Indice

1 Introduzione

Il λ-calcolo è un sistema formale sviluppato negli anni '30 da Alonzo Church. Lo scopo originario era quello di fondare la matematica, e non è stato raggiunto in quanto il sistema si rivelò inconsistente, come dimostrato da Kleene e Rosser nel 1936. Un sottoinsieme di tale sistema si è comunque sviluppato per la sua capacità di esprimere computazioni mediante astrazione su variabili e sostituzione. Lo studio del λ-calcolo è dunque lo studio di entità dette λ-termini che svolgono allo stesso tempo il ruolo di programmi e di dati su cui i programmi lavorano. Sui termini si considererà una relazione di ordine, detta riduzione, che rappresenta l'esecuzione dei programmi. All'interno del λ-calcolo, con opportune codifiche, è possibile rappresentare i numeri naturali e tutte le funzioni calcolabili. Poichè l'insieme delle funzioni calcolabili totali non è ricorsivamente enumerabile esistono dei termini la cui computazione non termina (diremo che non sono normalizzanti).

Il λ-calcolo tipato è una variante del λ-calcolo in cui ad ogni termine è associata un'entità sintattica detta tipo. Esso ha origine nei lavori di Haskell Curry (1934) e di Church (1940). La riduzione in questo caso è ridefinita aggiungendo vincoli sul come è possibile comporre i termini in base al loro tipo, e in particolare rende chiara la classe di dati su cui ciascun programma può operare. Come conseguenza si può dimostrare che questa variante ha la proprietà di normalizzazione, ovvero tutte le computazioni terminano e tutti i termini sono normalizzanti. I tipi sono studiati anche in ambito informatico per la verifica in modo automatico della presenza di alcuni errori che dovrebbero essere altrimenti cercati a mano dal programmatore.

In questa tesi ci occuperemo di una variante del calcolo detta sistema F, anche nota come λ-calcolo polimorfico o λ-calcolo del secondo ordine. Essa è stata sviluppata dal logico Jean-Yves Girard (1972) e dall'informatico John Charles Reynolds (1974). Il sistema F è essenzialmente una variante del λ-calcolo tipato in cui viene aggiunta una quantificazione universale sui tipi.

Anche per il sistema F vale la proprietà di normalizzazione. Troveremo dunque che le funzioni rappresentabili nel sistema F sono solo un sottoinsieme delle funzioni calcolabili totali, e ne daremo una caratterizzazione più precisa: esse sono esattamente le funzioni di cui l'aritmetica di Heyting del secondo ordine dimostra la totalità. Mostreremo quindi un esempio di funzione non rappresentabile nel sistema F e dedurremo dunque la consistenza dell'aritmetica di Peano del secondo ordine a partire dal risultato di normalizzazione.

Metteremo inoltre tale risultato a confronto con un risultato equivalente su un'altra variante del λ-calcolo, detta sistema T di Gödel, per cui vale ugualmente la proprietà di normalizzazione. Nel sistema T infatti le funzioni rappresentabili sono esattamente quelle che l'aritmetica di Heyting del primo ordine dimostra essere totali.

2 Aritmetiche di Peano e di Heyting

Iniziamo con l'introdurre alcune costruzioni fondamentali in logica di cui faremo uso nelle sezioni successive.

2.1 Primo Ordine

Definizione 2.1 Dato un linguaggio L formato da simboli di costante, di funzione di relazione, definiamo i termini induttivamente come

- Le costanti;
- Simboli $x_1, x_2, \ldots$ dette varibili individuali o semplicemente variabili;
- Elementi nella forma $f(t_1, \ldots, t_k)$, dove f è un simbolo di funzione k-aria e gli t_k sono termini.

Dato un simbolo di relazione k-aria R e k termini $t_1, \ldots, t_k$, diciamo che $R(t_1, \ldots, t_k)$ è una formula atomica. Inoltre anche il simbolo $\perp$, detto assurdo, denota una formula atomica.

Infine introduciamo le formule, definite induttivamente come:

- Le formule atomiche;
- $\varphi \wedge \psi$, $\varphi \vee \psi$ e $\varphi \to \psi$, dove φ e ψ sono formule.
- Se x è una variabile e φ è una formula, allora anche $\forall x \varphi$ e $\exists x \varphi$ sono formule.

Inoltre possiamo aggiungere una notazione per la negazione, ovvero $\neg\varphi = \varphi \to \perp$ per ogni formula φ.

Dato un insieme finito di formule Γ e una formula φ, un giudizio è la loro coppia, scritta nella forma $\Gamma \vdash \varphi$. Per comodità scriveremo $\vdash \varphi$ al posto di $\vdash \varphi$ e $\Gamma, \psi \vdash \varphi$ al posto di $\Gamma \cup \{\psi\} \vdash \varphi$.

Diciamo che una variabile x appare legata in una formula φ se $\forall x \eta$ compare nella costruzione induttiva di φ, per qualche formula η.

Una dimostrazione di un giudizio $\Gamma \vdash \varphi$ è una successione di giudizi $\Gamma_i \vdash \varphi_i$ con $i = 1, \vdash, n$ tale $\Gamma_n = \Gamma$ e $\varphi_n = \varphi$ e inoltre per ogni i l'i-esimo giudizio è un'istanza della regola (Ax), o esiste $j < i$ tale che

$$\frac{\Gamma_j \vdash \varphi_j}{\Gamma_i \vdash \varphi_i}$$

sia un'istanza delle regole della deduzione, oppure esistano $j, j' < i$ tali che

$$\frac{\Gamma_j \vdash \varphi_j \qquad \Gamma_{j'} \vdash \varphi_{j'}}{\Gamma_i \vdash \varphi_i}$$

sia un'istanza delle regole oppure un fatto equivalente con tre ipotesi. Elenchiamo in seguito alcune della regole per la deduzione naturale:

$$\frac{}{\Gamma, \varphi \vdash \varphi} \text{ Ax}$$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi} \to\text{I} \qquad \frac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi} \to\text{E}$$

$$\frac{\Gamma, \varphi \qquad \Gamma, \psi}{\Gamma \vdash \varphi \wedge \psi} \wedge\text{I} \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \wedge\text{E1} \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi} \wedge\text{E2}$$

$$\frac{\Gamma, \varphi}{\Gamma \vdash \varphi \vee \psi} \vee\text{I1} \qquad \frac{\Gamma, \psi}{\Gamma \vdash \varphi \vee \psi} \vee\text{I2} \qquad \frac{\Gamma, \varphi \vdash \theta \qquad \Gamma, \psi \vdash \theta \qquad \Gamma \vdash \varphi \vee \psi}{\Gamma \vdash \theta} \vee\text{E}$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x \varphi} \forall\text{I } (x \text{ non libera in } \Gamma) \qquad \frac{\Gamma \vdash \forall x \varphi}{\Gamma \vdash \varphi[a/x]} \forall\text{E}$$

$$\frac{\Gamma \vdash \varphi[a/x]}{\Gamma \vdash \exists x \varphi} \exists\text{I} \qquad \frac{\Gamma \vdash \exists x \varphi \qquad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi} \exists\text{E } (x \text{ non libera in } \Gamma \text{ nè in } \psi)$$

in cui a è un qualunque termine. Inoltre la scrittura $\varphi[a/x]$ indica la sostituzione di a in tutte le occorrenze libere di x nella formula φ. In questo caso si deve porre l'attenzione di non avere la cattura di variabili, ovvero se y è una variabile che compare in a essa non deve diventare legata in seguito alla sostituzione. Questo comunque non è restrittivo in quanto ci interessa considerare le formule a meno del rinominarne le variabili legate. Formule ottenute rinominando tali variabili si dicono α-equivalenti.

Aggiungendo a quelle presentate la regola

$$\frac{\Gamma, \varphi \to \bot \vdash \bot}{\Gamma \vdash \varphi} \bot\text{E}$$

otteniamo il sistema della logica classica, mentre aggiungendo la regola

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi} \bot\text{E}$$

otteniamo il sistema per la logica intuizionista.

Consideriamo ora un linguaggio, detto linguaggio dell'aritmetica, che ha come unica costante O, il simbolo di funzione successore S e una relazione binaria di uguaglianza $=$. Aggiungiamo inoltre simboli di funzione binaria per somma e moltiplicazione.

Una formula si dice Δ_0 se i quantificatori che vi compaiono sono solo della forma $\forall x(x < t \to \varphi)$ o $\exists x(x < t \wedge \varphi)$, dove la formula $x < t$ indica $\exists n(t = x + n)$. Una formula si dice Σ^0_1 se è nella forma $\exists x\eta$ in cui η è una formula Δ_0. Infine una formula si dice Π^0_1 se è nella forma $\forall x\psi$ dove ψ è una formula Σ^0_1.

Consideriamo il seguente sistema di assiomi:

- $\forall x(x = x)$;
- $\forall x\forall y(x = y \to y = x)$;
- $\forall x\forall y\forall z(x = y \to y = z \to x = z)$;
- $\forall x\forall y(x = y \to Sx = Sy)$;
- $\forall x\forall y(Sx = Sy \to x = y)$;
- $\forall x(Sx = O \to \bot)$;
- $\forall x(a + O = a)$;
- $\forall x\forall y(a + Sb = S(a + b))$;
- $\forall x(x \cdot O = O)$;
- $\forall x\forall y(x \cdot y = x \cdot y + x)$;
- $\forall x(\varphi(x) \to \varphi(Sx)) \to \varphi(O) \to \forall x\varphi(x)$.

I primi tre assiomi indicano che l'uguaglianza è una relazione di equivalenza. I successivi due esprimono la compatibilità tra successore e uguaglianza. Infine si definisce il comportamento di somma e prodotto e si dà l'assioma di induzione, valido per ogni formula φ.

Considerando tali assiomi nel sistema della logica classica formano l'aritmetica di Peano, considerandoli nel sistema della logica intuizionista formano l'aritmetica di Hayting. Presentiamo un risultato che tornerà utile e che approfondiremo nel caso della logica e delle aritmetiche del secondo ordine:

Teorema 2.2 Una formula Π^0_2 è dimostrabile nell'aritmetica di Hayting se e solo se è dimostrabile nell'aritmetica di Peano.

Risulta inoltre che l'aritmetica di Peano è equivalente al sistema della logica classica privata del quantificatore $\vee$ con gli assiomi di peano. Questo è vero perchè nella logica classica $\varphi \vee \psi$ è equivalente a $\neg(\neg\varphi \wedge \neg\psi)$. Ovvero ogni formula del primo sistema è dimostrabile se e solo se la formula ottenuta traducendo tutte le occorrenze di $\vee$ è dimostrabile nel secondo sistema.

È vero anche un fatto simile per l'aritmetica di Hayting, che però sfrutta per intero gli assiomi. Infatti in tale sistema $\varphi \vee \psi$ è equivalente a

$$\exists x((x = 0 \to \varphi) \wedge (\neg(a = 0) \to \psi)).$$

2.2 Secondo Ordine

Definizione 2.3 Il linguaggio per la logica del secondo ordine è lo stesso di quello del primo ordine con l'aggiunta per ogni naturale n di un insieme numerabile di simboli X_k^n con $k \in \mathbb{N}$, che chiameremo variabili di relazione.

Le formule atomiche sul linguaggio L sono $\perp$, espressioni della forma $X_k^n(t_1, \ldots, t_n)$, dove i t_i sono termini del linguaggio, oppure espressioni della forma $R(t_1, \ldots, t_n)$, dove R è un simbolo di relazione in L e i t_i sono nuovamente termini.

Le formule sono definite induttivamente come:

- Le formule atomiche.
- Date φ e ψ formule, sono formule anche $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$.
- Data una formula φ e una variabile x, sono formule anche $\forall x\varphi$ e $\exists x\varphi$.
- Data una formula φ e una variabile di relazione X, sono formule anche $\forall X\varphi$ e $\exists X\varphi$.

Si definisce inoltre la formula $\neg\varphi$ come $\varphi \to \perp$.

In modo naturale possiamo definire il concetto di variabili libere in una formula:

Definizione 2.4

- Le variabili libere di $X(t_1, \ldots, t_n)$ con X variabile di relazione n-aria, sono X e l'unione di tutte le variabili libere che compaiono nei termini t_i per ogni i.
- Le variabili libere di $r(t_1, \ldots, t_n)$ con r simbolo d relazione n-aria sono l'unione di tutte le variabili libere che compaiono nei termini t_i per ogni i.

- Le variabili libere di $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ sono l'unione delle variabili libere di φ e ψ.

- Le variabili libere di $\forall x\varphi$ e $\exists x\varphi$ con x variabile, sono le variabili libere di φ meno x.

- Le variabili libere di $\forall X\varphi$ e $\exists X\varphi$ con X variabile di relazione, sono le variabili libere di φ meno X.

La sostituzione di termini nelle variabili è la sostituzione standard, con l'attenzione di evitare la cattura delle variabili. Per sostituire invece relazioni al posto di variabili di relazione ci appoggeremo al concetto di specie.

Definizione 2.5 Sia φ una formula e $x_1, \ldots, x_n$ delle variabili individuali, allora l'espressione $\lambda x_1, \ldots, x_n.\varphi$ è una specie di arietà n. Si noti che le variabili x_i possono apparire o non apparire in φ.

Le variabili libere di $\lambda x_1, \ldots, x_n.\varphi$ sono le variabili libere di φ meno le variabili x_i.

Abbrevieremo inoltre l'espressione $\lambda x_1, \ldots x_n.X(x_1, \ldots, x_n)$ con semplicemente X e se $\underline{x} = x_1, \ldots, x_n$, abbrevieremo $\lambda x_1, \ldots, x_n.\varphi$ con $\lambda \underline{x}.\varphi$.

Definiamo inoltre induttivamente la sostituzione di una specie di arietà n $\lambda \underline{x}.\varphi$ in una variabile di relazione n-aria X:

- $\perp[\lambda \underline{x}.\varphi/X] = \perp$.

- $r(t_1, \ldots, t_n)[\lambda \underline{x}.\varphi/X] = r(t_1, \ldots, t_n)$ quando r è una relazione oppure una variabile di relazione diversa da X.

- $(X(\underline{t})[\lambda \underline{x}.\varphi/X] = \varphi[\underline{t}/\underline{x}]$.

- $(\eta \rightarrow \psi)[\lambda \underline{x}.\varphi/X] = \eta[\lambda \underline{x}.\varphi/X] \rightarrow \psi[\lambda \underline{x}.\varphi/X]$ e equivalentemente per $\eta \vee \psi$ e $\eta \wedge \psi$.

- $(\forall x\eta)[\lambda \underline{x}.\varphi/X] = \forall x\eta[\lambda \underline{x}.\varphi/X]$ per tutte le variabili individuali x che non appaiono libere in $\lambda \underline{x}.\varphi$. Equvalentemente per $\exists x\eta$.

- $(\forall Y\eta)[\lambda \underline{x}.\varphi/X] = \forall Y\eta[\lambda \underline{x}.\varphi/X]$ con Y variabile di relazione diversa da X e Y che non appare libera in $\lambda \underline{x}.\varphi$.

Per fare un esempio sostituendo $\lambda x.\varphi(x, a)$ al posto di X in $\forall yX(y)$ otteniamo $\forall y\varphi(y, a)$.

A questo punto presentiamo le regole della deduzione naturale per la logica del secondo ordine:

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall X \varphi}\ \forall^2\mathrm{I} \qquad\qquad \frac{\Gamma \vdash \forall X \varphi}{\Gamma \vdash \varphi[\lambda x.\psi/X]}\ \forall^2\mathrm{E}$$

$$\frac{\Gamma \vdash \varphi[\lambda x.\psi/X]}{\Gamma \vdash \exists X \varphi}\ \exists^2\mathrm{I} \qquad\qquad \frac{\Gamma \vdash \exists X \varphi \qquad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}\ \exists^2\mathrm{E}$$

in cui in $\forall^2 I$ la variabile di proposizione X non appare libera in Γ e in $\exists^2 E$ non appare libera né in Γ né in ψ.

Aggiungendo queste regole alle regole della deduzione naturale per la logica classica del primo ordine, si ottiene il sistema per la logica classica del secondo ordine. Equivalentemente, aggiungendole alle regole per la logica intuizionista del primo ordine si ottiene il sistema per la logica intuizionista del secondo ordine.

Notiamo che è possibile dimostrare in entrambi i tipi di logica il principio di comprensione:

$$\exists Y\ \forall x\ (\varphi(x) \leftrightarrow Y(x)).$$

per ogni formula φ.

Inoltre molti dei connettivi presentati sono ridondanti: infatti è possibile definirli tutti in termini dei soli $\rightarrow$, $\perp$ e $\forall$ (su variabili individuali e di relazione). In particolare, sia nella logica intuizionistica che in quella classica sono valide:

- $\varphi \vee \psi = \forall X((\varphi \rightarrow X) \rightarrow (\psi \rightarrow X) \rightarrow X)$.
- $\varphi \wedge \psi = \forall X((\varphi \rightarrow \psi \rightarrow X) \rightarrow X)$.
- $\exists x \varphi = \forall R(\forall x(\varphi \rightarrow R) \rightarrow R)$.
- $\exists X \varphi = \forall R(\forall X(\varphi \rightarrow R) \rightarrow R)$.

Inoltre, nell'ambito della logica intuizionista, anche il simbolo $\perp$ è superfluo perché può essere riscritto come $\perp = \forall X.X$.

Il prossimo passo è quello di mettere in evidenza un rapporto che sussiste tra le proposizioni derivabili dalla logica intuizionista e la logica classica.

Definizione 2.6 Data una formula φ definiamo induttivamente la sua traduzione di Kolmogorov $k(\varphi)$ come:

- $\neg\neg\varphi$ se φ è atomica.

- $\neg\neg(k(\eta) \to k(\psi))$ se $\varphi = \eta \to \psi$.
- $\neg\neg\forall x k(\psi)$ se $\varphi = \forall x\psi$ per la quantificazione universale del primo ordine.
- $\neg\neg\forall X k(\psi)$ se $\varphi = \forall X\psi$ per la quantificazione universale del secondo ordine.

Notiamo che se vale il principio del terzo escluso per una certa formula φ, ovvero se è possibile dimostrare che $\varphi \vee \neg\varphi$, allora si può dimostrare che $k(\varphi) \leftrightarrow \varphi$.

A questo punto possiamo enunciare il seguente risultato, che mette in relazione la logica intuizionista con quella classica:

Proposizione 2.7 La proposizione φ è un teorema della logica classica se e solo se $k(\varphi)$ è un teorema della logica intuizionista.

Introduciamo adesso le aritmetiche del secondo ordine, sul linguaggio dell'aritmetica. Consideriamo i seguenti assiomi per l'uguaglianza:

(U1) $\forall a(a = a)$;
(U2) $\forall ab(a = b \to b = a)$;
(U3) $\forall abc(a = b \to b = c \to a = c)$;
(U4) $\forall ab(a = b \to Sa = Sb)$,

in cui i primi tre sono i consueti assiomi per una relazione di equivalenza e l'ultimo è una sorta di passo induttivo.

Aggiungiamo ancora tre assiomi di Peano:

(P1) $\forall ab(Sa = Sb \to a = b)$;
(P2) $\forall a(Sa = 0 \to \bot)$;
(P3) $\forall a\ \mathrm{Int}(a)$,

dove $\mathrm{Int}(a) = \forall X(\forall b(X(b) \to X(Sb)) \to X(0) \to X(a))$ serve a sostituire lo schema di induzione.

Definizione 2.8 Gli assiomi (U1-4) e (P1-2) utilizzati con il sistema della logica classica del secondo ordine definiscono l'aritmetica di Peano del secondo ordine PA2. Quando invece sono utilizzati con il sistema della logica intuizionista del secondo ordine definiscono l'aritmetica di Heyting del secondo ordine HA2.

Notiamo che nelle teorie presentate abbiamo omesso l'assioma di induzione (P3). Questo perché, attraverso la formula Int è possibile relativizzare ogni quantificazione ai termini che la soddisfano. Per esempio una formula

$\forall x\varphi$ può essere relativizzata con $\forall x(\mathrm{Int}(x) \rightarrow \varphi)$. Infatti ogni formula è dimostrabile nel sistema che comprende (P3) se e solo se la sua relativizzata è dimostrabile nel sistema senza (P3).

Utilizzando la quantificazione al secondo ordine è anche possibili definire dei predicati per la somma e il prodotto (e anche per le funzioni primitive ricorsive).

Una cosa interessante da notare è che nell'ambito dell'aritmetica, le formule di complessità Δ_0 si comportano in un certo senso come le formule della logica classica, ovvero se φ è Δ_0 allora l'aritmetica di Heyting dimostra $\varphi \vee \neg\varphi$, il principio del terzo escluso. Ne consegue che la traduzione di Kolmogorov, non indebolisce tali formule, cioè è un teorema che $k(\varphi) \leftrightarrow \varphi$ quando φ è Δ_0. Questi fatti sono veri poichè è possibile dimostrare che per ogni termine chiuso t esiste un naturale n tale che l'aritmetica di Heyting dimostri $t = \overline{n}$. Da qui si può ricavare la proprietà per le formule atomiche. Per passare alle Δ_0, si ragiona induttivamente. Per esempio sapendo che $\varphi \vee \neg\varphi$ e $\psi \vee \neg\psi$, si può ricavare il terzo escluso per $\varphi \wedge \psi$. Gli altri connettivi binari si trattano similmente. Utilizzando l'assioma di induzione si tratta anche il caso con i quantificatori limitati, per induzione sulla loro limitazione.

Un teorema di fondamentale importanza per le discussioni che seguiranno (e di cui abbiamo già utilizzato una versione al primo ordine) è:

Teorema 2.9 Le funzioni dimostrabilmente totali in PA2 sono esattamente le funzioni dimostrabilmente totali in HA2.

Per dimostrare il teorema ci appoggeremo alla seguente definizione:

Definizione 2.10 Sia φ e ψ due formule. Denotiamo con φ^ψ la formula ottenuta sostituendo ogni sottoformula atomica (compresa $\perp$ quindi) η presente in φ con $\eta \vee \psi$, assumendo che le variabili libere di η non siano legate in φ.

Osserviamo che la richiesta sulle variabili libere di η non è restrittiva in quanto è possibile rinominare le variabili legate in φ per l'α-equivalenza.

Risulta utile verificare le seguenti proprietà per il costrutto appena definito.

Lemma 2.11 Per ogni coppia di formule φ e ψ valgono i seguenti punti:

(i) $\psi \rightarrow \varphi^\psi$ è un teorema intuizionista.

(ii) Se a è termine e x è una variabile che non occore libera in ψ, allora vale $(\varphi[a/x])^\psi = \varphi^\psi[a/x]$.

(iii) Siano X e $\lambda\overline{x}.\eta$ rispettivamente una variabile di proposizione e una specie di uguale arietà per cui X e $\overline{x}$ non occorrono libere in ψ. Allora la formula $(\varphi[\lambda\overline{x}.\eta/X])^\psi \leftrightarrow \varphi^\psi[\lambda\overline{x}.\eta^\psi/X]$ è un teorema della logica intuizionista.

(iv) Se $\Gamma \vdash \varphi$, allora $\Gamma^\psi \vdash \varphi^\psi$, dove $\Gamma^\psi = \{\gamma^\psi \mid \gamma \in \Gamma\}$.

(v) Se φ è un assioma di HA2, allora HA2 $\vdash \varphi^\psi$.

(vi) Se HA2 $\vdash \varphi$, allora anche HA2 $\vdash \varphi^\psi$

(vii) Se φ è una formula Δ_0, allora HA2 $\vdash \varphi^\psi \to \varphi \vee \psi$

Dimostrazione. (i) La dimostrazione di questo punto si può fare per induzione sulla complessità di φ.

(ii) Anche questo punto viene dimostrato per induzione sulla complessità di φ.

(iii) Nuovamente si tratta di un'induzione. In questo caso il passo base è interessante e fa uso dei due punti precedenti. Infatti se $\varphi = X(\overline{y})$, dove X è una variabile di proposizione, allora

$$(\varphi[\lambda\overline{x}.\eta/X])^\psi = (\eta[\overline{y}/\overline{x}])^\psi = \\ \eta^\psi[\overline{y}/\overline{x}] = \varphi[\lambda\overline{x}.\eta^\psi/X],$$

dove la seconda uguaglianza è giustificata dal primo punto di questo lemma. Notiamo inoltre che per il primo punto si ha che $\vdash \psi \to (\varphi[\lambda\overline{x}.\eta/X])^\psi$ e dunque anche $\vdash \psi \to \varphi[\lambda\overline{x}.\eta^\psi/X]$. Siccome $\overline{x}$ e X non compaiono libere in ψ, si ha che $\psi = \psi[\lambda\overline{x}.\eta^\psi/X]$. Da ciò si ottiene che

$$\vdash (X(\overline{y}) \vee \psi)[\lambda\overline{x}.\eta^\psi/X] \leftrightarrow \varphi[\lambda\overline{x}.\eta^\psi/X],$$

che è la formula che volevamo dimostrare.

(iv) Questo punto è ancora una semplice induzione.

(v) Qui procediamo per induzione, ma sulla complessità della dimostrazione. I punti più interessanti sono i casi in cui la dimostrazione termina con l'eliminazione di un quantificatore universale del primo o del secondo ordine, in cui si utilizzano rispettivamente il secondo e il terzo punto di questo lemma.

Nel caso in cui l'ultimo passo della dimostrazione di $\Gamma \vdash \varphi[\lambda\overline{x}.\eta/X]$ fosse fatto eliminando il quantificatore del secondo ordine a partire da $\Gamma \vdash \forall X\varphi$, allora per ipotesi induttiva si ha che $\Gamma^\psi \vdash (\forall X\varphi)^\psi = \forall X\varphi^\psi$. Ma allora $\Gamma^\psi \vdash \varphi[\lambda\overline{x}.\eta^\psi/X]$. Per il terzo punto si ottiene il risultato voluto $\Gamma^\psi \vdash (\varphi[\lambda\overline{x}.\eta/X])^\psi$.

Nel caso in cui la dimostrazione di $\Gamma \vdash \varphi[\overline{t}/\overline{x}]$ si concludesse con l'eliminazione del quantificatore universale del primo ordine a partire dal giudizio $\Gamma \vdash \forall x\varphi$, si avrebbe che per ipotesi induttiva $\Gamma^\psi \vdash \forall x\varphi^\psi$. Dunque $\Gamma^\psi \vdash \varphi^\psi[\overline{t}/\overline{x}]$ e la tesi si ottiene applicando il secondo punto.

(vi) Ancora una volta, la dimostrazione è fatta per induzione sulla complessità della formula φ. Vediamo il passo induttivo quando la formula φ è nella forma $\forall x < t\ \eta$. Vogliamo allora dimostrare che $\forall y[(\forall x < y\ \eta)^\psi \to (\forall x < y\ \eta) \vee \psi$. Questa formula la dimostriamo per induzione su y. Infatti il caso $(\forall x < 0\ \eta) \vee \psi$ vale a vuoto. Per il passo induttivo supponiamo che $\forall x[(x < Sy \vee \psi) \to \eta^\psi]$. Allora Utilizzando l'ipotesi induttiva sulla complessità della formula otteniamo che $\forall x[(x < Sy \vee \psi) \to \eta \vee \psi]$. Raccogliendo, si ha che $(\forall x < Sy\ \eta) \vee \psi$, e dunque applicando l'introduzione di $\to$, si ha la tesi.

Il caso con il quantificatore esistenziale viene trattato analogamente.

□

Infine dimostriamo una proposizione più forte del teorema.

Proposizione 2.12 Le formule chiuse Π^0_2 dimostrabili in HA2 sono esattamente quelle dimostrabili in PA2.

Dimostrazione. Sia $\varphi = \forall x\exists y\eta$ una formula chiusa dimostrabile in PA2 in cui η è una formula Δ_0. In particolare PA2 dimostra $\psi = \exists y\eta$. Allora per il lemma 2.11 e sfruttando l'equivalenza tra η e $k(\eta)$ vale che

$$\mathrm{HA2} \vdash \neg\neg\exists y\eta$$

Consideriamo ora $(\neg\neg\exists y\eta)^\psi$, e semplificando le sottoformule $\bot \vee \psi$ sostituendole semplicemente con ψ, si ottiene che

$$\mathrm{HA2} \vdash (\exists y(\eta \vee \psi)) \to \psi) \to \psi$$

e dunque si ricava

$$\mathrm{HA2} \vdash \psi,$$

ovvero la tesi, in quanto $\eta \vee \psi \to \psi$ è una tautologia. □

3 Il λ-calcolo Non Tipato

In questa tesi ci occuperemo di studiare alcune proprietà di diversi tipi di λ-calcoli. Iniziamo dunque definendo gli elementi fondamentali del λ-calcolo più semplice, ovvero quello non tipato [2].

Definizione 3.1 I termini del λ-calcolo semplice si definiscono induttivamente come:

- Le variabili $x_1, x_2, \ldots$ sono termini.
- Se t e v sono termini, allora anche l'applicazione (tv) è un termine.
- Se t è un termine e x è una variabile, l'astrazione $(\lambda x.t)$ è un termine.

Dato un termine t, i sottotermini sono tutti i termini che appaiono nella costruzione induttiva di t.

L'idea intuitiva dietro questa definizione è quella che un termine del tipo $\lambda x.t$ corrisponde a un programma con input x e corpo t. Inoltre un'applicazione della forma tv rappresenta un programma t quando eseguito con input v. Un fatto interessante da notare è che i termini possono svolgere indistintamente il ruolo di programma e di dato su cui un programma opera.

Per comodità e leggibilità delle notazioni, ometteremo spesso le parentesi sottointendendo che l'applicazione si associa a sinistra (e quindi $xyz = (xy)z$) e l'astrazione si associa a destra, usando un singolo simbolo λ, per esempio $\lambda xyz.yyxz = \lambda x.(\lambda y.(\lambda z.yyxz))$. Sempre per compattezza di notazione, se $\overline{x}$ è un vettore di variabili $x_1, \ldots, x_n$, scriveremo $\lambda\overline{x}$ per intendere $\lambda x_1 \cdots x_n$.

In modo simile a quanto si fa comunemente in logica è utile distinguere le occorrenze di una variabile in una formula tra occorrenze libere e legate. In particolare l'astrazione è una sorta di quantificatore che lega la variabile individuale a cui si riferisce. Più formalmente:

Definizione 3.2 Un'occorrenza della variabile x in un termine t si dice legata se esiste un sottotermine del tipo $\lambda x.t'$ che la contiene. Si dice libera altrimenti.

Inoltre, dato un termine t si definiscono induttivamente le sue variabili libere come:

- Se $t = x$ dove x è una variabile, allora è l'unica variabile libera d t.
- Se $t = uv$, allora le variabili libere di t sono tutte e sole le variabili libere che compaiono in u o in v.
- Se $t = \lambda x.u$, allora le variabili libere di t sono tutte le variabili libere di u con l'esclusione di x.

Nel seguito considereremo i termini modulo il rinominare le variabili legate. Questo corrisponde al fatto che in un programma è possibile rinominare i parametri formali delle funzioni (modificando consistentemente le loro occorrenze nei corpi di tali funzioni). La relazione di equivalenza che mette in relazione un termine con tutti i termini uguali a esso a meno del rinominare le variabili legate si chiama α-equivalenza. Da qui in avanti, per semplicità, abuseremo della notazione riferendoci alle classi di equivalenza con i loro elementi.

Con qualche accortezza per evitare la cattura delle variabili si può definire la sostituzione di un termine su una variabile.

Definizione 3.3 Se u e v sono termini e x è una variabile, allora la sostituzione di v su x in u è il termine $u[v/x]$ definito come:

- Se $u = x$ allora $u[v/x] = v$.
- Se $u = y$ con y una variabile distinta da x, allora $u[v/x] = y$.
- Se $u = tw$, allora $u[v/x] = (t[v/x])(w[v/x])$.
- Se $u = \lambda y.t$ e y è distinta da x e non appare libera in v, allora $u[v/x] = \lambda y.t[v/x]$.

A questo punto abbiamo presentato tutti gli strumenti per introdurre una relazione fondamentale sui termini del calcolo, ovvero la conversione.

Definizione 3.4 Dati due termini u e v, si dice che u si converte a v e scriverermo $u \leadsto_C v$, se v è ottenuto da u sostituendo un sottotermine nella forma $(\lambda x.u')u''$ con $u'[u''/x]$ (quando tale sostituzione è permessa e quindi non si ha cattura delle variabili).

Si dice che u si riduce a v se esistono n termini $u_1, \ldots, u_n$ con $u_1 = u$ e $u_n = v$ tali che $u_1 \leadsto_C u_2 \leadsto_C \ldots \leadsto_C u_n$. In tal caso scriveremo $u \leadsto v$.

Notiamo che la definizione dice essenzialmente che $\leadsto$ è la chiusura transitiva di $\leadsto_C$. Tali nozioni di conversione e riduzione sono chiamati in letteratura anche β-conversione e β-riduzione. La più piccola relazione di equivalenza che contiene la β-conversione è detta β-equivalenza.

L'idea di conversione corrisponde all'esecuzione di un passo del programma corrispondente al termine che viene convertito. Notiamo però che volendo ridurre un termine, la successione delle conversioni non è univocamente determinata, come nel termine $((\lambda x.xx)y)((\lambda x.x)z)$.

Facciamo ora due esempi importanti di riduzione.

Esempio 3.5

$$
\begin{gathered}
(\lambda f.\lambda x.f(fx))(\lambda y.yy) \rightsquigarrow_C \\
\lambda x.(\lambda y.yy)((\lambda y.yy)x) \rightsquigarrow_C \\
\lambda x.(\lambda y.yy)(xx) \rightsquigarrow_C \\
\lambda x.xxxx.
\end{gathered}
$$

Esempio 3.6

$$
\begin{gathered}
\Omega = (\lambda x.xx)(\lambda x.xx) \rightsquigarrow_C \\
(\lambda x.xx)(\lambda x.xx) \rightsquigarrow_C \\
\ldots
\end{gathered}
$$

Notiamo che il primo esempio finisce con un termine che non può essere ulteriormente convertito, mentre nel secondo la successione di conversioni è infinita. Questa distinzione è importante e conduce alle seguenti definizioni.

Definizione 3.7 Un termine si dice in forma normale se non può essere ulteriormente convertito. Un termine si dice normalizzante se può essere ridotto a un termine in forma normale.

Esistono comunque termini che pur essendo normalizzanti ammettono una successione infinita di conversioni:

Esempio 3.8

$$
(\lambda x.\lambda y.y)\Omega z \rightsquigarrow_C z
$$

ma convertendo ad ogni passo il termine Ω (si veda la sua definizione nell'esempio precedente), si ottiene che:

$$
\begin{gathered}
(\lambda x.\lambda y.y)\Omega z \rightsquigarrow_C \\
(\lambda x.\lambda y.y)\Omega z \rightsquigarrow_C \\
\ldots
\end{gathered}
$$

È allora utile introdurre le seguenti nozioni:

Definizione 3.9 Dato un termine t si definisce $\nu(t)$ il massimo numero di conversioni necessarie per portare t in forma normale, ossia

$$\nu(t) = \sup\{n \mid \exists u_1, \ldots, u_n \text{ per cui}$$
$$t \rightsquigarrow_C u_1 \rightsquigarrow_C \ldots \rightsquigarrow_C u_n \text{ e } u_n \text{ è in forma normale}\}.$$

Un termine t si dice fortemente normalizzante se $\nu(t) < \infty$.

Si osservi che se un termine è fortemente normalizzante allora è anche normalizzante, e non può essere convertito un numero infinito di volte.

Se un termine è normalizzante, vale che convertendo sempre il suo sottotermine non in forma normale più a sinistra si ottiene comunque sempre una forma normale.

Un'importante proprietà di cui gode la riduzione è detta proprietà di Church-Rosser, che implica l'unicità del termine in forma normale a cui si converte un termine normalizzante.

Teorema 3.10 Sia t un termine. Se u e v sono termini per cui $t \rightsquigarrow u$ e $t \rightsquigarrow v$, allora esiste un quarto termine w tale che $u \rightsquigarrow w$ e $v \rightsquigarrow w$.

I termini in forma normale hanno una struttura che presentiamo nel seguente teorema:

Proposizione 3.11 Un termine si dice in forma normale di testa se è nella forma

$$\lambda\overline{x}.yu_1 \cdots u_n,$$

dove gli u_i sono termini e y è una variabile (che a priori può non comparire in $\overline{x}$). Si noti che $\overline{x}$ potrebbe anche essere vuoto. Vale che un termine è in forma normale se e solo se è in forma normale di testa e gli u_i della notazione appena introdotta sono termini in forma normale per ogni i.

Dimostrazione. Ragioniamo per induzione sulla complessità dei termini. Se un termine è nella forma $\lambda x.u$, esso è in forma normale se e solo se u è in forma normale, ed è in forma normale di testa se e solo se u è in forma normale di testa. La tesi si ha dunque utilizzando l'ipotesi induttiva, che afferma l'equivalenza delle due proprietà su u.

Se invece consideriamo un termine nella forma uv, esso è in forma normale se e solo se sia u che v sono in forma normale e u non è un'astrazione. Inoltre uv è in forma normale di testa se e solo se v è normale e u è in forma normale di testa e non un'astrazione. Per ipotesi induttiva, si ha la tesi.

Infine se il termine è una variabile, la tesi è banalmente vera. □

3.1 Rappresentabilità nel λ-calcolo Non Tipato

Possiamo individuare alcuni termini del λ-calcolo per essere messi in corrispondenza con i numeri naturali. La scelta che faremo non è l'unica, ma sicuramente è conveniente e si presta facilmente ad essere riutilizzata in altre varianti del calcolo.

Definizione 3.12 Dato un numero naturale n definiamo il corrispondente numerale $\overline{n}$ come il termine $\lambda fx.f^n x$, dove il simbolo $f^n x$ indica $f(f(\cdots f(x)\cdots))$, in cui la f compare n volte.

Possiamo interpretare $\overline{n}$ come un termine che avendo in input una funzione f, la compone con se stessa n volte. Osserviamo che inoltre i numerali sono termini in forma normale, e dunque per la proprietà di Church-Rosser sono anche termini distinti modulo la β-equivalenza.

Definizione 3.13 Data una funzione (eventualmente parziale) $\varphi\colon \mathbb{N} \to \mathbb{N}$, si dice che un termine t rappresenta φ se per ogni coppia di naturali m e n vale che $\varphi(n) = m$ se e solo se $t\overline{n} \rightsquigarrow \overline{m}$ e $t\overline{n}$ non è normalizzante quando $\varphi(n) = \perp$ (ovvero quando $\varphi(n)$ è indefinita).

Facciamo ora alcuni esempi di funzioni rappresentabili:

Esempio 3.14 Il termine $A = \lambda pqfx.(pf)(qfx)$ rappresenta l'addizione. Per esempio:

$$
\begin{gathered}
A\overline{2}\ \overline{3} \rightsquigarrow \lambda fx.f^2(f^3x) = \\
\lambda fx.f^5x = \overline{5}.
\end{gathered}
$$

In modo simile esistono i termini $M = \lambda pqfx.q(pf)x$ e $E = \lambda pqfx.qpfx$ che rappresentano rispettivamente la moltiplicazione e l'esponenziazione. Questo ultimo termine è leggermente differente dagli altri due in quanto è l'unico in cui un numerale viene direttamente applicato ad un altro numerale. Vedremo che questa differenza sarà decisiva per la rappresentabilità in alcune varianti di λ-calcolo.

La classe di funzioni rappresentabili nel λ-calcolo non tipato è estremamente ampia, infatti vale il seguente teorema, che si trova in [5]:

Teorema 3.15 Le funzioni rappresentabili nel λ-calcolo non tipato sono esattamente le funzioni calcolabili.

Un nodo cruciale nella dimostrazione del precedente teorema è l'esistenza di un combinatore di punto fisso, ovvero di un termine t tale che per ogni termine u vale che $tu = u(tu)$, modulo la β-conversione. Un esempio di combinatore di punto fisso è il termine $Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$, come si può facilmente verificare. Esso è noto come combinatore di punto fisso di Curry. Notiamo che tale termine non è normalizzante, ma più in generale possiamo dimostrare che nessun combinatore di punto fisso t può essere normalizzante. Infatti se indichiamo con t' la forma normale di tx, dove x è una variabile, vale che $t' = xt'$, modulo la β-conversione. Ma entrambi questi termini sono in forma normale, e dunque abbiamo l'assurdo.

4 Il λ-calcolo Tipato Semplice

Nel λ-calcolo non tipato, esistono termini per cui la normalizzazione non corrisponde all'idea intuitiva di "semplificazione", come nell'esempio 3.6 oppure nel caso ancora peggiore seguente, in cui il numero di sottotermini cresce proseguendo nella riduzione.

Esempio 4.1

$$
\begin{gathered}
(\lambda x.xxx)(\lambda x.xxx) \rightsquigarrow \\
(\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx) \rightsquigarrow \\
\dots
\end{gathered}
$$

Il problema alla base di questo comportamento è il fatto che non vi è distizione tra dati e programmi e in particolare è permessa l'applicazione di un termine a se stesso. Versioni più sofisticate del calcolo puntano dunque a introdurre dei vincoli sull'applicazione dei termini, ed è per questo motivo che si introducono i tipi. L'idea è quella di associare ad ogni termine una struttura sintattica chiamata tipo, e permettere l'applicazione di termini solo se i loro tipi sono compatibili. Notiamo una forte somiglianza tra i tipi del λ-calcolo e i tipi presenti nei linguaggi di programmazione.

Definiamo allora una variante del λ-calcolo detta λ-calcolo tipato semplice.

Definizione 4.2 I tipi del λ-calcolo tipato semplice sono definiti induttivamente come:

- $U_1, U_2, \dots$ sono tipi, detti variabili di tipo oppure tipi atomici.
- Se U e V sono tipi, allora anche $(U \to V)$ è un tipo.

Per comodità, in assenza di parentesi, l'associatività di $\to$ è intesa essere a destra: per esempio $U \to V \to W = (U \to (V \to W))$.

A questo punto ricostruiamo i termini associando a ciascuno di essi un relativo tipo.

- Per ogni tipo U, le variabili $x_1^U, x_2^U, \ldots$ sono termini di tipo U.
- Se t e v sono termini di tipo rispettivamente $U \to V$ e U, allora l'applicazione (tv) è un termine di tipo V.
- Se t è un termine di tipo V e x è una variabile di tipo U, allora l'astrazione $(\lambda x.t)$ è un termine di tipo $U \to V$.

Per indicare che un termine t è di tipo U scriveremo anche t^U. Inoltre se il tipo delle variabili è chiaro dal contesto, ometteremo di indicarlo ad esponente.

Si vede dunque che il tipo $U \to V$ corrisponde all'idea delle funzioni dai termini di tipo U ai termini di tipo V, e che l'applicazione è consentita solo quando "il dominio della funzione e il tipo dell'argomento coincidono".

In modo identico a quanto già fatto per il λ-calcolo semplice (essendo i termini tipato semplice un sottoinsieme di quelli del non tipato, a meno di eticattare ogni termine con il suo tipo), è possibile definire le nozioni di conversione, riduzione e forma normale. Si noti che tali relazioni conservano il tipo dei termini.

4.1 Normalizzazione Forte per il Tipato Semplice

L'obiettivo di questa sezione è quella di dimostrare il seguente importante risultato:

Teorema 4.3 Tutti i termini del λ-calcolo tipato semplice sono fortemente normalizzanti.

Da questo fatto discende che l'espressività di questo calcolo è molto ridotta rispetto a quella del λ-calcolo tipato semplice. Per esempio non possiamo trovare nessun combinatore di punto fisso e vedremo che la classe di funzioni rappresentabili è anch'essa ridotta.

Introduciamo come prima cosa la nozione di riducibilità.

Definizione 4.4 Sia U un tipo, e t un termine di tipo U. Definiamo induttivamente il suo insieme di termini riducibili RED_U come:

- Se U è atomico, t è riducibile se e solo se è fortemente normalizzante.
- Se $T = V \to W$, t è riducibile se e solo se per ogni termine riducibile v di tipo V, il termine tv è riducibile di tipo W.

Osserviamo che sono termini riducibili tutte le variabili di tipo.

Definizione 4.5 Diciamo che un termine t è neutrale se è nella forma ab, dove a e b sono termini di tipo compatibile.

L'idea dietro alla neutralità è che se t è un termine neutrale e v un termine per cui $tv \rightsquigarrow_C u$, allora $u = t'v$ oppure $u = tv'$ dove t' e v' sono conversioni rispettivamente di t e v. In particolare non ci sono step di riduzione in cui v o un suo sottotermine viene sostituito in una variabile di t.

Dimostriamo ora che gli insiemi di riducibili godono di alcune proprietà, che saranno utili a dimostrare il teorema di questa sezione e torneranno anche utili nello studio della normalizzazione nel sistema F.

Proposizione 4.6

(CR1) Se $t \in \mathrm{RED}_U$, allora t è fortemente normalizzante.
(CR2) Se $t \in \mathrm{RED}_U$ e $t \rightsquigarrow u$, allora $u \in \mathrm{RED}_U$.
(CR3) Se t è neutrale di tipo U e per ogni t' per cui $t \rightsquigarrow_C t'$ vale che $t' \in \mathrm{RED}_U$, allora anche $t \in \mathrm{RED}_U$.

Notiamo che la prima proprietà indica che essere riducibile implica l'essere fortemente normalizzante. La seconda proprietà permette di conoscere la riducibilità di un termine data la riducibilità di un termine precedente in una catena di conversioni. Infine la terza proprietà permette di conoscere la riducibilità di un termine data quella delle sue conversioni.

Dimostrazione. La dimostrazione è per induzione sulla complessità dei tipi.

Iniziamo dal caso in cui il tipo U sia una variabile di tipo. Allora, poichè i riducibili di tipo U sono i termini fortemente normalizzanti, (CR1) è una tautologia. Se un termine t è fortemente normalizzante e $t \rightsquigarrow t'$, allora anche t' è fortemente normalizzante perché vale che $\nu(t') < \nu(t)$. Dunque anche (CR2) vale. Per (CR3), sia t un termine neutrale per cui tutte le conversioni sono fortemente normalizzanti. Allora vale che $\nu(t)$ è pari al massimo di $\nu(t')$ al variare di t' tra le conversioni di t, e dunque è finito.

Consideriamo adesso il tipo $U \to V$. Supponiamo che t sia un riducibile di tale tipo, e supponiamo che x sia una variabile di tipo U. Poichè x è neutrale e normale, essa è riducibile. Allora anche tx è riducibile, per la definizione di riducibilità sul tipo freccia. Osserviamo ora che $\nu(t) \leq \nu(tx)$, infatti ad ogni catena di riduzioni $t \rightsquigarrow t_1 \rightsquigarrow \cdots \rightsquigarrow t_n$ possiamo associare la catena $tx \rightsquigarrow t_1x \rightsquigarrow \cdots \rightsquigarrow t_nx$. Poichè $\nu(tx)$ è finito, t è fortemente normalizzante, e (CR1) è dimostrato.

Se consideriamo un termine t di tipo $U \to V$ riducibile e un termine t' tale che $t \rightsquigarrow t'$, allora per ogni termine u di tipo U vale che $tu \rightsquigarrow t'u$. Utilizzando l'ipotesi induttiva di (CR2) su V, otteniamo che anche $t'u$ è riducibile. Per cui anche t' è riducibile e (CR2) vale.

Supponiamo ora di avere t neutrale per cui tutte le sue conversioni siano riducibili. Sia u un riducibile di tipo U. L'obiettivo è mostrare che tu è riducibile. Per (CR1) per U, già sappiamo che u è fortemente normalizzante, per cui possiamo ragionare per induzione su $\nu(u)$. Notiamo che per neutralità di t, tu si può convertire soltanto in $t'u$, con t' conversione di t, oppure in tu', con u' conversione di u. Nel primo caso sappiamo che t' è riducibile, e dunque anche $t'u$ lo è. Nel secondo caso possiamo osservare che $\nu(u') < \nu(u)$ e dunque per induzione otteniamo nuovamente che $t'u$ è riducibile. Poichè tu si converte soltanto a riducibili, è anch'esso riducibile per ipotesi induttiva di (CR3). □

A questo punto dimostriamo un utile lemma:

Lemma 4.7 Se per tutti i termini riducibili u di tipo U, il termine $v[u/x]$ è riducibile, allora anche il termine $\lambda x.v$ è riducibile.

Dimostrazione. Supponiamo che $v[u/x]$ sia di tipo V, allora il termine $\lambda x.v$ è di tipo $U \to V$. Allora vogliamo dimostrare che per ogni termine riducibile u di tipo U vale che $(\lambda x.v)u$ è riducibile. Notiamo che v è riducibile, infatti x è riducibile di tipo U e v è ottenuto con la sostituzione banale di x al posto di x in v.

Ragioniamo per induzione sulla somma $\nu(v) + \nu(u)$ per dimostrare che tutte le conversioni di $(\lambda x.v)u$ sono riducibili. Il termine $(\lambda x.v)u$ si può convertire in:

- $v[u/x]$, che è riducibile per ipotesi.
- $(\lambda x.v')u$, con v' conversione di v. Allora si ha che v' è riducibile per (CR2), e vale $\nu(v') < \nu(v)$ e quindi per ipotesi induttiva $\lambda x.v'$ è riducibile.
- $(\lambda x.v)u'$, con u' conversione di u. In questo caso, similmente a prima, u' è riducibile, e vale $\nu(v') < \nu(v)$. Nuovamente $(\lambda x.v)u'$ è anch'esso riducibile per ipotesi induttiva.

Concludiamo per (CR3), che assicura che $\lambda x.v$ sia dunque riducibile. □

Adesso si dimostra una versione più forte del teorema principale.

Proposizione 4.8 Sia t un termine le cui variabili libere compaiono tra $x_1, \ldots, x_n = \underline{x}$, di tipo rispettivamente $U_1, \ldots, U_n$. Siano $u_1, \ldots, u_n = \underline{u}$ termini riducibili di tipo rispettivamente $U_1, \ldots, U_n$. Allora il termine $t[\underline{u}/\underline{x}]$ è riducibile. Intendiamo per $t[\underline{u}/\underline{x}]$ la sostituzione $t[u_1/x_1] \cdots [u_n/x_n]$.

Dimostrazione. Per induzione sulla complessità di t:

- Se $t = x_i$, allora la tesi è banalmente vera.
- Se $t = wv$, allora per l'ipotesi induttiva $w[\underline{u}/\underline{x}]$ e $v[\underline{u}/\underline{x}]$ sono riducibili. Ne consegue che $t[\underline{u}/\underline{x}] = w[\underline{u}/\underline{x}]v[\underline{u}/\underline{x}]$ è riducibile.
- Se $t = \lambda y.w$ di tipo $V \to W$, allora per ipotesi induttiva, $t[\underline{u}/\underline{x}][v/y]$ è riducibile per tutti i termini v di tipo V. Allora per il lemma 4.7 si ottiene che $\lambda y.w[\underline{u}/\underline{x}]$ è riducibile.

□

La dimostrazione del teorema di normalizzazione forte segue dall'ultima proposizione ponendo $\underline{u} = \underline{x}$.

4.2 Rappresentabilità per il Tipato Semplice

All'interno del λ-calcolo tipato semplice è possibile rappresentare i numeri naturali con gli stessi numerali presentati per il caso non tipato. Tuttavia, la scelta del tipo per le variabili che vi compaiono non è unica. Infatti per ogni tipo U possiamo costruire per ogni naturale n il corrispondente numerale

$$\overline{n} = \lambda f^{U \to U}.\lambda x^U.f^n x$$

di tipo $\mathrm{Int} = (U \to U) \to U \to U$. Avendo a disposizione i numerali, possiamo definire le funzioni rappresentabili in modo identico a quanto fatto con il λ-calcolo tipato semplice.

Possiamo associare dei tipi anche ai termini che avevamo usato nel λ-calcolo semplice per rappresentare la somma e la moltiplicazione, infatti

$$A = \lambda p^{\mathrm{Int}} q^{\mathrm{Int}} f^{U \to U} x^U.(pf)(qfx)$$

è la versione tipata per l'addizione e

$$M = \lambda p^{\mathrm{Int}} q^{\mathrm{Int}} f^{U \to U} x^U.q(pf)x$$

lo è per la moltiplicazione.

Possiamo dare un ulteriore esempio di funzione rappresentabile, che è quella corrispondente all'*if/then/else*, ovvero la funzione condizionale $f(x, y, z)$ che vale y se x è non nullo e vale z altrimenti. Essa è rappresentata dal termine

$$C = \lambda p^{\mathrm{Int}} q^{\mathrm{Int}} r^{\mathrm{Int}} f^{U \to U} x^U.p(\lambda y^U.qfx)(rfx).$$

Infatti se a, b, c sono numerali $Cabc$ si riduce a b se se a è non nullo e si riduce c altrimenti. Si noti che questa proprietà non è sempre vera nel caso in cui gli argomenti non fossero numerali ma generici termini di tipo Int.

Come importante conseguenza della proprietà di normalizzazione forte si ha una conseguente riduzione della classe delle funzioni rappresentabili, che devono essere per forza totali. Inoltre non tutte le funzioni totali sono rappresentabili, infatti se così fosse sarebbe possibile trovare una loro enumerazione con i termini del calcolo, ma ciò è assurdo perché l'insieme delle (codifiche delle) funzioni totali non è ricorsivamente enumerabile.

Per esempio il termine che avevamo usato per rappresentare l'esponenziazione non può essere tipato; infatti l'applicazione di un numerale a un altro numerale, ovvero un termine di tipo Int a un altro termine di tipo Int, non è permessa dalle regole del λ-calcolo tipato semplice.

Si può dire di più, perché nessun termine può rappresentare l'esponenziazione e più in generale vale il seguente teorema:

Teorema 4.9 Le funzioni rappresentabili nel λ-calcolo tipato semplice sono esattamente le funzioni generate dalle costanti 0 e 1 e dalle funzioni di somma, moltiplicazione e condizionale.

Un verso è immediato, avendo già trovato i termini A, M e C.

5 Il Sistema T di Gödel

Il grosso problema che rende in un certo senso limitante il calcolo tipato semplice è che i numerali non hanno un tipo canonico, e non è possibile definire funzioni per ricorsione primitiva. Tale calcolo ha dunque un'espressività molto bassa. Per ovviare a questo problema introduciamo una nuova variante del calcolo, il sistema T di Gödel, in cui vengono artificialmente inseriti tipi per gli interi e le coppie di termini, insieme a nuove costruzioni che rappresentano delle funzioni basilari su di essi. Vedremo che con queste semplici aggiunte il calcolo guadagna una forte potenza espressiva. In questo e nel succesivo capitolo useremo le notazioni di [1] e [**?**]

Per definire dunque il sistema T, estendiamo la definizione dei tipi e dei termini del λ-calcolo tipato semplice.

Definizione 5.1 I tipi del sistema T sono definiti induttivamente come:

- $U_1, U_2, \ldots$ sono tipi, detti variabili di tipo.
- Se U e V sono tipi, allora anche $(U \to V)$ è un tipo.
- Se U e V sono tipi, allora anche $(U \times V)$ è un tipo.

- Int è un tipo atomico.

I termini del sistema T sono definiti induttivamente come:

- Per ogni tipo U, le variabili $x_1^U, x_2^U, \ldots$ sono termini di tipo U.
- Se t e v sono termini di tipo rispettivamente $U \to V$ e U, allora l'applicazione (tv) è un termine di tipo V.
- Se t è un termine di tipo V e x è una variabile di tipo U, allora l'astrazione $(\lambda x.t)$ è un termine di tipo $U \to V$.
- Se u e v sono termini di tipo rispettivamente U e V, allora $\langle u, v\rangle$ è un termine di tipo $U \times V$.
- Se t è un termine di tipo $U \times V$, allora $\pi^1 t$ e $\pi^2 t$ sono termini di tipo rispettivamente U e V.
- O è un termine di tipo Int.
- Se t è un termine di tipo Int, allora anche St è un termine di tipo Int.
- Se u, v, e t sono termini rispettivamente di tipo U, $U \to \mathrm{Int} \to U$ e Int, allora $Ruvt$ è un termine di tipo U.

Il significato che vorremmo assegnare ai nuovi termini è il seguente: il tipo $U \times V$ corrisponde al tipo di una coppia formata da un termine u di tipo U e un termine v di tipo V in quest'ordine. La coppia è rappresentata dal termine $\langle u, v\rangle$; π^1 e π^2 sono le proiezioni. Il termine O e il simbolo S rappresentano rispettivamente lo zero e la funzione di successore, e pertanto scriveremo il numerale relativo a n come $S^n O$. Inoltre R è l'operatore di ricorsione primitiva, per cui $Ruv(St) = v(Ruvt)t$, dove l'uguaglianza è intesa a meno di β-equivalenza.

Con l'introduzione dei nuovi termini occorre estendere le regole per la conversione:

Definizione 5.2 Un termine u si converte a v quando v è ottenuto sostituendo in u un sottotermine u' con un termine v' tali che valga una delle seguenti:

- $u' = \pi^1\langle w, t\rangle$ e $v' = w$, per opportuni termini w e t.
- $u' = \pi^2\langle w, t\rangle$ e $v' = t$, per opportuni termini w e t.
- $u' = (\lambda x.w)t$ e $v' = w[t/x]$ dove x è una variabile e w e t sono termini tali che la sostituzione $w[t/x]$ sia permessa.

- $u' = RwtO$ e $v' = w$, per opportuni termini w e t.
- $u' = Rwt(Sn)$ e $v' = t(Rwtn)n$, per opportuni termini w, t e n.

Riassumiamo le regole nel seguente schema:

$$\boxed{\begin{array}{c} \pi^1\langle w, t\rangle \rightsquigarrow_C w \\ \pi^2\langle w, t\rangle \rightsquigarrow_C t \\ (\lambda x.w)t \rightsquigarrow_C w[t/x] \\ RwtO \rightsquigarrow_C w \\ Rwt(Sn) \rightsquigarrow_C t(Rwtn)n \end{array}}$$

Come negli altri calcoli si possono definire a questo punto la riduzione, la forma normale, e la rappresentabilità.

Come osservazione, notiamo che in generale non è vero che il termine $\langle\pi^1 t, \pi^2 t\rangle$ si riduce al termine t, e anzi potrebbero essere entrambi in forma normale e pertanto nemmeno β-equivalenti, in quanto anche in questa variante vale la proprietà di Church-Rosser.

Inoltre notiamo che il simbolo R può essere instanziato su qualunque altro tipo, ovvero

5.1 Normalizzazione Forte per il Sistema T

Anche per il sistema T, come per il λ-calcolo tipato semplice, vale la proprietà di normalizzazione forte. Lo scopo di questa sezione è appunto quella di dimostrare il seguente teorema:

Teorema 5.3 Tutti i termini del sistema T sono fortemente normalizzanti.

La dimostrazione è un adattamento di quella già vista per il tipato semplice. Estendiamo dunque la definizione di termine riducibile e di termine neutrale:

Definizione 5.4 Sia U un tipo, e t un termine di tipo U. Definiamo induttivamente il suo insieme di riducibilità RED_U come:

- Se U è atomico, t è riducibile se e solo se è fortemente normalizzante.
- Se $T = V \to W$, t è riducibile se e solo se per ogni termine riducibile v di tipo V, il termine tv è riducibile di tipo W.
- Se $T = V \times W$, t è riducibile se e solo se sia $\pi^1 t$ che $\pi^2 t$ sono riducibili.

Definizione 5.5 Un termine si dice neutrale quando non è in nessuna delle seguenti forme: $\langle u, v\rangle$, $\lambda x.v$, O o St.

Anche con questa definizione di neutralità vale l'osservazione fatta appena sotto la definizione 4.5.

Con queste definizioni estese possiamo riproporre senza modifiche le proprietà (CR1-3). Continua a valere la proposizione 4.6, di cui dobbiamo estendere la dimostrazione.

Proposizione 5.6 Per tutti i riducibili del sistema T valgono le proprietà (CR1-3).

Dimostrazione. Nonostante sia stato inserito un ulteriore tipo atomico, la dimostrazione fatta per essi continua a rimanere valida.

Studiamo allora il tipo prodotto. Supponiamo che il termine t di tipo $U \times V$ sia riducibile, il che significa che entrambe le sue proiezioni sono riducibili. Per ipotesi induttiva, $\pi^1 t$ è fortemente normalizzante. Similmente a quanto già fatto, notiamo che $\nu(t) \leq \nu(\pi^1 t)$ perché ad ogni catena $t \rightsquigarrow t_1 \rightsquigarrow \cdots \rightsquigarrow t_k$ possiamo associare la catena $\pi^1 t \rightsquigarrow \pi^1 t_1 \rightsquigarrow \cdots \rightsquigarrow \pi^1 t_k$. Ne consegue che $\nu(t)$ è finito e questo dimostra (CR1).

Per (CR2) osserviamo che se t e t' sono termini per cui $t \rightsquigarrow t'$, allora $\pi^1 t \rightsquigarrow \pi^1 t'$ e $\pi^2 t \rightsquigarrow \pi^2 t'$. Quindi see t è riducibile, allora lo sono anche le sue proiezioni e quindi, per ipotesi induttiva, lo sono anche le riduzioni delle proiezioni. Poichè t' ha entrambe le proiezioni riducibili, è anch'esso riducibile.

Infine sia t un termine neutrale per cui tutte le sue conversioni sono riducibili. Convertendo $\pi^1 t$, si può ottenere solo un termine della forma $\pi^1 t'$ con t' conversione di t, in quanto t è neutrale e non può essere una coppia. Allora t' è riducibile e quindi anche $\pi^1 t'$ lo è. Ma questo vuol dire che tutte le conversioni del termine neutrale $\pi^1 t$ sono riducibili, che implica che anche $\pi^1 t$ è riducibile. Ripetendo il ragionamento con la seconda proiezione si dimostra la riducibilità di t e quindi anche (CR3). $\square$

Notiamo che O è un termine di tipo atomico in forma normale, e pertanto è riducibile. Inoltre se t è un termine di tipo Int riducibile (quindi fortemente normalizzante) allora anche St è riducibile. Infatti vale che $\nu(St) = \nu(t)$.

Dimostriamo ora una coppia di lemmi simili al lemma 4.7.

Lemma 5.7 Se u e v sono riducibili, anche $\langle u, v\rangle$ è riducibile.

Dimostrazione. Ragioniamo per induzione sulla somma $\nu(u) + \nu(v)$ per dimostrare che $\pi^1\langle u, v\rangle$ si converte a soli termini riducibili, e quindi è anch'esso riducibile. Tale termine si può convertire in:

- u, che è riducibile.

- $\pi^1\langle u', v\rangle$ con u' conversione di u. In questo caso abbiamo che u' è riducibile per (CR2), e quindi poichè $\nu(u') < \nu(u)$ otteniamo che per ipotesi induttiva la riducbilità di $\pi^1\langle u', v\rangle$.

- $\pi^1\langle u, v'\rangle$, con v' conversione di v. Questo caso è identico al precedente.

Ripetendo i passaggi precedenti con $\pi^2\langle u, v\rangle$, otteniamo che la coppia $\langle u, v\rangle$ è riducibile. □

Lemma 5.8 Se u, v e t sono termini riducibili, allora, se i tipi dei sottotermini ne permettono la costruzione, anche $Ruvt$ è un termine riducibile.

Dimostrazione. Di nuovo ragioniamo per induzione per ottenere che tutte le conversioni del termine $Ruvt$ sono riducibili per poi concludere con (CR3). In questo caso l'induzione viene fatta su $\nu(u) + \nu(v) + \nu(t) + l(t)$, dove $l(t)$ indica il numero di simboli che appaiono nella forma normale di t.

Il termine $Ruvt$ si converte in:

- $Ru'v't'$, con u', v' e t' riduzioni rispettivamente di u, v e t. In questo caso vale che $\nu(u') + \nu(v') + \nu(t') < \nu(u) + \nu(v) + \nu(t)$ e $l(t) = l(t')$. Si può quindi usare l'ipostesi induttiva per concludere che $Ru'v't'$ è riducibile.

- u, riducibile per ipotesi. Questo accade quando $t = O$.

- $v(Ruvw)w$, dove $t = Sw$. In questo caso avevamo già osservato che $\nu(t) = \nu(w)$, ma vale anche che $l(w) < l(w)$. L'ipotesi induttiva dice dunque che $Ruvw$ è riducibile. Poichè anche v e w sono riducibili, per definizione di riducibilità del tipo freccia anche $v(Ruvw)w$ è riducibile.

Concludiamo che $Ruvt$ è riducibile. □

Occorre a questo punto estendere la 6.15 con ulteriori casi per gestire i termini introdotti nel sistema T.

Dimostrazione.

- Se $t = \pi^1 w$, allora per ipotesi induttiva $w[\overline{u}/\overline{x}]$ è riducibile. Per cui anche $\pi^1(w[\overline{u}/\overline{x}]) = t[\overline{u}/\overline{x}]$ è riducibile.

- Se $t = \pi^1 w$, allora il caso è identico al precedente.

- Se $t = \langle v, w\rangle$, allora per ipotesi induttiva $v[\overline{u}/\overline{x}]$ e $w[\overline{u}/\overline{x}]$ sono riducibili. La tesi qui si ha come conseguenza del 5.7.

- Se $t = O$, allora abbiamo già notato che è riducibile.
- Se $t = Sw$, abbiamo che se $w[\overline{u}/\overline{x}]$ è riducibile, allora $t[\overline{u}/\overline{x}]$ è fortemente normalizzante e quindi riducibile.
- Se $t = Rvwp$, allora per ipotesi anche le sostituzioni nei tre paramentri sono riducibili, e quindi per il 5.8, anche il termine $t[\overline{u}/\overline{x}]$ è riducibile.

□

Come nel caso del calcolo tipato semplice, anche in questo caso abbiamo il teorema come corollario dell'ultima proposizione.

5.2 Rappresentabilità nel Sistema T

Prima di studiare risultati generali sulla rappresentabilità di funzioni nel sistema T di Gödel, mostriamo qualche esempio di funzione rappresentabile.

La somma può essere rappresentata dal termine

$$A = \lambda p^{\mathrm{Int}} q^{\mathrm{Int}}.Rp(\lambda x^{\mathrm{Int}} y^{\mathrm{Int}}.Sx)q.$$

Infatti vediamo che soddisfa la definizione ricorsiva della somma:

$$ApO \rightsquigarrow Rp(\lambda x^{\mathrm{Int}} y^{\mathrm{Int}}.Sx)O \rightsquigarrow p$$

e

$$\begin{gathered} Ap(Sq) \rightsquigarrow Rp(\lambda x^{\mathrm{Int}} y^{\mathrm{Int}}.Sy)(Sq) \rightsquigarrow \\ (\lambda x^{\mathrm{Int}} y^{\mathrm{Int}}.Sy)(Rp(\lambda x^{\mathrm{Int}} y^{\mathrm{Int}}.Sy)q)q \rightsquigarrow \\ S(Apq) \end{gathered}$$

Inoltre nel sistema T è tipabile anche l'esponenziazione, con il termine

$$E = \lambda p^{\mathrm{Int}} q^{\mathrm{Int}}.R(SO)(\lambda x^{\mathrm{Int}} y^{\mathrm{Int}} Mpy)q$$

dove M è il termine che rappresenta la moltiplicazione. In modo simile è possibile rappresentare ogni funzione ricorsiva primitiva. Infatti, dato un termine H che rappresenta una funzione $h\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, allora il termine

$$F = \lambda p^{\mathrm{Int}}.R\overline{n}(\lambda x^{\mathrm{Int}} y^{\mathrm{Int}} Hxy)p$$

rappresenta la funzione f definita con $f(0) = n$ e $f(k+1) = h(f(k), k)$.

Notiamo che per la ricorsione primitiva abbiamo utilizzato il simbolo R soltanto istanziato sul tipo Int, infatti nei casi che abbiamo visto i suoi primi due argomenti sono sempre di tipo Int e Int $\to$ Int $\to$ Int. La definizione del sistema T permette di istanziare R su tipi più complessi, e ciò porta ad avere ulteriori funzioni rappresentabili. Per esempio la funzione di Ackermann, definita da

$$A_0(n) = n + 1$$
$$A_{m+1}(n) = A_m^{n+1}(1)$$

non è primitiva ricorsiva ma è rappresentabile nel sistema T. Infatti dato un naturale n e un termine f di tipo Int $\to$ Int possiamo costruire il suo iterato $f^n(1)$ con il termine $R\overline{1}(\lambda x^{\mathrm{Int}}y\mathrm{Int}.fx)\overline{n}$. Possiamo rappresentare anche la funzione che manda A_m in A_{m+1} con $\lambda n^{\mathrm{Int}}.R\overline{1}(\lambda x^{\mathrm{Int}}y^{\mathrm{Int}}.A_m x)(S\overline{n})$. Concludiamo iterando sulle m, per ottenere la funzione A_m:

$$R(\lambda x^{\mathrm{Int}}.Sx)[\lambda A^{\mathrm{Int}\to\mathrm{Int}}z^{\mathrm{Int}}.\lambda n^{\mathrm{Int}}.R\overline{1}(\lambda x^{\mathrm{Int}}y^{\mathrm{Int}}Ax)(Sn)]\overline{m}$$

in cui notiamo che l'ultimo R introdotto è instanziato sul tipo Int $\to$ Int.

Nel seguito daremo risultati più generali, e caratterizzeremo la classe di funzioni rappresentabili nel sistema T. Diamo prima una definizione fondamentale per tale studio:

Definizione 5.9 Data una teoria T nel linguaggio dell'aritmetica, una funzione $f\colon \mathbb{N} \to \mathbb{N}$ si dice dimostrabilmente totale in T se esiste un formula φ di complessità Σ^0_1 tale che nel modello standard dell'aritmetica valga $\varphi(n,m)$ se e solo se $f(n) = m$ e $T \vdash \forall n\ \exists m\ \varphi(n,m)$.

Notiamo che nel caso in cui f sia calcolabile, una formula Σ^0_1 con argomente n e m che sia vera solo quando $f(n) = m$ esiste sempre, e si può costruire tramite la funzione universale di Kleene.

Lo scopo del resto della sezione è quello di dimostrare il seguente teorema:

Teorema 5.10 Le funzioni rappresentabili nel sistema T sono tutte e sole le funzioni dimostrabilmente totali nell'aritmetica di Peano.

Una delle due implicazioni segue dalla dimostrazione di normalizzazione forte, e la racchiudiamo nella prossima proposizione.

Proposizione 5.11 Le funzioni rappresentabili nel sistema T sono dimostrabilmente totali nell'aritmetica di Peano.

Dimostrazione. Supponendo che esista un termine t di tipo Int $\to$ Int per cui $t\overline{n}$ si riduce a $\overline{m}$ ogniqualvolta valga $m = f(n)$, per dimostrare che f è totale basta dimostrare che $t\overline{n}$ sia fortemente normalizzante per ogni naturale n. A meno di codificare i termini del calcolo con opportuni numeri naturali, detti numeri di Gödel, per cui i predicati di conversione siano primitivi ricorsivi, possiamo sfruttare la dimostrazione della normalizzazione forte della sezione precedente. Infatti per ogni tipo, possiamo trovare un predicato che esprima la riducibilità dei termini di tale tipo (lo si può fare induttivamente sulla complessità dei tipi), e poi ragionare con tutte le induzioni che abbiamo già mostrato. Notiamo che tuttavia non è possibile trovare una formula che esprima uniformemente la riducibilità di tutti i termini del sistema T. Per esprimere tali insiemi occorre fare uso di una quantificazione del secondo ordine, che ovviamente non è presente nelle aritmetiche del primo ordine. □

Per mostrare l'altra implicazione, estendiamo i tipi del sistema T con un nuovo tipo $\mathbf{1}$ e con un termine $\imath$ di tipo $\mathbf{1}$. Ad ogni tipo U di questo sistema esteso associamo un tipo $|U|$ che sia o uguale a $\mathbf{1}$ o tale che $\mathbf{1}$ non vi compaia. Il tipo $|U|$ è definito induttivamente con le seguenti regole:

- Se $U = \mathbf{1} \times V$, allora $|U| = |V|$.
- Se $U = V \times \mathbf{1}$, allora $|U| = |V|$.
- Se $U = \mathbf{1} \to V$, allora $|U| = |V|$.
- Se $U = V \to \mathbf{1}$, allora $|U| = \mathbf{1}$.
- Se $U = V \times W$, con nè V nè W pari a $\mathbf{1}$ allora $|U| = |V| \times |W|$.
- Se $U = V \to W$, con nè V nè W pari a $\mathbf{1}$ allora $|U| = |V| \to |W|$.
- Se U è un tipo atomico, allora $|U| = U$.

Definiamo inoltre una mappa $\flat$ dalle formule dell'aritmetica (scritte senza la disgiunzione) ai tipi del sistema T esteso come segue:

$$\begin{aligned}
\flat(\perp) &= \mathbf{1};\\
\flat(s = t) &= \mathbf{1};\\
\flat(\exists x \varphi) &= |\mathrm{Int} \times \flat(\varphi)|;\\
\flat(\forall x \varphi) &= |\mathrm{Int} \to \flat(\varphi)|;\\
\flat(\varphi \wedge \psi) &= |\flat(\varphi) \times \flat(\psi)|;\\
\flat(\varphi \to \psi) &= |\flat(\varphi) \to \flat(\psi)|.
\end{aligned}$$

La definizione per i casi con i quantificatori è legata all'idea che una formula del tipo $\forall x\ \varphi$ dove la quantificazione universale viene relativizzata ai soli numeri naturali si scriverebbe come $\forall x\ (\mathrm{int}(x) \to \varphi)$. In modo equivalente la formula $\exists x\ \varphi$ si riscriverebbe come $\exists x\ (\mathrm{int}(x) \wedge \varphi)$, dove int esprime il predicato di appartenenza ai numeri naturali.

Definiamo ora la realizzabilità. Si dice che un termine M di tipo $\flat(\varphi)$ chiuso, che sia uguale a $\imath$ o in cui $\imath$ non compare, realizza una formula chiusa φ nelle seguenti situazioni:

- $M = \imath$ e φ è la formula $s = t$ e s e t rappresentano lo stesso numerale;
- φ è la formula $\forall x\ \varphi$ e $M\overline{n}$ realizza $\varphi[\overline{n}/x]$ per tutti gli n.
- φ è la formula $\exists x\ \varphi$ e $\pi^1 M$ si riduce a $\overline{n}$ e $\pi^2 M$ realizza $\varphi[\overline{n}/x]$ per qualche intero n.
- φ è la formula $\psi \wedge \eta$ e $\pi^1 M$ realizza ψ e $\pi^2 M$ realizza η.
- φ è la formula $\psi \to \eta$ e il termine MN realizza η per tutti i termini N che realizzano ψ.

Diciamo inoltre che M realizza un giudizio $\Gamma \vdash \varphi$ con $\Gamma = \{\gamma_1, \dots, \gamma_n\}$ se M realizza la formula $\gamma_1 \to \cdots \to \gamma_n \to \varphi$. Una formula si dice realizzabile se esiste un termine che la realizza.

Notiamo che non esiste alcun termine che realizzi $\perp$.

Dimostriamo ora il seguente importante teorema:

Teorema 5.12 Ogni giudizio dimostrabile nell'aritmetica di Heyting è realizzato da un termine del sistema T.

Dimostrazione. L'idea è quella di procedere induttivamente, dimostrando prima che gli assiomi dell'aritmetica di Heyting sono realizzabili, e poi che avendo a disposizione dei termini che realizzano le premesse delle regole della deduzione naturale, è possibile costruire un termine che ne realizzi la conclusione.

Gli assiomi per l'uguaglianza sono realizzabili, infatti $\forall x(x = x)$ è realizzato dal termine $\imath$, così come l'assioma $\forall x \forall y(x = y) \to (y = x)$. L'assioma di Peano $\forall x(Sx = O \to \perp)$ è sempre realizzato da $\imath$ perché la formula atomica $Sx = O$ non è realizzabile.

Mostriamo ancora la realizzabilità dell'assioma di induzione. Se $|\varphi| = \mathbf{1}$, allora l'assioma è realizzato da $\imath$. In caso contrario, sia t un termine che realizza il passo induttivo $\forall x(\varphi(x) \to \varphi(Sx))$ e sia u un termine che realizza il passo base $\varphi(O)$. Si ha allora che per ogni $n > 0$, il termine

$$t\overline{n-1}(t\overline{n-2}(\cdots t\overline{1}(t\overline{0}u)\cdots))$$

realizza $\varphi(\overline{n})$. Utilizziamo dunque il simbolo di ricorsione per definire il termine $v = \lambda pqn.Rpqn$ che realizza l'assioma di induzione, infatti per ogni n, $vut\overline{n}$ realizza $\varphi(\overline{n})$.

Completiamo la dimostrazione mostrando che i giudizi dimostrabili sono realizzabili. Supponiamo che $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$ con $|\gamma_i| = U_i$ per tutti gli indici i e che le variabili libere in Γ siano $a_1, \ldots, a_l$. Supponiamo inoltre che l'ultimo passo della dimostrazione sia l'introduzione della congiunzione, ovvero che termini con

$$\frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}.$$

Per ipotesi induttiva esistono due termini t e v che realizzano rispettivamente $\Gamma \vdash \varphi$ e $\Gamma \vdash \psi$. Allora il termine $\lambda\overline{x}.\lambda\overline{y}.\langle t\overline{xy}, v\overline{xy}\rangle$ realizza $\Gamma \vdash \varphi \wedge \psi$, dove $\overline{x}$ è un vettore di l variabili di tipo Int e $\overline{y}$ è un vettore di k variabili di tipo $U_1, \ldots, U_k$.

Se supponiamo invece che l'ultimo passo sia un'eliminazione del quantificatore universale, ovvero

$$\frac{\Gamma \vdash \forall z\varphi}{\Gamma \vdash \varphi[b/z]}$$

con z variabile libera in φ, allora per ipotesi induttiva esiste un termine t che rappresenta il giudizio in ipotesi. Sia $\lambda\overline{x}.v$ il termine che rappresenta b, intendendo che le variabili libere di t compaiano fra le a_i. Allora $\lambda\overline{xy}.t\overline{xy}v$ realizza il giudizio di tesi, poiché si sono astratte le variabili libere nei vari termini e si è sfruttata la definizione di realizzabilità. Le altre regole della deduzione naturale si trattano in modo simile. □

Osserviamo che la dimostrazione del teorema precedente fornisce una costruzione esplicita dei termini che realizzano le formule dimostrabili, data la loro dimostrazione.

A questo punto possiamo concludere la dimostrazione del teorema principale.

Proposizione 5.13 Le funzioni dimostrabilmente totali nell'aritmetica di Peano sono rappresentabili nel sistema T.

Dimostrazione. Notiamo che poichè le formule Π^0_2 dimostrabili in Peano sono esattamente quelle dimostrabili in Heyting, basta dimostrare la proposizione relativamente al caso intuizionista. Sia allora f una funzione dimostrabilmente totale. Per ipotesi abbiamo dunque una formula Σ^0_1, sia essa φ, tale che $\mathrm{HA} \vdash \forall n\, \exists m\, \varphi(n, m)$ e $\varphi(n, m)$ è vera quando $f(n) = m$. Per la proposizione 5.12, esiste un termine M che realizza tale formula. Dunque per ogni naturale n, $M\overline{n}$ realizza la formula $\exists m\, \varphi(\overline{n}, m)$. Si ha quindi che $\pi^1(M\overline{n})$ si riduce a un numerale $\overline{m}$ e che $\pi^2(M\overline{n})$ realizza $\varphi(\overline{n}, \overline{m})$. Diciamo che il termine $\lambda x.\pi^1(Mx)$ rappresenta f, infatti $\varphi(\overline{n}, \overline{m})$ è una formula vera, e quindi $f(n) = m$. □

6 Il Sistema F

In questa sezione introduciamo un'ulteriore variante del λ-calcolo tipato detta sistema F, o λ-calcolo polimorfico oppure λ-calcolo tipato del secondo ordine. Essa mantiene e anzi estende l'espressività del sistema T di Gödel e allo stesso tempo evita soluzioni ad hoc come l'inserimento esplicito fra i tipi di un tipo Int. Nel sistema F vedremo che esiste comunque un tipo corrispondente ai numeri naturali, ma la sua costruzione è assai più generale Il sistema permette anche, fra le tante cose, di definire tipi coppia e tipi unione oppure tipi corrispondenti a liste o alberi. La chiave per fare questo è l'introduzione di una quantificazione sui tipi.

Definizione 6.1 I tipi del sistema F sono definiti induttivamente come:

- $U_1, U_2, \ldots$ sono tipi, detti variabili di tipo o anche tipi atomici.
- Se U e V sono tipi, allora anche $(U \to V)$ è un tipo.
- Se X è una variabile di tipo e U è un tipo, allora anche $\Pi X.U$ è un tipo.

Avendo una sorta di quantificazione è utile avere anche la distinzione tra occorrenze libere e legate di variabili di tipo. Un'occorrenza di una variabile di tipo X è legata se X compare in una porzione del tipo nella forma $\Pi X.U$.

Definizione 6.2 I termini del sistema F si definiscono induttivamente come:

- Per ogni tipo U, le variabili $x_1^U, x_2^U, \ldots$ sono termini di tipo U.
- Se t e v sono termini di tipo rispettivamente $U \to V$ e U, allora l'applicazione (tv) è un termine di tipo V.
- Se t è un termine di tipo V e x è una variabile di tipo U, allora l'astrazione $(\lambda x.t)$ è un termine di tipo $U \to V$.
- Se u è un termine di tipo U e X è una variabile di tipo che non occorre libera nei tipi delle variabili libere di u, allora $\Lambda X.u$ è un termine di tipo $\Pi X.U$.
- Se t è un termine di tipo $\Pi X.U$ e V è un tipo allora tV è un termine di tipo $U[V/X]$, che indica il tipo U in cui sono state sostituite le occorrenze libere di X con V, con l'usuale attenzione per la cattura di variabili.

Occorre ancora definire il comportamento dei nuovi termini sotto l'azione della conversione.

Definizione 6.3 Un termine u si converte a v quando v è ottenuto sostituendo in u un sottotermine u' con un termine v' tali che valga una delle seguenti:

- $u' = (\lambda x.w)t$ e $v' = w[w/x]$ dove x è una variabile e w e t sono termini tali che la sostituzione $w[w/x]$ sia permessa.
- $u' = (\Lambda X.v)U$ e $v' = v[U/X]$, con v un termine, U un tipo e X una variabile di tipo.

Riassumendo, le regole del calcolo sono:

$$\boxed{\begin{array}{c}(\lambda x.w)t \rightsquigarrow_C w[t/x] \\ (\Lambda X.w)U \rightsquigarrow_C w[U/X]\end{array}}$$

Attraverso l'astrazione sui tipi, o generalizzazione, possiamo comporre programmi che operano uniformemente su dati di differenti tipi. L'esempio più facile è dato dal termine identità $\Lambda X.\lambda x^X.x$, di tipo $\Pi X.X$. Dato un termine u di tipo U, si ha che $(\Lambda X.\lambda x^X.x)U$ ha tipo U e si riduce a $\lambda x^U.x$ e quindi $(\Lambda X.\lambda x^X.x)Uu$ si riduce semplicemente a u.

Altri tipi che possono essere costruiti corrispondo al tipo prodotto e il tipo unione disgiunta. Per esempio un tipo prodotto tra U e V può essere rappresentato nel sistema F dal tipo $U \times V = \Pi X.(U \to V \to X) \to X$. Possiamo scrivere un termine che produce una coppia, ovvero

$$\mathrm{CPL} = \lambda x^U y^V.\Lambda X.\lambda p^{U \to V \to X}.pxy$$

e le conseguenti proiezioni

$$\begin{aligned}\pi^1 &= \lambda x^{U \times V}.xU(\lambda y^U z^V.y) \\ \pi^2 &= \lambda x^{U \times V}.xV(\lambda y^U z^V.z)\end{aligned}$$

rispettivamente sul primo e secondo fattore.

Mostriamo per esempio la riduzione di

$$\begin{gathered}\pi^1(\mathrm{CPL}uv) \rightsquigarrow \\ (\lambda x^{U \times V}.xU(\lambda y^U z^V.y))(\mathrm{CPL}uv) \rightsquigarrow \\ (\mathrm{CPL}uv)U(\lambda y^U z^V.y) \rightsquigarrow \\ (\Lambda X.\lambda p^{U \to V \to X}puv)U(\lambda y^U z^V.y) \rightsquigarrow \\ (\lambda y^U z^V.y)uv \rightsquigarrow u.\end{gathered}$$

Chiaramente tutti i termini precedenti possono essere generalizzati su U e su V.

Il tipo somma tra U e V è definito come il tipo $U + V = \Pi X.(U \to X) \to (V \to X) \to X$ con il costruttore con la variabile di tipo U come

$$\mathrm{UN1} = \lambda u^U.\Lambda X.\lambda p^{U\to X} q^{V \to X}.pu$$

e l'equivalente sulla variabile di tipo V come

$$\mathrm{UN2} = \lambda v^V.\Lambda X.\lambda p^{U\to X} q^{V \to X}.qv.$$

Mostriamo anche un termine di eliminazione per il tipo unione, ovvero

$$Duvt = tUuv$$

per cui vale la riduzione

$$\begin{gathered} Duv(\mathrm{UN1}r) = (\Lambda X.\lambda p^{U\to X} q^{V\to X}.pr)Uuv \rightsquigarrow \\ (\lambda p^{U\to X} q^{V\to X}.pr)uv \rightsquigarrow ur \end{gathered}$$

e l'altra equivalente $Duv(\mathrm{UN2}r) \rightsquigarrow vr$. L'idea intuitiva del tipo somma corrisponde alle union del linguaggio C, e il termine D serve per distinguere in quale dei due tipi l'ultimo parametro è utilizzato.

La costruzione di questi due tipi è in realtà generalizzabile a qualunque tipo di dato algebrico. Ne mostriamo un ultimo esempio, più chiaro. Supponiamo di voler costruire un tipo relativo agli alberi binari in cui ogni nodo contiene un dato di tipo U. L'albero è generato utilizzando due costruttori. Il primo, che chiamiamo E, è l'albero vuoto. Il secondo, R, ha come parametri il dato da mettere nella radice dell'albero e i due alberi figli della radice. Se chiamassimo X il tipo dell'albero binario, risulterebbe che i costruttori avrebbero tipo rispettivamente tipo X e $U \to X \to X \to X$. Da qui costruiamo l'effettivo tipo dell'albero

$$T = \Pi X.X \to (U \to X \to X \to X) \to X.$$

I due costruttori sono rappresentati dai termini:

$$\begin{gathered} E = \Lambda X.(\lambda x^X y^{U\to X\to X\to X}.x) \\ R = \lambda r^U l^T r^T \Lambda X.(\lambda x^X y^{U\to X\to X\to X}.yr(lXxy)(rXxy)) \end{gathered}$$

Se $u_1, u_2, \ldots$ sono termini di tipo U un esempio di albero binario è

$$Ru_1(Ru_2(Ru_4EE)E)(Ru_3EE).$$

È possibile operare su queste strutture con funzioni definite con pattern-matching. Per esempio se abbiamo un termine f di tipo $U \to V \to V \to V$ e un termine iniziale v di tipo V, possiamo definire l'iteratore di visita anticipata sull'albero t come:

$$Ivft = tVv(\lambda n^U l^V r^V.fnlr).$$

La funzione calcolata da Ivf è tale che sull'albero nullo dia risultato v, e sull'albero con nodo di valore u e sottoalberi sinistro e destro t_l e t_r ritorni fuv_lv_r, dove v_l e v_r sono i valori calcolati da Ivf sui due sottoalberi. Discussioni più dettagliate si possono trovare su [4].

Notiamo tuttavia che l'introduzione del sistema F porta con sè un difficoltà nel lavorare per induzione sulla complessità dei termini. Infatti nell'esempio dell'identità, il termine può essere istanziato su un qualunque tipo, e dunque l'istanziazione non riduce la complessità di un termine, ma potenzialmente la può aumentare. Per esempio questo accade nel termine $(\Lambda X.\lambda x^X.x)(\Pi X.X)$. Nello studio del sistema F ci imbatteremo in problemi di questo genere e presenteremo dei modi per aggirarli.

6.1 Normalizzazione per il Sistema F

Per dimostrare la normalizzazione forte nel sistema F, un primo tentativo può essere quello di estendere la dimostrazione già fatta per il λ-calcolo tipato semplice e per il sistema T. Questo però non è possibile, perchè nello spirito della definizione 4.4, vorremmo provare a definire i riducibili di tipo $\Pi X.V$ come i termini t tali che per ogni tipo U il termine tU è riducibile di tipo $V[U/X]$. Questo conduce a una definizione impredicativa, perchè per conoscere la riducibilità di un termine di tipo $V[U/X]$ occorre conoscere la riducibilità dei suoi sottotermini. Nel caso in cui per esempio si avesse $U = \Pi X.V$, le dimostrazioni per induzione fallirebbero. Occorre dunque un procedimento adatto ad aggirare il problema.

Come prima cosa, estendiamo la definizione di riducibilità del calcolo tipato semplice con i nuovi casi introdotti nel sistema F.

Definizione 6.4 Un termine t si dice neutrale se è in una delle seguenti forme: x, vu o vU, in cui x è una variabile, v e u sono termini e U è un tipo.

A questo punto possiamo definire i candidati di riducibilità. Essi sono insiemi di termini di uno stesso tipo per cui valgono le tre proprietà che avevamo dimostrato valere per i riducibili. L'idea quindi è quella di costruire degli ulteriori insiemi di termini detti riducibili parametrici, la cui costruzione induttiva sui tipi atomici e sui tipi freccia corrisponde a quella data per i riducibili nel tipato semplice, e sui tipi della forma $\Lambda X.V$ corrisponde alla proposta fatta sopra, però utilizzando come parametri, al posto dei riducibili, dei generici candidati.

Definizione 6.5 Un candidato di riducibilità (o semplicemente candidato) di tipo U è un insieme $\mathcal{R}$ di termini di tipo U per cui valgono:

(CR1) Se $t \in \mathcal{R}$ allora t è fortemente normalizzante.
(CR2) Se $t \in \mathcal{R}$ e t' è un termine ottenuto da una riduzione di t, cioè $t \rightsquigarrow t'$, allora $t' \in \mathcal{R}$.
(CR3) Se t è neutrale, e per ogni conversione di uno step di t si ottiene un termine $t' \in \mathcal{R}$, allora anche $t \in \mathcal{R}$.

Definiamo una notazione semplice per i candidati di tipo freccia costruiti esattamente come i riducibili del calcolo tipato semplice.

Definizione 6.6 Se $\mathcal{R}$ e $\mathcal{S}$ sono insiemi di termini di tipo rispettivamente U e V, si definisce l'insieme $\mathcal{R} \to \mathcal{S}$ come l'insieme dei termini di tipo $U \to V$ per cui per ogni termine $u \in \mathcal{R}$ si ha che $tu \in \mathcal{S}$.

Occorre dimostrare che questi insiemi siano effettivamente dei candidati, e lo facciamo nel seguente lemma.

Lemma 6.7 Se $\mathcal{R}$ e $\mathcal{S}$ sono candidati per i tipi U e V, allora $\mathcal{R} \to \mathcal{S}$ è candidato di tipo $U \to V$.

Dimostrazione. Per mostrare (CR1) prendiamo $t \in \mathcal{R} \to \mathcal{S}$ e una variabile x di tipo U. Poiché le variabili sono sia normali che che neutrali, $x \in \mathcal{R}$ e quindi $tx \in \mathcal{S}$. Inoltre $\nu(t) < \nu(tx)$, e quindi siccome tx è fortemente normalizzante, anche t lo è.

Per (CR2),se $t \rightsquigarrow t'$, per ogni $u \in \mathcal{R}$ si ha che $tu \rightsquigarrow t'u$. Usando la (CR2) su $\mathcal{S}$, si ottiene che $t'u \in \mathcal{S}$. Allora $t'\mathcal{R} \to \mathcal{S}$.

Infine consideriamo t neutrale di tipo $U \to V$ per cui per tutte le conversioni di uno step $t \rightsquigarrow t'$ si ha che $t' \in \mathcal{R} \to \mathcal{S}$. Sia $u \in \mathcal{R}$, e per induzione su $\nu(u)$ dimostriamo che tu si riduce in uno step a termini in $\mathcal{S}$. Infatti poiché t è normale, tu si può ridurre solo a $t'u$ o a tu' per opportuni termini t' e u'. Ma il primo appartiene a $\mathcal{S}$ perchè $t' \in \mathcal{R} \to \mathcal{S}$, e il secondo ci appartiene per ipotesi induttiva in quanto $\nu(u') < \nu(u)$. Per (CR3) su $\mathcal{S}$ allora $tu \in \mathcal{S}$. $\square$

Possiamo dunque definire cosa sono i riducibili parametrici:

Definizione 6.8 Sia $T[\underline{X}]$ un tipo con variabili libere in $\underline{X}$. Sia $\underline{U}$ un vettore di tipi della stessa lunghezza e siano $\mathcal{R}$ dei rispettivi candidati. Possiamo allora definire l'insieme $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$ di termini riducibili parametrici di tipo $T[\underline{U}/\underline{X}]$ nel modo seguente:

(1) Se $T = X_i$ per qualche indice i, allora $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}] = \mathcal{R}_i$.
(2) Se $T = V \to U$, allora $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}] \to \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$.
(3) Se $T = \Pi Y.W$, allora $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$ è l'insieme dei termini t di tipo $T[\underline{U}/\underline{X}]$ tali che per ogni tipo V e per ogni candidato $\mathcal{S}$ di tale tipo vale che $tV \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$.

Ci servirà nel seguito utilizzare induttivamente i riducibili parametrici come parametri di altri riducibili parametrici. Nel prossimo lemma dimostreremo che è possibile farlo in quanto i riducibili parametrici rispettano le proprietà di candidati.

Lemma 6.9 $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$ è un candidato di riducibilità di tipo $T[\underline{U}/\underline{X}]$.

Dimostrazione. Lo facciamo per induzione sulla complessità del tipo T. Nel caso in cui T è una variabile individuale, il teorema è una tautologia. Il caso in cui $T = V \to W$ lo abbiamo già fatto. Manca solo il caso in cui $T = \Pi Y.W$.

Verifichiamo (CR1). Sia $t \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$, V un tipo e $\mathcal{S}$ un suo candidato. Allora $tV \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$ per definizione. Usando l'ipotesi induttiva sul tipo W si ha che tV è fortemente normalizzante. Ma vale anche che $\nu(t) < \nu(tV)$. Quindi anche t è fortemente normalizzante.

Per (CR2), supponiamo di avere $t \rightsquigarrow t'$ con uno step di conversione. Allora, $tV \rightsquigarrow t'V$, per cui $t'V \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$ e quindi $t' \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

Infine, per (CR3), consideriamo t un qualunque termine di tipo T neutrale. Supponiamo che per ogni t' ottenuto dalla conversione di t in un singolo step si abbia t' riducibile parametrico. Allora per ogni tipo V e relativo condidato $\mathcal{S}$, le uniche conversioni di tV sono della forma $tV \rightsquigarrow t'V$. Usando l'ipotesi induttiva allora anche tV è riducibile parametrico, e quindi si ha la tesi. □

Il prossimo lemma è utile per studiare il comportamento della riducibilità parametrica rispetto alle instanziazioni.

Lemma 6.10 Sia T un tipo con variabili libere Y e $\underline{X}$ e V un tipo. Siano $\underline{\mathcal{R}}$ candidati per $\underline{X}$. Allora vale che $\mathrm{RED}_{T[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}][\mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y]$.

Dimostrazione. Come prima, facciamo un'induzione sulla complessità del tipo T. Per comodità, usiamo l'abbreviazione $A = \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$.

Iniziamo con il caso in cui $T = Z$ è una variabile individuale diversa da Y. Allora vale che

$$\mathrm{RED}_{T[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_Z[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_Z[\underline{\mathcal{R}}/\underline{X}][A/Y].$$

Se invece $T = Y$ si ha che

$$\mathrm{RED}_{T[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_Y[\underline{\mathcal{R}}/\underline{X}][\mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y].$$

Consideriamo ora il caso in cui $T = U \to W$. Vale che

$$\begin{aligned}\mathrm{RED}_{T[V/Y]}[\underline{\mathcal{R}}/\underline{X}] =& \mathrm{RED}_{U[V/Y]\to W[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \\ & \mathrm{RED}_{U[V/Y]}[\underline{\mathcal{R}}/\underline{X}] \to \mathrm{RED}_{W[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \\ & \mathrm{RED}_U[\underline{\mathcal{R}}/\underline{X}][A/Y] \to \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][A/Y] = \\ & \mathrm{RED}_{U\to W}[\underline{\mathcal{R}}/\underline{X}][A/Y].\end{aligned}$$

Sia Z come prima e svolgiamo il caso $T = \Pi Z.W$. Per definizione, $\mathrm{RED}_{\Pi Z.W[V/Y]}[\underline{\mathcal{R}}/\underline{X}]$ è l'insieme di tutti i termini t per cui per ogni tipo U e relativo candidato $\mathcal{S}$ vale che

$$tU \in \mathrm{RED}_{W[V/Y]}[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Z] = \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Z][A/Y].$$

Dunque si ottiene la tesi per la definizione di $\mathrm{RED}_{\Pi Z.W}$.

Infine il caso in cui $T = \Pi Y.W$ è semplice perchè Y non occorre libera in T. □

I seguenti due teoremi sono gli equivalenti per questa dimostrazione dei lemmi 4.7, 5.7 e 5.8.

Lemma 6.11 Se per ogni tipo V e per ogni candidato di riducibilità $\mathcal{S}$ per V vale che $w[V/Y] \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$ per ogni termine w di tipo W, allora $\Lambda Y.w \in \mathrm{RED}_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$.

Dimostrazione. Dimostriamo per induzione su $\nu(w)$ che tutte le conversioni in uno step di $(\Lambda Y.w)V$ sono in $\mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$. Una conversione di tali conversioni può essere soltanto di due forme. La prima è $(\Lambda Y.w')V$, con w' una conversione di w. Ma allora $\nu(w') < \nu(w)$ e si usa l'ipotesi induttiva. La seconda forma è del tipo $w[V/Y]$, e questa è riducibile parametrico per ipotesi del lemma.

Allora la dimostrazione si conclude per (CR3). □

Lemma 6.12 Se $t \in \mathrm{RED}_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$, allora $tV \in \mathrm{RED}_{W[V/Y]}[\underline{\mathcal{R}}/\underline{X}]$ per ogni tipo V.

Dimostrazione. Per la definizione di $\mathrm{RED}_{\Pi Y.W}$, per ogni candidato $\mathcal{S}$ per V vale che $tV \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$. Allora vale anche per $\mathcal{S} = \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$ e la tesi segue per il 6.10. $\square$

Lemma 6.13 Se per ogni $u \in \mathrm{RED}_U[\underline{\mathcal{R}}/\underline{X}]$ vale che $v[u/x] \in \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$, allora $\lambda x^U.v \in \mathrm{RED}_{U\to V}[\underline{\mathcal{R}}/\underline{X}]$.

Dimostrazione. Dimostriamo per induzione su $\nu(u)+\nu(v)$ che tutte le conversioni di $(\lambda x^U.v)u$ sono riducibili parametrici. Infatti tale termine si converte in $(\lambda x^U.v)u'$, con u' conversione di u, oppure in $(\lambda x^U.v')u$ con v' conversione di v, oppure in $v[u/x]$. I primi due casi si risolvono con l'ipotesi induttiva, il terzo con l'ipotesi del lemma.

Infine il teorema si dimostra per la proprietà (CR3). $\square$

Utilizziamo ora l'idea di riducibilità parametrica per definire i riducibili, nello spirito che a posteriori potremo dire che i riducibili sono esattamente i termini fortemente normalizzanti. In tal caso la definizione corrisponderebbe con quella data per il λ-calcolo tipato semplice.

Definizione 6.14 Un termine t di tipo T è riducibile se è in $\mathrm{RED}_T[\underline{\mathcal{SN}}/\underline{X}]$ dove $X_1, \ldots, X_m$ sono le variabili libere di T e $\underline{\mathcal{SN}}_i$ è l'insieme dei termini fortemente normalizzanti di tipo X_i.

Infine la proposizione che segue svolge lo stesso ruolo 6.15.

Proposizione 6.15 Sia t un termine di tipo T le cui variabili libere sono $x_1, \ldots, x_n$ di tipo rispettivamente $U_1, \ldots, U_n$. Supponiamo che le variabili libere dei tipi T e di tutti gli U_i siano $X_1, \ldots, X_m$. Siano $\mathcal{R}_1, \ldots, \mathcal{R}_m$ candidati di riducibilità per dei tipi $V_1, \ldots, V_m$ e siano inoltre $u_1, \ldots, u_n$ termini di tipo $U_1[\underline{V}/\underline{X}], \ldots U_n[\underline{V}/\underline{X}]$ presi nei rispettivi $\mathrm{RED}_{U_i}[\underline{\mathcal{R}}/\underline{X}]$. Allora $t[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

Dimostrazione. Per induzione sulla complessità di t. Distinguiamo allora i seguenti casi:

(i) $t = x_i$. Questo caso è una tautologia.
(ii) $t = wv$, con w di tipo $W \to T$ e v di tipo W. Per ipostesi induttiva vale che $w[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \mathrm{RED}_{W\to T}[\underline{\mathcal{R}}/\underline{X}]$ e che $v[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$. In questo caso la tesi segue dalla definizione di $\mathrm{RED}_{W\to T}$.
(iii) $t = wS$. Questo caso è una diretta conseguenza del lemma 6.12 sull'istanziazione.
(iv) $t = \Lambda Z.Y$. Questo discende dal lemma 6.11 sulla generalizzazione.
(v) $t = \lambda y^P.w$. Questo caso si fa con il lemma 4.7 sui tipi freccia.

$\square$

Come corollari otteniamo il seguente risultato e il teorema di normalizzazione forte per il sistema F.

Proposizione 6.16 Tutti i termini del sistema F sono riducibili.

Dimostrazione. Basta usare la proposizione precedente e prendere $\mathcal{R}_i = \mathcal{SN}_i$ e $u_i = x_i$ e concludere per (CR1). $\square$

Teorema 6.17 Tutti i termini del sistema F sono fortemente normalizzanti.

6.2 Rappresentabilità in F

Nel sistema F è presente un tipo corrispondente ai numeri naturali, ovvero il tipo

$$\mathrm{Int} = \Pi X.X \to (X \to X) \to X$$

dove si hanno i termini corrispondenti allo zero e al successore rispettivamente uguali a

$$O = \Lambda X.\lambda x^X.\lambda f^{X\to X}.x$$

$$S = \lambda n^{\mathrm{Int}}.\Lambda X.\lambda x^X.\lambda f^{X\to X}.f(nXxf).$$

Possiamo scrivere allora i numerali come le forme normali di S^nO per ogni n naturale. A questo punto dimostriamo il lemma:

Lemma 6.18 I numerali sono tutti e soli i termini chiusi in forma normale di tipo Int.

Dimostrazione. In modo equivalente a quanto fatto con il λ-calcolo non tipato (3.11), un termine chiuso e in forma normale di tipo Int deve necessariamente essere in forma normale di testa

$$t = \Lambda X.\lambda f^{X\to X}x^X.v.$$

Mostriamo per induzione sulla complessità di v che v deve essere della forma f^nx per qualche naturale n. Se infatti v fosse nella forma wu o wU con u termine distinto da f e U un tipo, per normalità w non potrebbe essere un'astrazione o una generalizzazione. Allora w dovrebbe essere nella forma $w'u'$ oppure $w'U'$, dove w' e u' sono termini e U' è un tipo. Ma il tipo di w'

dovrebbe allora essere più complessio di quello delle variabili f e x, e dunque w' deve necessariamente essere o un'astrazione o una generalizzazione. Questo è contraddittorio con il fatto che w è in forma normale.

Abbiamo allora ottenuto come risultato che v deve essere o una variabile o un termine nella forma fu. Nel primo caso, osservando i tipi, si ottiene che $v = x$ e dunque t sarà il numerale corrispondente allo zero. Nel secondo caso per ipotesi induttiva u è nella forma $f^n x$ e dunque t corrisponde al numerale $n + 1$. $\square$

In modo equivalente a quanto già fatto con le altre versioni del λ-calcolo è possibile definire la nozione di funzione rappresentabile, e poi dare una caratterizzazione di tali funzioni.

Teorema 6.19 Le funzioni dimostrabilmente totali in PA2 sono tutte e sole le funzioni rappresentabili nel sistema F.

Una cosa da notare è che affinchè nel lemma 6.10 abbia senso il predicato $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}][\mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y]$, occorre che $\mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$ non sia soltanto un predicato, ma anche un effettivo insieme. Notiamo anche che il principio di estensione ci permette di passare da predicati a insiemi senza problemi.

Iniziamo con la freccia più semplice, ovvero $\Leftarrow$. Come nel caso del sistema T, la dimostrazione della forte normalizzazione di un termine, può essere interpretata in PA2 come una dimostrazione della totalità della funzione corrispondente a tale termine. Infatti per la dimostrazione sono stati utilizzati due principi:

- Lo schema di comprensione, necessario a dimostrare che i riducibili parametrici sono candidati di riducibilità.
- Il principio di induzione.

Notiamo che tuttavia non è nuovamente possibile esprimere la riducibilità parametrica in modo uniforme su tutti i tipi del sistema F, ma soltanto caso per caso per ciascun tipo, in quanto per ogni generalizzazione occorre una quantificazione universale del secondo ordine.

L'obiettivo del resto della sezione è quello di dimostrare l'ultima freccia del teorema, ovvero la seguente proposizione.

Proposizione 6.20 Le funzioni dimostrabilmente totali in HA2 sono rappresentabili nel sistema F.

L'idea è quella di trovare una traduzione dalle dimostrazioni di totalità di una funzione in programmi del sistema F. Per fare ciò iniziamo con l'associare ad ogni formula φ in HA2 (dimostrabile o non dimostrabile) un tipo $[\![\varphi]\!]$ del sistema F. L'associazione viene fatta con le seguenti regole:

- $[\![a = b]\!] = S$, dove S è un tipo fissato per cui esiste almeno un termine di tale tipo, per esempio $S = \Pi X.X \to X$.
- $[\![\varphi \to \psi]\!] = [\![\varphi]\!] \to [\![\psi]\!]$.
- $[\![\forall x.\varphi]\!] = [\![\varphi]\!]$, per le quantificazioni del primo ordine.
- $[\![\forall X.\varphi]\!] = \Pi X.[\![\varphi]\!]$ per le quantificazioni del secondo ordine.

Notiamo che con queste regole le variabili del primo ordine scompaiono completamente, per cui $[\![\varphi[a/x]]\!] = [\![\varphi]\!]$.

Inoltre è possibile esprimere i connettivi di congiunzione e disgiunzione con le implicazioni e la quantificazione al secondo ordine, e associare ad essi un relativo tipo:

$$[\![\varphi \wedge \psi]\!] = [\![\forall X((\varphi \to \psi \to X) \to X)]\!] = \Pi X.([\![\varphi]\!] \to [\![\psi]\!] \to X) \to X.$$

Questo tipo corrisponde a quello che avevamo definito essere $[\![\varphi]\!] \times [\![\psi]\!]$. In modo simile vale che il tipo relativo a $\varphi \vee \psi$ è uguale al tipo $[\![\varphi]\!] + [\![\psi]\!]$. Notiamo tuttavia che i tipi $[\![\varphi \wedge \psi]\!]$ (rispettivamente $[\![\varphi \vee \psi]\!]$) sono differenti dai tipi $[\![\psi \wedge \varphi]\!]$ (rispettivamente $[\![\psi \vee \varphi]\!]$).

Analogamente possiamo trovare la traduzione della formula $\mathrm{Int}(x)$. Essa è infatti il tipo $\Pi X.X \to (X \to X) \to X = \mathrm{Int}$.

Ora possiamo definire una traduzione di ogni dimostrazione di una formula φ con un termine di tipo $[\![\varphi]\!]$. Utilizziamo la notazione $[\![\delta]\!]$ per indicare la traduzione di una dimostrazione δ. Tale traduzione avviene secondo le seguenti regole:

- Se δ è l'ipotesi η_i, allora $[\![\delta]\!]$ è la variabile x_i di tipo $[\![\eta_i]\!]$.
- Ciascun assioma viene tradotto in un qualunque termine chiuso fissato del tipo corretto.
- Se l'ultimo passo di δ è la deduzione di una formula $\eta \to \varphi$ con l'introduzione dell'implicazione fatta eliminando l'ipotesi η corrispondente alla variabile x e se chiamiamo ε la dimostrazione di φ, allora $[\![\delta]\!] = \lambda x^{[\![\eta]\!]}.[\![\varepsilon]\!]$. La situazione rappresentata è la seguente:

$$\dfrac{\begin{array}{c}\eta \\ \vdots\,\varepsilon \\ \varphi\end{array}}{\eta \to \varphi}\,\delta$$

- Se δ è la dimostrazione di una formula φ il cui ultimo passo è l'eliminazione dell'implicazione fatta a partire dalle formule ψ e $\psi \to \varphi$, dimostrate rispettivamente con ε e con γ, allora $[\![\delta]\!] = [\![\gamma]\!][\![\varepsilon]\!]$. La situazione è illustrata da:

$$\frac{\begin{array}{cc} \vdots\,\gamma & \vdots\,\varepsilon \\ \psi \to \varphi & \psi \end{array}}{\varphi}\,\delta$$

- Se l'ultimo passo di δ è l'introduzione o l'eliminazione di un quantificatore del primo ordine, allora la traduzione non fa nulla. In particolare se l'ultima inferenza di δ introduce la quantificazione universale $\forall x\varphi$ a partire da φ, formula dimostrata con ε, allora $[\![\delta]\!] = [\![\varepsilon]\!]$. Similmente, se l'ultima inferenza di δ dimostra $\varphi[a/x]$ a partire da $\forall x\varphi$, formula dimostrata con ε, allora si ha ancora $[\![\delta]\!] = [\![\varepsilon]\!]$. Le situazioni illustrate sono le seguenti:

$$\frac{\begin{array}{c} \vdots\,\varepsilon \\ \varphi \end{array}}{\forall x\varphi}\,\delta \qquad\qquad \frac{\begin{array}{c} \vdots\,\varepsilon \\ \forall x\varphi \end{array}}{\varphi[a/x]}\,\delta$$

- Se δ termina con l'introduzione di un quantificatore universale del secondo ordine per dimostrare una formula $\forall X\varphi$ a partire da φ, dimostrata a sua volta da ε, allora $[\![\delta]\!] = \Pi X.[\![\varepsilon]\!]$. La situazione corrispondente è:

$$\frac{\begin{array}{c} \vdots\,\varepsilon \\ \varphi \end{array}}{\forall X\varphi}\,\delta$$

- Infine se δ termina con l'eliminazione di una quantificazione del secondo ordine in cui partendo da una formula $\forall X.\varphi$, dimostrata con ε si deduce $\varphi[\lambda x.\psi/X]$, allora $[\![\delta]\!] = [\![\varepsilon]\!][\![\psi]\!]$ (in cui nel secondo membro la prima traduzione è quella di una dimostrazione, la seconda è quella di una formula). La situazione è la seguente:

$$\frac{\begin{array}{c} \vdots\,\varepsilon \\ \forall x\varphi \end{array}}{\varphi[\lambda x.\psi/X]}\,\delta$$

Osserviamo che le regole appena presentate rispettano i tipi di tutte le formule. Inoltre le variabili libere presenti nelle traduzioni corrispondono ad altrettante ipotesi utilizzate nella dimostrazione.

Le definizioni mettono in luce un legame molto forte tra il sistema F e le dimostrazioni nella logica intuizionista. Per fare un esempio supponiamo di avere la dimostrazione:

$$\dfrac{\dfrac{\begin{array}{c}\varphi\\ \vdots\,\varepsilon\\ \psi\end{array}}{\varphi \to \psi} \qquad \begin{array}{c}\vdots\,\delta\\ \varphi\end{array}}{\psi}$$

Essa può essere normalizzata alla semplice

$$\begin{array}{c}\vdots\,\delta\\ \varphi\\ \vdots\,\varepsilon\\ \psi\end{array}$$

Questo corrisponde a quanto accade alle relative traduzioni: $(\lambda x^{[\![\varphi]\!]}.[\![\varepsilon]\!])[\![\delta]\!]$ si riduce al termine $[\![\varepsilon]\!][[\![\delta]\!]/x]$. Si può interpretare $\lambda x^{[\![\varphi]\!]}.[\![\varepsilon]\!]$ come un termine che prende una dimostrazione x di φ e tramite ε dimostra ψ.

Occorre notare però che la definizione precedente nasconde una difficoltà: infatti non esiste alcun termine chiuso che può tradurre l'assioma $\forall x(Sx = O \to \bot)$. Infatti la sua traduzione, ricordando che $\bot$ è una scorciatoia sintattica per $\forall XX$, è di tipo $S \to (\Pi X.X)$.

Per risolvere il problema si potrebbe aggiungere al sistema F un termine artificiale Ω di tipo $\Pi X.X$ e tradurre l'assioma con $\lambda x^S.\Omega$.

Noi tuttavia seguiremo un'altra strada, che consiste nel sostituire l'assioma problematico con uno più debole, ovvero con $\forall x \forall y(Sx = O \to y = O)$. Chiamiamo HA2m la versione modificata di HA2 in cui è stata fatta tale sostituizione, e vedremo che nonostante HA2m sia più debole di HA2, è equivalente ad esso quando si tratta di dimostrare la totalità di funzioni. Pertanto ci basterà dimostrare il teorema con HA2m per avere il risultato che cerchiamo. Verifichiamo quanto detto nella seguente proposizione:

Proposizione 6.21 Una funzione è dimostrabilmente totale in HA2m se e solo se lo è in HA2.

Dimostrazione. Sia φ la formula che esprime la totalità di una funzione, nella forma $\forall n \exists m \psi$ in cui ψ è una formula Σ^0_1. Supponiamo che HA2 $\vdash \varphi$. Siano

$P = \forall x(Sx = O \to \bot)$ e $P' = \forall x \forall y(Sx = O \to y = O)$ i due assiomi. Notiamo che da P' è possibile dedurre $P \vee \forall y(y = O)$. Da entrambe le componenti siamo comunque in grado di dimostrare φ. Infatti dalla prima si può direttamente usare la dimostrazione fatta in HA2. Dalla seconda discende che φ è vera se e solo se è vera $\exists m\psi[O/n]$. Ma questa è una formula Σ_1^0 che esprime la terminazione di $f(0)$. Quindi è vera nel modello standard e dunque è anche dimostrabile. □

Il seguente lemma è un corrispettivo del lemma 6.18 nel caso della logica.

Lemma 6.22 Esiste un'unica deduzione normale di $\mathrm{Int}(S^n O)$, ovvero $\check{n}$.

Dimostrazione. Sia δ una deduzione normale di $\mathrm{Int}(S^n O)$. Dimostriamo come prima cosa che δ non può terminare con un'eliminazione, perchè partendo da soli assiomi (del primo ordine) non è possibile arrivare alla conclusione utilizzando soltanto eliminazioni. Quindi l'ultimo passo deve essere l'introduzione del quantificatore universale del secondo ordine, a partire da $X(O) \to \forall x(X(x) \to X(Sx)) \to X(\overline{n})$. Per le stesse ragioni di prima la dimostrazione di questa formula deve essere un'introduzione, e l'ultimo passo deve avere come premessa la dimostrazione di $\forall x(X(x) \to X(Sx)) \to X(\overline{n})$ a partire da $X(O)$. Poichè non si può applicare alcuna regola di eliminazione a quest'ultima formula, il passo ancora precedente è nuovamente un'introduzione. Risalendo ancora di un passo, si deve dimostrare $X(\overline{n})$ a partire da $X(O)$ e $\forall x(X(x) \to X(Sx))$. Questo non si può fare con un'introduzione, in quanto la formula da dimostrare è atomica. Se $\overline{n} = O$, allora abbiamo concluso, in quanto l'ultima formula da dimostrare è una delle ipotesi. In caso contrario si deve avere $\overline{n} = S\overline{n-1}$. L'unica eliminazione possibile che termina con $X(S\overline{n-1}$ passa attraverso $X(\overline{n-1}) \to X(S\overline{n-1}$, che a sua volta deriva da un'eliminazione sull'ipotesi $\forall x(X(x) \to X(Sx))$. La premessa minore dell'eliminazione dell'implicazione necessita di una dimostrazione della formula $X(\overline{n-1}$ a partire dalle due ipotesi che avevamo e dagli assiomi dell'aritmetica. In realtà il ragionamento è uguale a quello fatto per n, e questo conclude la dimostrazione. □

Infine concludiamo con la dimostrazione del risultato principale.

Dimostrazione. Consideriamo la formula $\varphi(x, y)$ che esprime il fatto che dato un algoritmo con input x termini con output $y = f(x)$, a meno di una codifica con numeral. Supponiamo di voler dimostrare in HA2

$$\forall n \in \mathbb{N}\; \exists m \in \mathbb{N}.\varphi(x, y)$$

ovvero
$$\forall x(\mathrm{Int}(x) \to \exists y.(\mathrm{Int}(y) \wedge \varphi(x, y))).$$
Chiamiamo δ tale dimostrazione. Ad essa associamo un termine $[\![\delta]\!]$ di tipo $\mathrm{Int} \to (\mathrm{Int} \times [\![\varphi]\!])$ e un termine $t = \lambda x^{\mathrm{Int}}.\pi^1([\![\delta]\!]x)$ che ne contiene il significato algoritmico.

Infatti per ogni naturale n, la forma normale della deduzione di $\exists y.(\mathrm{Int}(y) \wedge \varphi(S^n O, y))$ deve necessariamente terminare con l'introduzione

$$\frac{\mathrm{Int}(S^m O) \wedge \varphi(S^n O, S^m O)}{\exists y.(\mathrm{Int}(y) \wedge \varphi(S^n O, y))}\ \exists\mathrm{I}.$$

Dunque applicando l'eliminazione a sinistra alla deduzione di $\mathrm{Int}(S^m O) \wedge \varphi(S^n O, S^m O)$ si ottiene una deduzione di $\mathrm{Int}(S^m O)$, la cui traduzione è equivalente a $t\overline{n}$. Per il lemma 6.22, tale deduzione ha una forma normale $\check{m}$, per cui $t\overline{n}$ si normalizza a $\overline{m}$. Inoltre poiché la formula $\varphi(S^n O, S^m O)$ è dimostrabile in HA2, essa è vera nel modello standard e quindi $f(n) = m$. Questo conclude la dimostrazione della proposizione. $\square$

6.3 Conseguenze

Iniziamo con il presentare una funzione che non è rappresentabile nel sistema F. Se m è un numero naturale che codifica il termine t, indichiamo con $N(m)$ come la codifica della forma normale di t. Inoltre se m codifica t e n codifica u, indichiamo con $A(m, n)$ la codifica dell'applicazione tu. Inoltre definiamo la funzione B che ad ogni naturale n associa la codifica del numerale $\overline{n}$ e definiamo C come la sua funzione inversa. Il valore di queste funzioni con argomenti che non sono opportune codifiche non è importante, e possiamo assumere che sia zero.

Ora consideriamo la funzione

$$D(n) = C[N(A(n, B(n)))] + 1.$$

Se tutte le funzioni definite sopra sono rappresentabili, anche D è rappresentabile. Ma ciò è assurdo. Infatti supponiamo che t sia un termine di tipo $\mathrm{Int} \to \mathrm{Int}$ che rappresenta D, e sia n la codifica di t.

Allora $A(n, B(n))$ è la codifica di $t\overline{n}$, e $N(A(n, B(n)))$ è la codifica della sua forma normale. Ma per come è definito t si deve avere che $t\overline{n} \leadsto \overline{D(n)}$ e quindi $N(A(n, B(n))) = B(D(n))$. Da questo si ottiene $C[N(A(n, B(n)))] = D(n)$, e per la definizione di D si ha che $D(n) = D(n) + 1$.

Si ottiene dunque che o D o una delle funzioni tra N, A, B o C non è rappresentabile. Per le codifiche più naturali dei termini si ha che A, B e C

sono rappresentabili, dunque il problema è su N. Eppure è possibile trovare una formula che esprima il grafico della funzione di normalizzazione. Questo implica che esiste una funzione per cui l'aritmetica di Peano (o di Heyting) del secondo ordine non è in grado di dimostrare la totalità, e dunque tale teoria è coerente. Abbiamo dunque dimostrato che la normalizzazione forte del sistema F implica la coerenza di PA2.

Il ragionamente appena fatto può essere formalizzato nell'aritmetica di Peano, e pertanto una dimostrazione della normalizzazione forte in tale sistema porterebbe a una dimostrazione di coerenza. Otteniamo come risultato, per il secondo teorema di Gödel che la normalizzazione forte del sistema F è indipendente dagli assiomi di Peano del secondo ordine.

7 Bibliografia

Riferimenti bibliografici

[1] J. Girard, P. Taylor, Y. Lafont. *Proofs and Types*, Cambridge University Press, Cambridge, 1990.

[2] H. P. Barendregt. *Lambda Calculus, Its Syntax and Semantics*, Studies in logic and the foundation of mathematics, Volume 103, Elsevier Science B.V., 1984.

[3] M. H. Sørensen, P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, Elsevier Science B.V., Amsterdam, 2006.

[4] S. Fortune, D. Leivant, M. O'Donnel. *The Expressiveness of Simple and Second-Order Type Structures*, Journal of the Association for Computer Machinery, Vol 30, No 1, January 1983, pp. 151-185.

[5] J. Krivine. *Lambda-calculus, types and models*, DEA, Université Paris 7, 2002,