Valori booleani

Iniziamo a definire i valori booleani TRUE,FALSE e il costrutto IF

$$\mathrm{TRUE} \stackrel{\text{def}}{=} \lambda x.\lambda y.x$$
$$\mathrm{FALSE} \stackrel{\text{def}}{=} \lambda x.\lambda y.y$$
$$\mathrm{IF} \stackrel{\text{def}}{=} \lambda z.z$$

1. **TRUE:** dati due argomenti, restituisce sempre il **primo**
2. **FALSE:** dati due argomenti, restituisce sempre il **secondo**
3. **IF:** banalmente, è la funzione identità

Proposizioni booleane

1. $\mathrm{IF}\ \mathrm{TRUE}\ M\ N \Leftrightarrow M$
   Da intendersi come $((\mathrm{IF})\mathrm{TRUE})\mathrm{FALSE})M\ N$
2. $\mathrm{IF}\ \mathrm{FALSE}\ M\ N \Leftrightarrow N$

Dimostrazione 1° proposizione:

$\overset{IF}{\lambda z.\, z}\ \ \overset{TRUE}{\lambda x.\, \lambda y.\, x}\ \ M\ N \rightarrow ((((\overset{IF}{\lambda z.\, z})\overset{TRUE}{\lambda x.\, \lambda y.\, x})\ M)\ N) \rightarrow ((\overset{TRUE}{\lambda x.\, \lambda y.\, x})\ M)N \rightarrow (\lambda y.\, M)N \rightarrow M$

Dimostrazione 2° proposizione:

$\overset{IF}{\lambda z.\, z}\overset{FALSE}{\lambda x.\, \lambda y.\, y}\ \ M\ N \rightarrow ((((\overset{IF}{\lambda z.\, z})\overset{FALSE}{\lambda x.\, \lambda y.\, y})\ M)\ N) \rightarrow ((\overset{FALSE}{\lambda x.\, \lambda y.\, y})\ M)N \rightarrow (\lambda y.\, y)N \rightarrow N$

Operatori logici

Attraverso le definizione dei valori booleani, possiamo definirci sopra anche gli operatori logici:

$$\mathtt{AND} \stackrel{\text{def}}{=} \lambda x.\lambda y.\mathtt{IF}\ x\ y\ \mathtt{FALSE}$$
$$\mathtt{OR} \stackrel{\text{def}}{=} \lambda x.\lambda y.\mathtt{IF}\ x\ \mathtt{TRUE}\ y$$
$$\mathtt{NOT} \stackrel{\text{def}}{=} \lambda x.\mathtt{IF}\ x\ \mathtt{FALSE}\ \mathtt{TRUE}$$

Il loro funzionamento è lo stesso di qualsiasi altro linguaggio di

programmazione.
Accetta due(o uno) **parametri booleani** e restituisce il loro **AND/OR/NOT** logico.

Dunque le seguenti proposizioni sono valide:

$$\mathrm{AND\ TRUE\ TRUE} \Leftrightarrow \mathrm{TRUE}$$
$$\mathrm{AND\ TRUE\ FALSE} \Leftrightarrow \mathrm{FALSE}$$
$$\mathrm{AND\ FALSE\ TRUE} \Leftrightarrow \mathrm{FALSE}$$
$$\mathrm{AND\ FALSE\ FALSE} \Leftrightarrow \mathrm{FALSE}$$

Dimostrazione 2° proposizione:

$$\overset{AND}{((\lambda x.\,\lambda y.\,\underline{\overset{IF}{\lambda z.\,z}}\ x\ y\ \underline{\overset{FALSE}{\lambda x.\,\lambda y.\,y}})\ \underline{\overset{TRUE}{\lambda x.\,\lambda y.\,x}})\ \overset{FALSE}{\lambda x.\,\lambda y.\,y} \to_\beta}$$
$$\overset{IF}{\to_\beta ((\lambda y.\,\lambda z.\,z}\underline{\overset{TRUE}{\lambda x.\,\lambda y.\,x}}\ \ y\ \underline{\overset{FALSE}{\lambda x.\,\lambda y.\,y}})\ \overset{FALSE}{\lambda x.\,\lambda y.\,y}$$
$$\overset{IF}{\to_\beta ((\lambda y.\,\lambda z.\,z}\underline{\overset{TRUE}{\lambda x.\,\lambda y.\,x}}\quad \underline{\overset{FALSE}{\lambda x.\,\lambda y.\,y}}\overset{FALSE}{\lambda x.\,\lambda y.\,y})$$

ma sappiamo da prima che $\mathrm{IF\ TRUE}\ M \to M$ quindi
$\mathrm{IF\ TRUE\ FALSE\ FALSE}$ si riduce a $\mathrm{FALSE\ FALSE}$ ovvero FALSE

Coppie

Definiamo il costruttore delle **Coppie** e i metodi **First** e **Second** che restituiscono rispettivamente il **primo** e **secondo** elemento

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

Proposizioni

$$\mathrm{FST}\ (\mathrm{PAIR}\ M\ N) \Leftrightarrow M$$
$$\mathrm{SND}\ (\mathrm{PAIR}\ M\ N) \Leftrightarrow N$$

Dimostrazioni 1° Proposizione:

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

La dimostrazione è abbastanza intuitiva: espandendo le varie definizioni e effettuando una β-riduzione, otteniamo $\mathrm{PAIR\ M\ N\ TRUE}$.
Effettuando ulteriori β-riduzioni è facile vedere come il risultato sia M stesso.

Numeri naturali (Codifica di Churc)

L'idea di base è di rappresentare un qualsiasi numero n come una funzione f **applicata** ad un parametro x per n **volte**.
Esempio: il numero 5 sarebbe $f(f(f(f(f(5)))))$

Definizione

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

$M^0\ N = N$.

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

Il **successore** banalmente è $\underline{n+1}$
Dimostrazione di $\mathrm{SUCC}\ \underline{2}$**:**
$\underline{2} = \lambda f.\,\lambda x.\,f(f\ x)$
$\mathrm{SUCC}\ \underline{2} = \underbrace{(\lambda a.\,\lambda f.\,\lambda x.\,a\ f(f\ x))}^{SUCC}\ \ \underbrace{(\lambda f.\,\lambda x.\,f(f\ x)}^{\underline{2}} \to_\beta$
$\to_\beta \lambda f.\,\lambda x.\,(\lambda f.\,\lambda x.\,f(f\ x))f(f\ x)$
$\to_\beta \lambda f.\,\lambda x.\,(\lambda x.\,f(f\ x))(f\ x)$
$\to_\beta \lambda f.\,\lambda x.\,(f(f\ f(x))) = \underline{3}$

ADD,MUL ed EXP

Definizione

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

Dimostrazione di 1.

$$
\begin{aligned}
\texttt{ADD}\ \underline{m}\ \underline{n} &\Rightarrow \underline{n}\ \texttt{SUCC}\ \underline{m} && \text{definizione di } \texttt{ADD} \\
&\Rightarrow \texttt{SUCC}^n\ \underline{m} && \text{definizione di } \underline{n} \\
&\Leftrightarrow \underline{m+n} && \text{proprietà di } \texttt{SUCC}
\end{aligned}
$$

□

Dal momento che un numerale $\underline{n}$ di Church altro non è che una funzione applicata n volte, l'addizione di due numeri m e n la posso pensare come la funzione $SUCC$ applicata n volte sul numero m

Predecessore

L'idea è di calcolare una sequenza di n coppie
$(0,0);(0,1);(1,2);\ldots;(n-1,n)$
e poi estrarre $n-1$ dall'n-esima coppia.

Definizione (predecessore)

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

NEXT prende in input una coppia P_1 e ne crea un'altra prendendo, come primo elemento, il secondo elemento di P_1 e come secondo elemento, il successore del primo elemento preso.
PRED prende in input un numero n e crea una coppia formata da $(n-1,n)$, infine restituisce il primo elento della coppia.

Proposizione

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

Imponiamo che il predecessore di 0 sia 0 stesso

Numero 0

L'idea di base è iterare a volte la funzione costante $\lambda x.\, FALSE$, con caso base $TRUE$.
Se la itero 0 volte, banalmente la funzione scompare.

Definizione

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

Dimostrazione ISZERO 0:

$(\lambda a.\, a(\lambda x.\,\mathrm{FALSE})\mathrm{TRUE})\ \underline{0} \rightarrow_\beta$
$(\lambda f.\, \overset{\underline{0}}{\lambda x}\ x)\ (\lambda x.\,\mathrm{FALSE})\mathrm{TRUE} \rightarrow_\beta$
$(\lambda x.\, x)\mathrm{TRUE} = \mathrm{TRUE}$

Dimostrazione ISZERO 1:

$(\lambda a.\, a(\lambda x.\,\mathrm{FALSE})\mathrm{TRUE})\ \underline{1} \rightarrow_\beta$
$(\lambda f.\, \overset{\underline{1}}{\lambda x}\ f\ x)\ (\lambda x.\,\mathrm{FALSE})\mathrm{TRUE} \rightarrow_\beta$
$(\lambda x.\, (\lambda x.\,\mathrm{FALSE})x)\mathrm{TRUE} \rightarrow_\beta$
$(\lambda x.\,\mathrm{FALSE})\mathrm{TRUE} = \mathrm{FALSE}$

$\texttt{ISZERO}\ \underline{n+1}$	$\rightarrow$	$\underline{n+1}\ (\lambda x.\texttt{FALSE})\ \texttt{TRUE}$	def. di ISZERO
	$\Leftrightarrow$	$(\lambda x.\texttt{FALSE})^{n+1}\ \texttt{TRUE}$	def. di $\underline{n+1}$
	$=$	$(\lambda x.\texttt{FALSE})\ ((\lambda x.\texttt{FALSE})^{n}\ \texttt{TRUE})$	def. di M^{n+1}
	$\rightarrow$	$\texttt{FALSE}$	β-riduzione $\square$

Generalizzazione di ISZERO (N+1)

Ricorsione: fattoriale

Una prima idea per realizzare il fattoriale potrebbe essere:

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

E' quasi pseudocodice:
if(iszero(n)) then 1 else(MUL(1,Fact(n-1)))

Tuttavia non va bene come definizione, dal momento che stiamo definendo **FACT** in funzione di **FACT** stesso.

Semplificandola otteniamo

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

In questo modo abbiamo una redex ben definita

Dobbiamo trovare quindi $x = F(x)$ dove x è **FACT** mentre **F** è la mia redex definita prima.

Ovvero dobbiamo calcolare il punto fisso di una funzione($x = F(X)$).

Definiamo quindi una funzione ausiliaria **AUX:**

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

e definiamo **FACT** come **FIX AUX** dove **FIX** è una funzione per trovare i **punti fissi:**

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

Proposizione

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

Dimostrazione:

Dimostrazione.

$$\begin{array}{lcll} \texttt{FIX}\ M & \to & (\lambda x.M\ (x\ x))\ (\lambda x.M\ (x\ x)) & \beta\text{-riduzione} \\ & \to & M\ ((\lambda x.M\ (x\ x))\ (\lambda x.M\ (x\ x))) & \beta\text{-riduzione} \\ & \leftarrow & M\ (\texttt{FIX}\ M) & \beta\text{-riduzione} \end{array}$$

□

Ora possiamo tornare alla funzione **fattoriale:**

Definizione

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

Proposizione

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

Dimostrazione.

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

□