

Rappresentabilità di Funzioni nel Lambda-calcolo Polimorfico

Alessio Marchetti

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 si rivelò inconsistente, come dimostrato da Kleene e Rosser nel 1936. Un sottoinsieme del 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 andrà a considerare 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 le tutte funzioni calcolabili, parziali e totali. 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 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 distinzione tra dati e programmi. 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 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 in 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 Il λ -calcolo non tipato

Il λ -calcolo è un sistema formale per descrivere delle computazioni. [...]

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.

Definizione 2.1 I termini del λ -calcolo semplice si definiscono induttivamente come:

- Le variabili x_1, x_2, \dots 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 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 $\lambda xyz.yyxz = \lambda x.(\lambda y.(\lambda z.yyxz))$. **Vettori di variabili.**

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 su una variabile lega tale variabile. Più formalmente:

Definizione 2.2 Un'occorrenza della variabile x in un termine t si dice legata se esiste un sottoterminale del tipo $\lambda x.t'$ che la contiene. Si dice legata 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 di 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). In particolare 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 2.3 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 2.4 Dati due termini u e v , si dice che u si converte a v e scriveremo $u \rightsquigarrow_C v$, se v è ottenuto da u sostituendo un sottotermino 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, \dots, u_n con $u_1 = u$ e $u_n = v$ tali che $u_1 \rightsquigarrow_C u_2 \rightsquigarrow_C \dots \rightsquigarrow_C u_n$. In tal caso scriveremo $u \rightsquigarrow v$.

Notiamo che la definizione dice essenzialmente che \rightsquigarrow è la chiusura transitiva di \rightsquigarrow_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 2.5

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

Esempio 2.6

$$\begin{aligned} \Omega &= (\lambda x.xx)(\lambda x.xx) \rightsquigarrow_C \\ &(\lambda x.xx)(\lambda x.xx) \rightsquigarrow_C \\ &\dots \end{aligned}$$

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

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

Esempio 2.8

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

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

$$\begin{aligned} & (\lambda x. \lambda y. y) \Omega z \rightsquigarrow_C \\ & (\lambda x. \lambda y. y) \Omega z \rightsquigarrow_C \\ & \dots \end{aligned}$$

È allora utile introdurre le seguenti nozioni:

Definizione 2.9 Dato un termine t si definisce $\nu(t)$ il massimo numero di conversioni necessarie per portare t in forma normale, ossia

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

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.

Convertendo sempre la redex 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 in particolare implica l'unicità del termine in forma normale a cui si converte un termine normalizzante.

Teorema 2.10 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 $v \rightsquigarrow w$. **Disegno del diamante.**

2.1 Rappresentabilità nel λ -calcolo non tipato

Tra i termini del λ -calcolo ne possiamo individuare alcuni per metterli in corrispondenza con i numeri naturali.

Definizione 2.11 Dato un numero naturale n definiamo il corrispondente numerale \bar{n} come il termine $\lambda f x. f^n x$, dove il simbolo $f^n x$ indica $f(f(\dots f(x) \dots))$.

In particolare \bar{n} è un termine che presa una funzione f , la compone con se stessa n volte. Osserviamo che inoltre i numerali sono termini in forma normale, dunque per la proprietà di Church-Rosser sono anche termini distinti anche modulo la β -equivalenza.

Definizione 2.12 Data una funzione (eventualmente parziale) $\phi: \mathbb{N} \rightarrow \mathbb{N}$, si dice che un termine t rappresenta ϕ se per ogni coppia di naturali m e n vale che $\phi(n) = m$ se e solo se $t\bar{n} \rightsquigarrow \bar{m}$ e $t\bar{n}$ non è normalizzante quando $\phi(n) = \perp$.

Facciamo ora alcuni esempi di funzioni rappresentabili:

Esempio 2.13 Il termine $A = \lambda p q f x. (pf)(qfx)$ rappresenta l'addizione. Per esempio:

$$\begin{aligned} A\bar{2}\ \bar{3} &\rightsquigarrow \lambda f x. f^2(f^3x) = \\ &\lambda f x. f^5x = \bar{5}. \end{aligned}$$

In modo simile esistono i termini $M = \lambda p q f x. q(pf)x$ e $E = \lambda p q f x. qpfx$ 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.

Teorema 2.14 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 inoltre 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, allora anche t' è normalizzante. Inoltre vale che $t' = xt'$, modulo la β -conversione, ma entrambi questi termini sono in forma normale, e dunque abbiamo l'assurdo.

3 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 **quello con Ω** oppure nel caso ancora peggiore seguente.

Esempio 3.1

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

Il problema alla base di questo comportamento è il fatto che non vi è distinzione 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 un tipo, e permettere l'applicazione di termini solo se i loro tipi sono compatibili.

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

Definizione 3.2 I tipi del λ -calcolo tipato semplice sono definiti induttivamente come:

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

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

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, \dots sono termini di tipo U .
- Se t e v sono termini di tipo rispettivamente $U \rightarrow V$ 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 $U \rightarrow V$.

Per indicare che un termine t è di tipo U scriveremo anche t^U .

Si vede dunque che il tipo $U \rightarrow V$ corrisponde al tipo 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, è possibile definire le nozioni di conversione, riduzione e forma normale. Si noti che entrambe le relazioni conservano il tipo dei termini.

3.1 Normalizzazione Forte per il Tipato Semplice

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

Teorema 3.3 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.

Definiamo come prima cosa la nozione di riducibilità.

Definizione 3.4 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 $T = V \rightarrow W$, 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 3.5 Diciamo che un termine t è neutrale se è nella forma xy .

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 sottoterminale 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 3.6

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

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 $U \rightarrow V$. 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 conversioni $t \rightsquigarrow_C t_1 \rightsquigarrow \dots \rightsquigarrow t_n$ possiamo associare la catena $tx \rightsquigarrow_C t_1x \rightsquigarrow \dots \rightsquigarrow t_nx$. Poichè $\nu(tx)$ è finito, t è fortemente normalizzante, e (CR1) è dimostrato.

Se consideriamo un termine t di tipo $U \rightarrow V$ 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 l'ipotesi induttiva 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). \square

A questo punto dimostriamo un utile lemma.

Lemma 3.7 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 $U \rightarrow V$. 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 .

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 $\nu(u') < \nu(u)$. Nuovamente $(\lambda x.v)u'$ è anch'esso riducibile per ipotesi induttiva.

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

Adesso si dimostra una versione più forte del teorema.

Proposizione 3.8 Sia t un termine le cui variabili libere compaiono tra $x_1, \dots, x_n = \underline{x}$, di tipo rispettivamente U_1, \dots, U_n . Siano $u_1, \dots, u_n = \underline{u}$ termini riducibili di tipo rispettivamente U_1, \dots, U_n . Allora il termine $t[\underline{u}/\underline{x}]$ è riducibile. Intendiamo per $t[\underline{u}/\underline{x}]$ la sostituzione $t[u_1/P]x_1 \cdots [u_n/P]x_n$.

Dimostrazione. Per induzione sulla complessità di t :

- Se $t = x_i$, allora la tesi è una tautologia.
- 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 $V \rightarrow W$, 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 **quale lemma?** si ottiene che $\lambda y.w[\underline{u}/\underline{x}]$ è riducibile.

\square

La dimostrazione del teorema di normalizzazione forte segue da quella della proposizione ponendo $\underline{u} = \underline{x}$.

3.2 Rappresentabilità per il Tipato Semplice

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

$$\bar{n} = \lambda f^{U \rightarrow U}. \lambda x^U. f^n x$$

di tipo $\text{Int} = (U \rightarrow U) \rightarrow U \rightarrow U$. 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

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

è la versione tipata per l'addizione e

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

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

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

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 non è possibile 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 3.9 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 . **L'altro è da fare?**

4 Il Sistema T di Gödel

Il grosso problema del calcolo tipato semplice è che i numerali non hanno un tipo canonico, e non è possibile definire funzioni per ricorsione primitiva. Per ovviare a questo problema introduciamo un nuovo calcolo, il sistema T di Gödel, in cui vengono artificialmente inseriti tipi per gli interi e alcuni termini che rappresentano delle funzioni basilari su di essi. Vedremo che con queste semplici aggiunte il calcolo guadagna una forte potenza espressiva.

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

Definizione 4.1 I tipi del sistema T sono definiti induttivamente come:

- U_1, U_2, \dots sono tipi, detti variabili di tipo.
- Se U e V sono tipi, allora anche $(U \rightarrow V)$ 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, \dots sono termini di tipo U .
- Se t e v sono termini di tipo rispettivamente $U \rightarrow V$ 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 $U \rightarrow V$.
- 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 $U, U \rightarrow (\text{Int} \rightarrow U)$ e Int , allora $Ruv t$ 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 la struttura 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 $Ruv(St) = v(Ruv)t$.

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

Definizione 4.2 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 $v' = w[w/x]$ dove x è una variabile e w e t sono termini tali che la sostituzione $w[w/x]$ 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 .

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. **Qui ho usato Church-Rosser.**

4.1 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 4.3 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 4.4 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 $T = V \rightarrow W$, 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 4.5 Un termine si dice neutrale quando non è in nessuna delle seguenti forme: $\langle u, v \rangle$, $\lambda x.v$, O o St .

Con queste definizioni estese possiamo riproporre senza modifiche le proprietà (CR1-3). Continua a valere la proposizione 3.6??, di cui dobbiamo estendere la dimostrazione.

Proposizione 4.6 Per tutti i riducibili 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 $t \rightsquigarrow t_1 \rightsquigarrow \dots \rightsquigarrow t_k$ possiamo associare la catena $\pi^1 t \rightsquigarrow \pi^1 t_1 \rightsquigarrow \dots \rightsquigarrow \pi^1 t_k$. 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 allora $\pi^2 t \rightsquigarrow \pi^2 t'$. Se t è riducibile, allora lo sono anche le sue proiezioni e quindi, per ipotesi induttiva, lo sono anche le riduzioni delle proiezioni. Poiché 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 Lemma??.

Lemma 4.7 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 riducibilità 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 la dimostrazione con $\pi^2\langle u, v \rangle$, otteniamo che la coppia $\langle u, v \rangle$ è riducibile. \square

Lemma 4.8 Se u , v e t sono termini riducibili, allora, se i tipi dei sotto-termini ne permettono la costruzione, anche $Ruv t$ è un termine riducibile.

Dimostrazione. Di nuovo ragioniamo per induzione per ottenere che tutte le conversioni del termine $Ruv t$ 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 $Ruv t$ 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'ipotesi 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 $l(w) < l(t)$. 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 $Ruv t$ è riducibile. \square

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

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

- Se $t = \pi^1 w$, allora il caso è identico al precedente.
- Se $t = \langle v, w \rangle$, allora per ipotesi induttiva $v[\bar{u}/\bar{x}]$ e $w[\bar{u}/\bar{x}]$ sono riducibili. La tesi qui si ha come conseguenza del **Lemma più su**.
- Se $t = O$, allora abbiamo già notato che è riducibile.
- Se $t = Sw$, abbiamo che se $w[\bar{u}/\bar{x}]$ è riducibile, allora $t[\bar{u}/\bar{x}]$ è fortemente normalizzante e quindi riducibile.
- Se $t = Rvwp$, allora per ipotesi anche le sostituzioni nei tre parametri sono riducibili, e quindi per il **Lemma??**, anche il termine $t[\bar{u}/\bar{x}]$ è riducibile.

□

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

4.2 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

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

Infatti vediamo che soddisfa la definizione ricorsiva della somma:

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

e

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

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

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

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 $h: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, allora il termine

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

[controlla che sia giusto] rappresenta la funzione f definita con $f(0) = n$ e $f(k+1) = h(f(k), k)$. **Dimostro?**

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 , $\text{Int} \rightarrow \text{Int} \rightarrow \text{Int}$. La definizione del sistema T permette di stanziare R su tipi più complessi, e ciò porta ad avere ulteriori funzioni rappresentabili. Per esempio la funzione di **Ackermann**.

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 4.9 Data una teoria T nel linguaggio dell'aritmetica, una funzione $f: \mathbb{N} \rightarrow \mathbb{N}$ si dice dimostrabilmente totale in T se esiste un predicato ϕ di complessità Σ_1^0 tale che vale $\phi(n, m)$ se e solo se $f(n) = m$ e $T \vdash \forall n \exists! m \phi(n, m)$.

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

Teorema 4.10 Le funzioni rappresentabili nel sistema T sono tutte e sole le funzioni dimostrabilmente totali nell'aritmetica di Peano.

Estendiamo i tipi del sistema T con un nuovo tipo $\mathbf{1}$ e con un termine \mathbf{i} di tipo $\mathbf{1}$. Per ogni tipo U di questo sistema esteso definiamo 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 $U = \mathbf{1} \rightarrow V$, allora $|U| = |V|$.
- Se $U = V \rightarrow \mathbf{1}$, allora $|U| = \mathbf{1}$.
- Se $U = V \times W$, con $n \in V$ e $n \in W$ pari a $\mathbf{1}$ allora $|U| = |V| \times |W|$.
- Se $U = V \rightarrow W$, con $n \in V$ e $n \in W$ pari a $\mathbf{1}$ allora $|U| = |V| \rightarrow |W|$.
- Se U è un tipo atomico, allora $|U| = U$.

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

$$\begin{aligned}\flat(\perp) &= \mathbf{1}; \\ \flat(s = t) &= \mathbf{1}; \\ &\dots \text{continuare}\end{aligned}$$

La definizione per i casi con i quantificatori è legata all'idea che una formula del tipo $\forall x \phi$ dove la quantificazione universale viene ristretta ai soli numeri naturali si scriverebbe come $\forall x (\text{int}(x) \rightarrow \phi)$. In modo equivalente la formula $\exists x \phi$ si riscriverebbe come $\exists x (\text{int}(x) \wedge \phi)$.

Definiamo ora la realizzabilità. Si dice che un termine chiuso uguale a **i** o **n** cui **i** non compare M di tipo $\flat\phi$ realizza una formula chiusa ϕ nelle seguenti situazioni:

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

Notiamo che non esiste alcun termine che realizzi \perp .

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

Dimostriamo ora il seguente importante teorema:

Teorema 4.11 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 **i**, cos' **i** come l'assioma $\forall x \forall y(x = y) \rightarrow (y = x)$. L'assioma di Peano $\forall x(Sx = O \rightarrow \perp)$ è sempre realizzato da **i** perché la formula atomica $Sx = O$ non è realizzabile.

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

$$\overline{tn - 1}(t\overline{tn - 2}(\dots t\overline{1}(t\overline{0}u)\dots))$$

realizza $\phi(\bar{n})$. Utilizziamo dunque il simbolo di ricorsione per definire il termine $v = \lambda p q n. R p q n$ che realizza l'assioma di induzione. **non è conclusa** \square

A questo punto possiamo concludere la dimostrazione del teorema principale.

Proposizione 4.12 Le funzioni dimostrabilmente totali nell'aritmetica di Peano sono rappresentabili nel sistema T.

Dimostrazione. Notiamo che poichè le formule Π_2^0 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_1^0 , sia essa ϕ , tale che $\text{HA} \vdash \forall n \exists m \phi(n, m)$ e $\phi(n, m)$ è vera quando $f(n) = m$. Per la proposizione **XXX**, esiste un termine M che realizza tale formula. Dunque per ogni naturale n , $M\bar{n}$ realizza la formula $\exists m \phi(\bar{n}, m)$. Si ha quindi che $\pi^1(M\bar{n})$ si riduce a un numerale \bar{m} e che $\pi^2(M\bar{n})$ realizza $\phi(\bar{n}, \bar{m})$. Diciamo che il termine $\lambda x. \pi^1(Mx)$ rappresenta f , infatti $\phi(\bar{n}, \bar{m})$ è vera, e quindi $f(n) = m$. \square

5 Il sistema F

In questa sezione introduciamo un'ulteriore variante del λ -calcolo tipato detta sistema F. Essa mantiene e anzi estende l'espressività del sistema T di Gödel e allo stesso tempo evita soluzioni ad hoc come l'inserimento 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, e permette, 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 5.1 I tipi del sistema F sono definiti induttivamente come:

- U_1, U_2, \dots sono tipi, detti variabili di tipo e sono anche detti tipi atomici.
- Se U e V sono tipi, allora anche $(U \rightarrow V)$ 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 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 un tipo nella forma $\Pi X.U$.

Definizione 5.2 I termini del sistema F si definiscono induttivamente come:

- Per ogni tipo U , le variabili x_1^U, x_2^U, \dots sono termini di tipo U .
- Se t e v sono termini di tipo rispettivamente $U \rightarrow V$ 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 $U \rightarrow V$.
- 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 termine 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 5.3 Un termine u si converte a v quando v è ottenuto sostituendo in u un sottoterminale 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.

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 corrispondono 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 $U \times V = \Pi X.(U \rightarrow V \rightarrow X) \rightarrow X$. Possiamo scrivere un termine che produce una coppia, ovvero

$$\text{CPL} = \lambda x^U y^V . \Lambda X . \lambda p^{U \rightarrow V \rightarrow X} . puv$$

e le conseguenti proiezioni

$$\pi^1 = \lambda x^{U \times V} . xU(\lambda y^U z^V . y)$$

$$\pi^2 = \lambda x^{U \times V} . xV(\lambda y^U z^V . z)$$

rispettivamente sul primo e secondo fattore.

Mostriamo per esempio la riduzione di

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

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 $U + V = \Pi X.(U \rightarrow X) \rightarrow (V \rightarrow X) \rightarrow X$ con il costruttore con la variabile di tipo U come

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

e l'equivalente sulla variabile di tipo V come

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

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

$$Duv t = tU(\lambda x^U . u)(\lambda x^V . v)$$

tale che valga la riduzione

$$\begin{aligned} Duv(\text{UN1}r) &= (\Lambda X . \lambda p^{U \rightarrow X} q^{V \rightarrow X} . pr)U(\lambda x^U . u)(\lambda x^V . v) \rightsquigarrow \\ (\lambda p^{U \rightarrow X} q^{V \rightarrow X} . pr)(\lambda x^U . u)(\lambda x^V . v) &\rightsquigarrow \\ (\lambda x^U . u)r &\rightsquigarrow u[r/x] \end{aligned}$$

e l'altra equivalente $Duv(\text{UN2}r) \rightsquigarrow v[r/x]$. 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 $U \rightarrow X \rightarrow X \rightarrow X$. Da qui costruiamo l'effettivo tipo dell'albero

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

I due costruttori sono rappresentati dai termini:

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

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

$$Ru_1(Ru_2(Ru_4)E)(Ru_3EE).$$

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

$$Ivft = tVv(\lambda n^U l^V r^v. f(f(fl)n)r)$$

spiegare un minimo

Notiamo tuttavia che l'introduzione del sistema F porta con sé 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'istanziamento 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.

5.1 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 **la definizione di riducibilità**, vorremmo provare a definirli 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 5.4 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 candidat.

Definizione 5.5 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 5.6 Se \mathcal{R} e \mathcal{S} sono insiemi di termini di tipo rispettivamente U e V , si definisce l'insieme $\mathcal{R} \rightarrow \mathcal{S}$ come l'insieme dei termini di tipo $U \rightarrow V$ per cui per ogni termine $u \in \mathcal{R}$ si ha che $tu \in \mathcal{S}$.

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

Lemma 5.7 Se \mathcal{R} e \mathcal{S} sono candidati per i tipi U e V , allora $\mathcal{R} \rightarrow \mathcal{S}$ è candidato di tipo $U \rightarrow V$.

Dimostrazione. Per mostrare (CR1) prendiamo $t \in \mathcal{R} \rightarrow \mathcal{S}$ 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 $t'\mathcal{R} \rightarrow \mathcal{S}$.

Infine consideriamo t neutrale di tipo $U \rightarrow V$ per cui per tutte le conversioni di uno step $t \rightsquigarrow t'$ si ha che $t' \in \mathcal{R} \rightarrow \mathcal{S}$. 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 $t' \in \mathcal{R} \rightarrow \mathcal{S}$, 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 5.8 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 $\text{RED}_T[\mathcal{R}/\underline{X}]$ di termini riducibili parametrici di tipo $T[\underline{U}/\underline{X}]$ nel modo seguente:

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

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 5.9 $\text{RED}_T[\mathcal{R}/\underline{X}]$ è un candidato di riducibilità di tipo $T[\underline{U}/\underline{X}]$.

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

Verifichiamo (CR1). Sia $t \in \text{RED}_T[\mathcal{R}/\underline{X}]$, V un tipo e \mathcal{S} un suo candidato. Allora $tV \in \text{RED}_W[\mathcal{R}/\underline{X}][\mathcal{S}/Y]$ 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 $t'V \in \text{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$ e quindi $t' \in \text{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

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 candidato \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. \square

Il prossimo lemma è utile per studiare il comportamento delle riducibilità parametrica rispetto alle istanziazioni. Una cosa da notare è che affinché abbia senso il predicato $\text{RED}_T[\underline{\mathcal{R}}/\underline{X}][\text{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y]$, occorre che $\text{RED}_V[\underline{\mathcal{R}}/\underline{X}]$ 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.

Lemma 5.10 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 $\text{RED}_{T[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \text{RED}_T[\underline{\mathcal{R}}/\underline{X}][\text{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y]$.

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

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

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

Se invece $T = Y$ si ha che

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

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

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

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

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

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

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

I seguenti due teoremi sono gli equivalenti per questa dimostrazione dei lemmi **quei lemmi**.

Lemma 5.11 Se per ogni tipo V e per ogni candidato di riducibilità \mathcal{S} per V vale che $w[V/Y] \in \text{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$, allora $\Lambda Y.w \in \text{RED}_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$.

Dimostrazione. Dimostriamo per induzione su $\nu(w)$ che tutte le conversioni in uno step di $(\Lambda Y.w)V$ sono in $\text{RED}_W[\underline{\mathcal{R}}/\underline{X}][\mathcal{S}/Y]$. Una conversione di tali conversioni possono 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). \square

Lemma 5.12 Se $t \in \text{RED}_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$, allora $tV \in \text{RED}_{W[V/Y]}[\underline{\mathcal{R}}/\underline{X}]$ per ogni tipo V .

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

Lemma 5.13 Se per ogni $u \in \text{RED}_U[\underline{\mathcal{R}}/\underline{X}]$ vale che $v[u/x] \in \text{RED}_V[\underline{\mathcal{R}}/\underline{X}]$, allora $\lambda x^U.v \in \text{RED}_{U \rightarrow V}[\underline{\mathcal{R}}/\underline{X}]$.

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 5.14 Un termine t di tipo T è riducibile se è in $\text{RED}_T[\underline{\mathcal{SN}}/\underline{X}]$ dove X_1, \dots, 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 **della proposizione sul tipato semplice**.

Proposizione 5.15 Sia t un termine di tipo T le cui variabili libere sono x_1, \dots, x_n di tipo rispettivamente U_1, \dots, U_n . Supponiamo che le variabili libere dei tipi T e di tutti gli U_i siano X_1, \dots, X_m . Siano $\mathcal{R}_1, \dots, \mathcal{R}_m$ candidati di riducibilità per dei tipi V_1, \dots, V_m e siano inoltre u_1, \dots, u_n termini di tipo $U_1[\underline{V}/\underline{X}], \dots, U_n[\underline{V}/\underline{X}]$ presi nei rispettivi $\text{RED}_{U_i}[\underline{\mathcal{R}}/\underline{X}]$. Allora $t[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \text{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

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 $W \rightarrow T$ e v di tipo W . Per ipotesi induttiva vale che $w[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \text{RED}_{W \rightarrow T}[\underline{\mathcal{R}}/\underline{X}]$ e che $v[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \text{RED}_W[\underline{\mathcal{R}}/\underline{X}]$. In questo caso la tesi segue dalla definizione di $\text{RED}_{W \rightarrow T}$.
- (iii) $t = wS$. Questo caso è una diretta conseguenza del lemma **XXX** sull'istanziamento.
- (iv) $t = \Lambda Z.Y$. Questo discende dal lemma **XXX** sulla generalizzazione.
- (v) $t = \lambda y^P.w$. Questo caso si fa con il lemma **XXX** sui tipi freccia.

□

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

Proposizione 5.16 Tutti i termini del sistema F sono riducibili.

Dimostrazione. Basta usare la proposizione precedente e prendere $\mathcal{R}_i = \mathcal{SN}_i$ e $u_i = x_i$. □

Teorema 5.17 Tutti i termini del sistema F sono fortemente normalizzanti.

6 Aritmetiche di Peano e di Heyting

Definizione 6.1 Il linguaggio per la logica del secondo ordine è lo stesso di quello del primo ordine con l'aggiunta per ogni naturale n di numerabili simboli X^n , che chiameremo variabili di relazione.

Le formule atomiche sono \perp e espressioni della forma $X^n(t_1, \dots, t_n)$, dove i t_i sono termini del linguaggio.

Le formule sono definite induttivamente come:

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

Si definisce inoltre la formula $\neg\phi$ come $\phi \rightarrow \perp$.

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

Definizione 6.2

- Le variabili libere di $X(t_1, \dots, t_n)$ con X variabile di relazione n -aria, sono l'unione di tutte le variabili libere che compaiono nei termini t_i per ogni i e X .
- Le variabili libere di $r(t_1, \dots, t_n)$ con r simbolo di relazione n -aria sono l'unione di tutte le variabili libere che compaiono nei termini t_i per ogni i .
- Le variabili libere di $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$ sono l'unione delle variabili libere di ϕ e ψ .
- Le variabili libere di $\forall x\phi$ e $\exists x\phi$ con x variabile, sono le variabili libere di ϕ meno x .
- Le variabili libere di $\forall X\phi$ e $\exists X\phi$ 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 6.3 Sia ϕ una formula e x_1, \dots, x_n delle variabili individuali, allora l'espressione $\lambda x_1, \dots, x_n. \phi$ è una specie di arietà n . Si noti che le variabili x_i possono apparire o non apparire in ϕ .

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

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

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

- $\perp[\lambda\underline{x}.\phi/X] = \perp$.
- $r(t_1, \dots, t_n)[\lambda\underline{x}.\phi/X] = r(t_1, \dots, t_n)$ quando r è una relazione oppure una variabile di relazione diversa da X .
- $(X(\underline{t})[\lambda\underline{x}.\phi/X] = \phi[\underline{t}/\underline{x}]$.
- $(\eta \rightarrow \psi)[\lambda\underline{x}.\phi/X] = \eta[\lambda\underline{x}.\phi/X] \rightarrow \psi[\lambda\underline{x}.\phi/X]$ e equivalentemente per $\eta \vee \psi$ e $\eta \wedge \psi$.
- $(\forall x\eta)[\lambda\underline{x}.\phi/X] = \forall x\eta[\lambda\underline{x}.\phi/X]$ per tutte le variabili individuali x che non appaiono libere in $\lambda\underline{x}.\phi$. Equivalentemente per $\exists x\eta$.
- $(\forall Y\eta)[\lambda\underline{x}.\phi/X] = \forall Y\eta[\lambda\underline{x}.\phi/X]$ con Y variabile di relazione diversa da X e Y che non appare libera in $\lambda\underline{x}.\phi$.

A questo punto presentiamo le regole della deduzione naturale per la logica del secondo ordine: **regole per la deduzione, p.308**

$$\frac{\begin{array}{c} A \\ \vdots \\ A' \end{array} \quad B}{C} \text{ inf}$$

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:

$$\exists Y \forall x (\phi(x) \leftrightarrow x \in Y).$$

per ogni formula ϕ .

Inoltre molti dei connettivi presentati sono ridondanti: infatti è possibile definirli tutti in termini dei soli \rightarrow e \forall (su variabili individuali e di relazione). In particolare: **È corretto usare $=$?**

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

- $\exists X\phi = \forall R(\forall X(\phi \rightarrow R) \rightarrow R)$.

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 6.4 Data una formula ϕ definiamo induttivamente la sua traduzione di Gödel $k(\phi)$ come:

- $\neg\neg\phi$ se ϕ è atomica.
- $\neg\neg(k(\eta) \rightarrow k(\psi))$ se $\phi = \eta \rightarrow \psi$, e in modo di equivalentemente si definisce per gli altri connettivi binari.
- ...

Introduciamo adesso le aritmetiche del secondo ordine. Utilizziamo un linguaggio che ha come unica costante 0, il simbolo di funzione successore S e una relazione binaria di uguaglianza $=$.

A questo punto possiamo dare il seguente risultato:

Proposizione 6.5 La proposizione ϕ è un teorema della logica classica se e solo se $k(\phi)$ è un teorema della logica intuizionista. Dimostrazione?

Consideriamo i seguenti assiomi per l'uguaglianza:

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

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) $\forall ab(Sa = Sb \rightarrow a = b)$;
- (P2) $\forall a(Sa = 0 \rightarrow \perp)$;
- (P3) $\forall a \text{ Int}(a)$,

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

Definizione 6.6 Gli assiomi (U1-4) e (P1-3) 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.

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

Consideriamo T una teoria nel linguaggio dell'aritmetica. Sia inoltre \mathcal{U} la funzione universale, ovvero una formula primitiva ricorsiva tale che $\mathcal{U}(e, n, m)$ è vera se e solo se il programma con codifica e eseguito con input n ha output m .

Diciamo che una funzione è dimostrabilmente totale in T se esiste un programma con codifica e tale che

$$T \vdash \forall n \exists! m \mathcal{U}(e, n, m).$$

Aggiungere precisazioni per funzioni con più di una variabile?

Notiamo che la formula da dimostrare ha complessità Π_2^0 .

Vale il seguente teorema:

Teorema 6.7 Le funzioni dimostrabilmente totali in PA2 sono esattamente le funzioni dimostrabilmente totali in HA2. **Dimostrazione?**

7 Rappresentabilità in F

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

$$\text{Int} = \Pi X. X \rightarrow (X \rightarrow X) \rightarrow X$$

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

$$O = \Lambda X. \lambda x. \lambda f. x$$

$$S = \lambda n. \Lambda X. \lambda x. \lambda f. f(n X x f).$$

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

Lemma 7.1 I numerali sono tutti e soli i termini in forma normale di tipo Int.

In realtà la costruzione fatta per i numeri naturali a partire dai costruttori zero e successore può essere generalizzata a qualunque tipo di dato algebrico.

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

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 possibile esprimere la riducibilità in modo uniforme su tutti i tipi del sistema F, ma soltanto caso per caso per ciascun tipo.

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

Proposizione 7.3 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 $\llbracket \phi \rrbracket$ del sistema F. L'associazione viene fatta con le seguenti regole:

- $\llbracket a = b \rrbracket = S$, dove S è un tipo fissato per cui esiste almeno un termine di tale tipo, per esempio $S = \Pi X. X \rightarrow X$.
- $\llbracket \phi \rightarrow \psi \rrbracket = \llbracket \phi \rrbracket \rightarrow \llbracket \psi \rrbracket$.
- $\llbracket \forall x. \phi \rrbracket = \llbracket \phi \rrbracket$, per le quantificazioni del primo ordine.
- $\llbracket \forall X. \phi \rrbracket = \llbracket \phi \rrbracket$ per le quantificazioni del secondo ordine.

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

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:

$$\llbracket \phi \wedge \psi \rrbracket = \llbracket \forall X. ((\phi \rightarrow \psi \rightarrow X) \rightarrow X) \rrbracket = \Pi X. (\llbracket \phi \rrbracket \rightarrow \llbracket \psi \rrbracket \rightarrow X) \rightarrow X.$$

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

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

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

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

$$\frac{\begin{array}{c} \eta \\ \vdots \\ \phi \end{array} \epsilon}{\eta \rightarrow \phi} \delta$$

- 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 $\psi \rightarrow \phi$, dimostrate rispettivamente con ϵ e con γ , allora $\llbracket \delta \rrbracket = \llbracket \gamma \rrbracket \llbracket \epsilon \rrbracket$. La situazione è illustrata da:

$$\frac{\begin{array}{c} \vdots \gamma \\ \psi \rightarrow \phi \end{array} \quad \begin{array}{c} \vdots \epsilon \\ \psi \end{array}}{\phi} \delta$$

- 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 $\forall x \phi$ a partire da ϕ , formula dimostrata con ϵ , allora $\llbracket \delta \rrbracket = \llbracket \epsilon \rrbracket$. Similmente, se l'ultima inferenza di δ dimostra $\phi[a/x]$ a partire da $\forall x \phi$, formula dimostrata con ϵ , allora si ha ancora $\llbracket \delta \rrbracket = \llbracket \epsilon \rrbracket$. Le situazioni illustrate sono le seguenti:

$$\begin{array}{cc} \begin{array}{c} \vdots \epsilon \\ \phi \\ \hline \forall x \phi \end{array} \delta & \begin{array}{c} \vdots \epsilon \\ \forall x \phi \\ \hline \phi[a/x] \end{array} \delta \end{array}$$

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

$$\frac{\begin{array}{c} \vdots \\ \epsilon \end{array} \quad \phi}{\forall X\phi} \delta$$

- Infine se δ termina con l'eliminazione di una quantificazione del secondo ordine in cui partendo da una formula $\forall X.\phi$, dimostrata con ϵ si deduce $\phi[\lambda x.\psi/X]$, allora $\llbracket \delta \rrbracket = \llbracket \epsilon \rrbracket \llbracket \psi \rrbracket$ (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:

$$\frac{\begin{array}{c} \vdots \\ \epsilon \end{array} \quad \forall x\phi}{\phi[\lambda x.\psi/X]} \delta$$

Osserviamo che le regole appena presentate rispettano i tipi di tutte le formule.

Lemma 7.4 Esiste un'unica deduzione normale di $\text{Int}(S^n O)$, ovvero \check{n} .

Dimostrazione. Consideriamo la formula $\phi(x, y)$ 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

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

ovvero

$$\forall x(\text{Int}(x) \rightarrow \exists y.(\text{Int}(y) \wedge \phi(x, y))).$$

Chiamiamo δ tale dimostrazione. Ad essa associamo un termine $\llbracket \delta \rrbracket$ di tipo $\text{Int} \rightarrow (\text{Int} \times \llbracket \phi \rrbracket)$ e un termine $t = \lambda x^{\text{Int}}.\pi^1(\llbracket \delta \rrbracket x)$ che ne contiene il significato algoritmico.

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

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

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