def}}{=} \lambda f.\lambda a.\texttt{IF}\ (\texttt{ISZERO}\ a)\ \underline{1}\ (\texttt{MUL}\ a\ (f\ (\texttt{PRED}\ a)))$
2. definire una funzione **FIX** per calcolare punti fissi
3. definire **FACT** come **FIX AUX**

# operatore di punto fisso

## Definizione (operatore di punto fisso)

$\mathtt{FIX} \stackrel{\text{def}}{=} \lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$

## Proposizione

$\mathtt{FIX}\ M \Leftrightarrow M\ (\mathtt{FIX}\ M)$ $\qquad$ $\mathtt{FIX}\ M$ *è (convertibile a) un punto fisso di* $M$

## Dimostrazione.

$$
\begin{array}{llll}
\mathtt{FIX}\ M & \rightarrow & (\lambda x.M\ (x\ x))\ (\lambda x.M\ (x\ x)) & \beta\text{-riduzione} \\
& \rightarrow & M\ ((\lambda x.M\ (x\ x))\ (\lambda x.M\ (x\ x))) & \beta\text{-riduzione} \\
& \leftarrow & M\ (\mathtt{FIX}\ M) & \beta\text{-riduzione} \quad \square
\end{array}
$$

# usare FIX per definire funzioni ricorsive

## Definizione

$\mathtt{FACT} \stackrel{\text{def}}{=} \mathtt{FIX}\ \mathtt{AUX}$

## Proposizione

$\mathtt{FACT} \Leftrightarrow \lambda a.\mathtt{IF}\ (\mathtt{ISZERO}\ a)\ \underline{1}\ (\mathtt{MUL}\ a\ (\mathtt{FACT}\ (\mathtt{PRED}\ a)))$

## Dimostrazione.

$$\begin{aligned}
\mathtt{FACT} &= \mathtt{FIX}\ \mathtt{AUX} && \text{def. di } \mathtt{FACT} \\
&\Leftrightarrow \mathtt{AUX}\ (\mathtt{FIX}\ \mathtt{AUX}) && \text{prop. di } \mathtt{FIX} \\
&= \mathtt{AUX}\ \mathtt{FACT} && \text{def. di } \mathtt{FACT} \\
&\to \lambda a.\mathtt{IF}\ (\mathtt{ISZERO}\ a)\ \underline{1}\ (\mathtt{MUL}\ a\ (\mathtt{FACT}\ (\mathtt{PRED}\ a))) && \beta\text{-rid.} \quad \square
\end{aligned}$$