Il Sistema F

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 Kleen 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 svologono 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 funzioni calcolabili. Poichè l'insieme delle funzioni calcolabili totali non è ricorsivamente enumerabile esistono dei termini la cui computazione termina (diremo che non sono normalizzanti).

Il λ-calcolo tipato è una variante del 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 del-

le 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 sistem 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 questo documento 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 si definiscono induttivamente ...

L'idea intuitiva di questi termini 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 l'output di un programma t quando viene eseguito con input v. Un fatto interessante da notare è che i termini sono al contempo programmi e dati.

Definizione 2.2 Dato un termine t, i sottotermini sono tutti i termini che appaiono nella costruzione induttiva di t.

In un termine distinguiamo le occorrenze di ciascuna variabile tra occorrenze libere e legate. Un'occorrenza di una variabile x si dice legata se appare in un sottotermine della forma $\lambda x.t$. In modo più preciso diamo anche la definizione induttiva:

Definizione 2.3 Dato un termine t, l'insieme delle variabili libere di t $\mathrm{VL}(t)$ ...

Una variabile non libera si dice legata.

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).

Definizione 2.4 L'alpha equivalenza è ...

Indichiamo i termini modulo l'equivalenza con Λ.

In tutto il documento lavoreremo solo con rappresentanti delle classi di Λ, identificando i loro rappresentanti con esse.

Con qualche accortezza per evitare la cattura delle variabili si può definire la sostituzione.

Definizione 2.5 Se u e v sono termini e x è una variabile tale che ..., allora $u[v/x]$ ...

A questo punto possiamo introdurre una relazione fondamentale sui termini del calcolo, ovvero la conversione.

Definizione 2.6 Dati due termini u e v, si dice che u si converte a v e scriverermo $u \rightsquigarrow_C v$, se u e nella forma ...

Si dice che u si converte a v se esistono n termini $u_1, \ldots, u_n$ con $u_1 = u$ e $u_n = v$ tali che $u_1 \rightsquigarrow_C u_2 \rightsquigarrow_C \ldots \rightsquigarrow_C u_n$. 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.

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 2.7

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

Esempio 2.8

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

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 2.9 Un termine si dice in forma normale se non può essere 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 2.10

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

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

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

È allora utile introdurre le seguenti nozioni:

Definizione 2.11 Dato un termine t si definisce $\nu(t)$ il massimo numero di conversioni necessarie per portare t in forma normale, ossia

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

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 2.12 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.

Nella dimostrazione del teorema serve mostrare che sono rappresentabili alcune funzioni base. In ogni caso è utile mettere un esempio di: somma, prodotto, esponenziazione, predecessore?, uguaglianza?

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 2.13 Dato un numero naturale n definiamo il corrispondente numerale $\overline{n}$ come il termine ...

In particolare $\overline{n}$ 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 Church-Rosser sono anche termini distinti anche modulo la β-equivalenza.

Definizione 2.14 Data una funzione $\phi\colon \mathbb{N} \to \mathbb{N}$, si dice che un termine t rappresenta ϕ se per ogni coppia di naturali m e n tali che $\phi(n) = m$ si ha che $t\overline{n} \rightsquigarrow \overline{m}$ e per ogni n tale che $f(n) = \perp$ si ha che $t\overline{n}$ non ha forma normale.

Teorema 2.15 Le funzioni rappresentabili nel λ-calcolo non tipato sono esattamente le funzioni calcolabili.

3 Il λ-calcolo tipato semplice

Nel λ-calcolo non tipato, esistono termini per cui la normalizzazione non corrisponde all'idea intuitiva di "semplificazione" del termine, come nell'esempio quello con Ω oppure nel caso ancora peggiore seguente.

Esempio 3.1 il termine Ω_3.

Il problema alla base di questo comportamento è il fatto che non vi è distizione tra dati e porgrammi e inoltre 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 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.

Definizione 3.2 I tipi del tipato semplice sono...
I termini del tipato semplice sono...
Le conversioni nel tipato semplice sono...

Si vede dunque che il tipo $U \to V$ 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".

Avendo dato la nozione di conversione anche in questo calcolo, si possono definire in modo identico a quanto già fatto per il calcolo non tipato

le nozioni di riduzione, forma normale, termini normalizzanti e fortemente normalizzanti. Faremo la stessa cosa anche con le successive varianti del λ-calcolo.

3.1 Normalizzazione Forte per il Tipato Semplice

In questa sezione si dimostra che tutti i termini nel tipato semplice sono fortemente normalizzanti.

Pulire e incollare la dimostrazione in strong_norm.pdf

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 in numeri naturali non è unica.

Come importante conseguenza della proprietà di normalizzazione forte si ha una conseguente riduzione della classe delle funzioni rappresentabili, che devone essere per forza totali. Inoltre non tutte le funzioni totali sono rappresentabili, infatti...

Per esempio, i termini di somma e prodotto sono tipabili, l'esponenziale non lo è

Infatti vale il seguente risulatato che non dimostro:

Teorema 3.3 Le funzioni rappresentabili sono ...

4 Il Sistema T di Gödel

Il grosso problema del calcolo tipato semplice è che i naturali non hanno un tipo unico, 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, i booleani e alcuni termini che rappresentano delle funzioni basilari su di essi.

Definizione 4.1 I termini del Sistema T...

Inoltre si danno anche le regole per le conversioni.

Definizione 4.2 La riduzione per il Sistema T...

Si ha dunque che i termini O e S rappresentano rispettivamente lo zero e il successore, e T e F i valori booleani vero e falso. D rappresenta l'operatore *if/then/else*, mentre R è l'operatore di ricorsione primitiva.

4.1 Normalizzazione Forte per il Sistema T

Teorema 4.3 Tutti i termini del sistema T sono fortemente normalizzanti.

Definiamo come prima cosa la nozione di riducibilità.

Definizione 4.4 Sia U un tipo del sistema T, 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 $U = V \times W$, t è riducibile se e solo se $\pi^1 t$ e $\pi^2 t$ lo sono.
- 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 tipi primitivi e le costanti O, T, F,

Definizione 4.5 Diciamo che un termine t è neutrale se è in una delle seguenti forme: $x, \pi^1 x, \pi^2 x, xy, Rxyz$ oppure $Dxyz$.

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.

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 termne data quella delle sue conversioni.

A questo punto si dimostrano alcuni lemmi.

Lemma 4.7 Se u e v sono riducibili, allora anche $\langle u, v \rangle$ è riducibile.

Lemma 4.8 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.

Lemma 4.9 Se u, v, t sono termini riducibili, allora anche $Duvt$ è riducibile.

Lemma 4.10 Se u, v, t sono termini riducibili, allora anche $Ruvt$ è riducibile.

Adesso si dimostra una versione più forte del teorema.

Proposizione 4.11 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.

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

5 Il sistema F

5.1 Normalizzazione per il Sistema F

6 Aritmetiche di Peano e di Heyting

Introduzione sulla deduzione al primo ordine, linguaggio e assiomi di Heyting e peano al primo ordine.

Distinzione tra secondo ordine semplice e secondo ordine

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, i predicati per n variabili.

Le formule sono costruite con le stesse regole del primo ordine con l'aggiunta che ci sono le formule atomiche e i $\forall^2$...

Definizione 6.2 Le regole per la deduzione naturale sono quelle solite più le quattro per i due nuovi quantificatori

Si noti che nella logica del secondo ordine si possono definire $\bot, \wedge, \vee, \exists$ e $\exists^2$ ($\bot$ solo nel caso intuizionista).

Fare alcuni esempi su come si riscrivono i connettivi.

Inoltre nel secondo ordine, è possibile derivare lo schema di comprensione, ovvero per ogni formula A(x), vale che

$$\exists Y \ \forall x \ (A(x) \iff x \in Y).$$

Definizione 6.3 PA e HA, primo ordine

Per il secondo ordine non è necessario aggiungere nuovi assiomi alla teoria, infatti è possibile ottenere l'aritmetica di Peano e di Heyting

7 Rappresentabilità in T

7.1 Codifica dei termini

Lo scopo di questa sezione è quello di interpretare il sistema T all'interno delle teorie dell'aritmetica. Per fare ciò serve come prima cosa trovare una codifica con i numeri di Gödel per i termini del calcolo.

Definizione 7.1 Dato un tipo U, gli si associa un numero naturale...
Idem con i termini

Abbiamo a questo punto delle formule che esprimono la conversione, la riduzione, la normalizzazione di un termine, la forte normalizzazione di un termine.

7.2 Le funzioni rappresentabili sono dimostrabilmente totali in PA

Notiamo che le funzioni rappresentabili nel sistema T sono calcolabili. Infatti, dati una funzione f rappresentata da un termine t e un numero naturale n, per la tesi di Church, disponiamo di un algoritmo di calcolo. Si scrive il termine $t\overline{n}$, lo si riduce in forma normale t'. Essa sarà il numerale corrispondente al numero naturale $m = f(n)$.

Lemma 7.2 I termini normali di tipo int sono tutti e soli i numerali. Capire dove mettere questo lemma.

Sia T una teoria nel linguaggio dell'aritmetica. Sia inoltre $\mathcal{U}$ la funzione universale, ovvero una formula Δ_0 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

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

Si noti che la formula da dimostrare ha complessità Π^0_2.

Vogliamo dunque dimostrare il seguente teorema:

Teorema 7.3 Tutte le funzioni rappresentabili dal sistema T sono dimostrabilmente totali in PA.

L'idea di dimostrazione consiste nel ripercorrere la dimostrazione della normalizzazione forte su un singolo termine. Per fare ciò serve esprimere con un predicato la riducibilità di un termine, e poi ragionare per induzione sulle varie riducibilità.

7.3 Le funzioni dimostrabilmente totali in PA sono rappresentabili

8 Rappresentabilità in F

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

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

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

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

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

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

Lemma 8.1 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 8.2 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 generale, ma solo per specifici termini.