Estratto da

C. Bernardi e P. Pagli (a cura di), *Atti degli incontri di logica matematica* Volume 2, Siena 5-8 gennaio 1983, 6-9 aprile 1983, 9-12 gennaio 1984, 25-28 aprile 1984.

Disponibile in rete su http://www.ailalogica.it

UNA GENERALIZZAZIONE DEI FUNZIONALI RICORSIVI DI GÖDEL

Alessandro Berarducci (*)

Introduzione

I funzionali ricorsivi di Gödel di tipo finito basati sul tipo dei numeri naturali N (cfr.(7) e (8)) vengono generalizzati considerando come strutture base, oltre a N , qualsiasi algebra di termini eterogenea A senza generatori (cfr. (1) per una definizione di algebra eterogenea).

Si mostra in seguito come rappresentare tali funzionali nel λ-calcolo con tipi del primo ordine o, alternativamente, con tipi del secondo ordine.

Qusta ultima rappresentazione e' sostazialmente il risultato principale del lavoro (2) scritto in collaborazione con il Prof. C.Böhm.

La presente esposizione e' basata su una comunicazione da me tenuta nell' ambito del convegno su "Ricorsivita' e Sistemi Formali" (Univ. di Siena , Gennaio 1984), ma rispetto a questa ha un carattere piu' algebrico che tiene conto di risultati successivi ottenuti in collaborazione con il Prof. C.Böhm (4).

(*)Ricerca parzialmente svolta nell'ambito del contratto 82007760.97 del CNR,progetto finalizzato per l'informatica

Tipi

I tipi del primo ordine basati su N sono definiti da:

(i) N e' un tipo base

(ii) Un tipo base e' un tipo

(iii) Se α, β sono tipi, $\alpha \to \beta$ e' un tipo (rappresentante la classe delle funzioni da α a β)

Funzionali ricorsivi di Gödel

I funzionali ricorsivi di Gödel sono la piu' piccola classe di funzioni chiusa per composizione (rispettando i tipi) e contenente la funzione successore $s:N\to N$, lo zero $0:N$, e per ogni tipo α un funzionale Z^N_α di tipo $N\to(\alpha\to\alpha)\to\alpha$ che soddisfa le equazioni $Z^N_\alpha 0fu = u$, $Z^N_\alpha(sn)fu = f(Z^N_\alpha nfu)$

(cfr. (8) per una definizione equivalente basata sullo operatore di ricorsione R).

Ovviamente la funzione $g:N\to\alpha$ definita da $gn = Z^N_\alpha nfu$, soddisfa lo schema iterativo $g(sn) = f(gn)$, $g0 = u$.

I funzionali di Gödel sono quindi chiusi per definizioni iterative.

Si puo' anche dimostrare che i funzionali di Gödel sono chiusi per definizioni ricorsive, cioe' per definizioni della forma $g(sn) = fn(gn)$ dove $u:\alpha$,$f:N\to\alpha\to\alpha$,$g:N\to\alpha$

Se $\alpha = N$ otteniamo l'usuale shema per la ricursione primitiva sui numeri naturali, di conseguenza le funzioni ricorsive primitive sono funzionali di Gödel.

E' noto d'altra parte che, anche limitandosi ai funzionali di Gödel di tipo $N\to N$ si ottiene una classe molto piu' ampia della classe delle funzioni primitive unarie.

Generalizzazione

Vogliamo ora considerare funzionali ricorsivi basati su altre strutture oltre a quella dei numeri naturali.

Le strutture che considereremo saranno algebre eterogenee, cioe' sistemi $\underline{A} = \langle \{S_1,\dots,S_n\}, \{g_1,\dots,g_k\}\rangle$ dove $S_1,\dots,S_n$ sono insiemi sostegno, ed ogni $g\in\{g_1,\dots,g_k\}$ e' una funzione finitaria $g:R_1\times\dots\times R_p\to R_{p+1}$ con $p\geq 0$ e $\{R_1,\dots,R_{p+1}\}\subseteq$ $\subseteq\{S_1,\dots,S_n\}$ (funzioni base).

Nel seguito identificheremo il tipo $R_1\times\dots\times R_p\to R_{p+1}$ con $R_1\to\dots\to R_p\to R_{p+1}$.

Assumiamo inoltre che tali algebre siano algebre di termini senza generatori, algebre cioe' in cui ogni elemento e' una parola sull' alfabeto delle operazioni base (incluse le costanti).

Un esempio e' l'algebra $\underline{N} = \langle\{N\}, \{s:N\to N, 0:N\}\rangle$.

Data un' algebra $\underline{A} = \langle \{S_1,\dots,S_n\}, \{g_1,\dots,g_k\}\rangle$ definiamo itipi relativi ad $\underline{A}$ come nel paragrafo precedente considerando pero' $S_1,\dots,S_n$ anziche' N come tipi base.

Per ogni n-upla di tipi $\bar{\alpha} = \alpha_1,\dots,\alpha_m$ e per ogni $S \in \{S_1,\dots,S_n\}$ definiamo l'operatore $Z^S_{\bar{\alpha}}$ nel modo seguente:

per ogni $g_i \in \{g_1,\dots,g_k\}$, di tipo $R_1 \to \dots \to R_p \to R_{p+1}$ per ogni $x_1,\dots,x_p$ nel dominio di g_i, e per ogni $h_1,\dots,h_k$ tali che, per $q = 1,\dots,k$, il tipo di h_q si ottiene dal tipo di g_q sostituendo $\alpha_1,\dots,\alpha_n$ al posto di $S_1,\dots,S_n$ rispettivamente, vale la seguente equazione:

$$(eq_i):\quad Z^R_{\bar{\alpha}}\ (g_i x_1\dots x_p)h_1\dots h_k =$$

$$= h_i(Z^R_{\bar{\alpha}}\ x_1 h_1\dots h_k)\dots(Z^R_{\bar{\alpha}}\ x_p h_1\dots h_k)$$

(la scelta dei tipi di $h_1,\dots,h_k$ e' la piu' generale che renda le equazioni $eq_1,\dots,eq_k$ sensate).

E' chiaro che le k quazioni $eq_1,\dots,eq_k$ definiscono n operatori $Z^S_{\bar{\alpha}},\dots,Z^S_{\bar{\alpha}}$ con una induzione incrociata.

Nel caso dell'algebra $\underline{N}$ abbiamo di nuovo le due equazioni $Z^N_{\alpha}(sn)fu = f(Z^N_{\alpha}nfu)$, $Z^N_{\alpha}0fu = u$ (tante quante sono le operazioni base), e un solo operatore Z^N_{α} (tanti quanti sono gli insiemi sostegno della

algebra).

La definizione dei funzionali di Gödel si generalizza ora immediatamente, mutatis mutandis, a qualsiasi algebra $\underline{A}$ come sopra.

Una conseguenza immediata delle equazioni $eq_1,\dots,eq_k$ sono le n equazioni a punto fisso

(fix_j): $Z^{S_j}_{\bar{\alpha}} x g_1\dots g_k = x$ dove $\bar{\alpha} = S_1,\dots,S_n$, $S_j \in \{S_1,\dots,S_n\}$ ed x ha tipo S_j.

In particolare $Z^N_N n s 0 = n$ (per $n \in N$).

In effetti e' stata proprio l'idea di generalizzare questa ultima equazione che ci ha condotti alle equazioni $eq_1,\dots,eq_k$. Questa e' anche la ragione per cui e' stato preferito un trattamento basato sulla iterazione anziche' sulla ricursione.

<u>Applicazioni al λ-calcolo</u>

Vogliamo ora sviluppare un sistema di notazione o "linguaggio di programmazione" per i funzionali ricorsivi generalizzati. Considereremo due possibilita' alternative. La prima possibilita' e' di lavorare nel λ-calcolo con tipi del primo ordine esteso con l'aggiunta di nuove costanti che rappresentino gli iteratori $Z^S_{\bar{\alpha}}$ e le costanti base $g_1,\dots,g_k$ (nel senso che postuliamo nel λ-calcolo delle equazioni corrispondenti alle equazioni $eq_1,\dots,eq_k$).

In questo modo otteniamo un linguaggio tipato in cui si rappresentano tutti e soli i funzionali di Gödel generalizzati

L'estensione e' necessaria nel senso che non e' possibile rappresentare i funzionali di Gödel nel λ-calcolo con tipi del primo ordine non esteso (segue facilmente dai risultati sul potere espressivo del λ-calcolo del primo ordine esposti in (6)).

Intuitivamente questo dipende dal fatto che i tipi del primo ordine impediscono le definizioni a punto fisso e in particolare ricorsive.

La seconda possibilita', relativamente sorprendente date le considerazioni precedenti, e' di lavorare nel λ-calcolo con tipi del secondo ordine senza l'aggiunta di alcuna nuova costante (cfr.(4) per alcune applicazioni all'informatica teorica e per un confronto con altri lavori sulla rappresentazione di funzioni nel λ-calcolo basati sulla teoria della dimostrazione).

Per semplicita' consideriamo dapprima una versione senza tipi della nostra rappresentazione.

E' noto che nel λ-calcolo e' possibile rappresentare $s:N \to N$ e $0:N$ con termini $\underline{s},\underline{0}$ definiti da

$\underline{s}xfu = f(xfu)$, $\underline{0}xfu = u$

dove x, f, u sono variabili del λ-calcolo (cfr.(5)).

Il numero n e' quindi rappresentatodal termine

$\underline{n} = \underline{s}(\dots(\underline{s}\ \underline{0})$ (n volte).

Notiamo che le definizioni di $\underline{s}$, $\underline{0}$ ricordano le equazioni $Z^N_\alpha(sn)fu = f(Z^N_\alpha nfu)$, $Z^N_\alpha 0fu = u$ con la differenza che gli operatori iterativi Z^N_α sono stati omessi (informalmente e' come se avessimo posto $\underline{n} = Z^N_\alpha n$).

Questa considerazione ci suggerisce come rappresentare nel λ-calcolo qualsiasi algebra $\underline{A}$ come sopra.

Intuitivamente basta considerare le equazioni $eq_1,\dots,eq_k$ e sopprimere gli Z^S_α ; piu' esattamente dalla equazione eq_i otteniamo la seguente rappresentazione $\underline{g}_i$ del generatore g_i

$$(eq_i^*):\ \underline{g}_i x_1\dots x_p h_1\dots h_k = h_i(x_1h_1\dots h_k)\dots(x_ph_1\dots h_k)$$

dove $x_1,\dots,x_p,h_1,\dots,h_k$ sono variabili del λ-calcolo.

Poiche' abbiamo assunto che ogni elemento di $\underline{A}$ e' una parola sull'alfabeto delle operazioni base, la rappresentazione di $g_1,\dots,g_k$ induce in modo ovvio una rappresentazione di tutti gli elementi dell'algebra.

L'utilita' di questa rappresentazione delle algebre e' nel fatto che essa fornisce un metodo immediato per rappresentare i funzionali di Gödel generalizzati su tali algebre.

In particolare data $\underline{A} = \langle \{S_1,\dots,S_n\}, \{g_1,\dots,g_k\}\rangle$, il funzionale H di dominio S_j definito iterativamente da $Hx = Z_{\alpha}^{S_j} xh_1\dots h_k$ e' rappresentato dal termine $\underline{H} = \lambda x.(x\underline{h}_1\dots\underline{h}_k)$ dove $\underline{h}_1,\dots,\underline{h}_k$ rappresentano $h_1,\dots,h_k$ rispettivamente.

Intuitivamente per rappresentare unfunzionale $\underline{H}$ definito iterativamente, basta applicare il generico argomento x di H alle funzioni $h_1,\dots,h_k$ in termini delle quali H e' definito.

Tipi del secondo ordine

Accenniamo ora brevemente a come sia possibile introdurre i tipi del secondo ordine (cfr.(6)).

A tale scopo ci basiamo ancora sulle equazioni eq_i, ma modifichiamo le $eq_i{}^*$ in

$$(eq_i{}^{**}):\quad g_i x_1\dots x_p \alpha_1\dots\alpha_n h_1\dots h_k = h_i(x_1\alpha_1\dots\alpha_n h_1\dots h_k)\dots(x_p\alpha_1\dots\alpha_n h_1\dots h_k)$$

dove $\alpha_1,\dots,\alpha_n = \bar{\alpha}$ sono variabili di tipo e dove il tipo di g_i e' scelto nell'unico modo possibile affinche', se g_i e' una funzione base $g_i: S_{i(1)}\to\dots\to S_{i(p)}\to S_{i(p+1)}$ allora $(x_1\alpha_1\dots\alpha_n h_1\dots h_k)$ ha tipo $\alpha_{i(1)}$, ..., $(x_p\alpha_1\dots\alpha_n h_1\dots h_k)$ ha tipo $\alpha_{i(p)}$.

Analogamente se $Hx = Z_{\alpha}^{S} xh_1\dots h_k$ allora la funzione H e' rappresentata dal termine $\underline{H}$ = $= \lambda x.(x\alpha_1\dots\alpha_n\underline{h}_1\dots\underline{h}_k)$ dove $\underline{h}_1,\dots,\underline{h}_k$ rappresentano $h_1,\dots,h_k$ e il tipo di x e' scelto in modo da rendere il termine $\underline{H}$ correttamente tipato.

Riferimenti bibliografici

(1) G.Birkhoff e J.Lipson, Heterogeneous Algebras J.Comb.Theory 8, 1970

(2) C.Böhm e A.Berarducci, Towards an Automatic Synthesis of Polymorphic Typed Lambda Terms.ext. abstract, non pubblicato, Novembre 1983.

(3) C.Böhm e D.Kozen, Eliminating Recursion over Acyclic Data Structures in Functional Programs Fourth International Workshop on the Semantics of Programming Languages, Bad Honnef, March 14-18, 1983.

(4) C.Böhm e A. Berarducci, Automatic Synthesis of Typed Λ-Programs on Term Algebras, inviato per pubblicazione al Journal of Theoretical Computer Science, Maggio 1984.

(5) A.Church, <u>The Calculi of Lambda Conversion</u>
Princeton University Press 1941

(6) S.Fortune, D.Leivant e M.O'Donnell, <u>The Expressiveness of Simple and Second Order Type Structures</u>, JACM, vol.30 n.1, Gennaio 1983

(7) K.Gödel, <u>Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes</u>, Dialectica,12,1958

(8) R.Hindley, B.Lercher e J.Seldin, <u>Introduction to Combinatory Logic</u>, Cambridge University Press, 1972

(9) D.Leivant, <u>Reasoning about Functional Programs and Complexity Classes associated with Type Disciplines</u>, Twenty-fourth Annual Symposium on Foundation of Computer Science, 1983, pp.460-469