Topologie e lambda calcolo

Autore: Manzonetto Giulio

Mestre (Venezia - Italia), 21 ottobre 2004

Ai miei genitori.

Indice

Abstract

In questo elaborato vengono applicate allo studio del λ-calcolo e della teoria della computabilità alcune fra le più importanti topologie, quali la topologia di Scott, di Visser e di Ershov. Durante la trattazione vengono riformulate delle versioni per alberi di Böhm del teorema del parametro e dei teoremi di Rice. Tramite questi risultati viene dimostrato che la topologia di Scott e la topologia di Ershov coincidono se definite sul cpo degli alberi di Böhm. Tale eguaglianza fra topologie ci permette di formulare una dimostrazione breve ed elegante del teorema di Kleene per alberi di Böhm. Viene, inoltre, definita e studiata la topologia di Visser sull'insieme dei λ-termini e, grazie alle sue proprietà, viene fornita una nuova dimostrazione topologica del lemma di genericità.

Keywords: λ-calcolo, funzioni ricorsive, domini, reticoli, topologie, topologia di Scott, topologia di Visser, topologia di Ershov, topologia di Lawson, topologia upper interval, topologia lower interval, lemma di genericità, teoremi di Rice, alberi di Böhm.

Notazione

Poiché la notazione utilizzata in questo elaborato non sempre è *standard* (e non siamo nemmeno sicuri di poter affermare che *esista* una notazione *standard*) riportiamo qui di seguito i simboli che utilizzeremo con il corrispondente significato.

Teoria degli insiemi

$\emptyset, \mathbb{N}$	insieme vuoto, insieme dei numeri naturali
$\cup, \cap$	unione ed intersezione di due insiemi
$\bigcup, \bigcap$	unione ed intersezione di una famiglia di insiemi
$X \subseteq Y$	X è sottoinsieme di Y
$X \subsetneq Y$	X è sottoinsieme *proprio* di Y
$\mathscr{P}(X)$	l'insieme delle parti di X
$\lvert X \rvert$	cardinalità dell'insieme X
$X \setminus Y$	sottrazione insiemistica fra X e Y
$X/_R$	l'insieme X modulo la relazione di equivalenza R
X^C	complemento dell'insieme X rispetto all'insieme universo

Teoria classica della computabilità

Φ_n	funzione computata dall'n-esimo programma della macchina di Turing
$f \downarrow x$	la funzione f è definita su x
$f \uparrow x$	la funzione f è indefinita su x
$\simeq$	eguaglianza di Kleene definita sui termini parziali, ovvero $s \simeq t$ se e solo se $((s \downarrow) \wedge (t \downarrow) \wedge (s = t))$ oppure $((s \uparrow) \wedge (t \uparrow))$
$\sim$	la relazione così definita: $x \sim y$ se e solo se $\Phi_x = \Phi_y$

λ-calcolo

Λ^-	insieme dei pre-termini
Λ	insieme dei λ-termini
Λ^0	insieme dei λ-termini chiusi
$\mathcal{BT}$	insieme degli alberi di Böhm
$\mathcal{BT}^{re}$	insieme degli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere
$FV(M)$	variabili libere di M
$BT(M)$	albero di Böhm di un termine M
$ABT(M)$	approssimante di un termine M
$\mathcal{ABT}(M)$	insieme degli approssimanti di un termine M
$Node$	insieme di sequenze di naturali rappresentanti nodi di alberi
$Label$	insieme delle etichette degli alberi di Böhm
$\equiv$	eguaglianza sintattica
$/M/$	numero di Gödel del termine M
n^*	λ-termine con numero di Gödel n
$\lceil n \rceil$ oppure n	numerale di Church associato al naturale n
$\boldsymbol{E}$	combinatore tale che $(\forall M \in \Lambda^0).(\boldsymbol{E}/M/ \twoheadrightarrow_\beta M)$
$\mathcal{P}_{\mathcal{BT}}$	insieme dei programmi che calcolano alberi di Böhm
$\lvert M \rvert_{\mathcal{BT}}$	programma (codificato dal λ-termine M) che calcola un albero di Böhm
t_α	sottoalbero dell'albero t radicato nel nodo α
$parent(\alpha)$	nodo padre del nodo α
$\alpha * \beta$	giustapposizione fra le sequenze di numeri α e β
K_Λ	insieme dei λ-termini che possiedono una head normal form
$M^\star$	riduzione massima in parallelo a un passo di M

Topologia

τ_A	topologia di Alexandroff
τ_S	topologia di Scott
τ_E	topologia di Ershov
τ_L	topologia di Lawson
τ_V	topologia di Visser
τ_{up}	topologia upper interval
τ_{low}	topologia lower interval

Basi e reticoli limitati e distributivi

$\mathcal{B}_S$	base di Scott formata dalle chiusure verso l'alto delle funzioni finite
$\mathcal{F}_S$	reticolo limitato e distributivo su $\mathcal{FP}$ ottenuto dalla chiusura di $\mathcal{B}_S$ per unione finita
$\mathcal{RE}_{\mathscr{C}}$	reticolo limitato e distributivo su $\mathscr{C}$ composto da insiemi la cui antimmagine attraverso Φ è r.e.
$\mathcal{RE}_{\mathbb{N}}$	reticolo limitato e distributivo su $\mathbb{N}$ composto da insiemi r.e. che rispettano le funzioni
$\mathcal{RE}_{\mathcal{BT}}$	reticolo limitato e distributivo su $\mathcal{BT}^{re}$ composto da insiemi la cui antimmagine attraverso BT è r.e.

Ordinamenti

poset	insieme parzialmente ordinato
dcpo	ordinamento parziale completo diretto
cpo	ordinamento parziale completo
$X \subseteq_{dir} Y$	X è un sottoinsieme diretto di Y
$\bigvee X$	estremo superiore dell'insieme X ordinato con $\leq$
$\bigsqcup X$	estremo superiore dell'insieme X ordinato con $\sqsubseteq$
$X\!\uparrow$	il più piccolo insieme chiuso verso l'alto contenente X
$X\!\downarrow$	il più piccolo insieme chiuso verso il basso contenente X

Operazioni sulle funzioni

$\mathcal{FP}$	insieme delle funzioni parziali da $\mathbb{N}$ in $\mathbb{N}$
$\mathscr{C}$	insieme delle funzioni parziali calcolabili da $\mathbb{N}$ in $\mathbb{N}$
$dom(f)$	dominio della funzione f
$cod(f)$	codominio della funzione f
$f : X \to Y$	funzione con $dom(f) = X$ e $cod(f) \subseteq Y$
$f : X \rightharpoonup_p Y$	funzione parziale con $dom(f) \subseteq X$ e $cod(f) \subseteq Y$
$f^{-1}[U]$	denota l'insieme $\{x \mid f(x) \in U\}$
$f[X]$	denota l'insieme $\{f(x) \mid x \in X\}$
$f \circ g$	composizione di f e g, ovvero $(f \circ g)(x) = g(f(x))$

λ-teorie

$\boldsymbol{\lambda\beta}$	teoria che eguaglia i termini β-equivalenti
$\mathcal{H}$	teoria che eguaglia tutti gli unsolvable
$\mathcal{B}$	teoria che eguaglia i termini che hanno lo stesso albero di Böhm
$\lambda + \mathcal{T}$	teoria ottenuta aggiungendo gli assiomi di $\mathcal{T}$ a quelli di $\boldsymbol{\lambda\beta}$
$\mathcal{T}^+$	insieme delle equazioni chiuse provabili in $\lambda + \mathcal{T}$

Logica

$\wedge, \vee$	connettivi logici "*e*" e "*o*"
$\Rightarrow$	implicazione logica
$\iff$	se e solo se

Capitolo 1

Introduzione

Il λ-calcolo è un formalismo computazionale introdotto da Church nel 1932 con l'obiettivo di trovare una nuova teoria fondazionale della matematica. Il progetto iniziale del 1932 prevedeva una teoria più generale, rispetto a quella che conosciamo oggi, che comprendeva anche elementi di logica. Nel 1935, tuttavia, Kleene e Rosser, all'epoca studenti di Church, dimostrarono l'inconsistenza di tale sistema. L'anno seguente Church, con l'aiuto dei suoi due allievi, separò una parte consistente del sistema e definì quello che è oggi noto come λ-calcolo. Furono proprio Church e Rosser, nel 1936, a fugare ogni dubbio dimostrando la consistenza del nuovo sistema con una tecnica che sfruttava le proprietà della β-riduzione.

Nello stesso anno in cui Turing definì la nozione di computabilità tramite la macchina di Turing, Church avanzò l'ipotesi che ogni funzione intuitivamente computabile fosse anche λ-definibile. Tale congettura, conosciuta in letteratura come *tesi di Church* non è a tutt'oggi mai stata dimostrata né è mai stato trovato un controesempio che la neghi. Church e Turing forniscono esempi di predicati non decidibili, ossia la cui funzione caratteristica non sia computabile algoritmicamente, quali la normalizzazione di un λ-termine o la terminazione di un programma su un dato input. Altri esempi di tali problematiche e i risultati fondamentali della teoria classica della computabilità sono esposti in Appendice D.

Il λ-calcolo deve il suo nome al simbolo di astrazione con cui vengono definite le funzioni. Il simbolo che identifica l'astrazione funzionale in tale sistema è, infatti, la lettera greca λ. Ci si può, in effetti, chiedere quale sia la genesi di tale notazione e quali sono stati i motivi che hanno portato alla scelta di questo particolare simbolo. Molto probabilmente questa lettera deve la sua fortuna non tanto ad un suo significato recondito, quanto ad un aspetto sintattico. Russel e Whitehead, nel loro famoso *Principia Mathematica*, indicavano una funzione come $f(x) = 7x+4$ con $7\hat{x} + 4$. Ispirandosi a questa notazione Church deve aver inizialmente scritto le funzioni come $\hat{x}.7x + 4$. Per problemi tipografici dell'epoca la notazione venne trasformata in $\wedge x.7x + 4$ ed infine diventò $\lambda x.7x + 4$.

Nel Capitolo 2 di questa tesi viene definito ed analizzato il formalismo computazionale del λ-calcolo non tipato partendo dalla nozione di pre-termine ed estendendola poi a quella di λ-termine. L'idea intuitiva è quella di considerare indistinguibili i pre-termini che differiscono solo per i nomi delle variabili legate. Vengono, inoltre, introdotte le relazioni di riduzione più importanti come, ad esempio, la β-riduzione e la riduzione di testa, vengono descritte le relative semantiche e infine presentati i risultati fondamentali in questo campo.

In seguito studieremo la potenza espressiva del λ-calcolo introducendo i numerali di Church, formulando i teoremi del punto fisso e formalizzando il concetto di funzione λ-definibile. Oltre alle nozioni di termine solvable e unsolvable, di albero di Böhm e di approssimazione di un λ-termine, vedremo cosa sono le λ-teorie e quali sono le proprietà ad esse collegate.

Nel Capitolo 3 ci si propone di fornire, senza richiedere al lettore particolari preconoscenze, le nozioni topologiche fondamentali a partire dal concetto stesso di topologia al fine di garantire la comprensione di questo elaborato anche a chi si addentri per la prima volta nello studio di questa disciplina. Le topologie sono strumenti matematici che possono essere utili per trovare delle dimostrazioni eleganti e formali. Esse sono state inizialmente introdotte per dare una formalizzazione del concetto intuitivo di "vicinanza" fra elementi di un dato insieme. Pur nascendo da proprietà metriche si sono col tempo svincolate da esse. Uno spazio topologico è un insieme a cui è associata una collezione di sottoinsiemi chiamati *insiemi aperti*. Intuitivamente due elementi sono tanto più vicini fra loro quanti più sono gli insiemi aperti che li contengono entrambi. Illustreremo i principali metodi per generare una topologia, daremo alcuni esempi generali di topologia e introdurremo le proprietà principali degli spazi topologici.

Nel Capitolo 4 vengono introdotti i reticoli e gli ordinamenti parziali e vengono considerati alcuni casi di studio di ordini parziali completi particolarmente importanti in informatica teorica. A titolo di esempio riportiamo il cpo delle funzioni parziali e quello degli alberi di Böhm. Vengono introdotte e studiate, fra le altre, la topologia di Alexandroff, quella di Scott, quella di Visser, quella di Ershov e quella di Lawson.

Nel Capitolo 5, dove vengono dimostrati i risultati più importanti di questa tesi, vengono definite la topologia di Scott e la topologia di Ershov sul cpo delle funzioni calcolabili parziali e su quello degli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere. Viene inoltre dimostrato che le due topologie coincidono su questi cpo e viene fornita una dimostrazione topologica del teorema di Kleene per la teoria delle funzioni ricorsive parziali. Il teorema del parametro, il secondo ed il terzo teorema di Rice e il teorema di Kleene vengono riformulati e dimostrati in una versione per alberi di Böhm.

Nel Capitolo 6, partendo da una λ-teoria $\mathcal{T}$, viene definita la topologia di Visser sull'insieme dei λ-termini, ovvero la topologia che ha come base insiemi co-ricorsivamente enumerabili che siano chiusi rispetto alla $\mathcal{T}$-equivalenza. Si dimostra che tale spazio topologico è iperconnesso e che, grazie a questa caratteristica, è possibile formulare una nuova dimostrazione topologica del lemma di genericità, un risultato di particolare rilevanza per il λ-calcolo.

Nella seguente figura vengono indicate le interdipendenze fra i vari capitoli e appendici di questa tesi. Nel grafo è presente un arco orientato fra il Capitolo i e il Capitolo j se il contenuto del Capitolo i è necessario alla comprensione del Capitolo j. Chiaramente il Capitolo 1, in quanto introduttivo, non appare in questo schema.

$$\begin{tikzcd}
A & 2 \arrow[l] \arrow[dl] \arrow[d] \arrow[ddr] & 3 \arrow[r] \arrow[d] & C \\
B & 5 \arrow[d] & 4 \arrow[l] \arrow[d] & \\
D \arrow[ur] & 7 & 6 \arrow[l] &
\end{tikzcd}$$

Capitolo 2

λ-calcolo non tipato

In questo capitolo introdurremo i concetti fondamentali riguardanti il λ-calcolo e daremo alcuni cenni storici. Definiremo le più importanti congruenze e relazioni di riduzione e vedremo inoltre cos'è un albero di Böhm e una λ-teoria.

2.1 Introduzione al λ-calcolo

Il λ-calcolo è una teoria formale non tipata che fu proposta da Alonzo Church nel 1932 come teoria fondazionale per la matematica e che diventò di rilevante interesse nel campo dell'informatica teorica e della logica. Il sistema iniziale di Church, che prevedeva anche elementi di logica, era più generale di quello conosciuto attualmente. Sfortunatamente nel 1935 Kleene e Rosser, due studenti di Church, dimostrarono l'inconsistenza[1] di tale teoria. L'anno successivo Church, continuando i suoi studi coadiuvato dai suoi due allievi, riuscì ad isolarne un sottosistema consistente che è oggi conosciuto come λ-calcolo.

Il λ-calcolo è una teoria orientata alle funzioni ed è *non estensionale* cioè λ-termini distinti possono avere lo stesso comportamento funzionale. Uno degli aspetti che lo rende interessante è il fatto che ciascun termine

[1]Una teoria equazionale è detta *inconsistente* se tutti i termini vengono eguagliati dagli assiomi vigenti.

è, nel contempo, funzione e argomento senza distinzione alcuna; ciò permette di far lavorare un programma (inteso come funzione) su un altro in modo automatico, senza cioè essere costretti a cercare, come avviene invece nella teoria classica della computabilità, una funzione effettiva di codifica definita dall'insieme dei programmi a quello degli argomenti.

Numerose proprietà del λ-calcolo non verranno trattate in questo elaborato in quanto esulano dai nostri scopi; si consiglia a chi fosse interessato ad approfondire tali argomenti la consultazione di [Barendregt, 1984], [Krivine, 1990] e [Sòrensen and Urzyczyn, 1998].

2.2 Interpretazione Informale del λ-calcolo

Prima di fornire una corretta definizione formale del λ-calcolo conviene spiegare in maniera meno matematica ma sicuramente più intuitiva le idee che stanno alla base di questo formalismo computazionale per far sì che anche chi si avvicina per la prima volta a tali concetti riesca a comprenderli efficacemente. Partiamo dunque dalla nozione di funzione ricordando come viene solitamente espressa nei formalismi matematici. Consideriamo una espressione generica:

$$E$$

che, auspicabilmente, conterrà delle occorrenze della variabile x. Per poter costruire una funzione basandoci sulla nostra espressione E siamo soliti introdurre un nuovo simbolo, ad esempio f, e porre:

$$f(x) = E.$$

Il nuovo simbolo f è il nome della funzione.

Nel λ-calcolo la stessa funzione si scrive in tal modo:

$$\lambda x.E$$

in cui λ è l'operatore di astrazione funzionale ed è lo strumento che permette di rappresentare le funzioni unarie, ossia dipendenti da un solo parametro. Si noti che, in questo modo, abbiamo costruito una funzione senza averle attribuito nome alcuno. Possiamo calcolare il valore che assume $\lambda x.E$ su un argomento a sostituendo ogni occorrenza della variabile x in E con a. Tale operazione è svolta da $\rightarrow_\beta$.

Si consideri, ad esempio, l'espressione aritmetica:

$$x^2 + x + 1$$

e si costruisca la funzione:

$$\lambda x.x^2 + x + 1$$

come visto in precedenza.

L'applicazione di un parametro, quale potrebbe essere 4, a $(\lambda x.x^2 + x + 1)$ produrrà le seguenti computazioni:

$$(\lambda x.x^2 + x + 1)4 \rightarrow_\beta (x^2 + x + 1)[x := 4] \;=\; 4^2 + 4 + 1 = 21$$

dove con $(x^2+x+1)[x := 4]$ si intende l'espressione x^2+x+1 in cui si sostituisce 4 ad ogni occorrenza della variabile x. La rappresentazione delle funzioni di arietà maggiore di uno in λ-calcolo è consentita da una proprietà dimostrata da Schönfinkel in [Schönfinkel, 1924] secondo la quale qualsiasi funzione a più variabili può essere ottenuta applicando sequenzialmente più funzioni unarie.

Consideriamo una funzione a due argomenti $f(x, y)$. Vediamo in che modo è possibile scomporre f in due funzioni di arietà uno che chiameremo F e F_x:

$$F_x \equiv (\lambda y.f(x, y)) \quad F \equiv \lambda x.F_x$$

$$((F\ x)\ y) = (F_x\ y) = f(x, y).$$

2.3 Pre-termini

Gli oggetti di studio nel λ-calcolo sono i λ-*termini* ma, per poterli definire correttamente, è conveniente introdurre in primo luogo la nozione di *pre-termine*.

Definizione 2.3.1 *Siano dati:*

$$V = \{v_0, v_1, \ldots\}$$

un insieme che denota un alfabeto infinito di variabili, il simbolo "λ" detto operatore di astrazione*, il simbolo "·" detto* operatore di applicazione *(che viene per brevità omesso e sostituito con la semplice giustapposizione) e i simboli "(" e ")" che consentono la* parentesizzazione*. Viene chiamato Λ^- l'insieme dei* pre-termini *cioè l'insieme di tutte le stringhe definite tramite la seguente grammatica espressa in BNF*[2]*:*

$$\Lambda^- ::= V|(\Lambda^-\Lambda^-)|(\lambda V\Lambda^-)$$

Le stringhe seguenti, ad esempio, sono pre-termini.

(i) $(((v_0\ v_1)\ v_2)\ v_3)$;

(ii) $(\lambda v_0(v_0\ v_1))$;

(iii) $((\lambda v_0\ v_0)\ v_1)$;

(iv) $(\lambda v_0(v_0\ v_1))(\lambda v_2(v_2\ v_3))$.

Per denotare elementi arbitrari di Λ^- si è soliti utilizzare lettere maiuscole quali K, L, M, N, O, P, Q, R. Verranno riservate, invece, lettere minuscole quali x, y, v, z, per denotare elementi arbitrari di V.

Terminologia 2.3.2 *Per convenzione:*

(i) Un pre-termine nella forma x, con $x \in V$, si dice variabile*;*

(ii) Un pre-termine nella forma $(\lambda x\ M)$ si dice λ-astrazione *su x;*

[2] Ovvero Backus-Naur Form, una notazione introdotta da J. Backus agli inizi degli anni '60. Naur era il curatore di una rivista, *l'Algol 60 Report*, in cui - per la prima volta - comparve la sintassi di un linguaggio di programmazione formalizzata in modo preciso.

(iii) Un pre-termine nella forma $(M\ N)$ *si dice* applicazione *di* M *a* N.

La parentesizzazione completa è spesso sconveniente per problemi di leggibilità. Si preferisce introdurre, quindi, una convenzione per l'eliminazione delle parentesi ridondanti. In via informale conviene pensare che l'operatore di applicazione associ verso sinistra mentre la λ-astrazione associ quanto più a destra è consentito.

Notazione 2.3.3 *Siano utilizzate le seguenti abbreviazioni:*

(i) $(M\ P\ Q)$ *al posto di* $((M\ P)\ Q)$;

(ii) $(\lambda x\ \lambda y\ M)$ *al posto di* $(\lambda x\ (\lambda y\ M))$;

(iii) $(\lambda x\ M\ N)$ *al posto di* $(\lambda x\ (M\ N))$;

(iv) $(M\ \lambda x\ N)$ *al posto di* $(M\ (\lambda x\ N))$.

Possiamo dunque omettere tali parentesi.

A questo punto è utile introdurre un'altra piccola variazione nella notazione adottata.

Notazione 2.3.4 *Scriveremo:*

$$\lambda x_1 \ldots x_n.M \quad \textit{in luogo di} \quad \lambda x_1 \ldots \lambda x_n\ M\ .$$

Come caso specifico di questa regola generale con $\lambda x.M$ *si intende* $\lambda x\ M$.

Con questi nuovi accorgimenti sintattici possiamo riscrivere i pretermini esemplificativi precedentemente elencati come segue:

(i) $v_0\ v_1\ v_2\ v_3$;

(ii) $\lambda v_0.v_0\ v_1$;

(iii) $(\lambda v_0.v_0)v_1$;

(iv) $(\lambda v_0.v_0\ v_1)\ \lambda v_2.v_2\ v_3$.

Spesso si continuerà a preferire

$$(\lambda v_0.v_0\ v_1)(\lambda v_2.v_2\ v_3)$$

in luogo di $(\lambda v_0.v_0\ v_1)\ \lambda v_2.v_2\ v_3$ per consentire una maggior leggibilità.

Definizione 2.3.5 *Sia $M \in \Lambda^-$, allora si definisce induttivamente l'insieme $FV(M)$ delle* variabili libere *di M come segue:*

$$\begin{array}{lcl} FV(x) & = & \{x\}; \\ FV(\lambda x.P) & = & FV(P)\backslash\{x\}; \\ FV(P\ Q) & = & FV(P) \cup FV(Q). \end{array}$$

Un pre-termine $M \in \Lambda^-$ viene detto *chiuso* se $FV(M) = \emptyset$. Forniamo ora la nozione di *sostituzione* che ci sarà utile, come abbiamo visto nel Paragrafo 2.2, per svolgere le computazioni quando applichiamo ad una funzione del λ-calcolo un argomento.

Definizione 2.3.6 *Siano $M,\ N \in \Lambda^-$ e $x \in V$ allora si dice* sostituzione *di N a x in M e si scrive $M[x := N] \in \Lambda^-$ l'operazione così definita induttivamente:*

$$\begin{array}{lcll} x[x := N] & = & N; & \\ y[x := N] & = & y & \textit{con } x \neq y; \\ (M\ P)[x := N] & = & M[x := N]\ P[x := N]; & \\ (\lambda x.M)[x := N] & = & \lambda x.M; & \\ (\lambda y.M)[x := N] & = & \lambda y.M[x := N] & \textit{se } y \notin FV(N) \textit{ o } x \notin FV(M); \\ (\lambda y.M)[x := N] & = & \lambda z.M[y := z][x := N] & \textit{se } y \in FV(N) \textit{ e } x \in FV(M) \end{array}$$

con $z \in \{v_i \in V|\ v_i \notin FV(M) \cup FV(N)\}$.

In quest'ultima definizione potrebbe non risultare evidente perché la regola da utilizzare nel caso di un pre-termine del tipo $(\lambda y.M)$ a cui applichiamo la sostituzione $[x := N]$ dipenda dall'appartenenza di x e y rispettivamente agli insiemi delle variabili libere di N e di M. Forniamo, quindi, un esempio significativo delle problematiche che si potrebbero riscontrare se tale distinzione non venisse effettuata. Si consideri il pre-termine $(\lambda y.xyy)$ a cui applichiamo la sostituzione $[x := yy]$. Se provassimo ad utilizzare la regola $(\lambda y.M)[x := N] = \lambda y.M[x := N]$ con $M \equiv xyy$ e $N \equiv yy$ anche in questo caso (si noti che $y \in FV(N)$) si otterrebbe:

$$(\lambda y.xyy)[x := yy] = \lambda y.xyy[x := yy] = \lambda y.yyyy.$$

Questo risultato è errato in quanto la variabile y che occorreva libera in N diventa legata dopo la sostituzione. La scorrettezza di tale sostituzione si può anche desumere dal fatto che il pre-termine $(\lambda y.xyy)$ ha lo stesso comportamento funzionale di $(\lambda z.xzz)$ quindi $(\lambda z.xzz)[x := yy]$ dovrebbe dare un risultato che ha, a sua volta, lo stesso comportamento funzionale di $(\lambda y.xyy)[x := yy]$. Invece

$$(\lambda z.xzz)[x := yy] = \lambda z.yyzz$$

che è il risultato corretto ed è stato ottenuto rinominando, prima di eseguire la sostituzione, la variabile y in $z \notin FV(N)$ che poi è l'operazione eseguita dall'ultima regola della Definizione 2.3.6. Problemi analoghi si verificano anche se $x \in FV(M)$.

Come abbiamo visto nel Paragrafo 2.2, i pre-termini rappresentano funzioni. Il risultato che otteniamo applicando un pre-termine ad un argomento non dipende dal nome delle variabili legate in esso contenute. È quindi necessario introdurre una relazione di congruenza che eguagli i termini che differiscono fra loro solo per il nome delle variabili legate.

Definizione 2.3.7 *Sia detta* α-equivalenza *e si scriva* $=_\alpha$ *la più piccola relazione su* Λ^-*, tale che:*

$$\begin{array}{ll} M =_\alpha M & \textit{per ogni } M \in \Lambda^-; \\ \lambda x.M =_\alpha \lambda y.M[x := y] & \textit{se } y \notin FV(M), \end{array}$$

ed inoltre sia chiusa rispetto alle seguenti regole:

$$\begin{array}{lll} M =_\alpha M' & \Rightarrow & (\forall x \in V).(\lambda x.M =_\alpha \lambda x.M'); \\ M =_\alpha M' & \Rightarrow & (\forall Q \in \Lambda^-).(M\ Q =_\alpha M'\ Q); \\ M =_\alpha M' & \Rightarrow & (\forall Q \in \Lambda^-).(Q\ M =_\alpha Q\ M'); \\ M =_\alpha M' & \Rightarrow & M' =_\alpha M; \\ M =_\alpha M' \ \wedge\ M' =_\alpha M'' & \Rightarrow & M =_\alpha M''. \end{array}$$

La relazione di α-equivalenza viene anche detta in letteratura relazione di *α-congruenza.*

Osservazione 2.3.8 *La relazione $=_\alpha$ di α-equivalenza è decidibile cioè la* funzione caratteristica[3] *di tale relazione è rappresentabile tramite il formalismo delle funzioni ricorsive.*

Sebbene sia semplice comprendere il perché di questa osservazione ne riportiamo una dimostrazione in Appendice A.

Definizione 2.3.9 *Si definisce* rinomina delle variabili legate *di M, con $M \equiv \lambda x.N$*[4]*, l'operazione ρ attraverso la quale si sostituisce ogni occorrenza in M della variabile legata x con una qualche variabile z che non sia già presente in M, cioè, $\rho(x, M) = \lambda z.N[x := z]$ con $z \notin FV(M)$.*

Diremo che è stata effettuata una rinomina delle variabili libere di M anche nel caso in cui M non sia una λ-astrazione ma sia comunque un termine che ne contenga. Siano considerati due termini arbitrari $M, N \in \Lambda^-$. È semplice verificare che M è α-equivalente ad N se e solo se esiste un numero finito di operazioni di rinomina delle variabili legate di M che lo rende sintatticamente uguale a N.

2.4 λ-termini

In questa sezione introduciamo i λ-termini che sono gli effettivi oggetti di studio del λ-calcolo. L'idea intuitiva è quella di considerare equivalenti tutti i pre-termini che differiscono solo per il nome delle variabili legate.

Definizione 2.4.1 *Definiamo per ogni $M \in \Lambda^-$ la* classe di equivalenza *$[M]_\alpha$ in tal modo:*

$$[M]_\alpha = \{N \in \Lambda^- \mid M =_\alpha N\}$$

[3]Per funzione caratteristica di una relazione binaria si intende la funzione $f(x, y)$ che vale 1 se x è in relazione con y, 0 altrimenti.

[4]Si noti che con $\equiv$ si intende l'eguaglianza sintattica.

Allora ci è possibile definire l'insieme Λ dei λ-termini come:

$$\Lambda \quad = \quad \Lambda^-/_{=_\alpha} \quad = \quad \{[M]_\alpha | M \in \Lambda^-\}$$

Per brevità, nel seguito della trattazione, scriveremo M in luogo di $[M]_\alpha$ eccetto quando verrà esplicitamente dichiarato il contrario. Ecco degli esempi di λ-termini.

(i) $\mathbf{K} \equiv \lambda xy.x$;

(ii) $\mathbf{S} \equiv \lambda xyz.xz(yz)$;

(iii) $\mathbf{I} \equiv \lambda x.x$;

(iv) $\mathbf{1} \equiv \lambda xy.xy$;

(v) $\mathbf{Y} \equiv \lambda f.(\lambda x.f(x\ x))(\lambda x.f(x\ x))$;

(vi) $\omega \equiv \lambda x.xx$;

(vii) $\Omega \equiv \omega\omega \equiv (\lambda x.xx)(\lambda x.xx)$.

Allo stesso modo di come abbiamo fatto per i pre-termini diamo le nozioni di variabili libere, variabili legate e chiusura di un termine anche per i λ-termini.

Osservazione 2.4.2 *Si noti che:*

(i) dati $M, N \in \Lambda$, allora $M =_\alpha N \Rightarrow FV(M) = FV(N)$;

(ii) I λ-termini ereditano le nozioni di insieme delle variabili libere e occorrenza di una variabile libera dai pre-termini.

Sia $M \in \Lambda$, allora una variabile x che occorre in M si dice *vincolata* o *legata* se $x \notin FV(M)$. Un λ-termine $M \in \Lambda$ viene detto *chiuso* se $FV(M) = \emptyset$. Denotiamo l'insieme di tutti i λ-termini chiusi con Λ^0.

Definizione 2.4.3 *Sia $M \in \Lambda$, allora si definisce induttivamente l'insieme dei sottotermini $Sub(M)$ come segue:*

$$
\begin{aligned}
&Sub(x) = \{x\}\\
&Sub(\lambda x.P) = \{\lambda x.P\} \cup Sub(P)\\
&Sub(P\ Q) = \{P\ Q\} \cup Sub(P) \cup Sub(Q)
\end{aligned}
$$

Siano $M,\ N \in \Lambda$ allora si è soliti dire che M è *sottotermine* di N se $M \in Sub(N)$.

Osservazione 2.4.4 *Anche la nozione di sostituzione definita per i pretermini viene ereditata dai λ-termini con l'accorgimento che se M, N, M', $N' \in \Lambda$ e $M =_\alpha M', N =_\alpha N'$, allora $M[x := N] =_\alpha M'[x := N']$.*

Forniamo di seguito alcuni esempi di sostituzione per λ-termini.

(i) $(\lambda x.x\ y)[y := z] = (\lambda x.x\ z)$;

(ii) $(x\ (y\ z))[y := \lambda k.k] = (x\ ((\lambda k.k)\ z))$.

2.5 Contesti

Precedentemente abbiamo visto come, applicando l'operatore di sostituzione a un λ-termine, sia impossibile vincolare una variabile che occorreva libera all'interno di esso prima della sostituzione. Ciò nonostante talvolta possiamo essere interessati ad un comportamento come quello descritto e per questo definiamo i contesti. Intuitivamente un contesto è un λ-termine in cui sono presenti dei "buchi", ovvero delle meta variabili che possono essere sostituite brutalmente con qualsiasi λ-termine. Queste variabili, dette *variabili di contesto*, vengono denotate con le lettere iniziali dell'alfabeto quali: $a, b \ldots$

Definizione 2.5.1 *Sia V un insieme di variabili del λ-calcolo e W un insieme di variabili di contesto tali che $V \cap W = \emptyset$. Un contesto C viene definito induttivamente come segue:*

- *x è un contesto, per ogni variabile $x \in V$,*

- *a è un contesto, per ogni variabile di contesto $a \in W$,*
- *se C è un contesto $\Rightarrow \lambda x.C$ è un contesto, per ogni variabile $x \in V$,*
- *se C_1 e C_2 sono contesti $\Rightarrow C_1C_2$ è un contesto.*

Con la notazione $C[a_1, \ldots, a_n]$ si fa riferimento ad un contesto C in cui occorrono le variabili di contesto $a_1, \ldots, a_n$. Con $C[M_1, \ldots, M_n]$ denotiamo il contesto in cui le variabili di contesto $a_1, \ldots, a_n$ sono state sintatticamente sostituite dai termini $M_1, \ldots, M_n$.

La sostituzione che avviene quando si introduce un certo λ-termine al posto della meta variabile corrispondente è di tipo sintattico ed è possibile che alcune variabili libere in esso risultino vincolate a sostituzione avvenuta come nell'esempio che segue.

Esempio 2.5.2 *Consideriamo il contesto $C[a] \equiv \lambda x.xa$. Allora si ha che $C[xy] = \lambda x.xxy$ e, quindi, la variabile di contesto viene sostituita con un'occorrenza del termine xy. Si noti come x, che occorreva libera nel termine xy, si trovi ad essere vincolata dopo la sostituzione.*

Ora che abbiamo dato un'idea intuitiva di come avviene la sostituzione nel caso dei contesti ne riportiamo una definizione formale.

Definizione 2.5.3 *Sia $C[a_1, \ldots, a_n]$ un contesto allora si dice sostituzione di $M_1, \ldots, M_n$ ad $a_1, \ldots, a_n$ in C, e si scrive $C[M_1, \ldots, M_n]$, l'operazione definita come segue:*

$$
\begin{array}{lcl}
a_i[M_1, \ldots, M_n] & = & M_i \\
x[M_1, \ldots, M_n] & = & x \\
(C_1C_2)[M_1, \ldots, M_n] & = & C_1[M_1, \ldots, M_n]C_2[M_1, \ldots, M_n] \\
(\lambda x.C)[M_1, \ldots, M_n] & = & \lambda x.(C[M_1, \ldots, M_n])
\end{array}
$$

2.6 Riduzione

Nel seguito di questo paragrafo introdurremo il concetto di *riduzione* sui λ-termini. L'operatore di riduzione è quello che permette di svolgere le computazioni quando viene applicato un argomento a un λ-termine. È, in realtà, possibile definire vari tipi di relazioni di riduzione, a seconda della semantica che si vuole loro attribuire. Definiremo quindi, in primo luogo, il concetto più generale di riduzione per poi andare a vedere nel dettaglio alcune istanze.

Definizione 2.6.1 *Una relazione binaria R definita su Λ è detta* relazione di riduzione *se valgono le seguenti proprietà:*

$$
\begin{array}{lll}
(M,M) \in R & & \textit{(riflessività)} \\
(M,N) \in R \Rightarrow & (MZ,NZ) \in R & \\
& (ZM,ZN) \in R & \\
& (\lambda x.M, \lambda x.N) \in R & \textit{(compatibilità)} \\
(M,N) \in R, (N,Z) \in R & \Rightarrow (MZ) \in R & \textit{(transitività)}
\end{array}
$$

Ora che sappiamo cos'è una relazione di riduzione possiamo introdurre la più nota relazione fra λ-termini, ovvero la β-riduzione.

Definizione 2.6.2 *Sia $\rightarrow_\beta$ la più piccola relazione su Λ tale che valga:*

$$(\lambda x.M)N \rightarrow_\beta M[x := N],$$

e che sia chiusa rispetto alle seguenti regole:

$$
\begin{array}{lll}
M \rightarrow_\beta M' & \Rightarrow & (\forall x \in V).(\lambda x.M \rightarrow_\beta \lambda x.M') \\
M \rightarrow_\beta M' & \Rightarrow & (\forall N \in \Lambda).(M\ N \rightarrow_\beta M'\ N) \\
M \rightarrow_\beta M' & \Rightarrow & (\forall N \in \Lambda).(N\ M \rightarrow_\beta N\ M').
\end{array}
$$

*Si dice che un termine nella forma $(\lambda x.M)N$ è un β-*redex *mentre $M[x := N]$ viene chiamato β-*contractum.

Un termine M è detto in *forma β-normale* (β-NF) se non esiste alcun termine N tale che $M \rightarrow_\beta N$ cioè se M non contiene β-redex.

Proposizione 2.6.3 *Esistono termini che non hanno una forma β-normale in quanto contengono β-redex il cui contractum contiene sempre un β-redex. In questo caso la riduzione continua all'infinito.*

Consideriamo questo esempio. Sia:

$$\Omega \equiv \ \omega\omega \ \equiv (\lambda x.xx)(\lambda x.xx).$$

Se si prova a svolgere la riduzione contenuta in Ω si ottiene:

$$\Omega \equiv (\lambda x.xx)(\lambda x.xx) \rightarrow_\beta (x\ x)[x := \lambda x.xx] = (\lambda x.xx)(\lambda x.xx) \equiv \Omega$$

dunque $\Omega \rightarrow_\beta \Omega \rightarrow_\beta \ldots \rightarrow_\beta \Omega$. Tale termine, dunque, non solo non è in forma β-normale poiché contiene un β-redex ma addirittura non possiede una forma β-normale.

Consideriamo ora il termine $\mathbf{KI}\Omega$, esso contiene due redex, Ω e $\mathbf{KI}$, e a seconda di quello che si sceglie di ridurre si otterranno risultati diversi. Proviamo, ad esempio, a ridurre il primo:

$$\mathbf{KI}\Omega \rightarrow_\beta \mathbf{KI}\Omega \rightarrow_\beta \ldots \rightarrow_\beta \mathbf{KI}\Omega$$

risulta evidente che le computazioni si protrarranno all'infinito. Vediamo, dunque, cosa accade se riduciamo il secondo:

$$\mathbf{KI}\Omega \rightarrow_\beta \mathbf{I}$$

con $\boldsymbol{I} \equiv \lambda x.x$ che è una β-NF.

Per indicare che da un λ-termine M si può giungere ad un λ-termine N con zero o un numero finito di applicazioni di $\rightarrow_\beta$ si può usare la relazione $\twoheadrightarrow_\beta$ che viene formalmente definita come segue.

Definizione 2.6.4 *La relazione $\twoheadrightarrow_\beta$ viene chiamata β-*riduzione transitiva *altresì detta β-*riduzione a più passi *ed è la chiusura riflessiva e transitiva di $\rightarrow_\beta$ ovvero $\twoheadrightarrow_\beta$ è la più piccola relazione chiusa rispetto le seguenti regole:*

$$\begin{array}{lcl} M \rightarrow_\beta M' & \Rightarrow & M \twoheadrightarrow_\beta M'; \\ M \twoheadrightarrow_\beta M' \ \wedge \ M' \twoheadrightarrow_\beta M'' & \Rightarrow & M \twoheadrightarrow_\beta M''; \\ M \twoheadrightarrow_\beta M. & & \end{array}$$

Un termine $M \in \Lambda$:

(i) si dice *β-normalizzabile* se è possibile ridurlo a una forma β-normale, cioè se esiste una β-NF N tale che $M \twoheadrightarrow_\beta N$;

(ii) si dice *fortemente β-normalizzabile* se, da esso, non è possibile generare una β-riduzione infinita;

(iii) si dice *β-infinito* se non è β-normalizzabile.

Definizione 2.6.5 *La relazione $=_\beta$ viene chiamata* β-equivalenza *o* β-congruenza *ed è la chiusura riflessiva, simmetrica e transitiva di $\twoheadrightarrow_\beta$, cioè $=_\beta$ è la più piccola relazione chiusa rispetto le seguenti regole:*

$$\begin{array}{lll}
M =_\beta M; & & \\
M \to_\beta M' & \Rightarrow & M =_\beta M'; \\
M =_\beta M' \ \wedge \ M' =_\beta M'' & \Rightarrow & M =_\beta M''; \\
M =_\beta M' & \Rightarrow & M' =_\beta M.
\end{array}$$

Ecco alcuni esempi:

(i) $(\lambda x.xy)(\lambda z.z) \to_\beta (x\ y)[x := (\lambda z.z)] = (\lambda z.z)y \to_\beta z[z := y] = y$;

(ii) $(\lambda x.xx)(\lambda y.y) \twoheadrightarrow_\beta (\lambda z.z)$;

(iii) $(\lambda x.xx)y =_\beta y\ ((\lambda z.z)y)$.

In modo analogo a quanto fatto per la β-riduzione è possibile definire per un'arbitraria relazione di riduzione R la sua chiusura riflessiva e transitiva e la R-equivalenza che verranno indicate rispettivamente con $\twoheadrightarrow_R$ e $=_R$.

Definizione 2.6.6 *Si dice che una relazione binaria R fra λ-termini gode della* proprietà del diamante *se vale la seguente proprietà:*

$$(M \to_R N_1) \wedge (M \to_R N_2) \Rightarrow (\exists Z \in \Lambda).((N_1 \to_R Z) \wedge (N_2 \to_R Z)).$$

Il nome di questa proprietà deriva dalla forma romboidale che impone alle riduzioni:

$$
\begin{tikzcd}
 & M \arrow[dl, "R"'] \arrow[dr, "R"] & \\
N_1 \arrow[dr, "R"'] & & N_2 \arrow[dl, "R"] \\
 & Z &
\end{tikzcd}
$$

Sia data una relazione di riduzione R. Se $\twoheadrightarrow_R$ soddisfa la proprietà del diamante si dice che R gode della *proprietà di Church-Rosser*.

Teorema 2.6.7 *Sia R una relazione che gode della proprietà di Church-Rosser, allora:*

$$M =_R N \Rightarrow (\exists Z \in \Lambda).((M \twoheadrightarrow_R Z) \wedge (N \twoheadrightarrow_R Z)).$$

L'indubbia importanza di questo teorema deriva dal fatto che ci assicura che due termini R-equivalenti abbiano un ridotto comune, come si può notare dal seguente schema:

$$
\begin{tikzcd}
M \arrow[dr, two heads, "R"'] & & N_1 \arrow[dl, two heads, "R"] \arrow[dr, two heads, "R"'] & & \cdots & & N_{n-1} \arrow[dl, two heads, "R"] \arrow[dr, two heads, "R"'] & & N \arrow[dl, two heads, "R"] \\
 & M_1 \arrow[dr, two heads, "R"'] & & M_2 \arrow[dl, two heads, "R"] \arrow[dr, two heads, "R"'] & \cdots & M_{n-1} \arrow[dl, two heads, dashed, "R"] \arrow[dr, two heads, "R"] & & M_n \arrow[dl, two heads, "R"] & \\
 & & Z_1 \arrow[dr, two heads, "R"'] & & Z_2 \arrow[dl, two heads, "R"] \arrow[dr, two heads, "R"'] & \cdots & Z_{n-1} \arrow[dl, two heads, dashed, "R"] & & \\
 & & & P_1 \arrow[dr, two heads, dashed, "R"'] & & P_2 \arrow[dl, two heads, dashed, "R"] & & & \\
 & & & & Z & & & &
\end{tikzcd}
$$

Proposizione 2.6.8 *Se una relazione binaria R gode della proprietà di Church-Rosser allora ogni λ-termine ammette al massimo una forma R-normale*[5].

[5]Analogamente alla forma β-normale, un termine M è detto in forma *R-normale* se non esiste alcun termine N tale che $M \rightarrow_R N$.

Teorema 2.6.9 (Church Rosser)

(i) $\to_\beta$ *gode della proprietà di Church-Rosser;*

(ii) $M =_\beta N \Rightarrow (\exists Z \in \Lambda).((M \twoheadrightarrow_\beta Z) \wedge (N \twoheadrightarrow_\beta Z))$.

Poiché la dimostrazione di questo teorema è molto intessante ma discretamente lunga si è ritenuto conveniente esporla in Appendice B. Un'altra interessante proprietà correlata al Teorema 2.6.9 è quella che stabilisce che due termini sono β-equivalenti se esiste un cammino a "zig-zag" di β-riduzione transitiva che porta dall'uno all'altro. Ovvero $M =_\beta N$ implica che esiste $n \in \mathbb{N}$ ed esistono $N_1, \ldots, N_{n-1}, M_1, \ldots, M_n \in \Lambda$ tali che:

$$\begin{tikzcd}
M \arrow[dr, two heads, "\beta"'] & & N_1 \arrow[dl, two heads, "\beta"] \arrow[dr, two heads, "\beta"'] & & \cdots & & N_{n-1} \arrow[dl, two heads, "\beta"] \arrow[dr, two heads, "\beta"'] & & N \arrow[dl, two heads, "\beta"] \\
& M_1 & & M_2 & \cdots & M_{n-1} & & M_n &
\end{tikzcd}$$

2.7 Ricorsione e punti fissi

Essendo il λ-calcolo un sistema Turing equivalente[6], e di questo se ne può trovare una dimostrazione nel Capitolo 3.8 di [Manzonetto, 2002], è possibile esprimere la ricorsione e, quindi, formulare un teorema del punto fisso.

Teorema 2.7.1 *(del punto fisso) Per ogni* $F \in \Lambda$ *esiste un* λ*-termine* X *tale che:*

$$FX =_\beta X$$

esiste, infatti un λ*-termine* $\boldsymbol{Y}$ *tale che, per ogni* $F \in \Lambda$*:*

$$F(\boldsymbol{Y}F) = \boldsymbol{Y}F$$

Dim. Poniamo

$$\boldsymbol{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

[6]Ovvero ogni funzione rappresentabile tramite il formalismo delle funzioni ricorsive è definibile nel λ-calcolo.

e supponiamo senza perdita di generalità che x non occorra libera in F, allora

$$\begin{array}{rcll} \boldsymbol{Y}F & \equiv & (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))F & \text{per definizione di } \boldsymbol{Y} \\ & =_\beta & (\lambda x.F(xx))(\lambda x.F(xx)) & \text{per la } \beta\text{-riduzione} \\ & =_\beta & F((\lambda x.F(xx))(\lambda x.F(xx))) & \text{per la } \beta\text{-riduzione} \\ & =_\beta & F((\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))F) & \text{per astrazione + applicazione} \\ & =_\beta & F(YF) & \text{per definizione di } \boldsymbol{Y} \end{array}$$

come richiesto. $\square$

Il termine $\boldsymbol{Y}$ viene chiamato in letteratura *combinatore di punto fisso di Curry*.

Corollario 2.7.2 *Sia $M \in \Lambda$, allora esiste un termine $F \in \Lambda$ tale che:*

$$F =_\beta M[f := F]$$

Dim. Poniamo

$$F \equiv \boldsymbol{Y}(\lambda f.M)$$

allora

$$\begin{array}{rcl} F & \equiv & \boldsymbol{Y}(\lambda f.M) \\ & =_\beta & (\lambda f.M)(\boldsymbol{Y}\lambda f.M) \\ & \equiv & (\lambda f.M)F \\ & =_\beta & M[f := F] \end{array}$$

come richiesto. $\square$

Si noti che il Corollario 2.7.2 ci permette di scrivere definizioni ricorsive di λ-termini; possiamo, infatti, definire F come un λ-termine che soddisfi l'*equazione al punto fisso* $F =_\beta \lambda x.M$ dove vi possono essere occorrenze del termine F all'interno di M. Chiaramente possono esistere più termini F che soddisfano tale equazione.

2.8 Numerali di Church

Per poter rappresentare all'interno del formalismo del λ-calcolo i numeri naturali introduciamo i così detti numerali di Church e una funzione $\lceil\cdot\rceil : \mathbb{N} \to \Lambda$ che associa ad ogni numero naturale il numerale corrispondente.

Definizione 2.8.1 *(Numerali di Church)*

> *(i) per ogni $n \in \mathbb{N}$ e $F, M \in \Lambda$ definiamo induttivamente $F^n(M)$ (applicazione iterata n volte di F a M) come segue:*
>
> $$\begin{array}{lcl} F^0(M) & = & M \\ F^{n+1}(M) & = & F(F^n M) \end{array}$$
>
> *(ii) Per ogni $n \in \mathbb{N}$, il* numerale di Church *$\lceil n \rceil$ è il λ-termine*
>
> $$\lceil n \rceil = \lambda sz.s^n z$$
>
> *e $\lceil n \rceil$ è il numero n rappresentato nel formalismo del λ-calcolo.*

Indicheremo con $\Lambda_{\mathbb{N}}$ l'insieme dei λ-termini rappresentanti i numerali di Church. Per alleggerire la notazione non faremo distinzione fra gli elementi di $\mathbb{N}$ e di $\Lambda_{\mathbb{N}}$. Scriveremo, quindi, $n \in \mathbb{N}$ sia per indicare il naturale n, che il λ-termine $\lceil n \rceil$.

I seguenti λ-termini sono numerali di Church:

(i) $0 \equiv \lambda sz.z$;

(ii) $1 \equiv \lambda sz.sz$;

(iii) $2 \equiv \lambda sz.s(sz)$;

(iv) $3 \equiv \lambda sz.s(s(sz))$;

(v) $4 \equiv \lambda sz.s(s(s(sz)))$.

2.9 Gödelizzazione

Abbiamo visto nel paragrafo precedente come sia possibile codificare i numeri naturali con dei λ-termini. In realtà è addirittura possibile, avendo Λ e $\mathbb{N}$ la stessa cardinalità, definire una funzione biiettiva fra l'insieme dei λ-termini e quello dei numerali di Church. La funzione che associa ad ogni λ-termine un numero viene chiamata funzione di gödelizzazione in onore del matematico Gödel.

Definizione 2.9.1 *(Gödelizzazione)*

(i) È semplice da definire, con le tecniche usualmente utilizzate, una funzione effettiva biiettiva $/ \cdot / : \Lambda \to \mathbb{N}$*,*

(ii) dato un λ*-termine* M*, il numerale* $/M/$ *viene detto il numero di Gödel di* M*,*

(iii) denoteremo con * *la funzione inversa di* $/\cdot/$*, ovvero quella funzione tale che, per ogni* $M \in \Lambda$*, vale* $/M/^* = M$*.*

La gödelizzazione, come dimostrato da Goldberg nel suo articolo [Goldberg, 2000], è necessariamente una nozione meta-linguistica nel λ-calcolo, ovvero non è possibile definire un combinatore che associ ogni λ-termine al suo numero di Gödel.

Proposizione 2.9.2 *Non esiste un combinatore* G *tale che, per ogni* λ*-termine* M*,*

$$(GM) = /M/.$$

Dim. Supponiamo che tale combinatore G esista. Allora $/\boldsymbol{I}/ = (G\boldsymbol{I}) = (G(\boldsymbol{II})) = /(\boldsymbol{II})/$. Questo è impossibile per il Teorema 2.6.9 di Church-Rosser, siamo quindi giunti ad un assurdo. $\square$

L'idea intuitiva di questa dimostrazione è che un combinatore non riuscirebbe ad associare numeri distinti a λ-termini β-equivalenti. Vediamo ora il seguente teorema.

Teorema 2.9.3 *Esiste un termine* $\boldsymbol{E} \in \Lambda^0$ *tale che*

$$(\forall M \in \Lambda^0).(\boldsymbol{E}/M/ \twoheadrightarrow_\beta M).$$

La dimostrazione di questo teorema viene qui omessa per la sua eccessiva lunghezza ma può essere trovata in [Barendregt, 1984, Teorema 8.1.6]. Si noti che, se $n = /M/$, allora $\lambda\beta \vdash \boldsymbol{E}n = M$. Bisogna fare, però, attenzione a non confondere i numerali e il numero di Gödel dei numerali. Vediamo un esempio chiarificatore.

Esempio 2.9.4 *Supponiamo di definire la funzione* $/M/$ *in modo che sia biiettiva e che:*

- *se* $M \equiv n$*, dove* n *è un numero,* $/M/ = 2n$*,*
- *altrimenti ad* M *viene associato un numero dispari.*

Ad esempio, con la funzione di gödelizzazione appena definita, il numero di Gödel associato al λ*-termine* 3 *è* 6*, ovvero* $/3/ = 6$*. Da questa considerazione ne deriva che* $\boldsymbol{E}6 \twoheadrightarrow_\beta 3$.

2.10 λ-definibilità

Vediamo ora come ogni λ-termine individua, in modo naturale, una funzione intesa in senso insiemistico. Più precisamente, dato un λ-termine $M \in \Lambda$, la funzione di arietà k *indotta* da M è la funzione $f : \Gamma \to \mathbb{N}$ tale che:

(i) $\Gamma = \{(y_1, \ldots, y_k) \in \mathbb{N}^k \mid (\exists i \in \mathbb{N}).(My_1 \ldots y_k =_\beta i)\}$;

(ii) per ogni $(y_1, \ldots, y_k) \in \Gamma$, definiamo $f(y_1, \ldots, y_k) = i$ se e solo se $My_1 \cdots y_k =_\beta i$ per un certo i (per la proprietà di Church-Rosser tale i è unico).

Consideriamo un sottoinsieme Γ di $\mathbb{N}^k$ ed una funzione $f : \Gamma \rightharpoonup_p \mathbb{N}$. Si dice che $F \in \Lambda$ *rappresenta* f in λ-calcolo sse la funzione di arietà k indotta da F coincide con f.

Definizione 2.10.1 *Si dice che f è* rappresentabile *o λ-definibile se esiste $F \in \Lambda$ che rappresenta f.*

Si noti che nel caso delle funzioni totali la Definizione 2.10.1 si riconduce alla seguente e ben più nota definizione di λ-definibilità che riportiamo di seguito.

Definizione 2.10.2 *Si dice che una funzione totale $f : \mathbb{N}^m \longrightarrow \mathbb{N}$ è* λ-definibile *se esiste un $F \in \Lambda$ tale che:*

$$Fn_1 \cdots n_m =_\beta f(n_1, \ldots, n_m)$$

per ogni $n_1 \ldots n_m \in \mathbb{N}$.

Proposizione 2.10.3

(i) Sia data una funzione ricorsiva f di arietà k. Allora esiste un λ-termine $M \in \Lambda$ che rappresenta f;

(ii) sia dato un λ-termine $M \in \Lambda$, allora esiste una funzione di arietà k indotta da M e tale funzione è rappresentabile nella teoria delle funzioni ricorsive.

Possiamo dunque dedurre che il potere espressivo del λ-calcolo è equivalente a quello della teoria delle funzioni ricorsive parziali. Possiamo quindi formulare il secondo teorema del punto fisso.

Teorema 2.10.4 *(Secondo teorema del punto fisso)*

$$(\forall F \in \Lambda)(\exists X \in \Lambda).(F/X/ = X).$$

Dim. Poiché $/\cdot/$ è una funzione effettiva, esistono due funzioni calcolabili che chiameremo Ap e Num tali che $Ap(/M/, /N/) = /MN/$ e $Num(n) = /n/$. Siano $\boldsymbol{Ap} \in \Lambda^0$ e $\boldsymbol{Num} \in \Lambda^0$ i termini che λ-definiscono Ap e Num rispettivamente. Allora:

$$\boldsymbol{Ap}\ /M//N/ = /MN/, \quad \boldsymbol{Num}\ /n/ = //n//;$$

quindi, in particolare,

$$\boldsymbol{Num}\ /M/ = //M//.$$

Dato un termine $F \in \Lambda$, definiamo:

$$W \equiv \lambda x.F(\boldsymbol{Ap}\ x(\boldsymbol{Num}\ x)).$$

Il termine X che cerchiamo è

$$X \equiv W/W/,$$

infatti:

$$\begin{aligned} X &\equiv W/W/ = F(\boldsymbol{Ap}\ /W/(\boldsymbol{Num}\ /W/)) \\ &= F/W/W// \equiv F/X/ \end{aligned}$$

□

2.11 Termini solvable e unsolvable

In precedenza abbiamo distinto i λ-termini che possiedono una forma normale da quelli che non la possiedono. Ora introdurremo una classe di termini che hanno un tipo differente di forma normale: la *forma normale di testa*, altresì detta *head normal form* (abbreviato *hnf*).

Cominciamo col darne una definizione formale.

Definizione 2.11.1 *Sia $M \in \Lambda$. Si dice che M è in* head normal form *se:*

$$M \equiv \lambda x_1 \ldots x_n.xM_1 \cdots M_m, \qquad m, n \geq 0$$

in questo caso x viene chiamata variabile di testa *o* head variable*.*

Ecco alcuni esempi di termini che sono in head normal form.

Esempio 2.11.2

(i) xM,

(ii) $\lambda x.x$,

(iii) $\lambda xy.x$,

(iv) $\lambda xy.z\Omega$.

I seguenti termini, invece, *non* sono in head normal form.

Esempio 2.11.3

(i) $(\lambda x.xyz)\boldsymbol{K}$,

(ii) $(\lambda x.x)(\lambda x.x)$,

(iii) $(\lambda xy.x)\Omega$.

Diremo che un termine M possiede una head normal form se esiste un termine M' che è in head normal form tale che $M =_\beta M'$. Se un termine $M \equiv \lambda x_1 \dots x_n.(\lambda x.M_0)M_1 \cdots M_m$ con $n \geq 0$ e $m \geq 1$ allora il sottotermine $(\lambda x.M_0)M_1$ viene chiamato *head redex* di M.

Si noti che l'head redex di un termine è anche il redex più a sinistra mentre non è vero l'opposto, ovvero non è detto che il redex più a sinistra di un termine sia un head redex, come si può notare dal controesempio che segue: $\lambda x.x((\lambda y.y)z)$.

Un'utile osservazione inerente a questi concetti appena introdotti è che la classe dei termini che hanno una head normal form è un sovrainsieme proprio di quella dei termini che possiedono una forma β-normale. È possibile affermare che un termine β-infinito è privo di contenuto informativo poiché ogni suo cammino di β-riduzione continua all'infinito. Se, però, tale termine possiede una head normal form, possiamo avere un qualche tipo di informazione su esso, sappiamo infatti che esiste un cammino di riduzione che lo può portare nella forma $\lambda x_1 \dots x_n.xM_1 \cdots M_m$ e che, per quanto si continui a β-ridurlo, il sottotermine $\lambda x_1 \dots x_n.x$ rimarrà immutato. Vale, infatti, la seguente proposizione.

Proposizione 2.11.4 *Ogni sequenza di β-riduzione che parte da un λ-termine in head normal form*

$$\lambda x_1 \dots x_n.xM_1 \cdots M_p$$

consiste di un intervallarsi di riduzioni fra loro indipendenti che si svolgono all'interno dei termini $M_1, \ldots, M_p$. Con questo intendiamo che se $\lambda x_1 \ldots x_n.xM_1 \cdots M_p \twoheadrightarrow_\beta P$ allora esistono $N_1, \ldots, N_p$ tali che

$$P \equiv \lambda x_1 \ldots x_n.xN_1 \cdots N_p$$

e per ogni $i \leq p$ vale $M_i \twoheadrightarrow_\beta N_i$.

Possiamo dire di aver estrapolato informazioni sulla *struttura* di tale termine sebbene il suo comportamento funzionale possa rimanere indefinito. Chi fosse interessato a una dimostrazione della proposizione precedente può consultare [Hindley and Seldin, 1986].

Una domanda che può venire spontanea a questo punto della trattazione è se vi sia un modo effettivo per trovare, nel qual caso esista, la head normal form di un termine. A questo scopo è necessario definire una relazione di riduzione opportuna.

Definizione 2.11.5

(i) Sia $M \in \Lambda$ un termine contenente Δ come head redex. Scriviamo

$$M \rightarrow_h N$$

se $M \rightarrow_\beta N$ e N è il risultato della contrazione di Δ in M. La relazione di riduzione $\rightarrow_h$ viene chiamata riduzione di testa a un passo*;*

(ii) al solito con $\twoheadrightarrow_h$ intendiamo la chiusura riflessiva e transitiva di $\rightarrow_h$;

(iii) il percorso di riduzione di testa *di M è la sequenza $M_0, M_1, \ldots$ univocamente determinata da:*

$$M \equiv M_0 \rightarrow_h M_1 \rightarrow_h \ldots$$

Nel qual caso si giunga ad un M_n in head normal form diremo che il percorso di riduzione di testa di M termina *con M_n, altrimenti diremo che M ha una riduzione di testa infinita.*

Teorema 2.11.6 *Sia $M \in \Lambda$, allora M ha una head normal form se e solo se il suo percorso di riduzione di testa termina.*

Questo teorema, di cui si può trovare una dimostrazione in [Barendregt, 1984, Corollario 11.4.8], ci dice che la proprietà per un termine di avere una head normal form è semidecidibile e ci fornisce una descrizione algoritmica di come procedere per trovare tale forma normale. Determinare se un termine *non* ha una head normal form è, invece, non semidecidibile.

Un'altra nozione intimamente legata a quella di termine in head normal form è quella di termine *solvable*.

Definizione 2.11.7

(i) Sia $M \in \Lambda^0$. Allora M si dice solvable *se*

$$\exists n \in \mathbb{N}, \exists N_1 \cdots N_n \in \Lambda \qquad MN_1 \cdots N_n = \mathbf{I};$$

(ii) Un arbitrario $M \in \Lambda$ è solvable se la chiusura di M, cioè $\lambda\vec{x}.M$ con $\{\vec{x}\} = FV(M)$, è solvable;

(iii) $M \in \Lambda$ è unsolvable *se non è solvable.*

È possibile dimostrare che un termine M è solvable se e solo se ha una head normal form: i due concetti sono, dunque, completamente equivalenti. Chi fosse interessato ad una dimostrazione può consultare [Barendregt, 1984, Teorema (Wadsworth) 8.3.14].

2.12 Alberi di Böhm

Abbiamo visto come la strategia di riduzione di testa, nel caso di termini solvable, riduca il termine fino ad ottenere una head normal form attraverso computazioni parziali che possono essere viste come approssimazioni di tale risultato. La situazione è analoga a quella dei numeri irrazionali che sono composti da un infinito numero di cifre dopo la virgola e di cui è possibile dare solo approssimazioni finite ovvero la sequenza iniziale dei numeri che li compongono. Un metodo per strutturare l'informazione può, quindi, essere quello di costruire un albero i cui nodi siano etichettati con le "teste" dei sottotermini che compongono il termine rappresentato. Queste etichette rappresentano un'informazione finita e stabile poiché non dipende da riduzioni future.

I termini unsolvable, sprovvisti cioè di head normal form, sono quindi totalmente privi di informazione e per questo vengono rappresentati con un simbolo indicante l'indefinito.

Definizione 2.12.1 *Sia $M \in \Lambda$ definiamo l'*albero di Böhm *di M che denotiamo con $BT(M)$ come segue:*

- *se M è unsolvable, allora $BT(M)$ è composto da un solo nodo foglia con etichetta $\perp$ (bottom), ovvero:*

$$BT(M) = \perp$$

- *se M è solvable e il suo percorso di riduzione di testa termina con $M \twoheadrightarrow_h \lambda x_1 \dots x_n.xM_1 \dots M_m$, con $m, n \geq 0$, allora la radice dell'albero di Böhm di M è etichettata con $\lambda x_1 \dots x_n.x$ e possiede m nodi figli:*

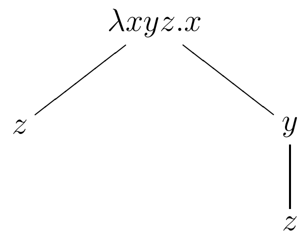
$$BT(M) = \begin{array}{ccc} & \lambda x_1 \dots x_n.x & \\ / & & \backslash \\ BT(M_1) & \cdots & BT(M_m) \end{array}$$

Nel seguito della trattazione indicheremo l'insieme di tutti gli alberi di Böhm con $\mathcal{BT}$.

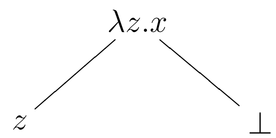
Si noti che questa definizione non è algoritmica in quanto la proprietà di essere unsolvable, per un termine, è non semidecidibile. Si osservi inoltre che se M ha $\lambda x_1 \dots x_n.xM_1 \dots M_m$ come head normal form, può accadere che M_i sia anche più complicato di M stesso. Ciò è equivalente ad affermare che la Definizione 2.12.1 non è una definizione induttiva, piuttosto è possibile dire che essa descrive un processo che può portare anche alla creazione di alberi infiniti.

Esempio 2.12.2

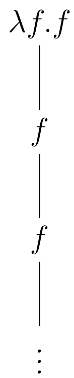
(i) $BT(\lambda xyz.xz(yz)) =$



(ii) $BT((\lambda xyz.xzy)x\Omega) =$



(iii) $BT(\lambda f.((\lambda x.f(xx))(\lambda x.f(xx)))) =$



È possibile dimostrare, per induzione sulla profondità dell'albero, che la definizione data degli alberi di Böhm è compatibile con la β-equivalenza, ovvero che vale la seguente proposizione.

Proposizione 2.12.3 *Siano $M, N \in \Lambda$, allora:*

$$M =_\beta N \Rightarrow BT(M) = BT(N)$$

Una dimostrazione di questa proposizione può essere trovata in [Barendregt, 1984, Proposizione 10.1.6]. Si noti che non vale l'opposto, ovvero esistono λ-termini che hanno lo stesso albero di Böhm ma *non* sono β-equivalenti, come si può constatare dall'esempio che segue.

Esempio 2.12.4 *Sia* $N \equiv \lambda yx.x(yz)$ *e* $M \equiv \boldsymbol{Y}N$. *Allora:*

$$
\begin{array}{rcll}
M & \equiv & \boldsymbol{Y}N & \textit{per def. di } M \\
 & =_\beta & N(\boldsymbol{Y}N) & \textit{per def. di } \boldsymbol{Y} \\
 & \equiv & (\lambda yx.x(yz))(\boldsymbol{Y}N) & \textit{per def. di } N \\
 & =_\beta & \lambda x.x((\boldsymbol{Y}N)z) & \textit{per } \beta\textit{-riduzione} \\
 & \equiv & \lambda x.x(Mz) & \textit{per def. di } M
\end{array}
$$

Quindi $M \twoheadrightarrow_h \lambda x.x(Mz)$. *Consideriamo ora i termini* $N' \equiv \lambda yx.x(yv)$ *e* $M' \equiv \boldsymbol{Y}N'$ *chiaramente è impossibile dimostrare che* $M =_\beta M'$ *in quanto* M *ed* M' *non hanno un comune ridotto, ciò nonostante gli alberi di Böhm di* M *e* M' *sono uguali.*

$$
BT(M) = \begin{array}{c} \lambda x.x \\ | \\ \lambda x.x \\ | \\ \lambda x.x \\ | \\ \vdots \end{array}
\qquad\qquad
BT(M') = \begin{array}{c} \lambda x.x \\ | \\ \lambda x.x \\ | \\ \lambda x.x \\ | \\ \vdots \end{array}
$$

Nel processo di costruzione dell'albero di Böhm le variabili z *e* v *che occorrono rispettivamente in* M *ed* M' *vengono "spinte" all'infinito.*

Da queste considerazioni si evince che, in generale, dato un λ-termine M vale che $FV(BT(M)) \subseteq FV(M)$. Ciò corrisponde ad osservare che la relazione di eguaglianza fra alberi di Böhm partiziona l'insieme dei λ-termini in classi più ampie rispetto alla β-equivalenza, ovvero, per ogni termine $N \in \Lambda$:

$$\{\ M \mid M =_\beta N\ \} \subseteq \{\ M \mid BT(M) = BT(N)\ \}.$$

2.12.1 Alberi di Böhm ricorsivamente enumerabili

Ci si può a questo punto domandare se per ogni albero di Böhm $t \in \mathcal{BT}$ esista un λ-termine $M \in \Lambda$ tale che $BT(M) = t$. Per poter rispondere abbiamo bisogno di fornire la nozione di *funzione che rappresenta un albero di Böhm* e di albero di Böhm *ricorsivamente enumerabile*.

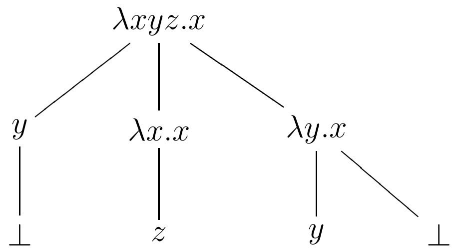
Definizione 2.12.5 *Sia $Node$ l'insieme delle sequenze di numeri naturali, si noti che possiamo associare induttivamente a tali sequenze i possibili nodi di un albero nel seguente modo:*

- *alla radice dell'albero corrisponde la sequenza vuota, che indicheremo con ϵ,*
- *al figlio $(n+1)$-esimo di un nodo rappresentato dalla sequenza α corrisponde la sequenza $\alpha * n$,*

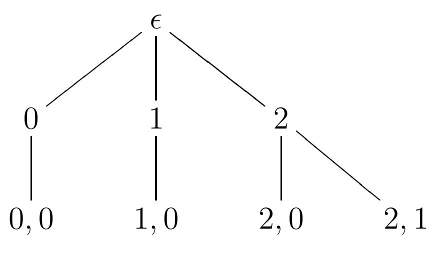
dove con $*$ intendiamo l'operazione di giustapposizione fra sequenze di numeri. Indicheremo con $parent : Node \to Node$ la funzione che associa ad ogni nodo α diverso dalla radice il padre di α e alla radice, la radice stessa. Si noti, quindi, che un albero viene univocamente determinato da un sottoinsieme T di $Node$ tale che $\epsilon \in T$ e $(\forall \alpha \in Node).(\alpha \in T \Rightarrow parent(\alpha) \in T)$.

Esempio 2.12.6 *Prendiamo, ad esempio, in considerazione l'albero di Böhm del λ-termine $M \equiv \lambda xyz.x(y\Omega)(\lambda x.xz)(\lambda y.xy\Omega)$, ovvero:*

$BT(M) =$



La sequenza di numeri naturali che assegnamo ad ogni nodo, concordemente con quanto detto nella definizione precedente, è



Definizione 2.12.7

- *Denoteremo con Label l'insieme delle possibili etichette degli alberi di Böhm, ovvero l'insieme delle stringhe della forma:*

$$\lambda x_1 \dots x_n.x$$

  con $n \in \mathbb{N}$ *e* $x_1, \dots, x_n, x \in V$ *(dove* V *ricordiamo essere l'alfabeto delle variabili del* λ*-calcolo).*

- *Diremo, inoltre, che una funzione* $f : Node \rightharpoonup_p Label \times \mathbb{N}$ rappresenta un albero di Böhm *se per ogni nodo* $\alpha \in Node$ *vale*

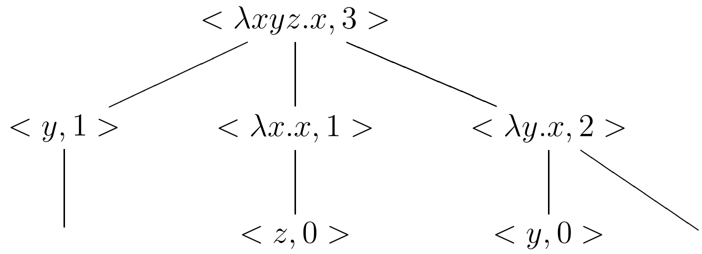
$$\begin{array}{lll} (f \downarrow \alpha) & \Rightarrow & (f \downarrow parent(\alpha)) \wedge (f(\alpha) = < \lambda x_1 \dots x_n.y, m >) \wedge \\ & & \wedge (f \uparrow (\alpha * k) \text{ per ogni } k \geq m.) \end{array}$$

  dove $\lambda x_1 \dots x_n.y$ *è l'etichetta del nodo* α *e* m *il numero di figli del nodo* α*.*

Esempio 2.12.8 *La funzione che rappresenta l'albero di Böhm del* λ*-termine* $M \equiv \lambda xyz.x(y\Omega)(\lambda x.xz)(\lambda y.xy\Omega)$*, ad esempio, è la seguente:*

$$f_M(\alpha) = \begin{cases} < \lambda xyz.x, 3 > & \text{se } \alpha = \epsilon \\ < y, 1 > & \text{se } \alpha = 0 \\ < \lambda x.x, 1 > & \text{se } \alpha = 1 \\ < \lambda y.x, 2 > & \text{se } \alpha = 2 \\ < z, 0 > & \text{se } \alpha = 1, 0 \\ < y, 0 > & \text{se } \alpha = 2, 0 \\ \uparrow & \text{altrimenti.} \end{cases}$$

ovvero la funzione f_M *etichetta l'albero dell'Esempio 2.12.6 come segue:*



Si noti che la funzione f_M è indefinita sia sulle sequenze che identificano nodi inesistenti dell'albero, sia su quelle in corrispondenza di $\perp$. Ciò non genera ambiguità perché siamo a conoscenza del numero di figli di ogni nodo in cui la funzione è definita.

Vediamo ora alcuni concetti che ci saranno utili per lavorare sugli alberi di Böhm e sulle funzioni che li rappresentano, come, ad esempio, la nozione di variabile libera e legata di un albero di Böhm.

Definizione 2.12.9 *Sia data una funzione f che rappresenta un albero di Böhm t.*

(i) Si consideri un nodo $\alpha \in Node$, se $f(\alpha) =< \lambda x_1 \ldots x_n.y, m >$, useremo questa notazione:

$$\vec{x}_\alpha = x_1, \ldots, x_n, \quad y_\alpha = y, \quad m_\alpha = m.$$

Se, invece, $f \uparrow \alpha$, allora $\vec{x}_\alpha = y_\alpha = m_\alpha =\uparrow$;

(ii) se $\alpha \in Node$, allora l'insieme delle variabili legate del percorso dalla radice ad α, che indicheremo con $\vec{x}_{\overline{\alpha}}$, viene definito induttivamente come segue:

$$\vec{x}_{\overline{\epsilon}} = \vec{x}_\epsilon, \quad \vec{x}_{\overline{\alpha * k}} = \vec{x}_{\overline{\alpha}}, \vec{x}_{\alpha * k}.$$

Ad esempio $\vec{x}_{\overline{1,0}} = \vec{x}_\epsilon, \vec{x}_1, \vec{x}_{1,0}$;

(iii) se $\alpha \in Node$, allora l'insieme delle variabili libere di un nodo α, che denoteremo con $FV_f(\alpha)$, viene definito come:

$$FV_f(\alpha) = \begin{cases} \{y_\alpha\} \setminus \{\vec{x}_{\overline{\alpha}}\} & \textit{se } f \downarrow \alpha \\ \emptyset & \textit{altrimenti;} \end{cases}$$

(iv) l'insieme delle variabili libere dell'albero t, che denoteremo con $FV(t)$, viene così definito:

$$FV(t) = \bigcup_{\alpha \in Node} FV_f(\alpha);$$

(v) t viene detto chiuso *se $FV(t) = \emptyset$.*

Possiamo finalmente dare una definizione formale di *albero di Böhm ricorsivamente enumerabile.*

Definizione 2.12.10 *Un albero di Böhm viene detto ricorsivamente enumerabile se la funzione che lo rappresenta (dopo aver codificato in modo effettivo dominio e codominio con numeri naturali) è calcolabile.*

D'ora in avanti denoteremo con $\mathcal{BT}^{re}$ l'insieme di tutti gli alberi di Böhm ricorsivamente enumerabili in cui occorre un numero *finito* di variabili libere. Possiamo quindi enunciare la seguente proposizione.

Proposizione 2.12.11 *Sia $t \in \mathcal{BT}^{re}$ allora esiste $M \in \Lambda$ tale che $BT(M) = t$ ed è possibile calcolare tale λ-termine in modo algoritmico.*

Dim. Sia f la funzione parziale calcolabile che rappresenta l'albero t e sia $FV(t) = \vec{a}$. Definiamo induttivamente $\vec{z_\alpha}$, al variare di $\alpha \in Node$, come segue:

$$\begin{array}{ll} \vec{z_\epsilon} & = \vec{a} \quad \text{se } f(\epsilon)\downarrow \\ & =\uparrow \quad \text{altrimenti.} \end{array}$$

$$\begin{array}{ll} \vec{z}_{\alpha * k} & = \vec{z}_\alpha, \vec{x}_\alpha \quad \text{se } f(\alpha)\downarrow \\ & =\uparrow \qquad \text{altrimenti.} \end{array}$$

Si noti che, per ogni $\alpha \in Node$ tale che $f \downarrow \alpha$, vale

$$\vec{z}_\alpha, \vec{x}_\alpha = \vec{a}, \vec{x}_\alpha = \vec{z}_{\alpha * k} \text{ per ogni } k \leq m_\alpha. \tag{2.1}$$

Per il Teorema 2.9.3 esiste un λ-termine $\boldsymbol{E} \in \Lambda^0$ tale che, se $M \in \Lambda^0$, $\boldsymbol{E}/M/ \twoheadrightarrow_\beta M$. Si può assumere che, se M è unsolvable, anche $\boldsymbol{E}/M/$ è unsolvable (questo è, ad esempio, vero per il combinatore costruito nel di [Barendregt, 1984, Teorema 8.1.6]). Se $f \downarrow \alpha$, definiamo

$$\begin{array}{rl} B_\alpha \equiv & \lambda m \vec{z}_\alpha \vec{x}_\alpha . y_\alpha \big(\boldsymbol{E} m/\alpha * 0/\vec{z}_{\alpha*0}\big) \cdots \\ & \cdots \big(\boldsymbol{E} m/\alpha * (m-1)/\vec{z}_{\alpha*(m-1)}\big). \end{array}$$

Si noti che, per la 2.1, si ha $B_\alpha \in \Lambda^0$ per ogni α tale che $f \downarrow \alpha$. Quindi f è una funzione parziale calcolabile, allora esiste una funzione ψ parziale e calcolabile tale che

$$\psi(\alpha) = \begin{cases} /B_\alpha/ & \text{se } f \downarrow \alpha \\ \uparrow & \text{altrimenti.} \end{cases}$$

Sia F il termine che λ-definisce ψ. Per il secondo teorema del punto fisso (Teorema 2.10.4), esiste un termine $M \in \Lambda^0$ tale che

$$M \twoheadrightarrow_\beta \lambda s.\boldsymbol{E}(Fs)/M/.$$

Se $\psi \downarrow \alpha$, allora

$$\begin{array}{ll} M/\alpha/\vec{z}_\alpha & \twoheadrightarrow_\beta \boldsymbol{E}(F/\alpha/)/M/\vec{z}_\alpha \\ & \twoheadrightarrow_\beta \boldsymbol{E}/B_\alpha//M/\vec{z}_\alpha \\ & \twoheadrightarrow_\beta B_\alpha/M/\vec{z}_\alpha \\ & \twoheadrightarrow_\beta \lambda\vec{x}_\alpha.y_\alpha\big(\boldsymbol{E}/M//\alpha * 0/\vec{z}_{\alpha*0}\big)\cdots \\ & \twoheadrightarrow_\beta \lambda\vec{x}_\alpha.y_\alpha\big(/M//\alpha * 0/\vec{z}_{\alpha*0}\big)\cdots \end{array} \tag{2.2}$$

Similarmente, se $f \uparrow \alpha$, allora $M/\alpha/\vec{z}_\alpha$ è unsolvable. Vogliamo dimostrare ora che, per ogni nodo α, si ha $BT(M/\alpha/\vec{z}_\alpha) = t_\alpha$, dove con t_α denotiamo il sottoalbero radicato nel nodo α. Dunque, se $f \uparrow \alpha$, si ha che t_α e $BT(M/\alpha/\vec{z}_\alpha)$ sono entrambi l'albero vuoto e quindi il termine $M/\alpha/\vec{z}_\alpha$ è unsolvable. Se $f \downarrow \alpha$, allora mostreremo, per induzione sulla lunghezza della sequenza β, che[7]

$$BT(M/\alpha/\vec{z}_\alpha)(\beta) = f(\alpha * \beta).$$

Se $\beta = \epsilon$, per la 2.2, si ha che

$$BT(M/\alpha/\vec{z}_\alpha)(\epsilon) =< \lambda\vec{x}_\alpha.y_\alpha, m_\alpha >= f(\alpha).$$

Supponiamo $\beta = k * \beta'$. Se $k < m_\alpha$, allora

$$\begin{array}{ll} BT(M/\alpha/\vec{z}_\alpha)(\beta) = & \\ = BT(M/\alpha * k/\vec{z}_{\alpha*k})(\beta') = & \text{per def. di } BT(\cdot) \text{ e per la 2.2,} \\ = f(\alpha * k * \beta') = & \text{per ipotesi induttiva} \\ = f(\alpha * \beta) & \end{array}$$

[7]Per brevità, nei passaggi che seguono, non faremo distinzione fra un albero di Böhm e la funzione che lo rappresenta.

Se $k \geq m_\alpha$, allora

$$BT(M/\epsilon/\vec{z}_\alpha) = t_\epsilon = t,$$

quindi esiste un λ-termine che ha come albero di Böhm t. Essendo tutte le operazioni eseguite nella dimostrazione per costruire tale termine effettive, è possibile scrivere un algoritmo che, preso in input un programma che calcola un albero di Böhm con un numero finito di variabili libere, restituisca in output un λ-termine che ha quell'albero. $\square$

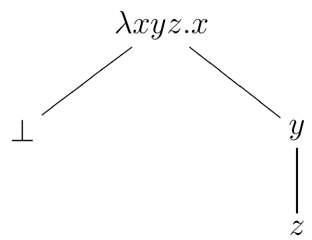
Questa proposizione è di grande utilità per i nostri scopi in quanto ci sarà indispensabile quando, nel Capitolo 5, riformuleremo i teoremi di Rice per gli alberi di Böhm.

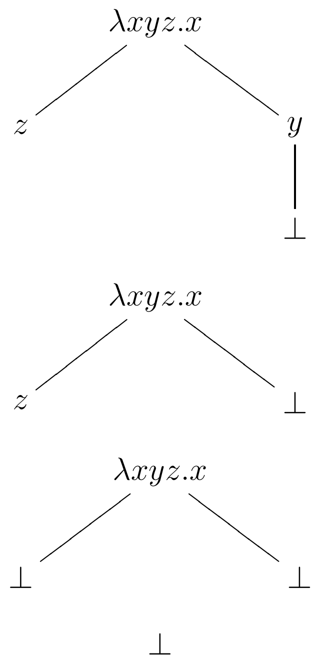
2.12.2 Approssimazioni

Alla luce dell'esistenza di alberi di Böhm infiniti è utile definire delle approssimazioni di questi ultimi ossia degli alberi finiti che abbiano un contenuto informativo ridotto rispetto all'albero che approssimano. Poiché l'informazione deve rimanere coerente, è necessario che la struttura dell'albero rimanga inalterata. Tramite l'eliminazione di un nodo (che diventa etichettato con $\perp$) e dell'eventuale sottoalbero in esso radicato è possibile creare un nuovo albero, consistente col precedente, che lo approssimi. Il simbolo $\perp$ assume ora due valenze: può rappresentare l'albero di un termine unsolvable e può indicare la mancata computazione di un sottoalbero.

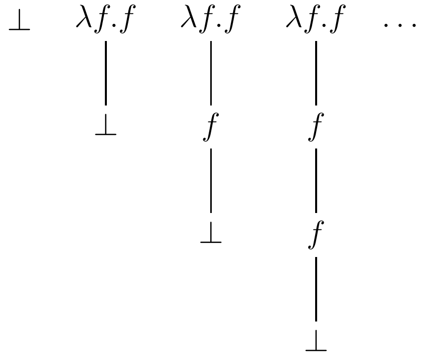
Seguono alcuni esempi.

Esempio 2.12.12 *(i) Le approssimazioni del combinatore $\boldsymbol{S} \equiv \lambda xyz.xz(yz)$ sono in numero finito; ciò deriva dal fatto che l'albero di Böhm di $\boldsymbol{S}$ è anch'esso finito.*





(ii) Le approssimazioni del termine $\boldsymbol{Y}$ *sono chiaramente infinite e si distinguono solo per il grado di profondità dell'albero:*



Dall'esempio precedente risulta evidente che un albero infinito può essere visto come il limite a cui tendono le sue approssimazioni finite. Si noti, inoltre, che esiste una ben determinata relazione fra un dato albero e un suo approssimante che può essere formalizzata come segue.

Definizione 2.12.13 *Siano t_1, t_2 due alberi di Böhm, allora $t_1 \sqsubseteq t_2$ se e solo se è possibile ottenere t_1 eliminando da B alcuni sottoalberi ed etichettando la radice dei sottoalberi eliminati con $\perp$. In questo caso ci riferiremo a t_1 come albero* approssimante *e a t_2 come* approssimato*.*

Definizione 2.12.14 *Sia $M \in \Lambda$, Definiamo l'approssimante di M come l'albero finito costruito nel seguente modo:*

- *se M non è in head normal form*

$$ABT(M) = \perp$$

- *se M è in head normal form, ovvero $M \equiv \lambda x_1 \dots x_n.xM_1 \dots M_n$, allora*

$$ABT(M) = \quad \begin{array}{ccc} & \lambda x_1 \dots x_n.x & \\ \swarrow & & \searrow \\ ABT(M_1) & \cdots & ABT(M_m) \end{array}$$

Si noti che la differenza fra l'albero di Böhm di un termine e il suo approssimante si basa sulla differenza fra *essere* in head normal form e *avere* una head normal form. In questo modo ad ogni termine corrisponde una ed una sola approssimazione e tale approssimazione è calcolabile in modo algoritmico. D'altra parte noi preferiamo che termini β-equivalenti siano caratterizzati dallo stesso insieme di approssimazioni, ne diamo dunque la seguente caratterizzazione.

Definizione 2.12.15 *L'insieme delle approssimazioni dell'albero di Böhm del termine M è*

$$\mathcal{ABT}(M) = \{ABT(N) \mid N =_\beta M\}.$$

Nell'esempio fatto in precedenza, le cinque approssimazioni di $BT(\boldsymbol{S})$ sono le approssimazioni dell'albero di cinque λ-termini. Essi devono contenere dei redex in posizione coerente con quella del simbolo $\perp$ dell'albero e devono essere β-equivalenti al termine $\boldsymbol{S}$. Ecco un esempio di cinque termini che soddisfano queste proprietà:

(i) $\lambda xyz.x(\boldsymbol{I}z)(yz)$,

(ii) $\lambda xyz.xz(y(\boldsymbol{I}z))$,

(iii) $\lambda xyz.xz(\boldsymbol{I}(yz))$,

(iv) $\lambda xyz.x\boldsymbol{I}z\boldsymbol{I}(yz)$,

(v) $\boldsymbol{IS}$.

Chiaramente ne esistono infiniti, per trovarne altri è sufficiente sostituire a $\boldsymbol{I}$ un termine nella forma $\boldsymbol{I}^n \equiv (\lambda x.x)(\lambda x.x)\dots(\lambda x.x)$ ovvero $\boldsymbol{I}$ applicato a se stesso n volte.

In questa tesi si è deciso di partire dalla definizione di albero di Böhm per un termine M e di introdurre in seguito il suo albero approssimante $ABT(M)$. È, in realtà, possibile definire l'albero di Böhm di un termine a partire dalla nozione di albero approssimante in questo modo:

$$BT(M) = \bigsqcup\{\ ABT(N) \mid M \twoheadrightarrow_\beta N\ \}$$

Questo è l'approccio utilizzato, ad esempio, in [Amadio and Curien, 1998]. In questo elaborato torneremo in seguito su questi concetti definendo formalmente e dimostrando l'esistenza di tale estremo superiore.

2.13 Le λ-teorie

Una teoria è un insieme chiuso di equazioni. Con *chiuso* intendiamo che ogni equazione che è conseguenza logica degli assiomi che definiscono la teoria fa parte della teoria stessa. Una teoria stabilisce i criteri in base ai quali due termini vengono considerati semanticamente uguali, ovvero dotati dello stesso comportamento operazionale.

2.13.1 La teoria $\lambda\beta$

La teoria più debole che può essere definita su Λ viene chiamata $\boldsymbol{\lambda\beta}$-teoria. In tale teoria si considerano indistinguibili tutti i termini che sono β-congruenti. Le altre teorie sono tutte estensioni del sistema assiomatico di $\boldsymbol{\lambda\beta}$.

Definizione 2.13.1 *I seguenti assiomi definiscono la* $\boldsymbol{\lambda\beta}$*-teoria:*

(i) $(\lambda x.M)\ N = M[x := N]$ $\quad\beta - congruenza;$

(ii) $(\lambda x.M) = (\lambda y.M)[x := y], con\ y \notin FV(M)$ $\quad\alpha - conversione;$

(iii)
$$\begin{array}{ll} M = M & prop.\ simmetrica; \\ M = N \Rightarrow N = M & prop.\ riflessiva; \\ M = N, N = P \Rightarrow M = P & prop.\ transitiva; \end{array}$$

(iv)
$$\begin{array}{ll} M = N \Rightarrow MP = NP & prop.\ applicazione\ 1; \\ M = N \Rightarrow PM = PN & prop.\ applicazione\ 2; \\ M = N \Rightarrow \lambda x.M = \lambda x.N & regola\ \xi. \end{array}$$

Definizione 2.13.2 *Due termini M e N si dicono* convertibili *e si scrive $\lambda\beta \vdash M = N$ se la loro eguaglianza è provabile a partire dagli assiomi della* $\boldsymbol{\lambda\beta}$*-teoria.*

Siano dati $M, N \in \Lambda$. Si noti che scrivere $\lambda\beta \vdash M = N$ corrisponde a dire che M è β-congruente a N ovvero $M =_\beta N$.

Un aspetto fondamentale delle teorie, che è utile per stabilirne la sensatezza, è il concetto di *consistenza*.

Definizione 2.13.3

(i) Un'equazione è una formula del tipo $M = N$ con $M, N \in \Lambda$ e viene detta chiusa *se $M, N \in \Lambda^0$;*

(ii) Una teoria $\mathcal{T}$ si dice consistente *e si scrive $Con(\mathcal{T})$ se esistono due termini la cui eguaglianza non può essere provata a partire dagli assiomi della teoria stessa;*

(iii) Una teoria $\mathcal{T}$ si dice inconsistente *se non è consistente;*

(iv) Sia $\mathcal{T}$ un insieme di equazioni. Allora si indica con $\lambda + \mathcal{T}$ la teoria ottenuta aggiungendo gli assiomi contenuti in $\mathcal{T}$ a quelli di $\boldsymbol{\lambda\beta}$*;*

(v) Si dice che $\mathcal{T}$ è consistente e si scrive $Con(\mathcal{T})$ se $Con(\lambda + \mathcal{T})$.

Proposizione 2.13.4 *La teoria $\boldsymbol{\lambda\beta}$ è consistente.*

L'idea intuitiva della dimostrazione di tale proposizione consiste nel considerare due termini $M, N \in \Lambda$ che ammettano forma β-normale distinta. Se tali termini fossero eguagliabili tramite gli assiomi della $\boldsymbol{\lambda\beta}$-teoria essi sarebbero β-equivalenti. Ma, poiché $\twoheadrightarrow_\beta$ gode della proprietà di Church-Rosser, esisterebbe un ridotto comune delle due forme β-normali e questo creerebbe una contraddizione.

Sia $\mathcal{T}$ un insieme di equazioni chiuse. Allora:

(i) $\mathcal{T}^+$ è l'insieme delle equazioni chiuse provabili in $\lambda + \mathcal{T}$;

(ii) $\mathcal{T}$ è una $\boldsymbol{\lambda}$-teoria se $\mathcal{T}$ è consistente e $\mathcal{T}^+ = \mathcal{T}$.

Ne deriviamo che ogni teoria è identificabile con l'insieme delle equazioni chiuse che sono provabili in essa. In genere, data una λ-teoria $\mathcal{T}$ denoteremo con $=_\mathcal{T}$ la relazione di eguaglianza fra λ-termini indotta da $\mathcal{T}$.

2.13.2 La teoria $\mathcal{H}$

Un classico esempio di λ-teoria è quella che eguaglia tutti i termini unsolvable:

$$\mathcal{H}_0 = \{M = N \mid M, N \in \Lambda^0, \text{unsolvable}\}$$

$$\mathcal{H} = \mathcal{H}_0^+.$$

Il motivo per il quale si è pensato di definire tale teoria è che tutti i termini unsolvable possono essere considerati privi di contenuto informativo. La teoria $\mathcal{H}$, quindi, non solo non crea problemi di incompatibilità fra λ-termini ma è molto utile per lo studio del λ-calcolo. Proprio in base al ruolo che rivestono i λ-termini unsolvable, infatti, viene data una classificazione delle λ-teorie.

Definizione 2.13.5 *Sia $\mathcal{T}$ una λ-teoria. Allora:*

(i) $\mathcal{T}$ si dice sensibile se $\mathcal{H} \subseteq \mathcal{T}$,

(ii) $\mathcal{T}$ *si dice semi-sensibile se* $=_{\mathcal{T}}$ *non eguaglia termini solvable e termini unsolvable.*

Il legame fra la nozione di teoria sensibile e semisensibile viene esplicitato dalla seguente proposizione.

Proposizione 2.13.6 *Sia* $\mathcal{T}$ *una* λ*-teoria. Allora:*

$$\mathcal{T} \text{ è sensibile} \Rightarrow \mathcal{T} \text{ è semi-sensibile.}$$

2.13.3 La teoria $\mathcal{B}$

La λ-teoria $\mathcal{B}$ è quella che eguaglia tutti i termini che hanno lo stesso albero di Böhm.

$$\mathcal{B} = \{M = N \mid (M, N \in \Lambda^0) \wedge (BT(M) = BT(N))\}$$

È possibile dimostrare che tale teoria è consistente e sensibile. Abbiamo infatti dimostrato nella Proposizione 2.12.3 che termini β-equivalenti hanno lo stesso albero di Böhm e, in particolare, che tutti i termini unsolvable sono associati tramite $BT(\cdot)$ all'albero $\perp$. Vale, inoltre, la seguente proposizione.

Proposizione 2.13.7

(i) $\mathcal{B} \vdash M = N \iff BT(M) = BT(N)$,

(ii) $\mathcal{B}$ *è una* λ*-teoria tale che* $\mathcal{H} \subseteq \mathcal{B}$.

Chi fosse interessato ad una dimostrazione può consultare la Proposizione 16.4.2 di [Barendregt, 1984].

2.13.4 La teoria $\lambda\eta$

Uno degli aspetti che caratterizzano il λ-calcolo è la sua non estensionalità (altresì detta *intensionalità*). In un formalismo estensionale due termini sono uguali se hanno lo stesso comportamento funzionale, cioè se applicando gli stessi argomenti si ottengono gli stessi risultati. Due termini, invece, si dicono uguali dal punto di vista intensionale se individuano lo stesso processo di computazione.

Consideriamo, ad esempio, un λ-termine M. Poiché il λ-calcolo è non estensionale, M e $\lambda x.Mx$ (con $x \notin FV(M)$) sono diversi anche se il loro comportamento funzionale è il medesimo. Definiamo allora la teoria $\boldsymbol{\lambda\eta}$ nella quale considereremo indistinguibili i λ-termini che hanno comportamento computazionale identico. Tale teoria sarà, dunque, estensionale.

Definizione 2.13.8

(i) Regola di estensionalità (denotata con $\boldsymbol{ext}$*):*

$$\forall M, N \in \Lambda \qquad Mx = Nx \;\Rightarrow\; M = N;$$

(ii) Denoteremo la teoria $\boldsymbol{\lambda\beta}$ *estesa con la regola* $\boldsymbol{ext}$ *con* $\boldsymbol{\lambda} + \boldsymbol{ext}$.

Per dotare il λ-calcolo della nozione di estensionalità è possibile, in alternativa, introdurre un nuovo assioma. La cosiddetta regola ξ che definisce la $\boldsymbol{\eta}$-conversione:

$$\forall M \in \Lambda \qquad \lambda x.Mx = M \quad (\text{con } x \notin FV(M)).$$

La teoria che aggiunge l'$\boldsymbol{\eta}$-conversione agli assiomi di $\boldsymbol{\lambda\beta}$ viene solitamente denotata con $\boldsymbol{\lambda\eta}$.

Teorema 2.13.9 *(di Curry) Le teorie* $\boldsymbol{\lambda} + \boldsymbol{ext}$ *e* $\boldsymbol{\lambda\eta}$ *sono equivalenti.*

Dim. Per prima cosa mostriamo che $\boldsymbol{\lambda} + \boldsymbol{ext} \vdash \boldsymbol{\eta}$. Effettivamente

$$(\lambda x.Mx)x = Mx$$

quindi, se $x \notin FV(M)$, per $\boldsymbol{ext}$

$$\lambda x.Mx = Mx.$$

Nel senso opposto si osservi che la teoria $\boldsymbol{\lambda\eta}$ è chiusa rispetto la regola $\boldsymbol{ext}$. Sia, dunque,

$$Mx = Nx$$

con $x \notin FV(MN)$. Allora, per la regola ξ:

$$\lambda x.Mx = \lambda x.Nx.$$

Quindi, per la $\boldsymbol{\eta}$ si ottiene $M = N$. $\square$

Si noti che la regola ξ gioca un ruolo essenziale nell'equivalenza fra $\boldsymbol{\lambda} + \boldsymbol{ext}$ e $\boldsymbol{\lambda\eta}$. Proprio per questo in letteratura ci si riferice alla regola ξ chiamandola *regola di estensionalità debole*.

Capitolo 3

Topologie

In questo capitolo introduciamo alcune delle nozioni che stanno alla base dello studio delle topologie tra cui il concetto stesso di spazio topologico. Illustreremo inoltre i principali metodi per generare una topologia, daremo alcuni esempi e introdurremo le prime proprietà degli spazi topologici. Le definizioni e i risultati qui enunciati sono necessari per la comprensione dei capitoli che seguiranno. Come testo di riferimento per gli argomenti di carattere topologico il lettore può consultare [Kelley, 1955] e [Bredon, 1993].

3.1 Spazi topologici

Le topologie vengono introdotte allo scopo di dare una formalizzazione del concetto intuitivo di "vicinanza" fra elementi di un dato insieme e nascono da proprietà metriche anche se, col tempo, si sono svincolate da esse. Uno spazio topologico è un insieme a cui è associata una collezione di sottoinsiemi chiamati *insiemi aperti*. Intuitivamente due elementi sono tanto più vicini fra loro quanti più sono gli insiemi aperti che li contengono entrambi.

Definizione 3.1.1 *Sia A un insieme arbitrario. Una* topologia τ *definita su A è una famiglia di sottoinsiemi di A che verifica le seguenti proprietà:*

(i) $\emptyset \in \tau$ *e* $A \in \tau$*,*

(ii) τ *è chiusa per intersezione finita,*

(iii) τ *è chiusa per unione arbitraria.*

Uno spazio topologico *è una coppia* (A, τ) *che denota un insieme* A *dotato di una topologia* τ*.*

Parleremo, in caso non sia fonte di ambiguità, di A come spazio topologico sottointendendo la topologia su esso definita. Gli elementi della topologia τ definita su A vengono chiamati *insiemi aperti* o *aperti* di A. Un sottoinsieme $C \subseteq A$ è detto *insieme chiuso* o *chiuso* se il suo complementare $A \setminus C$ è un aperto. A partire dalla Definizione 3.1.1 è possibile dare una prima caratterizzazione degli insiemi chiusi:

Proposizione 3.1.2 *La famiglia* σ *di chiusi di uno spazio topologico* (A, τ) *soddisfa le seguenti proprietà:*

(i) $\emptyset \in \sigma$ *e* $A \in \sigma$*,*

(ii) σ *è chiusa per unione finita,*

(iii) σ *è chiusa per intersezione arbitraria.*

Si noti che queste proprietà derivano naturalmente dalla Definizione 3.1.1 utilizzando le formule di De Morgan. È evidente che, data una famiglia di sottoinsiemi di A soddisfacente queste proprietà, la famiglia formata dai complementari di questi sottoinsiemi è una topologia su A. Abbiamo, ovvero, trovato un modo alternativo di definire una topologia specificandone gli insiemi chiusi.

Uno spazio topologico (A, τ) viene detto *non banale* se esistono due insiemi X e Y, con $\emptyset \subsetneq X,\ Y \subsetneq A$, tali che X è un aperto mentre Y non lo è. Negli esempi che seguono mostriamo le uniche topologie banali definibili su un insieme A.

Esempio 3.1.3 *Sia* A *un insieme arbitrario, allora l'insieme delle parti di* A*, denotato con* $\mathscr{P}(A)$*, è una topologia su* A*. Questa topologia viene chiamata* topologia discreta *ed è possibile definirla su qualsiasi insieme.*

Esempio 3.1.4 *È altresì possibile, dato un insieme A, definire quella che viene chiamata, in contrapposizione alla appena definita topologia discreta,* topologia indiscreta *ovvero la famiglia di insiemi contenente come unici elementi l'insieme vuoto e l'insieme stesso (solitamente chiamato* insieme universo*), cioè* $\{\emptyset, A\}$.

Si può intuitivamente essere indotti a pensare che questi siano degli esempi limite, ovvero che siano rispettivamente la più grande e la più piccola topologia definibile su un insieme. È lecito domandarsi se sia possibile dare una formalizzazione del concetto di "più grande" o "più piccolo" applicato alle topologie, se esista quindi una relazione di ordinamento fra topologie.

Definizione 3.1.5 *Una topologia τ_1 è detta* più grande *(risp.* piccola*) di una topologia τ_2 se ogni aperto di τ_2 (risp. τ_1) è anche un aperto di τ_1 (risp. τ_2) ovvero se $\tau_1 \supseteq \tau_2$ (risp. $\tau_1 \subseteq \tau_2$).*

Utilizzeremo la notazione $(A, \tau_1) \subseteq (A, \tau_2)$ per indicare che $\tau_1 \subseteq \tau_2$. Poiché le potenzialità di queste semplici nozioni possono non essere a tutti evidenti ad una prima lettura riportiamo in Appendice C una dimostrazione topologica di un teorema famoso nel campo della teoria elementare dei numeri.

3.2 Costruzione di Topologie

La definizione di spazio topologico è piuttosto astratta e non è molto semplice lavorare con essa. In generale, è difficile caratterizzare una topologia elencandone tutti gli insiemi aperti sebbene questo sia talvolta possibile (ad esempio per topologie definite su insiemi finiti). In pratica, usualmente, si è soliti costruire una topologia in modo differente ovvero specificandone le basi o eventualmente le sottobasi. Basi e sottobasi sono, dunque, concetti fondamentali nello studio degli spazi topologici ed è quindi opportuno darne subito una definizione. Per farlo è però necessario formalizzare cosa si intende per ricoprimento di un insieme.

Definizione 3.2.1 *Una famiglia σ di sottoinsiemi di un insieme A viene detta essere un* ricoprimento *di A se:*

$$(\forall x \in A)(\exists X \in \sigma).(x \in X).$$

Definizione 3.2.2 *Una famiglia $\mathcal{B}$ di sottoinsiemi di un insieme A viene detta* base topologica *se:*

(i) $\mathcal{B}$ è un ricoprimento di A;

(ii) $\forall x \in A$, $\forall B_1, B_2 \in \mathcal{B}$:

$$x \in B_1 \cap B_2 \Rightarrow \exists B_3 \in \mathcal{B}.((B_3 \subseteq B_1 \cap B_2) \wedge (x \in B_3)).$$

Una famiglia di sottoinsiemi di un insieme A viene detta una sottobase topologica *se verifica solo la condizione (i), ossia se l'unione dei suoi elementi è uguale ad A.*

Fornendo una descrizione in metalinguaggio di questa definizione possiamo dire che la base topologica dev'essere un ricoprimento dell'insieme A e che, se x appartiene all'intersezione fra due elementi della base, ne esiste un terzo, sottoinsieme di questi due, a sua volta contenente x. È, ad esempio, semplice da verificare che $\mathcal{B} = \big\{\{x\} \mid x \in X\big\}$ (ovvero la famiglia dei singoletti degli elementi di X) è una base della topologia discreta definita su un insieme X. È possibile, inoltre, convertire una sottobase in una base prendendo tutte le intersezioni finite dei suoi elementi. A questo punto sorge spontaneo chiedersi in che modo una famiglia di insiemi può generare una topologia e come siano correlati i concetti di base e spazio topologici.

Definizione 3.2.3 *Siano dati un insieme A e un sottoinsieme $\mathcal{C}$ di $\mathscr{P}(A)$, diremo che τ è la* topologia generata da $\mathcal{C}$ *se τ è la minima topologia contenente $\mathcal{C}$.*

Si osservi che tale topologia minima (rispetto all'ordinamento topologico introdotto nella Definizione 3.1.5) può essere trovata prendendo l'intersezione di tutte le topologie definibili sull'insieme A che contengano $\mathcal{C}$. È facilmente verificabile, infatti, che l'intersezione di una famiglia di topologie definite su un dato insieme è a sua volta una topologia ed è più piccola di qualunque altra appartenente a tale famiglia.

Teorema 3.2.4 *Sia $\mathcal{B}$ una famiglia di insiemi per cui valgano le proprietà (i) e (ii) della Definizione 3.2.2 di base topologica rispetto ad un insieme A, allora ogni aperto della topologia generata da $\mathcal{B}$ può essere scritto come unione di elementi di $\mathcal{B}$ (ovvero è la topologia che si ottiene chiudendo la famiglia $\mathcal{B}$ per unione arbitraria).*

Dim. Sia τ_{min} la topologia generata da $\mathcal{B}$, ovvero la minima topologia definibile su A contenente $\mathcal{B}$, costruiamo ora una famiglia di insiemi τ come segue:

$$\tau = \{O \subseteq A \mid O = \bigcup_{i \in I} \mathcal{A}_i \text{ per opportuni insiemi } \mathcal{A}_i \in \mathcal{B}\}.$$

Si noti che se proviamo che:

(a) τ è una topologia,

(b) $\mathcal{B} \subseteq \tau$,

(c) $\tau = \tau_{min}$,

abbiamo terminato la dimostrazione.
(a) Cominciamo col domandarci se τ è una topologia. La proprietà (i) della Definizione 3.1.1 di topologia è banale da verificare, così come la proprietà (iii), ed esse discendono dalla costruzione di τ. Per dimostrare che vale la proprietà (ii), invece, è necessario provare che

$$\Big(\bigcup_{i \in I} B_i\Big) \cap \Big(\bigcup_{j \in J} C_j\Big) \tag{3.1}$$

dove B_i, C_j, al variare di i e j, sono elementi della base, è un insieme aperto. Supponiamo che un elemento x appartenga a tale insieme, allora esisteranno due indici k e ℓ tali che $x \in B_k$ e $x \in C_\ell$. Dunque

$$x \in B_k \cap C_\ell$$

e, per la proprietà (ii) della Definizione 3.2.2, esiste un insieme $D_{k\ell} \in \mathcal{B}$ tale che $x \in D_{k\ell}$ e

$$D_{k\ell} \subseteq B_k \cap C_\ell \subseteq \Big(\bigcup_{i \in I} B_i\Big) \cap \Big(\bigcup_{j \in J} C_j\Big).$$

Un discorso analogo può essere fatto per ogni elemento dell'insieme 3.1 che, quindi, è un aperto in quanto unione degli aperti $D_{k\ell}$ trovati in questo modo.
(b) Il fatto che $\mathcal{B} \subseteq \tau$ deriva direttamente dalla costruzione di τ.
(c) Chiaramente vale $\tau_{min} \subseteq \tau$ per definizione di topologia minima, per dimostrare che $\tau \subseteq \tau_{min}$ dobbiamo provare che per ogni topologia τ' contenente $\mathcal{B}$ vale $\tau \subseteq \tau'$ e questo è semplice da verificare in quanto se τ' contiene $\mathcal{B}$ ed è chiuso per unione arbitraria deve anche contenere l'unione arbitraria degli elementi di $\mathcal{B}$. $\square$

Si noti che mentre una base caratterizza in modo univoco una topologia, una topologia può avere più basi. Conoscere una base di una topologia, quindi, significa sapere da quali aperti è composta la topologia stessa. Per questo, come vedremo in seguito, individuare una base si rivela utile in molti contesti. Si noti inoltre che, poiché ogni aperto di una topologia (A, τ) può essere scritto come unione di elementi della base $\mathcal{B}$ e l'intero insieme A è un aperto, la base si rivela essere un ricoprimento dello spazio topologico, ossia:

$$\bigcup_{B \in \mathcal{B}} B = A$$

non vale invece il contrario, ovvero non ogni ricoprimento di A, pur essendo una sottobase, è anche una base topologica. Il concetto di sottobase torna utile per definire l'*unione* fra due topologie.

Definizione 3.2.5 *Siano dati due spazi topologici* (X, τ_1) *e* (X, τ_2). *Chiameremo* unione topologica *di* τ_1 *e* τ_2 *lo spazio* (X, τ), *dove* τ *è la topologia generata dalla sottobase* $\tau_1 \cup \tau_2$.

Tutti i differenti metodi che abbiamo visto e che vedremo per costruire una topologia vanno presi in considerazione dal momento che ogni prospettiva comporta proprietà differenti.

3.3 Intorni

Ci accingiamo ora a definire la nozione di *intorno* che vedremo essere intimamente correlata a quella di aperto.

Definizione 3.3.1 *Sia X uno spazio topologico e sia U un sottoinsieme di X. Un sottoinsieme $S \subseteq X$ si dice* intorno *di U se e solo se esiste un insieme aperto A contenuto in S e contenente U, ossia tale che valga la relazione $U \subseteq A \subseteq S$.*

Quindi un insieme è un intorno di un punto $x \in X$ se esso è intorno dell'insieme $\{x\}$. Da questa definizione ne scaturisce, chiaramente, che un aperto è intorno di ogni suo elemento e, viceversa, se un insieme è intorno di ogni suo punto esso è un aperto. Quest'ultimo enunciato è facilmente verificabile, si noti infatti che se per ogni elemento $x \in S$ vale che S è intorno di x, allora deve esistere un aperto $A_x \subseteq S$ contenente x e l'unione degli aperti $\bigcup_{x \in S} A_x$ è proprio S che è a sua volta un aperto (in quanto unione di aperti). Possiamo quindi enunciare il seguente teorema.

Teorema 3.3.2 *Sia X uno spazio topologico e S un sottoinsieme di X. Allora:*

$$S \text{ è aperto} \iff S \text{ è intorno di ogni suo punto.}$$

3.4 Chiusura topologica

La chiusura di un insieme è un'operazione che ha a che fare sia con la nozione di intorno che con quella di chiuso ed è interessante in quanto consente di dare una caratterizzazione alternativa degli insiemi chiusi.

Definizione 3.4.1 *Sia A uno spazio topologico.*

(i) Definiamo la chiusura *di un sottoinsieme $U \subseteq A$, e la denotiamo con $\overline{U}$, come il più piccolo insieme chiuso tale che $U \subseteq \overline{U} \subseteq A$;*

(ii) sia $a \in A$. Un insieme I viene chiamato intorno *di a in A se esiste un insieme aperto U tale che $a \in U \subseteq I$.*

Si noti che $a \in \overline{U}$ se $U \cap V \neq \emptyset$ per ogni intorno aperto V di a. *Par abus de langage* d'ora in poi scriveremo $\overline{b}$ per indicare la chiusura del singoletto contenente b ovvero dell'insieme $\{b\}$. Vediamo ora alcuni esempi.

Esempio 3.4.2

1. *Sia D uno spazio dotato della topologia indiscreta, allora per ogni sottoinsieme $S \subseteq D$ non vuoto vale $\overline{S} = D$. Viceversa, se per ogni $S \neq \emptyset$ si ha che $\overline{S} = D$ allora la topologia definita su D è quella indiscreta.*

2. *Uno spazio D è dotato della topologia discreta se e solo se per ogni sottoinsieme S vale $\overline{S} = S$.*

Grazie all'operatore di chiusura, come già accennato in precedenza, è possibile fornire una caratterizzazione positiva dei chiusi di una topologia; in particolare valgono i seguenti risultati.

Proposizione 3.4.3 *Sia X uno spazio topologico. Un sottoinsieme $S \subseteq X$ è chiuso se e solo se $S = \overline{S}$ ed $\overline{S}$ è il più piccolo insieme chiuso contenente S.*

Le proprietà che caratterizzano l'operatore di chiusura sono il fatto che sia crescente, ovvero $S \subseteq \overline{S}$, il fatto che sia finitamente additivo in quanto $\overline{\emptyset} = \emptyset$ e $(\overline{S \cup T}) = \overline{S} \cup \overline{T}$, e la sua idempotenza, ossia il fatto che $\overline{\overline{S}} = \overline{S}$. Vale, inoltre, il seguente teorema.

Teorema 3.4.4 *Sia dato un insieme X. Per ogni sottoinsieme $S \subseteq X$ sia dato il sottoinsieme $\overline{S}$ in modo tale che siano rispettate le seguenti proprietà:*

(i) $S \subseteq \overline{S}$,

(ii) $\overline{\emptyset} = \emptyset$,

(iii) $(\overline{S \cup T}) = \overline{S} \cup \overline{T}$,

(iv) $\overline{\overline{S}} = \overline{S}$,

allora esiste una e una sola topologia nella quale l'insieme $\overline{S}$ è la chiusura di S.

3.5 Funzioni Continue

Le funzioni continue definite fra spazi topologici sono di rilevante interesse per lo studio delle topologie. In via del tutto informale è possibile pensare alle funzioni continue come trasformazioni che associano elementi vicini ad elementi vicini senza "strappi". In questa sezione ci proponiamo di darne una definizione formale e di studiarne le proprietà.

Definizione 3.5.1 *Siano X e Y spazi topologici e $f : X \to Y$ una funzione. f viene detta* continua *se l'immagine inversa di un insieme aperto è a sua volta un aperto, ovvero se $f^{-1}[U]$ è un insieme aperto di X per ogni U insieme aperto di Y.*

Seguendo questa definizione la continuità di una funzione coincide con la proprietà della sua inversa[1] di preservare gli insiemi aperti. Si noti che per dimostrare la continuità di una funzione è sufficiente verificare che la proprietà richiesta nella definizione appena introdotta valga per gli elementi di una base della topologia. È possibile definire la nozione di continuità anche in termini di intorno, di insieme chiuso o rispetto a un dato elemento, come si può vedere dalle seguenti proposizioni.

Proposizione 3.5.2 *Una funzione $f : X \to Y$ fra spazi topologici è continua se e solo se per ogni intorno U di $f(x)$ esiste un intorno V di x tale che $f(V) \subseteq U$.*

Proposizione 3.5.3 *Una funzione $f : X \to Y$ fra spazi topologici è continua se e solo se l'immagine inversa di un chiuso di Y è, a sua volta, un chiuso di X.*

Proposizione 3.5.4 *Sia $f : X \to Y$ una funzione fra spazi topologici e x un elemento di X, allora f è continua in x se e solo se per ogni intorno U di $f(x)$ esiste un intorno V di x tale che $f(V) \subseteq U$.*

Diremo nell'ultimo caso che f è continua se lo è rispetto ad ogni $x \in X$. Consideriamo ora alcuni esempi di funzioni continue.

---

[1] Con *inversa* in questo testo intendiamo la funzione f^{-1} tale che $f^{-1}[U] = \{x \mid f(x) \in U\}$. Non è necessario quindi fare assunzioni sulla biiettività di f.

Esempio 3.5.5

- *Sia X un insieme arbitrario dotato della topologia discreta, allora qualsiasi funzione $f : X \to X$ è continua,*
- *sia X un insieme arbitrario dotato della topologia discreta. Una funzione $f : X \to Y$ è continua se per ogni aperto A di Y vale*
$$A \cap f(X) \neq \emptyset \Rightarrow f(X) \subseteq A.$$

Un corollario sulla continuità evidente ma piuttosto importante è il seguente.

Corollario 3.5.6 *Se f è una funzione continua da X a Y e g è una funzione continua da Y a Z allora la composizione di f e g, ovvero $f \circ g : X \to Z$, è a sua volta continua.*

Dim. Bisogna dimostrare che se U è un intorno di $(f \circ g)(x) = g(f(x))$ in Z allora $(f \circ g)^{-1}[U] = f^{-1}[g^{-1}[U]]$ è un intorno di x in X. Ciò è vero in quanto la continuità di g in $f(x)$ implica che $g^{-1}[U]$ è un intorno di $f(x)$ in Y e la continuità di f implica che $f^{-1}[g^{-1}[U]]$ è un intorno di x. $\square$

Come abbiamo visto, l'inversa di una funzione continua $f : X \to Y$ preserva gli insiemi aperti, questo significa che anche se ampliamo la topologia definita sull'insieme X con altri aperti o se eliminiamo degli aperti dalla topologia di Y essa resta continua. Vale, dunque, la seguente proposizione.

Proposizione 3.5.7 *Sia $f : X \to Y$ una funzione continua tra due spazi topologici X e Y, essa rimane continua se si sostituisce la topologia di X con una più grande e quella di Y con una più piccola.*

Dim. Siano τ_X e τ'_X due topologie per l'insieme X con $\tau_X \subseteq \tau'_X$ e siano τ_Y e τ'_Y due topologie per Y con $\tau'_Y \subseteq \tau_Y$. Sia, inoltre, $f : (X, \tau_X) \to (Y, \tau_Y)$ una funzione continua fra X e Y allora per ogni insieme aperto A di (Y, τ_Y) si ha che $f^{-1}[A]$ è un insieme aperto di (X, τ_X).

Consideriamo ora un insieme aperto A di (Y, τ'_Y) allora A è anche un aperto di (Y, τ_Y), da ciò segue che l'insieme $f^{-1}[A]$ è un aperto di (X, τ_X) dunque $f^{-1}[A]$ è anche un aperto di (X, τ'_X). $\square$

3.6 Preordine di specializzazione

Una topologia su un insieme A induce sui suoi elementi un preordine, ovvero una relazione riflessiva e transitiva, che viene chiamato *preordine di specializzazione*. Lo studio di tale preordine fornisce varie informazioni sulle caratteristiche della topologia, sull'organizzazione della famiglia di aperti, sugli intorni in essa presenti e sulle funzioni continue definibili. È anche possibile studiare le topologie a partire da un dato ordine di specializzazione.

Definizione 3.6.1 *(preordine di specializzazione)*
Sia (A, τ) uno spazio topologico, allora esso induce un preordine $\leq_\tau$ *su A definito da:*

$$a \leq_\tau b \iff \forall O \in \tau.(a \in O \Rightarrow b \in O).$$

In altre parole, $a \leq_\tau b$ se ogni intorno di a è anche un intorno di b.

Si noti che $\leq_\tau$ rispetta la definizione di preordine essendo riflessiva e transitiva. A partire da questa definizione è facile comprendere perché si parla di specializzazione, infatti l'elemento b è più specializzato dell'elemento a poiché è contenuto in più insiemi aperti. Questo è particolarmente intuitivo se si pensa all'appartenenza agli insiemi aperti come una proprietà che gli elementi possono o non possono avere. È semplice da verificare che la definizione data di preordine di specializzazione può essere riformulata ne seguente modo:

$$a \leq_\tau b \iff a \in \overline{b} \iff \overline{a} \subseteq \overline{b}.$$

Nella prossima sottosezione introdurremo alcune definizioni di tipi di spazi topologici che vedremo essere collegati all'ordine di specializzazione.

Proprietà degli Spazi Topologici

Poiché ci saranno utili nel futuro enunceremo ora alcune proprietà degli spazi topologici. I seguenti assiomi vengono chiamati in letteratura *assiomi di separazione*.

Definizione 3.6.2 *Uno spazio A viene detto:*

(i) T_0 se e solo se per ogni $a, b \in A$ esiste un aperto O tale che $O \cap \{a, b\}$ è un singoletto,

(ii) T_1 se ogni sottoinsieme con un singolo elemento è chiuso,

(iii) T_2 o Hausdorff *se, per ogni $a, b \in A$, esistono due insiemi aperti U e V tali che $a \in U, b \in V$ e $U \cap V = \emptyset$,*

(iv) $T_{2_{1/2}}$ o completamente Hausdorff *se, per ogni a, b elementi distinti di A, esistono due insiemi aperti U e V con $a \in U, b \in V$ e $\overline{U} \cap \overline{V} = \emptyset$.*

Il nome di tali assiomi deriva dal fatto che essi impongono un diverso grado di separazione fra gli intorni di due punti distinti dello spazio topologico.

Si noti che ogni funzione continua fra spazi T_0 è necessariamente monotona. Il grado di separazione di uno spazio è, inoltre, collegato con l'ordine di specializzazione. Infatti è dimostrabile che uno spazio è:

- T_0 se ha come preordine di specializzazione un ordine parziale,
- T_1 se il suo preordine di specializzazione è discreto[2].

Risulta evidente sia dalle definizioni date che dal tipo di ordine indotto che ogni spazio T_1 è anche T_0. I precedenti assiomi possono essere riformulati per coppie di elementi, ad esempio, a e b vengono detti $T_{2_{1/2}}$-separabili se esistono due insiemi aperti U e V tali che $a \in U, b \in V$ e $\overline{U} \cap \overline{V} = \emptyset$. In modo analogo possono essere definite le nozioni di T_2, T_1, T_0-separabilità.

Definizione 3.6.3 *Sia (A, τ) uno spazio topologico.*

(i) Un sottoinsieme di A viene detto clopen *se è, al contempo, un aperto e un chiuso della topologia;*

[2]Ovvero per ogni $a, b \in A$ si ha che $a \leq_\tau b \iff a = b$.

(ii) uno spazio (A, τ) si dice connesso *se $\emptyset, A$ sono gli unici clopen di tale spazio;*

(iii) un sottoinsieme Y di A si dice connesso *se Y, considerato come sottospazio di A, è connesso;*

(iv) la componente connessa *di un elemento $a \in A$ è il più grande sottoinsieme connesso di A contenente a;*

(v) uno spazio (A, τ) viene detto iperconnesso *se per ogni U, V insiemi aperti non vuoti vale $U \cap V \neq \emptyset$;*

Le componenti connesse di uno spazio (A, τ) formano una partizione di A.

3.6.1 Topologia debole

La nozione di *topologia debole* si rivela, per noi informatici teorici, di particolare utilità essendo legata alla teoria delle numerazioni precomplete (si veda per un approfondimento [Ershov, 1999]). Nel caso della teoria classica della computabilità, ad esempio, si è soliti utilizzare una funzione totale che associ ad ogni numero naturale $n \in \mathbb{N}$ la funzione ϕ_n, ovvero la funzione computata dall'n-esimo programma della macchina di Turing. Come vedremo nel Capitolo 5, quando andremo a definire una topologia τ sull'insieme delle funzioni parziali calcolabili, essa "produrrà", nel modo che vedremo ora, una topologia su $\mathbb{N}$ individuata dalla famiglia di aperti $\{\phi^{-1}[A] \mid A \in \tau\}$ al variare di $x \in \mathbb{N}$. Questa non è altro che un'istanza della definizione di topologia debole.

Definizione 3.6.4 *Siano X, Y due insiemi e $f : X \to Y$ una funzione. Se τ_Y è una topologia su Y, allora il sottoinsieme τ di $\mathscr{P}(X)$ definito da:*

$$\tau = \{f^{-1}[A] \mid A \in \tau_Y \}$$

ovvero, per essere più formali,

$$\tau = \{U \mid (\exists A \in \tau_Y) \wedge (f^{-1}[A] = U)\}$$

è la più piccola topologia su X per cui f è continua. La topologia τ così definita viene detta topologia debole *rispetto ad f.*

La definizione può essere estesa in modo naturale ad una intera famiglia di funzioni, a patto che il dominio sia lo stesso. Per dare un'idea dell'importanza della topologia debole si consideri il fatto che solitamente, se non viene specificato altrimenti, è convenzione, dati due spazi topologici definiti su due insiemi X e Y, pensare alla *topologia prodotto* su $X \times Y$ come la topologia debole rispetto alle proiezioni canoniche[3] su X ed Y. È possibile dimostrare che la topologia debole è la più piccola topologia che rende continua la funzione f. La definizione di topologia debole non necessita che la funzione f citata sia suriettiva ma nei principali casi di studio e in particolare in quelli che considereremo in questo testo essa lo sarà sempre. Vale, inoltre, il seguente lemma.

Lemma 3.6.5 *Siano A e B due insiemi arbitrari, $\mathcal{G}$ una base topologica su B e sia $f : A \to B$ una funzione totale. Allora la famiglia*

$$\mathcal{F} = \{U \mid (\exists V \in \mathcal{G}).(f^{-1}[V] = U)$$

è una base della topologia debole rispetto ad f definita su A.

Dim. Per dimostrare questo lemma dobbiamo mostrare che

(a) $\mathcal{F}$ è una base topologica su A,

(b) la topologia generata da $\mathcal{F}$ è la topologia debole rispetto ad f.

(a) Che $\mathcal{F}$ sia un ricoprimento di A deriva direttamente dal fatto che f è totale e quindi vale $f^{-1}[B] = A$. Siano dati ora $x \in A$, e $B_1, B_2 \in \mathcal{F}$ vogliamo dimostrare che $x \in B_1 \cap B_2 \Rightarrow \exists B_3 \in \mathcal{F}$ tale che $x \in B_3 \subseteq B_1 \cap B_2$. Per definizione di $\mathcal{F}$ se $x \in B_1 \cap B_2$ esistono $V_1, V_2 \in \mathcal{G}$ tali che $f^{-1}[V_1] = B_1$ e $f^{-1}[V_2] = B_2$, quindi $f(x) \in V_1 \cap V_2$. Allora, poiché $\mathcal{G}$ è una base topologica, esiste $V_3 \in \mathcal{G}$ tale che $V_3 \subseteq V_1 \cap V_2$ con $f(x) \in V_3$ e questo implica che $x \in f^{-1}[V_3] \in \mathcal{F}$.

[3]Con *proiezioni canoniche* intendiamo le due funzioni $pr_X : X \times Y \to X$ e $pr_Y : X \times Y \to Y$ definite da:

$$pr_X(x, y) = x, \qquad pr_Y(x, y) = y.$$

(***b***) Chiamiamo τ_A la topologia generata da $\mathcal{F}$, τ_B la topologia generata da $\mathcal{G}$ e τ la topologia debole su A rispetto ad f. Vogliamo dimostrare $\tau_A = \tau$. Che la topologia debole τ contenga $\mathcal{F}$ è facilmente dimostrabile e deriva dal fatto che contiene tutte le antimmagini degli aperti di τ_B quindi, in particolare, di quelli della base $\mathcal{G}$. Ora, per terminare la dimostrazione, ci è sufficiente osservare che ogni aperto di τ è anche aperto di τ_A. Sia, dunque, $O \in \tau$, allora esiste $V \in \tau_B$ tale che $O = f^{-1}[V]$ ma essendo V un aperto di τ_B è possibile scriverlo come unione arbitraria di elementi di $\mathcal{G}$ ovvero $V = \bigcup_{i \in I} B_i$ per opportuni $B_i \in \mathcal{G}$. Quindi $O = f^{-1}[\bigcup_{i \in I} B_i] = \bigcup_{i \in I} f^{-1}[B_i]$ e poiché al variare di B_i vale, per definizione di $\mathcal{F}$, $f^{-1}[B_i] \in \mathcal{F}$ è possibile scrivere O come unione arbitraria di elementi di $\mathcal{F}$. $\square$

Un altro concetto legato a quello di topologia debole è quello di topologia indotta. La topologia indotta viene utilizzata quando si ha una topologia τ definita su un insieme A e si è interessati a restingere la topologia ad un sottoinsieme di A.

Definizione 3.6.6 *Sia dato uno spazio topologico* (A, τ) *e un sottoinsieme* X *di* A*. Allora la famiglia*

$$\tau' = \{S \mid (S \subseteq X) \wedge (\exists B \in \tau).(S = B \cap X)\}$$

viene detta topologia indotta da τ *su* X*.*

È facilmente verificabile che la topologia indotta da τ su X coincide con la topologia debole rispetto a $\iota : X \to A$, dove ι è la proiezione canonica.

Capitolo 4

Topologie e Ordinamenti

Le topologie sono uno strumento essenziale per trattare determinati aspetti dell'informatica teorica. Per contro, le problematiche derivanti dallo studio dell'informatica teorica hanno fornito nuovi e interessanti stimoli per i topologi. Queste problematiche hanno, inoltre, incrementato l'interazione fra le topologie e alcune branche della matematica quali la teoria degli ordinamenti e l'algebra topologica. In questo capitolo verranno prese in considerazione alcune delle più importanti topologie che, ormai da tempo, vengono applicate allo studio del λ-calcolo e della teoria classica della computabilità. Definiremo inoltre, in quanto correlati a tali topologie, gli insiemi parzialmente ordinati e i reticoli, concetti che ci torneranno utili nel seguito della trattazione.

4.1 Ordinamenti Parziali

In informatica teorica ordinamenti e topologie costituiscono gli strumenti di base per la *teoria dei domini*. Lo studio della teoria dei domini è cominciata alla fine degli anni sessanta quando Scott (si veda ad esempio [Scott, 1972]) ha trovato i primi modelli matematici per il λ-calcolo non tipato fornendo in questo modo, tra le altre cose, uno strumento per l'analisi semantica dei linguaggi di programmazione a paradigma funzionale. Un altro concetto fondamentale della teoria dei domini è la nozione di *approssimazione*. L'idea principale è quella di pensare gli oggetti "infiniti" di un poset come gli estremi superiori delle loro approssimazioni finite. Per questo vengono solitamente considerati cpo algebrici in cui ogni elemento può essere approssimato essenzialmente dai suoi elementi più piccoli. Vediamo, quindi, alcune definizioni.

Definizione 4.1.1 *(Ordini parziali, cpo e dcpo)*

(i) Un ordine parziale *su un insieme D è una relazione binaria $\leq\ \subseteq D \times D$ riflessiva, antisimmetrica e transitiva. Un insieme con un ordinamento parziale viene detto* insieme parzialmente ordinato *(*poset *in breve). In seguito diremo impropriamente che $(D, \leq)$ è un ordine parziale intendendo che D è un insieme parzialmente ordinato con la relazione $\leq$;*

(ii) dato un ordine parziale $(D, \leq)$, un sottoinsieme $S \subseteq D$ si dice diretto *se è non vuoto e:*

$$(\forall x, y \in S)(\exists z \in S).((x \leq z) \wedge (y \leq z)).$$

Come notazione utilizzeremo $\Delta \subseteq_{dir} D$ per indicare che l'insieme Δ è un sottoinsieme diretto di D (ometteremo "dir" a pedice in caso sia chiaro dal contesto);

(iii) un ordine parziale completo diretto*, in breve dcpo, è un ordine parziale $(D, \leq)$ in cui ogni sottoinsieme diretto S di D ha un estremo superiore (che denoteremo con $\bigvee S$);*

(iv) un ordine parziale completo*, in breve cpo, è un dcpo $(D, \leq)$ in cui esiste un elemento minimo che indicheremo con $\perp$ (bottom), ovvero un elemento tale che $(\forall x \in D).(\perp \leq x)$.*

Si noti che nella nostra definizione un insieme diretto deve essere non vuoto. Sebbene dovrebbe far parte delle preconoscenze di ogni matematico che si avventuri nella lettura di una tesi ricordiamo che un elemento $x \in D$ è detto *estremo superiore* o *supremum* di A se e solo se è il minimo dei maggioranti di A, dove a sua volta con *maggiorante* intendiamo un elemento $m \in A$ tale che valga

$(\forall y \in A).(m \leq y)$. Un semplice esempio di cpo è l'insieme delle parti $\mathscr{P}(X)$ di un insieme arbitrario X ordinato tramite la relazione di inclusione insiemistica; in questo caso l'elemento minimo è l'insieme vuoto.

Definizione 4.1.2 *Sia $(A, \leq)$ un insieme parzialmente ordinato. Due elementi a e b di A si dicono:*

(i) comparabili *se $a \leq b$ oppure $b \leq a$,*

(ii) compatibili *se hanno un maggiorante, cioè, se esiste un elemento z tale che $a \leq z$ e $b \leq z$.*

Si noti che gli insiemi diretti che abbiamo introdotto precedentemente sono, per definizione, composti da elementi compatibili. Porremo ora la nostra attenzione sugli elementi compatti che sono utilizzati per modellare la nozione di informazione finita.

Definizione 4.1.3 *(compatto) Sia $(D, \leq)$ un cpo. Un elemento $x \in D$ si dice* compatto *(o* finito*) se e solo se per ogni insieme diretto $X \subseteq_{dir} D$ vale*

$$x \leq \bigvee X \Rightarrow x \leq x_0 \text{ per qualche } x_0 \in X.$$

Come vedremo in seguito gli elementi compatti assumono un ruolo di fondamentale importanza nella definizione della topologia di Scott. Tramite essi è infatti possibile dare una caratterizzazione di una base di tale topologia. Uno degli ultimi concetti che introdurremo in questa sottosezione è quello di cpo algebrico, altresì detto *dominio*.

Definizione 4.1.4 *(cpo algebrico) Un cpo $(D, \leq)$ viene detto* algebrico *se per ogni elemento $x \in D$ l'insieme $\{y \mid (y \leq x) \wedge (y \text{ compatto})\}$ è diretto e*

$$x = \bigvee \{y \mid (y \leq x) \wedge (y \text{ compatto})\}.$$

Come già accennato, quindi, in un cpo algebrico ogni elemento può essere visto in tal modo come l'estremo superiore delle sue approssimanti finite.

Definizione 4.1.5 *(funzione monotona, cpo continua)*

(i) Siano $(D, \leq)$ e $(D', \leq)$ degli ordini parziali. Una funzione $f : D \to D'$ viene chiamata monotona *se:*

$$(\forall x, y \in D).(x \leq y \Rightarrow f(x) \leq f(y)).$$

(ii) Siano $(D, \leq)$ e $(D', \leq)$ due dcpo, una funzione $f : D \to D'$ viene detta cpo continua *se è monotona e preserva gli estremi superiori degli insiemi diretti ovvero:*

$$(\forall \Delta \subseteq_{dir} D).\Big(f(\bigvee \Delta) = \bigvee f(\Delta)\Big).$$

Si osservi che una funzione monotona trasforma insiemi diretti in insiemi diretti. In letteratura, solitamente, le funzioni *cpo continue* vengono semplicemente dette continue. In questo testo si è preferito battezzarle come "cpo continue" per evitare di confonderle con le funzioni topologicamente continue definite nella Sezione 3.5 a pagina 61. Un'importante conseguenza di questa definizione è che ogni funzione continua definita da un cpo a sè stesso ammette un minimo punto fisso.

Definizione 4.1.6 *Sia $(D, \leq)$ un cpo e $f : D \to D$ una funzione cpo continua allora un elemento $d \in D$ viene detto* punto fisso *di f se $f(d) = d$.*

Il seguente teorema, semplice ma importante, ci fornisce esplicitamente un metodo per trovare il minimo punto fisso di una funzione cpo continua f definita su un cpo D.

Teorema 4.1.7 *(Teorema del punto fisso) Siano $(D, \leq)$ un cpo e $f : D \to D$ una funzione cpo continua. Definiamo*

$$fix(f) = \bigvee_{n \in \mathbb{N}} f^n(\bot)$$

allora:

(i) $f(fix(f)) = fix(f)$ *e*

(ii) se $f(d) \leq d$ *allora* $fix(f) \leq d$.

ovvero $fix(f)$ *è il minimo punto fisso di* f.

Dim. *(i)* Per la continuità di f vale

$$\begin{array}{rcl} f(fix(f)) & = & f(\bigvee_{n\in\mathbb{N}} f^n(\perp)) \\ & = & \bigvee_{n\in\mathbb{N}} f^{n+1}(\perp) \\ & = & (\bigvee_{n\in\mathbb{N}} f^{n+1}(\perp)) \vee \{\perp\} \\ & = & \bigvee_{n\in\mathbb{N}} f^n(\perp) \\ & = & fix(f). \end{array}$$

quindi $fix(f)$ è un punto fisso.
(ii) Supponiamo d sia un elemento tale che $f(d) \leq d$. Certamente $\perp \leq d$. Per la monotonicità di f vale $f(\perp) \leq f(d)$ ma poiché $f(d) \leq d$ varrà anche $f(\perp) \leq d$ e, per induzione su n, $f^n(\perp) \leq d$. Quindi $fix(f) = \bigvee_{n\in\mathbb{N}} f^n(\perp) \leq d$ ovvero $fix(f)$ è il minimo punto fisso di f. $\square$

4.1.1 Esempio: Il cpo delle funzioni parziali

Un esempio di ordine parziale completo importante per i nostri scopi è quello definito sull'insieme di tutte le funzioni parziali fra due insiemi arbitrari. Consideriamo l'insieme di tutte le funzioni parziali definite da un insieme X a un insieme Y ovvero:

$$\{ \ f \mid f : X \rightharpoonup_p Y \ \}$$

dotato della seguente relazione $\leq$ di ordinamento:

$$f \leq g \iff (\forall x \in X).(f \downarrow x \Rightarrow ((g \downarrow x) \wedge (f(x) = g(x))))$$

dove ($\downarrow$ significa "è definita"). Si noti che, intuitivamente, $f \leq g$ significa che il grafico di g *estende* quello di f. Volendo essere più formali possiamo affermare che $f \leq g \iff graph(f) \subseteq graph(g)$ dove $graph(f) = \{ \ (x,y) \mid (f \downarrow x) \wedge (f(x) = y) \ \}$.

Proposizione 4.1.8 *Siano dati due insiemi arbitrari X e Y. La coppia $(\{ \ f \mid f : X \rightharpoonup_p Y \ \}, \leq)$ è un ordine parziale completo algebrico.*

Questa proposizione è facilmente verificabile. Si noti infatti che esiste un elemento minimo, la funzione vuota, che denoteremo con $f_\emptyset$, ovvero quella funzione tale che $Dom(f_\emptyset) = \emptyset$. Le funzioni con dominio finito sono gli elementi compatti del cpo; tale cpo è, inoltre, algebrico in quanto ogni funzione parziale è l'estremo superiore delle sue approssimazioni finite, ovvero per ogni f vale:

$$f = \bigvee \{\theta \mid (\theta \leq f) \wedge (\theta \text{ compatto}) \ \}.$$

Si osservi anche che le funzioni totali fungono da elementi massimali del cpo, cioè:

$$(f \leq g) \wedge (Dom(f) = X) \Rightarrow f = g.$$

4.1.2 Esempio: Il cpo degli alberi di Böhm

È possibile dimostrare che l'insieme $\mathcal{BT}$ dotato della relazione di ordinamento $\sqsubseteq$, precedentemente introdotta nella Definizione 2.12.13 a pagina 46, individua anch'esso un cpo algebrico.

Proposizione 4.1.9 $(\mathcal{BT}, \sqsubseteq)$ *è un ordine parziale completo algebrico.*

Anche in questo caso la proposizione precedente è facilmente verificabile. Si noti che l'insieme $\mathcal{BT}$ è fornito di un elemento minimo, ovvero l'albero $\bot$ e che gli alberi finiti fungono da elementi compatti. Ogni insieme diretto ammette un *supremum*, come si può vedere dall'esempio che segue:

$$\bigsqcup\left\{\begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ \bot & & y\end{array}, \begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ x & & y\end{array}\right\} = \begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ x & & y\end{array}.$$

Anche questo cpo è algebrico, infatti ogni albero può essere pensato come estremo superiore dei suoi sottoalberi finiti. Ovvero:

$$BT(M) = \bigsqcup\{\ BT(N) \mid (BT(N) \sqsubseteq BT(M)) \wedge (BT(N) \text{ compatto})\}.$$

Vediamo, ad esempio, il caso dell'albero del λ-termine $\lambda xy.x(xy)$:

$$\begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ x & & y\end{array} = \bigsqcup\left\{\begin{array}{c} \bot, \begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ \bot & & \bot\end{array}, \begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ \bot & & y\end{array}, \\ \begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ x & & \bot\end{array}, \begin{array}{ccc} & \lambda xy.x & \\ / & & \backslash \\ x & & y\end{array} \end{array}\right\}.$$

Gli elementi massimali del cpo sono gli alberi che non contengono $\bot$ come sottoalbero.

4.2 Reticoli

La teoria dei reticoli viene naturalmente coinvolta in varie aree della matematica, incluse la teoria delle categorie, la topologia generale, la teoria computazionale e l'algebra topologica e analitica. Per approfondire lo studio della teoria dei reticoli si consiglia la lettura di [Burris and Sankappanavar, 1981] e [Gierz et al., 1980] che viene da molti considerato il testo di riferimento in questo campo. Ci sono due metodi *standard* per definire i reticoli: il primo permette di darne la caratterizzazione algebrica come si è soliti fare con i gruppi o con gli anelli, il secondo è invece basato sulla nozione di ordinamento.

Definizione 4.2.1 *(reticolo) Un insieme non vuoto L con due operazioni binarie $\wedge$ e $\vee$ (che vengono chiamate rispettivamente* "join" *e* "meet"*) su L viene detto* reticolo *se soddisfa le seguenti proprietà:*

$$
\begin{array}{lll l}
L1: & (a) & x \vee y \approx y \vee x & \\
 & (b) & x \wedge y \approx y \wedge x & \textit{(commutatività)} \\
L2: & (a) & x \vee (y \vee z) \approx (x \vee y) \vee z & \\
 & (b) & x \wedge (y \wedge z) \approx (x \wedge y) \wedge z & \textit{(associatività)} \\
L3: & (a) & x \vee x \approx x & \\
 & (b) & x \wedge x \approx x & \textit{(idempotenza)} \\
L4: & (a) & x \approx x \vee (x \wedge y) & \\
 & (b) & x \approx x \wedge (x \vee y) & \textit{(assorbimento)}
\end{array}
$$

La classe di reticoli che, negli anni, è stata studiata in modo più esaustivo è sicuramente quella dei reticoli distributivi.

Definizione 4.2.2 *(reticolo distributivo) Un* reticolo distributivo *è un reticolo che soddisfa una (e quindi, come vedremo, entrambe) di queste due proprietà:*

$$
\begin{array}{llcl}
D1: & x \wedge (y \vee z) & \approx & (x \wedge y) \vee (x \wedge z) \\
D2: & x \vee (y \wedge z) & \approx & (x \vee y) \wedge (x \vee z)
\end{array}
$$

Come già accennato durante la definizione precedente la proprietà $D1$ viene soddisfatta se e solo se lo è anche $D2$. Supponiamo infatti che valga $D1$. Allora

$$\begin{array}{rcll} x \vee (y \wedge z) & \approx & (x \vee (x \wedge z)) \vee (y \wedge z) & \text{(per L4(a))} \\ & \approx & x \vee ((x \wedge z) \vee (y \wedge z)) & \text{(per L2(a))} \\ & \approx & x \vee ((z \wedge x) \vee (z \wedge y)) & \text{(per L1(b) due volte)} \\ & \approx & x \vee (z \wedge (x \vee y)) & \text{(per D1)} \\ & \approx & x \vee ((x \vee y) \wedge z) & \text{(per L1(b))} \\ & \approx & (x \wedge (x \vee y)) \vee ((x \vee y) \wedge z) & \text{(per L4(b))} \\ & \approx & ((x \vee y) \wedge x) \vee ((x \vee y) \wedge z) & \text{(per L1(b))} \\ & \approx & (x \vee y) \wedge (x \vee z) & \text{(per D1)} \end{array}$$

Quindi vale anche $D2$. Una dimostrazione analoga può essere fatta per dimostrare il *vice versa*. Possiamo ora fornire una definizione alternativa di reticolo.

Definizione 4.2.3 *(reticolo II) Un insieme parzialmente ordinato (poset) L è un* reticolo *se e solo se per ogni elemento $a, b \in L$ esistono $\bigvee\{a, b\}$ e $\bigwedge\{a, b\}$ in L.*

Ci accingiamo ora a dimostrare che le due definizioni date di reticolo sono equivalenti nel senso che, se si costruisce un reticolo L che soddisfa una delle due definizioni, è sempre possibile costruire sullo stesso insieme L, in modo semplice e generale, un reticolo soddisfacente l'altra definizione; le due costruzioni sono invertibili.

Per prima cosa procediamo col descrivere le due costruzioni:

(a) se L è un reticolo per la prima definizione, allora definiamo la relazione $\leq$ su L in questo modo:

$$a \leq b \iff a = a \wedge b;$$

(b) se L è un reticolo per la seconda definizione, allora definiamo le operazioni $\vee$ e $\wedge$ come

$$\begin{array}{rcl} a \vee b & = & \bigvee\{a, b\} \\ a \wedge b & = & \bigwedge\{a, b\}. \end{array}$$

È facile verificare come queste due costruzioni siano invertibili. Ciò nonostante in seguito useremo la parola “reticolo” intendendo un insieme che soddisfa la Definizione 4.2.3 pur essendo spesso conveniente pensare alla corrispondente struttura con le due operazioni *join* e *meet*.

Definizione 4.2.4 *Un reticolo viene detto* limitato *o* bounded *se possiede un massimo e un minimo.*

L'elemento massimo viene spesso indicato con **1** mentre il minimo con **0**. Se x è un elemento del reticolo limitato allora ogni elemento y che soddisfa le proprietà $x \wedge y = \mathbf{0}$ e $x \vee y = \mathbf{1}$ viene detto il *complemento* di x. Un reticolo limitato in cui ogni elemento ha un, non necessariamente unico, complemento viene chiamato *reticolo complementato*.

4.3 Topologie: casi di studio

In questa sezione considereremo le più importanti topologie che vengono applicate allo studio del λ-calcolo e della teoria classica della computabilità e le proprietà topologiche ad esse correlate.

4.3.1 Topologia di Alexandroff

La prima topologia che vedremo è quella di Alexandroff ed è definibile sugli insiemi parzialmente ordinati che sono stati introdotti precedentemente nella Definizione 4.1.1. Essa è una delle più semplici topologie definibili su un insieme parzialmente ordinato poiché gli aperti vengono costruiti in modo tale da rispettare l'ordine imposto dalla relazione di ordinamento del poset.

Definizione 4.3.1 *Sia $(D, \leq)$ un poset. Il sottoinsieme $O \subseteq D$ è un aperto della* topologia di Alexandroff*, che indicheremo con τ_A se:*

$$(\forall y \in D).((x \in O) \wedge (x \leq y) \Rightarrow y \in O)$$

Si osservi che questa topologia è ben definita in quanto, banalmente, D e $\emptyset$ appartengono a τ_A che, inoltre, è chiusa per unione arbitraria e intersezione arbitraria. Una possibile base per la topologia τ_A, che denoteremo con $\mathcal{B}_{\tau_A}$, è data dalla seguente famiglia di insiemi:

$$\mathcal{B}_{\tau_A} = \{x\!\uparrow \ \mid x \in D\}$$

dove con $x\!\uparrow$, che chiameremo informalmente chiusura "verso l'alto" di x, denotiamo l'insieme

$$x\!\uparrow = \{y \mid (y \in D) \wedge (x \leq y)\}.$$

Poiché ogni aperto di τ_A è chiuso verso l'alto esso deve essere il risultato dell'unione di insiemi del tipo $x\!\uparrow$. L'insieme $x\!\uparrow$ risulta, quindi, essere il più piccolo aperto contenente x.

Anche le definizioni dei chiusi e delle funzioni continue sono altrettanto semplici, infatti i chiusi, essendo complementari degli aperti, sono gli insiemi chiusi verso il basso[1] e una funzione $f : D \to D$ è continua rispetto alla topologia di Alexandroff se rispetta l'ordinamento indotto da $\leq$ nel senso che dati due elementi $d_0, d_1 \in D$ si ha che $d_0 \leq d_1 \Rightarrow f(d_0) \leq f(d_1)$, ossia se f è *monotona*.

4.3.2 Topologia di Scott

La topologia di Scott è definibile sui cpo che abbiamo introdotto nella Definizione 4.1.1. Vediamo ora in che modo è possibile dare una caratterizzazione generale degli aperti di tale topologia.

Definizione 4.3.2 *Sia $(D, \leq)$ un cpo, il sottoinsieme $O \subseteq D$ è un aperto della* topologia di Scott*, che denoteremo con τ_S, se:*

(i) $(\forall y \in D).((x \in O) \wedge (x \leq y) \Rightarrow y \in O)$

(ii) $\bigvee X \in O,\ \ con\ X \subseteq_{dir} D \Rightarrow X \cap O \neq \emptyset$

[1] Analogamente agli insiemi chiusi verso l'alto intendiamo per insieme chiuso verso il basso quell'insieme $x\!\downarrow$ definito come $x\!\downarrow = \{y \mid (y \in D) \wedge (y \leq x)\}$.

È facile verificare che la famiglia così definita è chiusa per unione arbitraria e $\emptyset$ e D sono aperti di tale famiglia. Dati due aperti O_1 e O_2, inoltre, l'insieme $O_1 \cap O_2$ è un aperto in quanto, nella condizione (ii) della definizione precedente, si richiede che X sia diretto. Quindi la Definizione 4.3.2 individua effettivamente una topologia. Si noti che la definizione di aperto di Scott differisce da quella di aperto di Alexandroff solo per la seconda condizione. La seconda condizione appare più chiara se si pensa all'estremo superiore dell'insieme X come al limite a cui tendono gli elementi di X. Essa, infatti, stabilisce che se $\bigvee X$ appartiene ad un aperto O allora almeno una sua approssimazione deve appartenervi. Si osservi che questa condizione assume una valenza significativa solo quando si lavora su cpo infiniti; nel caso di cpo finiti, infatti, essa è banalmente verificata poiché l'estremo superiore di un insieme finito e diretto appartiene sempre all'insieme stesso e, dunque, la topologia di Scott e quella di Alexandroff coincidono.

A causa della seconda condizione, in generale, la base per τ_S non può essere la famiglia di tutti gli insiemi chiusi verso l'alto come invece accadeva per quella di Alexandroff. Gli insiemi $x\!\uparrow$, infatti, non formano una base per la topologia di Scott e non sono necessariamente neppure degli aperti della topologia a meno che l'elemento x non sia compatto. In questo caso, in effetti, l'insieme $x\!\uparrow$ soddisfa banalmente la prima condizione della Definizione 4.3.2 e, se supponiamo che $\bigvee X \in x\!\uparrow$, con X insieme diretto, allora, poiché x compatto, deve esistere un x_0 appartenente sia ad X che a $x\!\uparrow$ (e quindi all'intersezione fra i due) come richiesto dalla seconda condizione. Queste considerazioni ci permettono di enunciare la seguente proposizione.

Proposizione 4.3.3 *Sia D un cpo su cui è stata definita la topologia di Scott. Se $x \in D$ è compatto, allora l'insieme $x\!\uparrow$ è un aperto di tale topologia.*

Per individuare una base della topologia di Scott dobbiamo vincolare ancora di più il dominio su cui la topologia viene definita imponendo che esso sia un cpo algebrico. Se la topologia τ_S viene definita su un cpo algebrico, infatti, possiamo notare che per ogni elemento $x \in A$, dove A è un insieme aperto, si ha che $x = \bigvee\{y \mid (y \leq x) \wedge (y \text{ compatto})\}$ e quindi esiste un certo y_0 compatto che appartiene ad A tale che $x \in y_0\!\uparrow\subseteq A$. Vale ovvero la seguente proposizione.

Proposizione 4.3.4 *Sia D un ordine parziale completo algebrico. La famiglia di insiemi*

$$\mathcal{B} = \{x\!\uparrow\ \mid (x \in D) \wedge (x \textit{ compatto})\}$$

è una base della topologia di Scott.

Dim. Gli insiemi $d\!\uparrow$, con d elemento compatto, sono aperti di Scott per definizione di spazio compatto. Dobbiamo mostrare che se $d\!\uparrow \cap d'\!\uparrow \neq \emptyset$, allora $d''\!\uparrow\subseteq d\!\uparrow \cap d'\!\uparrow$, per qualche d'', ovvero che $d \leq d''$ e $d' \leq d''$. Sia $x \in d\!\uparrow \cap d'\!\uparrow$, cioè $d, d' \in \{e \mid (e \leq x) \wedge (e \text{ compatto})\}$. L'elemento d'' esiste in quanto tale insieme è diretto. Dobbiamo, inoltre, mostrare che se U è un aperto e $x \in U$, allora $x \in d\!\uparrow\subseteq U$ per un qualche elemento d. Questo segue banalmente dalla definizione di insieme aperto e algebricità. $\square$

Si noti che la base di Scott appena definita è chiusa per intersezione finita nel momento in cui l'estremo superiore di due elementi compatti e compatibili del cpo è, a sua volta, un elemento compatto. Gli ordini parziali completi per cui questa proprietà è verificata vengono chiamati in letteratura *domini di Scott*. Un risultato importante che si ottiene grazie alla topologia di Scott riguarda le funzioni continue.

Proposizione 4.3.5 *Siano D, D' cpo e $f : D \to D'$. Allora*

$$f \textit{ è Scott continua} \iff (\forall X \subseteq_{dir} D).f\Big(\bigvee X\Big) = \bigvee f(X).$$

Formulata in modo diverso questa proprietà stabilisce che una funzione continua rispetto alla topologia di Scott definita sul rispettivo poset è una funzione che preserva gli estremi superiori degli insiemi diretti. Si noti inoltre che le funzioni Scott continue sono, in particolare, monotone.

Considerazioni sugli Ordini di Separazione

Enunciamo ora alcune proprietà inerenti l'ordine di separazione degli spazi dotati della topologia di Scott o quella di Alexandroff.

Proposizione 4.3.6 *Sia* $(D, \leq)$ *un insieme parzialmente ordinato, allora:*

(i) la famiglia degli insiemi aperti di Scott forma una topologia T_0 *su* D*;*

(ii) se $x, y \in D$ *ed esiste un qualche aperto contenente* x *ma non contenente* y*, allora* $x \not\leq y$*.*

Dim. Sia D un insieme parzialmente ordinato. Chiaramente l'unione degli insiemi chiusi verso l'alto di D è a sua volta un insieme chiuso verso l'alto. Se inoltre $D' \subseteq_{dir} D$ e $\bigvee D' \in \bigcup_{i \in I} U_i$ dove U_i sono insiemi aperti al variare di i in un qualche insieme di indici I allora esiste un $k \in I$ tale che $\bigvee D' \in U_k$. Quindi U_k è un aperto di Scott, ne segue che $D' \cap U_k \neq \emptyset$ e quindi lo stesso vale anche per $D' \cap \left(\bigcup_{i \in I} U_i\right)$.

Se $x \not\leq y \in D$ allora la definizione di aperto di Scott implica che $y\downarrow$ è un chiuso di Scott. Dal fatto che $x \notin y\downarrow$ ne deriva $x \in D \setminus y\downarrow$ che è un aperto di Scott, quindi la topologia di Scott è T_0. Questo dimostra il punto (i).

Per il punto (ii) si noti che se $x, y \in D$ e $x \in U$ allora $x \leq y$ e $U = U\uparrow$ implica che $y \in U$ come volevamo. Quindi $y \notin U$ deve implicare $x \not\leq y$. $\square$

Una proposizione che segue direttamente dal punto (ii) della precedente ed è importante da tener presente è la seguente.

Proposizione 4.3.7 *Sia* $(D, \leq)$ *un insieme parzialmente ordinato, allora le seguenti proposizioni sono inoltre equivalenti:*

(a) la topologia di Scott è T_1*,*

(b) D *è dotato dell'ordine discreto,*

(c) la topologia di Scott è T_2*,*

(d) la topologia di Scott è discreta.

4.3.3 Topologia di Ershov e di Visser

Per poter definire la topologia di Ershov e quella di Visser è conveniente introdurre preliminarmente i reticoli limitati distributivi su un dato insieme e un operatore di complemento definito sui reticoli.

Definizione 4.3.8 *Un* reticolo limitato distributivo *definito su A è una famiglia $\mathcal{F}$ di sottoinsiemi di A che verifica le seguenti proprietà:*

(i) $\emptyset \in \mathcal{F}$ *e* $A \in \mathcal{F}$,

(ii) $\mathcal{F}$ *è chiusa per intersezione finita,*

(iii) $\mathcal{F}$ *è chiusa per unione finita.*

Una famiglia $\mathcal{F}$ di insiemi che ha queste proprietà rispetto un dato insieme A viene chiamata reticolo limitato e distributivo su A in quanto $(\mathcal{F}, \subseteq)$ è un poset che verifica la Definizione 4.2.3 di reticolo dove l'unione insiemistica svolge il ruolo di $\bigvee$ e l'intersezione di $\bigwedge$. Tale reticolo è *limitato* poiché possiede un elemento minimo, ovvero $\emptyset$, e un massimo, l'insieme A, ed è *distributivo* in quanto valgono le proprietà $D1$ e $D2$ della Definizione 4.2.2. Chiaramente ogni spazio topologico (A, τ) è anche un reticolo limitato distributivo su A.

Definizione 4.3.9 *Sia dato un reticolo limitato distributivo $\mathcal{F}$ definito su un insieme A, definiamo un operatore unario di complemento che indicheremo con $-$ come segue:*

$$-\mathcal{F} = \{B \subseteq A \mid B^C \in \mathcal{F}\}.$$

Dove con B^C intendiamo il complemento dell'insieme B rispetto all'insieme universo (in questo caso A), ovvero $A \setminus B$. Si noti, inoltre, che vale il seguente lemma.

Lemma 4.3.10 *Sia $\mathcal{F}$ un reticolo limitato distributivo, allora $-\mathcal{F}$ è, a sua volta, un reticolo limitato distributivo.*

Dim. La dimostrazione segue direttamente dalle regole di De Morgan, ovvero dal fatto che, per ogni $A, B \in -\mathcal{F}$, vale $A \cup B = (A^C \cap B^C)^C$ e $A \cap B = (A^C \cup B^C)^C$. $\square$

Un altro risultato che ci tornerà utile sui reticoli limitati e distributivi è il seguente.

Lemma 4.3.11 *Siano A e B due insiemi arbitrari, $\mathcal{F}$ un reticolo limitato distributivo su A e $f : A \to B$ una funzione totale. Allora la famiglia*

$$f[\mathcal{F}] = \{U \mid (U \subseteq B) \wedge (f^{-1}[U] \in \mathcal{F})\}$$

è un reticolo limitato distributivo su B.

Dim. Si noti che, poiché la funzione f è totale vale $f^{-1}[B] = A \in \mathcal{F}$, quindi $B \in f[\mathcal{F}]$ e che $\emptyset \in f[\mathcal{F}]$ in quanto $\emptyset \subseteq B$ e $f^{-1}[\emptyset] = \emptyset \in \mathcal{F}$. La famiglia $f[\mathcal{F}]$ è chiusa per intersezione finita in quanto due insiemi $U', U'' \in f[\mathcal{F}]$ se $f^{-1}[U'] \in \mathcal{F}$ e $f^{-1}[U''] \in \mathcal{F}$ ma allora, poiché $\mathcal{F}$ è chiusa per intersezione finita anche $f^{-1}[U'] \cap f^{-1}[U''] \in \mathcal{F}$ e $f^{-1}[U'] \cap f^{-1}[U''] = f^{-1}[U' \cap U''] \in \mathcal{F}$. La famiglia $f[\mathcal{F}]$ è chiusa per unione finita in quanto due insiemi $U', U'' \in f[\mathcal{F}]$ se $f^{-1}[U'] \in \mathcal{F}$ e $f^{-1}[U''] \in \mathcal{F}$ ma allora, poiché $\mathcal{F}$ è chiusa per unione finita anche $f^{-1}[U'] \cup f^{-1}[U''] \in \mathcal{F}$ e $f^{-1}[U'] \cup f^{-1}[U''] = f^{-1}[U' \cup U''] \in \mathcal{F}$. $\square$

A questo punto possiamo finalmente enunciare la definizione della topologia di Ershov e di quella di Visser.

Definizione 4.3.12 *(Topologie di Ershov e di Visser) Sia $\mathcal{F}$ un reticolo limitato distributivo definito su un insieme A.*

(i) La $\mathcal{F}$-topologia di Ershov *sull'insieme A è definita come la famiglia di tutti i sottoinsiemi U di A soddisfacente la seguente proprietà:*

per ogni $a \in U$ esiste $V \subseteq U$ tale che $a \in V \in \mathcal{F}$.

(ii) La $\mathcal{F}$-topologia di Visser *sull'insieme A è definita come la famiglia di tutti i sottoinsiemi U di A soddisfacente la seguente proprietà:*

$$\textit{per ogni } a \in U \textit{ esiste } V \subseteq U \textit{ tale che } a \in V \in -\mathcal{F}.$$

Si osservi che la $\mathcal{F}$-topologia di Ershov è uguale alla $-\mathcal{F}$-topologia di Visser, mentre la $-\mathcal{F}$-topologia di Ershov è uguale alla $\mathcal{F}$-topologia di Visser. Si noti, inoltre, che la Definizione 4.3.12 è equivalente a dire che dato un reticolo $\mathcal{F}$ la $\mathcal{F}$-topologia di Ershov è la topologia generata dal reticolo $\mathcal{F}$ e la $\mathcal{F}$-topologia di Visser è la topologia generata dal reticolo $-\mathcal{F}$. Vediamo, ad esempio, il caso della topologia di Ershov.

Proposizione 4.3.13 *Sia A un insieme arbitrario e $\mathcal{F}$ un reticolo limitato distributivo su A, allora la* $\mathcal{F}$-topologia di Ershov *ovvero la famiglia*

$$\tau_E = \{(U \subseteq A) \mid (\forall a \in U)(\exists V \subseteq U).(a \in V \in \mathcal{F})\} \tag{4.1}$$

ha le seguenti caratteristiche:

(a) è una topologia,

(b) la $\mathcal{F}$-topologia di Ershov è uguale alla topologia generata da $\mathcal{F}$.

Dim. (a) La proprietà (i) della definizione di topologia è verificata in quanto $\emptyset \in \mathcal{F}$ poiché l'insieme vuoto verifica banalmente la Proprietà 4.1 e $A \in \mathcal{F}$ in quanto $\mathcal{F}$ è un ricoprimento di A. Dimostriamo ora la proprietà (ii) della definizione di topologia, ovvero mostriamo che dati due insiemi $U, U' \in \tau_E$ l'insieme $U \cap U' \in \tau_E$. Se vale $a \in U \cap U'$ allora, per definizione, esistono $V_1, V_2 \in \mathcal{F}$ tali che $a \in V_1$ e $a \in V_2$ cioè $a \in V_1 \cap V_2 \in \mathcal{F}$. La chiusura per unione arbitraria di τ_E deriva dal fatto che se $a \in \bigcup_{i \in I} U_i$ per opportuni $U_i \in \tau_E$, allora esiste un indice $k \in I$ tale che $a \in U_k \in \tau_E$ e quindi per definizione $\exists V \subseteq U_k$ con $a \in V \in \mathcal{F}$ ma allora V è contenuto anche in $\bigcup_{i \in I} U_i$ e quindi $\left(\bigcup_{i \in I} U_i\right) \in \tau_E$.
(b) Chiamiamo τ_{min} la topologia generata da $\mathcal{F}$. Dimostrare $\tau_{min} \subseteq \tau_E$ è semplice in quanto $(\forall U \in \mathcal{F})(a \in U) \Rightarrow \exists V \in \mathcal{F}$ tale che $a \in V \in \mathcal{F}$ e come V possiamo prendere proprio l'insieme U. Per dimostrare $\tau_E \subseteq \tau_{min}$

basti notare che dato un $U \in \tau_E$ e chiamato $V_a \in \mathcal{F}$ un insieme tale che $a \in V_a \subseteq U$ vale $\bigcup_{a \in U} V_a = U$ e $U \in \tau_{min}$, poiché τ_{min} è la chiusura dell'unione arbitraria degli elementi di $\mathcal{F}$. □

Delle considerazioni analoghe possono essere fatte per la $\mathcal{F}$-topologia di Visser. Queste osservazioni sono utili per focalizzare meglio il fatto che, nel caso della topologia di Ershov e di quella di Visser, non stiamo fornendo una definizione generale degli aperti che le caratterizzano ma stiamo solamente dando una caratterizzazione di una loro base.

È inoltre possibile definire, dato un reticolo limitato e distributivo $\mathcal{F}$ su un dato insieme, il concetto di funzione $\mathcal{F}$-computabile che, come vedremo, è un requisito più forte rispetto a quello di funzione continua che abbiamo già introdotto in precedenza.

Definizione 4.3.14 *Sia $\mathcal{F}$ un reticolo limitato distributivo su A. Una funzione parziale $f : A \rightharpoonup_p A$ viene chiamata* $\mathcal{F}$-computabile *se*

$$(\forall U \in \mathcal{F}).(f^{-1}[U] \in \mathcal{F})$$

ovvero, se l'immagine inversa di un elemento del reticolo è a sua volta elemento del reticolo.

Proposizione 4.3.15 *Sia $\mathcal{F}$ un reticolo limitato distributivo su A, allora una funzione f è $\mathcal{F}$-computabile se e solo se è anche $-\mathcal{F}$-computabile.*

Dim. Questo deriva banalmente dal fatto che $(\forall U \in \mathcal{F}).(f^{-1}[U^C] = (f^{-1}[U])^C)$. □

Come abbiamo già detto la nozione di funzione $\mathcal{F}$-computabile è un requisito più restrittivo rispetto a quello di funzione continua, ovvero ogni funzione $\mathcal{F}$-computabile è anche continua rispetto alla topologia generata da $\mathcal{F}$ ma non vale il viceversa. Vale infatti la seguente proposizione.

Proposizione 4.3.16 *Sia $\mathcal{F}$ un reticolo limitato distributivo su A. Se una funzione parziale $f : A \rightharpoonup_p A$ è* $\mathcal{F}$-computabile *allora è anche continua rispetto alla $\mathcal{F}$-topologia di Ershov e alla $\mathcal{F}$-topologia di Visser.*

Dim. Sappiamo, dalla Definizione 3.5.1, che una funzione è continua se l'immagine inversa di un insieme aperto è, a sua volta, un aperto della topologia. Abbiamo visto che, se questa proprietà vale per gli elementi di una base, vale in genere per la topologia generata da quella base. Per definizione, se f è $\mathcal{F}$-computabile, l'immagine inversa di un elemento di $\mathcal{F}$ è a sua volta elemento di $\mathcal{F}$ e, quindi, è anche un aperto della topologia generata da $\mathcal{F}$. Poiché questa proprietà vale per ogni elemento di $\mathcal{F}$, che è una base della $\mathcal{F}$-topologia di Ershov, f è Ershov-continua. Sappiamo, inoltre, che una funzione $\mathcal{F}$-computabile è anche $-\mathcal{F}$-computabile, per analoghe considerazioni, dunque, la funzione f è anche Visser-continua. □

4.3.4 La topologia di Lawson

Quando si vogliono applicare strumenti topologici a insiemi parzialmente ordinati risulta naturale considerare topologie il cui ordine di specializzazione rispetta l'ordinamento di cui è dotato il poset. Questa condizione solitamente implica che lo spazio topologico ottenuto soddisfa l'assioma di separazione T_0 ma non altri assiomi più forti. È possibile dimostrare che esiste una topologia più piccola di ogni altra definibile su un insieme parzialmente ordinato e la chiameremo *topologia lower interval*. Definiremo tale topologia, al contrario di quanto fatto finora, descrivendo i chiusi che la caratterizzano.

Definizione 4.3.17 *Sia $(D, \leq)$ un poset, la topologia lower interval, che indicheremo con τ_{low}, è la più piccola topologia in cui un insieme $C \subseteq D$ è chiuso se:*

$$(x \in C) \wedge (y \leq x \Rightarrow y \in C).$$

Una sottobase di chiusi che genera questa topologia è la famiglia di insiemi $x{\downarrow}$ per ogni $x \in D$, mentre una possibile base di aperti è quella famiglia che ha come elementi insiemi del tipo $D \setminus \{x_1{\downarrow} \cup \ldots \cup x_n{\downarrow} \mid (x_1, \ldots, x_n \in D) \wedge (n \in \mathbb{N})\}$.

Confrontando gli aperti di τ_{low} con quelli della topologia di Alexandroff definita sullo stesso poset D, possiamo notare che ogni aperto di

τ_{low} è anche un aperto di τ_A. Se un elemento, infatti, appartiene ad un aperto di τ_{low}, anche ogni elemento maggiore di esso vi appartiene. Non vale, invece, il *vice versa* poiché ai chiusi di τ_A possono appartenere anche elementi fra loro non confrontabili. Da ciò ne deriva che la topologia lower interval è più piccola di quella di Alexandroff.

Vediamo ora un risultato che collega il preordine di specializzazione con la topologia lower interval e quella di Alexandroff.

Proposizione 4.3.18 *Sia $(D, \leq)$ un insieme parzialmente ordinato e τ una topologia su D. La topologia τ è T_0 con ordine di specializzazione $\leq$ se e solo se $\tau_{low} \subseteq \tau \subseteq \tau_A$.*

Dim. $(\Rightarrow)$ Se τ induce $\leq$ come preordine di specializzazione allora tutti gli insiemi aperti sono crescenti e tutti i chiusi sono decrescenti rispetto alla relazione $\leq$. Da questo ne deriva che $\tau \subseteq \tau_A$. Mostriamo ora che $\tau_{low} \subseteq \tau$ ovvero che gli insiemi della forma $x\downarrow$ sono chiusi di τ. Siano x, y due elementi tali che $y \not\leq x$, allora per l'assioma T_0 esiste un aperto O di τ che contiene y ma non contiene x. Poiché O è un insieme crescente che non contiene x vale $O \cap x\downarrow = \emptyset$.
$(\Leftarrow)$ Denotiamo con $\leq_\tau$ il preordine di specializzazione indotto da τ. Sappiamo che $x \leq_\tau y$ se e solo se $x \in \overline{\{y\}}$. L'insieme chiuso $\overline{\{y\}}$ è uguale all'insieme $y\downarrow$ in quanto i chiusi di τ sono insiemi decrescenti per il fatto che $\tau \subseteq \tau_A$. Possiamo quindi concludere che $x \leq_\tau y$ se e solo se $x \in \overline{\{y\}}$ ovvero se $x \leq y$ dunque τ induce $\leq$ come preordine di specializzazione. La topologia τ verifica l'assioma di separazione T_0 poiché $\leq$ è, per definizione, un ordine parziale. $\Box$

Da questo risultato deriva che, essendo la topologia di Scott T_0, vale $\tau_{low} \subseteq \tau_S \subseteq \tau_A$. La topologia duale rispetto all'ordinamento della lower interval viene chiamata upper interval e viene definita come segue.

Definizione 4.3.19 *Sia $(D, \leq)$ un poset, la topologia upper interval, che indicheremo con τ_{up}, è la più piccola topologia in cui un insieme $C \subseteq D$ è chiuso se:*

$$(x \in C) \wedge (x \leq y \Rightarrow y \in C).$$

Una sottobase di chiusi che genera questa topologia è la famiglia di insiemi $x\uparrow$ per ogni $x \in D$, mentre una possibile base di aperti è quella famiglia che ha come elementi insiemi del tipo $D \setminus \{x_1\uparrow \cup \ldots \cup x_n\uparrow \, | n \in \mathbb{N}\}$.

Si osservi che l'ordine di specializzazione della topologia upper interval è $(D, \geq)$, ovvero l'ordine duale di D. La topologia generata dalla unione topologica fra la topologia upper interval e la lower interval viene generalmente chiamata *topologia di Frink* (si veda, per approfondimenti [Frink, 1942]) mentre si è soliti riferirsi in letteratura all'unione topologica fra la topologia upper interval e la topologia di Scott con il nome di *topologia di Lawson*. In questo testo indicheremo la topologia di Lawson con τ_L.

Capitolo 5

Topologie e Informatica Teorica

Il primo teorema di ricorsione è uno dei risultati più importanti nel campo della teoria classica della computabilità. Dopo il lavoro di Scott la dimostrazione di tale teorema è stata riscritta in modo semplice e conciso facendo uso delle topologie. Tale teorema, come vedremo, può essere riformulato in modo naturale per il λ-calcolo e dimostrato con analoghi strumenti topologici. In questo capitolo mostreremo un nuovo risultato che abbiamo ottenuto ovvero che la topologia di Ershov e quella di Scott, definite sull'insieme degli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere, sono uguali e questo ci permette di semplificare in modo significativo alcune dimostrazioni. Poiché per la comprensione di questo capitolo sono necessarie preconoscenze di teoria classica della computabilità riportiamo in Appendice D un compendio con i principali risultati e i concetti fondamentali di tale disciplina.

5.1 Topologie e funzioni ricorsive parziali

Un'ottica alternativa e più generale attraverso la quale è possibile vedere i risultati della teoria classica della computabilità è considerare l'insieme $\mathcal{FP} = \{f \mid f : \mathbb{N} \rightharpoonup_p \mathbb{N}\}$ di tutte le funzioni parziali unarie come uno spazio topologico seguendo l'approccio di Uspenskii [Uspenskii, 1955] e Nerode [Nerode, 1957].

5.1.1 Scott sulle funzioni ricorsive parziali

Si noti che l'insieme $\mathcal{FP}$, dotato dell'ordinamento $\leq$ introdotto nell'Esempio 4.1.1 a pagina 74 è un ordine parziale completo algebrico. Questa affermazione segue direttamente dalla Proposizione 4.1.8 (essendone un caso particolare) ponendo $X = \mathbb{N}$ e $Y = \mathbb{N}$. Essendo, dunque, $(\mathcal{FP}, \leq)$ un ordine parziale completo è possibile definire su di esso la topologia di Scott, più precisamente vale la seguente proposizione.

Proposizione 5.1.1 *La famiglia $\mathcal{B}_S$ composta dagli insiemi:*

$$\theta\!\uparrow = \{\varphi \mid (\theta \leq \varphi) \wedge (\theta \textit{ funzione finita})\}$$

è una base della topologia di Scott su $\mathcal{FP}$ ed è chiusa per intersezione finita.

Dim. Che $\mathcal{B}_S$ sia una base della topologia di Scott segue direttamente dalla Proposizione 4.3.4. Dimostriamo ora che $\mathcal{B}_S$ è una famiglia di insiemi chiusa per intersezione finita. Si noti che, date due funzioni finite $\theta, \psi \in \mathcal{FP}$, possiamo definire:

$$\theta\!\uparrow \cap\, \psi\!\uparrow \;=\; \begin{cases} (\theta \cup \psi)\!\uparrow & \text{se } \theta, \psi \text{ sono compatibili,} \\ \emptyset & \text{altrimenti} \end{cases}$$

dove $(\theta \cup \psi)$ è la funzione che vale y se $\theta(x) = \psi(x) = y$ e che è indefinita sull'input x se $\theta(x)$ e $\psi(x)$ sono entrambe indefinite o definite ma differenti. $\square$

La famiglia $\mathcal{B}_S$ non è, invece, chiusa per unione in quanto, date due funzioni finite θ e ψ, non è in generale possibile scrivere l'insieme $(\theta\!\uparrow \cup \psi\!\uparrow)$ come la chiusura verso l'alto di una qualche funzione finita. Vale inoltre la seguente proposizione.

Proposizione 5.1.2 *La famiglia $\mathcal{F}_S$ che otteniamo chiudendo per unione finita l'insieme $\mathcal{B}_S$ è un reticolo limitato distributivo su $\mathcal{FP}$.*

La dimostrazione di tale proposizione è talmente semplice che viene omessa in questa sede in quanto deriva direttamente dalla definizione di reticolo distributivo limitato e dalla costruzione di $\mathcal{F}_S$. Si noti che $\emptyset$ è contenuto in $\mathcal{F}_S$ in quanto, prese due funzioni finite θ e ψ non compatibili, vale $\theta\!\uparrow \cup \psi\!\uparrow = \emptyset$ e che $\mathcal{FP} \in \mathcal{F}_S$ poiché $f_\emptyset\!\uparrow = \mathcal{FP}$. La famiglia $\mathcal{F}_S$ sarà, quindi, definita come segue:

$$\mathcal{F}_S = \{\emptyset, \mathcal{FP}\} \cup \{(\theta_1\!\uparrow \cup \ldots \cup \theta_n\!\uparrow) \mid \theta_1, \ldots \theta_n \text{ funzioni finite}\}.$$

Come già visto se, inoltre, chiudiamo la famiglia $\mathcal{F}_S$ anche per unione arbitraria otteniamo la topologia di Scott su $\mathcal{FP}$.

5.1.2 Scott sulle funzioni calcolabili

Poiché, però, l'oggetto di studio dell'informatica teorica non è tutto l'insieme $\mathcal{FP}$ delle funzioni parziali ma solo il sottoinsieme delle funzioni computabili (che denoteremo con $\mathscr{C}$) siamo costretti a restringere la topologia così costruita a tale insieme per poterla così utilizzare come strumento per lo studio della teoria classica della computabilità. Ricordiamo a tal proposito che una funzione viene detta computabile se esiste un programma, inteso come stringa sintatticamente corretta di un linguaggio di programmazione, che la calcola. Dato un linguaggio di programmazione Turing equivalente è possibile trovare una funzione di codifica effettiva, ovvero calcolabile e totale, che associ ad ogni numero naturale un programma scritto in tale linguaggio. Come notazione scriveremo P_n intendendo il programma associato tramite tale codifica al numero naturale n e Φ_n intendendo la funzione calcolata dal programma P_n. Si osservi, dunque, che la funzione $\Phi : \mathbb{N} \to \mathscr{C}$ è totale e suriettiva e induce sull'insieme $\mathbb{N}$ una relazione di equivalenza. È infatti possibile definire una relazione $\sim \subseteq \mathbb{N} \times \mathbb{N}$ tale che

$$n \sim m \iff \Phi_n = \Phi_m$$

che divide l'insieme $\mathbb{N}$ in classi di equivalenza. Intuitivamente una classe di equivalenza contiene tutti gli indici dei programmi che calcolano la stessa funzione. Diremo che un insieme $I \subseteq \mathbb{N}$ *rispetta le funzioni* se è possibile costruirlo tramite l'unione di queste classi di equivalenza ovvero se vale $(\Phi_x = \Phi_y) \Rightarrow ((x, y \in I) \vee (x, y \in I^C))$.

Nel seguito della trattazione considereremo quindi la topologia di Scott τ_S ristretta all'insieme $\mathscr{C}$ delle funzioni computabili (ovvero la topologia indotta da τ_S su $\mathscr{C}$) e la corrispondente topologia debole su $\mathbb{N}$ rispetto alla funzione Φ.

5.1.3 Ershov e Visser sulle funzioni calcolabili

Allo scopo di poter definire la topologia di Ershov e quella di Visser sulle funzioni computabili dobbiamo introdurre un nuovo reticolo limitato e distributivo che chiameremo $\mathcal{RE}_{\mathscr{C}}$.

Proposizione 5.1.3 *Sia $\mathcal{RE}_{\mathscr{C}}$ la famiglia composta dai sottoinsiemi $O \subseteq \mathscr{C}$ tali che*

$$\Phi^{-1}[O] = \{x \mid (x \in \mathbb{N}) \wedge (\Phi_x \in O)\}$$

è ricorsivamente enumerabile. Allora $\mathcal{RE}_{\mathscr{C}}$ è un reticolo limitato e distributivo su $\mathscr{C}$.

Dim. L'insieme $\mathscr{C} \in \mathcal{RE}_{\mathscr{C}}$ in quanto Φ è suriettiva e $\emptyset \in \mathcal{RE}_{\mathscr{C}}$ poiché rispetta banalmente la condizione di appartenenza. La famiglia $\mathcal{RE}_{\mathscr{C}}$ è chiusa per intersezione finita e unione finita in quanto l'unione e l'intersezione di due insiemi ricorsivamente enumerabili sono, a loro volta, insiemi ricorsivamente enumerabili. $\square$

Essendo $\mathcal{RE}_{\mathscr{C}}$ un reticolo limitato distributivo su $\mathscr{C}$ ci è possibile definire la $\mathcal{RE}_{\mathscr{C}}$-topologia di Ershov e la $\mathcal{RE}_{\mathscr{C}}$-topologia di Visser come visto nella Definizione 4.3.12. Quindi $\mathcal{RE}_{\mathscr{C}}$ è una base della $\mathcal{RE}_{\mathscr{C}}$-topologia di Ershov su $\mathscr{C}$ e $-\mathcal{RE}_{\mathscr{C}}$ è una base della $\mathcal{RE}_{\mathscr{C}}$-topologia di Visser su $\mathscr{C}$. Sarà per noi interessante considerare anche la topologia debole di Eshov sull'insieme $\mathbb{N}$. Per poter lavorare più semplicemente su questi concetti ci conviene caratterizzare in modo più chiaro gli aperti della topologia debole di Ershov.

Proposizione 5.1.4 *La famiglia $\mathcal{RE}_{\mathbb{N}}$ composta dai sottoinsiemi $O \subseteq \mathbb{N}$ tali che $O \in \mathcal{RE}_{\mathbb{N}}$ se:*

(i) O rispetta le funzioni,

(ii) O è un insieme ricorsivamente enumerabile,

è un reticolo limitato e distributivo su $\mathbb{N}$ ed è una base della topologia debole di Ershov rispetto a Φ.

Dim. Si osservi che $\emptyset, \mathbb{N} \in \mathcal{RE}_{\mathbb{N}}$ in quanto rispettano banalmente le funzioni e sono ricorsivamente enumerabili. Considerati inoltre $O_1, O_2 \in \mathcal{F}$ si verifica che anche la loro intersezione $O_1 \cap O_2 \in \mathcal{F}$ poiché, per il Teorema D.3.9, l'intersezione di due insiemi ricorsivamente enumerabili è, a sua volta, un insieme ricorsivamente enumerabile. Inoltre se $x \in O_1 \cap O_2$ allora $x \in O_1$ e $x \in O_2$ quindi, poiché O_1, O_2 rispettano le funzioni, $\{y \mid xRy\} \subseteq O_1$ e $\{y \mid xRy\} \subseteq O_2$ dunque l'intera classe di equivalenza sarà contenuta anche nell'intersezione di O_1, O_2. La famiglia $\mathcal{F}$ è inoltre chiusa per unione finita in quanto, per il Teorema D.3.9, l'unione di due insiemi ricorsivamente enumerabili è un insieme ricorsivamente enumerabile ed è facilmente verificabile che l'unione di due insiemi che rispettano le funzioni è un insieme che rispetta le funzioni. Dal Lemma 3.6.5, inoltre, si ottiene che $\mathcal{RE}_{\mathbb{N}}$ è una base della topologia debole di Ershov rispetto a Φ. $\square$

Si noti che i reticoli costruiti nelle due proposizioni precedenti non solo sono chiusi per unione finita ma anche per unione infinita ricorsivamente enumerabile. Questa proprietà deriva dal fatto che l'unione ricorsivamente enumerabile di insiemi ricorsivamente enumerabili è un insieme ricorsivamente enumerabile.

5.1.4 Ershov = Scott sulle funzioni calcolabili

In questa sezione dimostreremo che la topologia di Scott e quella di Ershov, definite sull'insieme delle funzioni parziali computabili, coincidono. Questo risultato è in realtà già noto in letteratura (viene ad esempio citato in [Odifreddi, 1989]), ci proponiamo in questo testo di farne una trattazione più comprensibile e dettagliata. Il teorema che vogliamo dimostrare è, quindi, il seguente.

Teorema 5.1.5 *La $\mathcal{RE}_{\mathscr{C}}$-topologia di Ershov sull'insieme $\mathscr{C}$ è uguale alla topologia di Scott sull'insieme $\mathscr{C}$.*

Dim. La dimostrazione di questo teorema è divisa in due parti. Per prima cosa mostreremo come ogni aperto della base $\mathcal{B}_S$ della topologia di Scott è anche un aperto della $\mathcal{RE}_{\mathscr{C}}$-topologia di Ershov e poi il *vice versa*, ovvero che ogni aperto della base $\mathcal{RE}_{\mathscr{C}}$ della $\mathcal{RE}_{\mathscr{C}}$-topologia di Ershov è anche un aperto della topologia di Scott.

(a) (Scott $\subseteq$ Ershov) Questa implicazione è banale, consideriamo infatti un aperto O della base di Scott. Essendo $O \in \mathcal{B}_S$ esisterà un'opportuna funzione θ con dominio finito tale che esso possa essere scritto nella forma $\theta\!\uparrow$. L'insieme $\Phi^{-1}[\theta\!\uparrow]$ è un insieme ricorsivamente enumerabile in quanto, poiché il dominio di θ è un insieme finito, per semidecidere l'appartenenza di un numero naturale x all'insieme $\Phi^{-1}[O]$ è sufficiente controllare che Φ_x e θ coincidano sul dominio finito di θ. Dunque l'insieme O è anche un aperto di Ershov.

(b) (Ershov $\subseteq$ Scott) Sia $A \subseteq \mathbb{N}$ un insieme ricorsivamente enumerabile che rispetta le funzioni tale che $A \neq \emptyset, \mathbb{N}$ allora l'insieme $X_A = \{\phi_x \mid x \in A\}$ è, per definizione, un aperto della base della topologia di Ershov. Si noti che, essendo A ricorsivamente enumerabile, non possiamo applicare ad A il secondo teorema di Rice, questo implica quindi che una delle ipotesi del teorema deve essere falsa. Nella fattispecie, poiché A rispetta le funzioni ed è diverso da $\emptyset, \mathbb{N}$, deve essere falsa l'ipotesi che stabilisce che esistono due funzioni f, g tali che $f \leq g$ e gli indici dei programmi che calcolano f sono contenuti in A mentre gli indici dei programmi che calcolano g sono contenuti in A^C. Questo significa che se A contiene l'indice di un programma che calcola una funzione f contiene anche gli indici di tutti i programmi che calcolano le funzioni che estendono f quindi l'insieme X_A è un insieme chiuso verso l'alto, ovvero $X_A\!\uparrow = X_A$. Per motivi analoghi il fatto che non sia applicabile il terzo teorema di Rice ci permette di concludere che se A contiene l'indice di un programma che calcola una funzione f contiene anche gli indici di tutti i programmi che calcolano una opportuna approssimazione finita di f. Esisteranno, quindi, delle funzioni con dominio finito θ_i e un insieme ricorsivamente enumerabile I, tali che $X_A = \bigcup_{i \in I} \theta_i \uparrow$ cioè che $A = \bigcup_{i \in I} \phi^{-1}(\theta_i \uparrow)$ è un aperto di Scott. Nei casi in cui $A = \mathbb{N}$ o $A = \emptyset$, gli unici che non abbiamo considerato prima, l'insieme X_A è uguale, rispettivamente, ad $\mathscr{C}$ e $\emptyset$ che sono aperti di Scott. $\square$

5.1.5 Teorema di Kleene

Dalle considerazioni fatte nella sottosezione precedente ne scaturisce una semplice dimostrazione del primo teorema di ricorsione, altresì detto *teorema di Kleene*. Vediamo, per prima cosa, l'enunciato di tale teorema.

Teorema 5.1.6 *(Teorema di Kleene) Sia $h : \mathbb{N} \to \mathbb{N}$ una funzione computabile e totale che soddisfi la seguente proprietà di estensionalità:*

$$\phi_x = \phi_y \;\Rightarrow\; \phi_{h(x)} = \phi_{h(y)} \quad \textit{per ogni } x, y \in \mathbb{N}.$$

Allora il funzionale ρ, definito da:

$$\rho(\phi_x) = \phi_{h(x)} \quad \textit{per ogni } x \in \mathbb{N},$$

ammette un minimo punto fisso computabile.

Dim. Si noti, innanzitutto, che la dimostrazione del teorema, per il Teorema 4.1.7 del punto fisso, coincide col dimostrare che il funzionale ρ è continuo rispetto alla topologia di Scott. Il funzionale ρ è banalmente continuo rispetto alla $\mathcal{RE}_{\mathscr{C}}$-topologia di Ershov in quanto, se consideriamo un aperto arbitrario della base di Ershov, ovvero un insieme O tale che $\Phi^{-1}[O]$ sia ricorsivamente enumerabile, possiamo facilmente verificare che anche $\rho^{-1}[O]$ è un aperto della base di Ershov in quanto l'immagine inversa rispetto ad una funzione calcolabile di un insieme ricorsivamente enumerabile (nel nostro caso l'insieme $\rho^{-1}[O]$) è, a sua volta, ricorsivamente enumerabile. Abbiamo però appena dimostrato che la topologia di

Ershov è uguale a quella di Scott, questo implica che ρ è anche continuo anche rispetto alla topologia di Scott. La dimostrazione del fatto che il minimo punto fisso di ρ sia computabile è quella classica. □

Come visto in precedenza, dato un reticolo distributivo limitato, è possibile definire il concetto di funzione computabile rispetto a quel reticolo richiedendo che l'antimmagine di un elemento del reticolo sia a sua volta un elemento del reticolo. Sappiamo dal teorema di Kleene appena enunciato che il funzionale ρ è continuo rispetto alla topologia di Scott, ricordando che la famiglia $\mathcal{F}_S$ che otteniamo chiudendo per unione finita la base di Scott sulle funzioni parziali è un reticolo limitato distributivo possiamo ora domandarci se il funzionale ρ è anche $\mathcal{F}_S$-computabile. Vediamo dunque alcuni esempi.

Esempio 5.1.7 *Fissato un arbitrario valore $c \in \mathbb{N}$, e definiamo una funzione $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ nel seguente modo:*

$$g(y, x) = c.$$

Essendo g una funzione calcolabile possiamo applicare il teorema del parametro, esisterà quindi una funzione calcolabile e totale h tale che $\Phi_{h(y)}(x) = g(y, x)$. Consideriamo il funzionale ρ tale che $\rho(\Phi_y) = \rho(\Phi_{h(y)})$ per ogni $y \in \mathbb{N}$. Si noti che, in questo caso, ρ trasforma qualsiasi funzione calcolabile nella funzione con output costante c che denoteremo con f_c, ovvero:

$$\rho(\Phi_y)(x) = f_c(y) = c.$$

Il dominio di ρ è dunque $\mathscr{C}$ e il suo codominio $\{f_c\}$. Consideriamo ora un elemento $B \in \mathcal{F}_S$, se $f_c \notin B$ allora $\rho^{-1}[B] = \emptyset$ che è un elemento di $\mathcal{F}_S$, se invece $f_c \in B$ l'antimmagine di B attraverso ρ è tutto l'insieme $\mathscr{C}$ e, per definizione di reticolo limitato e distributivo, $\mathscr{C} \in \mathcal{F}_S$.

Nel semplice esempio appena visto, dunque, il funzionale ρ è $\mathcal{F}_S$-computabile.

Esempio 5.1.8 *Definiamo una funzione g come segue.*

$$g(y, x) = \begin{cases} \Phi_y(\Phi_y(x + 11)) & \textit{se } x \leq 100, \\ x - 10 & \textit{altrimenti.} \end{cases}$$

Essendo g una funzione calcolabile possiamo applicare il teorema del parametro, esiste quindi una funzione h calcolabile e totale tale che $\Phi_{h(y)}(x) = g(y, x)$. Possiamo quindi definire il funzionale $\rho(\Phi_y) = \Phi_{h(y)}$ e chiederci se anche in questo caso ρ è $\mathcal{F}_S$-computabile. Consideriamo ora l'elemento $\psi\!\uparrow \in \mathcal{F}_S$ dove

$$\psi(x) = \begin{cases} 11 & \text{se } x = 0, \\ \uparrow & \text{altrimenti,} \end{cases}$$

ci chiediamo se $\rho^{-1}[\psi\!\uparrow]$ è un elemento di $\mathcal{B}_S$. Definiamo a questo scopo, al variare di $n \in \mathbb{N}$, una famiglia di funzioni f_n come segue:

$$f_n(x) = \begin{cases} 11 & \text{se } x = n, \\ n & \text{se } x = 11, \\ x & \text{altrimenti.} \end{cases}$$

Si noti che $(\forall n \in \mathbb{N}).(\rho(f_n) \in \psi\!\uparrow)$ poiché $\rho(f_n)(0) = f_n(f_n(11)) = f_n(n) = 11$. Perché una funzione f appartenga all'insieme $\rho^{-1}[\psi\!\uparrow]$ deve valere la condizione

$$(f \downarrow 11) \wedge (f \downarrow f(11)) \wedge (f(f(11)) = 11)$$

è dunque possibile scrivere $\rho^{-1}[\psi\!\uparrow] = \bigcup_{i \in \mathbb{N}} \theta_i\!\uparrow$ dove

$$\theta_i(x) = \begin{cases} 11 & \text{se } x = i, \\ i & \text{se } x = 11, \\ \uparrow & \text{altrimenti.} \end{cases}$$

L'insieme $\rho^{-1}[\psi\!\uparrow]$ non può quindi essere scritto come unione finita di elementi di $\mathcal{B}_S$ in quanto, essendo le funzioni finite θ_i non compatibili fra loro, ogni insieme $\theta_i\!\uparrow$ è un elemento indispensabile dell'unione degli insiemi che formano $\rho^{-1}[\psi\!\uparrow]$. In questo caso il funzionale ρ non è dunque $\mathcal{F}_S$-computabile.

Abbiamo trovato un esempio in cui il funzionale ρ è $\mathcal{F}_S$-computabile e un esempio in cui invece non lo è, sarebbe interessante vedere se è possibile caratterizzare in modo generale quando ρ è $\mathcal{F}_S$-computabile. Il funzionale ρ è invece sempre $\mathcal{RE}_{\mathscr{C}}$-computabile, come si può notare dalla proposizione che segue.

Proposizione 5.1.9 *Sia ρ il funzionale definito da:*

$$\rho(\phi_x) = \phi_{h(x)} \quad \textit{per ogni } x \in \mathbb{N},$$

dove $h : \mathbb{N} \to \mathbb{N}$ è una funzione computabile, totale ed estensionale. Allora ρ è $\mathcal{RE}_{\mathscr{C}}$-computabile.

Dim. Sia $O \in \mathcal{RE}_{\mathscr{C}}$, ovvero un sottoinsieme di $\mathscr{C}$ tale che

$$f^{-1}[O] = \{x \mid (x \in \mathbb{N}) \wedge (\Phi_x \in O)\}$$

sia ricorsivamente enumerabile. Allora, poiché l'immagine inversa rispetto ad una funzione calcolabile di un insieme ricorsivamente enumerabile è, a sua volta, ricorsivamente enumerabile anche $\rho^{-1}[O]$ sarà ricorsivamente enumerabile e quindi $\rho^{-1}[O] \in \mathcal{RE}_{\mathscr{C}}$. $\square$

Abbiamo visto, quindi che il funzionale ρ del teorema di Kleene è $\mathcal{RE}_{\mathscr{C}}$-computabile, dalla Proposizione 4.3.15 ne deriva che ρ è anche $-\mathcal{RE}_{\mathscr{C}}$-computabile.

5.2 Topologie e alberi di Böhm

Come già osservato nell'Esempio 4.1.2 del Capitolo 4, l'insieme $\mathcal{BT}$ degli alberi di Böhm, con la relazione di ordinamento $\sqsubseteq$ (ovvero, dati due alberi t_1 e t_2, vale $t_1 \sqsubseteq t_2$ se e solo se è possibile ottenere t_1 eliminando da t_2 alcuni sottoalberi ed etichettando la radice dei sottoalberi eliminati con $\perp$), è un ordine parziale completo algebrico. Possiamo dunque dotare l'insieme $\mathcal{BT}$ della topologia di Scott procedendo in modo analogo a quanto fatto per l'insieme $\mathcal{FP}$ delle funzioni ricorsive parziali. Per prima cosa, notando che in questo caso gli elementi compatti dell'ordine parziale completo sono gli alberi finiti, possiamo scrivere una base di tale topologia come segue.

Proposizione 5.2.1 *La famiglia $\mathcal{B}_{\mathcal{BT}}$ composta dagli insiemi:*

$$t{\uparrow} = \{t' \mid (t \sqsubseteq t') \wedge (t \textit{ albero finito})\}$$

è una base della topologia di Scott su $\mathcal{BT}$ ed è chiusa per intersezione finita.

Dim. Che $\mathcal{B}_{\mathcal{BT}}$ sia una base della topologia di Scott segue direttamente dalla Proposizione 4.3.4. Dimostriamo ora che $\mathcal{B}_{\mathcal{BT}}$ è una famiglia di insiemi chiusa per intersezione finita. Si noti che, dati due alberi finiti $t_1, t_2 \in \mathcal{BT}$, possiamo definire:

$$t_1\uparrow \cap\, t_2\uparrow = \begin{cases} (\bigsqcup\{t_1, t_2\})\uparrow & \text{se } t_1, t_2 \text{ sono compatibili,} \\ \emptyset & \text{altrimenti.} \end{cases}$$

Possiamo quindi affermare che $\mathcal{B}_{\mathcal{BT}}$ è chiuso per intersezione finita. □

Tuttavia quello che a noi interessa non è tanto l'analisi dell'insieme degli alberi di Böhm, quanto quella dei λ-termini che sono associati a questi alberi. Definita la topologia di Scott su $\mathcal{BT}$, studieremo quindi la topologia debole sull'insieme Λ costruita rispetto alla funzione $BT : \Lambda \to \mathcal{BT}$ che associa ad ogni λ-termine il suo albero di Böhm. Anche in questo caso per poter definire la topologia di Ershov sull'insieme degli alberi di Böhm dobbiamo restringere la nostra attenzione all'insieme $\mathcal{BT}^{re}$ degli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere. Solo in questo caso, infatti, la funzione BT diventa suriettiva. Questo ci permette di costruire il seguente reticolo.

Proposizione 5.2.2 *Sia $\mathcal{RE}_{\mathcal{BT}}$ la famiglia composta dai sottoinsiemi $O \subseteq \mathcal{BT}^{re}$ tali che l'insieme*

$$BT^{-1}[O] = \{M \mid (M \in \Lambda) \wedge (BT(M) \in O)\}$$

è ricorsivamente enumerabile. Allora $\mathcal{RE}_{\mathcal{BT}}$ è un reticolo limitato e distributivo su $\mathcal{BT}^{re}$.

Dim. L'insieme $\mathcal{BT}^{re} \in \mathcal{RE}_{\mathcal{BT}}$ in quanto Φ è suriettiva e $\emptyset \in \mathcal{RE}_{\mathcal{BT}}$ poiché rispetta banalmente la condizione di appartenenza. La famiglia $\mathcal{RE}_{\mathcal{BT}}$ è chiusa per intersezione finita e unione finita in quanto l'unione e l'intersezione di due insiemi ricorsivamente enumerabili sono, a loro volta, insiemi ricorsivamente enumerabili. □

Poiché, quindi, $\mathcal{RE}_{\mathcal{BT}}$ è un reticolo limitato e distributivo possiamo definire la $\mathcal{RE}_{\mathcal{BT}}$-topologia di Ershov su $\mathcal{BT}^{re}$ e la $\mathcal{RE}_{\mathcal{BT}}$-topologia di Visser su $\mathcal{BT}^{re}$. In questo capitolo dimostreremo un risultato analogo a quello ottenuto precedentemente sull'insieme $\mathscr{C}$ delle funzioni parziali calcolabili, ovvero vedremo che la topologia indotta di Scott sull'insieme $\mathcal{BT}^{re}$ coincide con la $\mathcal{RE}_{\mathcal{BT}}$-topologia di Ershov. Nel Paragrafo 5.1.4, per dimostrare il Teorema 5.2.8, abbiamo fatto uso dei teoremi di Rice della teoria classica della computabilità, se vogliamo poter fare una dimostrazione simile dobbiamo quindi trovare una versione di tali teoremi applicabile agli alberi di Böhm.

5.2.1 Teoremi di Rice per alberi di Böhm

In questa sezione riformuleremo il secondo e il terzo teorema di Rice e il teorema del parametro in modo che siano applicabili agli alberi di Böhm. Per prima cosa dovremo fare alcune considerazioni. Per alleggerire la notazione in questo capitolo non faremo distinzione fra gli alberi di Böhm e le funzioni che li rappresentano che sono state introdotte nel Capitolo 2 nella Definizione 2.12.7 di pagina 11. Questo ci è possibile in quanto esiste una corrispondenza biunivoca fra l'insieme $\mathcal{BT}^{re}$ di tutti gli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere e l'insieme delle funzioni calcolabili che li rappresentano. Definiremo, inoltre, un insieme semidecidibile K_Λ, il cui complementare sia non semidecidibile, che svolgerà il ruolo dell'insieme $K = \{x \mid \Phi_x \downarrow x\}$ nella teoria classica della computabilità.

Definizione 5.2.3 *$K_\Lambda \subseteq \Lambda$ è l'insieme dei λ-termini che possiedono una head normal form.*

Chiamiamo $\mathcal{P}_{\mathcal{BT}}$ l'insieme dei programmi che rappresentano alberi di Böhm. È semplice da verificare che l'insieme Λ ha la stessa cardinalità dell'insieme $\mathcal{P}_{\mathcal{BT}}$, possiamo dunque definire una funzione effettiva di codifica che associ ad ogni λ-termine un programma per calcolare una funzione rappresentante un albero di Böhm.

Definizione 5.2.4 *Sia* $|\cdot|_{\mathcal{BT}} : \Lambda \to \mathcal{P}_{\mathcal{BT}}$ *una funzione di codifica calcolabile e totale. Denotiamo, inoltre, con* $q : \Lambda \to \Lambda$ *la funzione calcolata dall'algoritmo descritto nella dimostrazione della Proposizione 2.12.11 ovvero quello che, preso in input un* λ*-termine* M*, restituisce un* λ*-termine che ha come albero di Böhm quello calcolato da* $|M|_{\mathcal{BT}}$*.*

In analogia con quanto fatto in teoria della computabilità denoteremo con Φ_M, con $M \in \Lambda$, la funzione calcolata dal programma $|M|_{\mathcal{BT}}$. Scriviamo, quindi, una versione *ad hoc* del teorema di parallelizzazione che ci consenta di lavorare sui concetti appena introdotti.

Teorema 5.2.5 *Teorema s-m-n (o del parametro) per alberi di Böhm.*
Sia $f : \Lambda \times Node \rightharpoonup_p Label \times \mathbb{N}$ *una funzione calcolabile tale che* $f(M, \alpha)$ *calcoli, fissato un* M *arbitrario, un albero di Böhm. Allora esiste una funzione calcolabile e totale* $s : \Lambda \to \Lambda$ *tale che per ogni* $M \in \Lambda$ *e* $\alpha \in Node$*:*

$$\Phi_{s(M)}(\alpha) = f(M, \alpha).$$

Dim. Essendo f computabile esiste un programma P_f che calcola f. Il programma $|s(M)|_{\mathcal{BT}}$ sarà, dunque, quel programma che prende in input α, predispone M e α come input del programma P_f ed esegue P_f. $\square$

Diremo inoltre che un insieme $I \subseteq \Lambda$ *rispetta gli alberi di Böhm* se vale $(\forall M, N \in \Lambda).(BT(M) = BT(N)) \Rightarrow ((M, N \in I) \vee (M, N \in I^C))$. Vediamo, quindi, come può essere riformulato il secondo teorema di Rice.

Teorema 5.2.6 *(Secondo teorema di Rice per alberi di Böhm)*
Sia $I \subseteq \Lambda$ *un insieme che rispetta gli alberi di Böhm e* $I \neq \emptyset, \Lambda$*, allora se esistono due alberi di Böhm* $f, g \in \mathcal{BT}^{re}$ *tali che:*

(i) $\{M \mid BT(M) = f\} \subseteq I$,

(ii) $\{M \mid BT(M) = g\} \subseteq I^C$,

(iii) $f \sqsubseteq g$,

I non è semidecidibile.

Dim. Definiamo una funzione $h : \Lambda \times Node \to Label \times \mathbb{N}$ come segue

$$h(M, \alpha) = \begin{cases} g(\alpha) & \text{se } M \text{ è solvable} \\ f(\alpha) & \text{altrimenti.} \end{cases}$$

Poiché g, f sono due alberi di Böhm ricorsivamente enumerabili esistono due programmi che calcolano $g(\alpha)$ e $f(\alpha)$ per ogni $\alpha \in Node$. Esponiamo ora un possibile algoritmo per calcolare h. Si esegue in parallelo il programma che semidecide l'appartenenza di M a K_Λ e il programma che computa la funzione f (tale programma esiste in quanto f calcolabile per ipotesi) sull'input α. Si possono verificare tre casi:

- termina il programma che semidecide l'appartenenza di M all'insieme K_Λ. Allora $M \in K_\Lambda$ e si può eseguire il programma che computa la funzione g sull'input α;

- termina prima il programma che calcola f sull'input α con, come risultato, $f(\alpha)$. Poiché l'albero calcolato dalla funzione g estende quello calcolato dalla funzione f non è necessario attendere la fine della computazione del programma che calcola g sull'input α in quanto il risultato dev'essere il medesimo. Possiamo quindi fornire come output $f(\alpha)$;

- nessuno dei due programmi termina. Allora $M \notin K_\Lambda$ ed anche il programma che calcola h, come richiesto, non termina.

Essendo, dunque, h una funzione computabile che, fissato un arbitrario $M \in \Lambda$, rappresenta un albero di Böhm possiamo applicare il teorema del parametro per alberi di Böhm. Esiste dunque una funzione $s : \Lambda \to \Lambda$ computabile e totale tale che:

$$\phi_{s(M)}(\alpha) = h(M, \alpha) = \begin{cases} g(\alpha) & \text{se } M \text{ è solvable} \\ f(\alpha) & \text{altrimenti.} \end{cases}$$

Componendo la funzione q della Definizione 5.2.4 con la funzione s appena trovata otteniamo un λ-termine $q(s(M))$ il cui albero di Böhm è lo stesso generato dal programma $|s(M)|_{\mathcal{BT}}$. Vale, dunque:

- $M \in K_\Lambda \Rightarrow \phi_{s(M)}(\alpha) = g(\alpha) \Rightarrow BT(q(s(M))) = g \Rightarrow q(s(M)) \in I^C$
- $M \in K_\Lambda^C \Rightarrow \phi_{s(M)}(\alpha) = f(\alpha) \Rightarrow BT(q(s(M))) = f \Rightarrow q(s(M)) \in I$.

Quindi, poiché la funzione $(q \circ s)$ è totale e computabile, possiamo dire che vale $K_\Lambda \leq_T I^C$ e $K_\Lambda^C \leq_T I$. $\Box$

Teorema 5.2.7 *(Terzo teorema di Rice per alberi di Böhm).*
Sia $I \subseteq \Lambda$ un insieme che rispetta gli alberi di Böhm e $I \neq \emptyset, \Lambda$, allora se esiste un albero di Böhm $f \in \mathcal{BT}^{re}$ tale che:

(i) $\{M \mid BT(M) = f\ \} \subseteq I$,

(ii) $\{M \mid (BT(M) \sqsubseteq f) \wedge (BT(M)$ *albero finito*$)\} \subseteq I^C$,

allora I non è semidecidibile.

Dim. Sia $\leq_\ell$ la relazione di ordinamento lessicografico e $\ulcorner\cdot\urcorner : Node \to \mathbb{N}$ una funzione di codifica che associa ad ogni nodo di un albero di Böhm un numero naturale tale che $(\forall \alpha, \beta \in Node).(\alpha \leq_\ell \beta \;\Rightarrow\; \ulcorner\alpha\urcorner < \ulcorner\beta\urcorner)$. Definiamo la funzione h come segue.

$$h(M, \alpha) = \begin{cases} f(\alpha) & \text{se } M \twoheadrightarrow_h^{\ulcorner\alpha\urcorner} M' \notin hnf \\ \uparrow & \text{altrimenti.} \end{cases}$$

Con la notazione $M \twoheadrightarrow_h^n M'$ intendiamo che M' è il λ-termine ottenuto con un cammino di riduzione di testa di lunghezza n (completato con passi $\twoheadrightarrow_h$ nel qual caso si giunga ad una head normal form in meno di n passi). Il fatto che h calcoli un'approssimazione dell'albero f segue da come è stata definita la funzione di codifica $\ulcorner\cdot\urcorner$. Se, infatti, per un certo nodo α è vero che $M \twoheadrightarrow_h^{\ulcorner\alpha\urcorner} M' \in hnf$, sarà vero anche per tutti i nodi successivi. Essendo, dunque, h una funzione computabile, fissato un $M \in \Lambda$ arbitrario, che rappresenta un albero di Böhm possiamo applicare il teorema del parametro per alberi di Böhm, allora esiste una funzione calcolabile e totale $s : \Lambda \to \Lambda$ tale che:

$$\phi_{s(M)}(\alpha) = \begin{cases} f(\alpha) & \text{se } M \twoheadrightarrow_h^{\ulcorner\alpha\urcorner} M' \notin hnf \\ \uparrow & \text{altrimenti.} \end{cases}$$

Componendo la funzione q della Definizione 5.2.4 con la funzione s appena trovata otteniamo un λ-termine $q(s(M))$ il cui albero di Böhm è lo stesso generato dal programma $|s(M)|_{\mathcal{BT}}$. Vale, dunque:

- $M \in K_\Lambda \Rightarrow (\exists c_0 \in \mathbb{N}).(M \in K_\Lambda \Rightarrow M \twoheadrightarrow_h^{c_0} M' \in hnf \Rightarrow$
  $\Rightarrow$ esiste un numero finito di nodi α tali che $\phi_{s(M)} \downarrow \alpha \Rightarrow$
  $\Rightarrow$ $BT(q(s(M)))$ è un'approssimazione finita di $f \Rightarrow q(s(M)) \in I^C$

- $M \in K_\Lambda^C \Rightarrow M \twoheadrightarrow_h^{\ulcorner\alpha\urcorner} M' \notin hnf$ per ogni $\ulcorner\alpha\urcorner \Rightarrow$
  $\Rightarrow BT(q(s(M)))$ è l'albero di Böhm $f \Rightarrow q(s(M)) \in I$.

Quindi, poiché la funzione $(q \circ s)$ è totale e computabile, possiamo dire che vale $K_\Lambda \leq_T I^C$ e $K_\Lambda^C \leq_T I$. $\square$

5.2.2 Ershov = Scott sugli alberi di Böhm

In questa sezione dimostreremo che la topologia di Scott e quella di Ershov, definite sull'insieme degli alberi di Böhm computabili, coincidono. Il teorema che vogliamo dimostrare è, quindi, il seguente.

Teorema 5.2.8 *La $\mathcal{RE}_{\mathcal{BT}}$-topologia di Ershov sull'insieme $\mathcal{BT}^{re}$ è uguale alla topologia di Scott sull'insieme $\mathcal{BT}^{re}$.*

Dim. La dimostrazione di questo teorema è divisa in due parti. Per prima cosa mostreremo come ogni aperto della base $\mathcal{B}_{\mathcal{BT}}$ della topologia di Scott è anche un aperto della $\mathcal{RE}_{\mathcal{BT}}$-topologia di Ershov e poi il *vice versa*, ovvero che ogni aperto della base $\mathcal{RE}_{\mathcal{BT}}$ della $\mathcal{RE}_{\mathcal{BT}}$-topologia di Ershov è anche un aperto della topologia di Scott.

(a) (Scott $\subseteq$ Ershov) Questa implicazione è banale, consideriamo infatti un aperto O della base di Scott. Essendo $O \in \mathcal{B}_{\mathcal{BT}}$ esisterà un opportuno albero di Böhm t finito tale che esso possa essere scritto nella forma $t\uparrow$. L'insieme $BT^{-1}[t\uparrow]$ è un insieme ricorsivamente enumerabile in quanto, poiché t è un albero finito, per semidecidere l'appartenenza di un elemento M all'insieme $BT^{-1}[O]$ è sufficiente controllare che $BT(M)$ e t coincidano su un numero finito di nodi. Dunque l'insieme O è anche un aperto di Ershov.

(b) (Ershov $\subseteq$ Scott) Sia $A \subseteq \Lambda$ un insieme ricorsivamente enumerabile chiuso che rispetta gli alberi di Böhm tale che $A \neq \emptyset, \Lambda$ allora l'insieme $X_A = \{BT(M) \mid M \in A\}$ è, per definizione, un aperto della base della topologia di Ershov. Si noti che, essendo A ricorsivamente enumerabile, non possiamo applicare ad A il secondo teorema di Rice per alberi di Böhm, questo implica quindi che una delle ipotesi del teorema deve essere falsa. Nella fattispecie, poiché A rispetta gli alberi di Böhm ed è diverso da $\emptyset, \Lambda$, deve essere falsa l'ipotesi che stabilisce che esistono due alberi f, g tali che $f \sqsubseteq g$ e i λ-termini che hanno f come albero di Böhm sono contenuti in A mentre i λ-termini che hanno g come albero di Böhm sono contenuti in A^C. Questo significa che se A contiene un λ-termine che ha f come albero di Böhm contiene anche i lambda termini il cui albero di Böhm estende f quindi l'insieme X_A è un insieme chiuso verso l'alto, ovvero $X_A\uparrow = X_A$. Per motivi analoghi il fatto che non sia applicabile il terzo teorema di Rice ci permette di concludere che se A contiene un λ-termine che ha f come albero di Böhm contiene anche tutti i λ-termini che hanno come albero di Böhm un'approssimazione finita di f quindi esisteranno degli alberi finiti t_i e un insieme ricorsivamente enumerabile I, tali che $X_A = \bigcup_{i \in I} t_i\uparrow$ cioè che $A = \bigcup_{i \in I} BT^{-1}(t_i\uparrow)$ è un aperto di Scott. Nei casi in cui $A = \Lambda$ o $A = \emptyset$, gli unici che non abbiamo considerato prima, l'insieme X_A è uguale, rispettivamente, ad $\mathcal{BT}^{re}$ e $\emptyset$ che sono aperti di Scott. $\square$

5.2.3 Teorema di Kleene per alberi di Böhm

Il teorema di Kleene della teoria classica della computabilità può essere riscritto per gli alberi di Böhm come segue.

Teorema 5.2.9 *Sia $h : \Lambda \to \Lambda$ una funzione computabile e totale che soddisfa la seguente proprietà di estensionalità:*

$$BT(M) = BT(N) \Rightarrow BT(h(M)) = BT(h(N)).$$

Allora il funzionale ρ definito da

$$\rho(BT(M)) = \rho(BT(h(M))), \text{ per ogni } \lambda\text{-termine } M,$$

ammette un minimo punto fisso computabile.

Dim. Si noti, innanzitutto, che la dimostrazione del teorema, per il Teorema 4.1.7 del punto fisso, coincide col dimostrare che il funzionale ρ è continuo rispetto alla topologia di Scott. Il funzionale ρ è banalmente continuo rispetto alla topologia di Ershov in quanto se consideriamo un aperto arbitrario della base di Ershov, ovvero un insieme O tale che $\{M \mid (M \in \Lambda) \wedge (BT(M) \in O)\}$ sia ricorsivamente enumerabile, possiamo facilmente verificare che anche $\rho^{-1}[O]$ è un aperto della base di Ershov in quanto l'immagine inversa rispetto ad una funzione calcolabile di un insieme ricorsivamente enumerabile (nel nostro caso l'insieme $\rho^{-1}[O]$) è, a sua volta, ricorsivamente enumerabile. Abbiamo però appena dimostrato che la topologia di Ershov è uguale a quella di Scott, questo implica che ρ è anche continuo anche rispetto alla topologia di Scott. La dimostrazione del fatto che il minimo punto fisso di ρ sia computabile è analoga a quella classica. $\square$

Capitolo 6

Lemma di genericità

In questo capitolo formuleremo una nuova dimostrazione topologica del lemma di genericità, un risultato molto importante per il λ-calcolo. Definiremo, a tale scopo, la topologia di Ershov e quella di Visser sull'insieme dei λ-termini e ne studieremo alcune proprietà.

6.1 λ-teorie ricorsivamente enumerabili

Siamo interessati alle topologie di Ershov e di Visser che sono generate rispettivamente da reticoli limitati e distributivi che hanno come elementi insiemi ricorsivamente enumerabili e co-ricorsivamente enumerabili. La nozione di ricorsivamente enumerabile è, però, strettamente collegata all'insieme $\mathbb{N}$ e quindi, nel momento in cui vogliamo estendere tale proprietà a insiemi di λ-termini dobbiamo trovare una funzione biiettiva fra Λ e $\mathbb{N}$. Ricordiamo che tale funzione, una volta identificato $\mathbb{N}$ con $\Lambda_{\mathbb{N}}$, è già stata introdotta nel Capitolo 2 in Definizione 2.9.1 con il nome di funzione di Gödelizzazione e viene denotata con $/\cdot/$.

Ci è, quindi, ora possibile definire la nozione di ricorsivamente enumerabile per insiemi di λ-termini e λ-teorie.

Definizione 6.1.1 *Un sottoinsieme $A \subseteq \Lambda$ viene detto ricorsivamente enumerabile se*

$$/A/ = \{/M/ \mid M \in A\}$$

è ricorsivamente enumerabile.

Definizione 6.1.2 *Una λ-teoria $\mathcal{T}$ viene detta* ricorsivamente enumerabile *se, per ogni termine $M \in \Lambda$, l'insieme*

$$M_{\mathcal{T}} = \{P \mid P =_{\mathcal{T}} M\}$$

è ricorsivamente enumerabile.

6.2 Topologia di Visser e di Ershov su Λ

In generale, data una λ-teoria $\mathcal{T}$, è possibile definire le topologie di Ershov e di Visser partendo dal seguente reticolo limitato e distributivo su Λ.

Proposizione 6.2.1 *Sia $\mathcal{T}$ una λ-teoria, allora la famiglia $\mathcal{RE}_{\mathcal{T}}$ di insiemi $O \subseteq \Lambda$ tale che*

(i) O è chiuso rispetto alla relazione $=_{\mathcal{T}}$,

(ii) O è ricorsivamente enumerabile,

è un reticolo limitato e distributivo su Λ.

Si noti che $\mathcal{RE}_{\mathcal{T}}$ è un sottoreticolo limitato e distributivo dell'algebra booleana di tutti i sottoinsiemi di Λ. A partire da questo reticolo possiamo, al solito, definire la $\mathcal{RE}_{\mathcal{T}}$-topologia di Ershov e la $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser.

Definizione 6.2.2 *(Topologie di Ershov e di Visser su Λ)*

(i) La $\mathcal{RE}_{\mathcal{T}}$-topologia di Ershov su Λ è la topologia generata dalla base di insiemi aperti $\mathcal{RE}_{\mathcal{T}}$,

(ii) la $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser su Λ è la topologia generata dalla base di insiemi chiusi $\mathcal{RE}_{\mathcal{T}}$.

In via informale è possibile affermare che la topologia di Scott sta alla teoria degli ordinamenti parziali come le topologie di Visser e di Ershov stanno alla teoria della computabilità. Osserviamo che, se $\mathcal{T}$ è una λ-teoria ricorsivamente enumerabile, allora la $\mathcal{RE}_\mathcal{T}$-topologia di Ershov corrisponde alla topologia discreta in quanto le classi di equivalenza su Λ determinate da $\mathcal{T}$ sono ricorsivamente enumerabili. Diremo che una funzione $f : \Lambda \to \Lambda$ *rispetta* $=_\mathcal{T}$ se

$$(\forall M, N \in \Lambda).(M =_\mathcal{T} N \Rightarrow (f(M) =_\mathcal{T} f(N)).$$

Ricordiamo, dalla Definizione 2.13.5, che una λ-teoria $\mathcal{T}$ si dice semi-sensibile se non eguaglia termini solvable e termini unsolvable.

Proposizione 6.2.3 *Sia $\mathcal{T}$ una teoria semi-sensibile e $f : \Lambda \to \Lambda$ una funzione calcolabile e totale che rispetta $=_\mathcal{T}$, allora f è $\mathcal{RE}_\mathcal{T}$-computabile e, quindi, è continua rispetto alla $\mathcal{RE}_\mathcal{T}$-topologia di Visser.*

Dim. Sia O un aperto della base contenente $f(M)$. Definiamo $f^{-1}[O] = \{N \mid f(N) \in O\}$. Poiché O^C è un insieme ricorsivamente enumerabile $f^{-1}[O^C] = \{N \mid f(N) \in O^C\}$ è ricorsivamente enumerabile. L'insieme $f^{-1}[O^C]$ è chiuso rispetto ad $=_\mathcal{T}$ in quanto f rispetta $=_\mathcal{T}$ e quindi $f^{-1}[O]$ è un aperto della $\mathcal{RE}_\mathcal{T}$-topologia di Visser. $\Box$

Corollario 6.2.4 *In particolare, data una teoria $\mathcal{T}$ semi-sensibile e un contesto $C[a]$, la funzione $C[\cdot]$ che associa ad ogni $M \in \Lambda$ il termine $C[M]$ è continua rispetto alla $\mathcal{RE}_\mathcal{T}$-topologia di Visser.*

Dim. Poiché $C[\cdot]$, se $\mathcal{T}$ è una teoria semi-sensibile, è calcolabile, totale e rispetta $=_\mathcal{T}$, per la Proposizione 6.2.3, si ha che $C[\cdot]$ è continua rispetto alla $\mathcal{RE}_\mathcal{T}$-topologia di Visser. $\Box$

Proposizione 6.2.5 *Se $\mathcal{T}$ è una λ-teoria semi-sensibile, l'insieme di tutti gli unsolvable, che denoteremo con $\mathscr{U}$, è un aperto della base della $\mathcal{RE}_\mathcal{T}$-topologia di Visser.*

Dim. Poiché $\mathcal{T}$ è una teoria semi-sensibile nessun λ-termine solvable viene eguagliato da $=_{\mathcal{T}}$ ad un termine unsolvable. Sia l'insieme $\mathscr{U}$ che il suo complementare, dunque, sono chiusi rispetto ad $=_{\mathcal{T}}$. Ricordando dalla Sezione 2.11 del Capitolo 2 che l'insieme di tutti i termini solvable, ovvero il complementare di $\mathscr{U}$, è un insieme ricorsivamente enumerabile possiamo affermare che $\mathscr{U}$ è un elemento della base della $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser. $\square$

Un altro risultato importante per i nostri scopi è il fatto che, se $\mathcal{T}$ è una λ-teoria semi-sensibile, allora l'insieme Λ, dotato della $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser, è uno spazio iperconnesso. Per poter procedere con la dimostrazione di questo enunciato dobbiamo preliminarmente richiamare alla memoria alcuni concetti ed enunciare e dimostrare il seguente lemma. Ricordiamo che con $/M/$ indichiamo il numero di Gödel del λ-termine M e con n^* denotiamo quel λ-termine tale che $/n^*/ = n$. Rammentiamo, inoltre, che quando parliamo di funzione da Λ in Λ calcolabile intendiamo che lo è sotto opportuna codifica effettiva del dominio e del codominio e che, per alleggerire la notazione, non facciamo distinzione fra numeri naturali e numerali.

Lemma 6.2.6 *Per ogni funzione computabile $\varphi : \mathbb{N} \rightharpoonup_p \mathbb{N}$ esiste una funzione computabile totale $\psi : \mathbb{N} \to \mathbb{N}$ tale che, per ogni $n \in \mathbb{N}$ vale $\varphi \downarrow n \Rightarrow \varphi(n)^* =_\beta \psi(n)^*$.*

Dim. Sia F il combinatore che λ-definisce la funzione computabile φ. Sappiamo, dal Teorema 2.9.3, che esiste un combinatore $\boldsymbol{E}$ tale che, per ogni $M \in \Lambda^0$, vale $\boldsymbol{E}/M/ \twoheadrightarrow_\beta M$ (per una definizione di $\boldsymbol{E}$ si veda [Barendregt, 1984], Sezione 8.1). Si noti che, se n è un naturale, allora $\lambda\beta \vdash \boldsymbol{E}n = n^*$. Definiamo la seguente funzione computabile $\psi(n) = /\boldsymbol{E}(Fn)/$ per ogni numero n. Assumiamo che la funzione φ sia definita su n e $\varphi(n) = m$. Ne segue che

$$
\begin{array}{ll}
\psi(n)^* = & (\text{poiché } \psi(n) = /\boldsymbol{E}(Fn)/) \\
\boldsymbol{E}(Fn) =_\beta & (\text{in quanto } Fn =_\beta m) \\
\boldsymbol{E}m =_\beta & (\text{poiché } \boldsymbol{E}m \twoheadrightarrow_\beta m^*) \\
m^* \equiv & (\text{per def. di } \varphi) \\
\varphi(n)^* & \square
\end{array}
$$

Siamo finalmente in grado di dimostrare il seguente teorema.

Teorema 6.2.7 *Sia $\mathcal{T}$ una teoria semi-sensibile. L'insieme Λ dotato della $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser è uno spazio iperconnesso.*

Dim. Dobbiamo dimostrare che l'intersezione di due elementi della base è sempre non vuota o, in alternativa, che l'unione dei loro complementi è sempre differente da Λ. Siano dunque U e V due chiusi distinti tali che i loro complementi siano elementi della base. Assumiamo, per assurdo, che $U \cup V = \Lambda$, consideriamo gli elementi $a \in U \setminus V$ e $b \in V \setminus U$ e definiamo la funzione φ nel seguente modo:

$$\varphi(x) = \begin{cases} /a/ & \text{se } x \in A \\ /b/ & \text{se } x \in B \\ \uparrow & \text{altrimenti} \end{cases}$$

dove A, B sono insiemi ricorsivamente inseparabili (A e B vengono detti *ricorsivamente inseparabili* se $A \cap B = \emptyset$ e non esiste un insieme C decidibile tale che $A \subseteq C$ e $B \subseteq C^C$). Sia ψ una funzione che estende φ rendendola totale (come nel Lemma 6.2.6). Si noti che gli insiemi $A' = \psi^{-1}[/U/]$ e $B' = \psi^{-1}[/V/]$ sono due insiemi ricorsivamente enumerabili la cui unione è $\mathbb{N}$ e che, per costruzione, $A \subset A'$, $A \cap B' = \emptyset$, $B \subset B'$ e $B \cap A' = \emptyset$. Fissiamo due programmi $P_{A'}$ e $P_{B'}$ che enumerano rispettivamente A' e B' e costruiamo un programma P_C come segue. P_C, preso in input un naturale x, lancia in parallelo $P_{A'}$ e $P_{B'}$. Essendo $A' \cup B' = \mathbb{N}$ il numero x comparirà sicuramente nell'enumerazione di A' o di B' (o di entrambe). Il programma P_C fornirà, quindi, 1 come output se x viene enumerato prima da $P_{A'}$, 0 altrimenti. Si noti che il programma P_C è totale e calcola la funzione caratteristica di un insieme C che è quindi decidibile e, per costruzione, $A \subset A' \subseteq C$ e $B \subset B' \subseteq C^C$. Siamo quindi riusciti a separare A e B trovando, così, un assurdo. $\square$

Corollario 6.2.8 *Sia $\mathcal{T}$ una teoria semi-sensibile, allora ogni aperto della $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser contiene un λ-termine unsolvable.*

Dim. Per la Proposizione 6.2.5 sappiamo che l'insieme degli unsolvable è un aperto della $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser e dal Teorema 6.2.7 evinciamo che l'intersezione di due insiemi aperti è sempre non vuota. Da questo si ricava che ogni aperto della $\mathcal{RE}_{\mathcal{T}}$-topologia di Visser contiene un λ-termine unsolvable. $\Box$

Si ricorda che denotiamo con $\mathcal{H}$ la λ-teoria introdotta nella Sezione 2.13.2, ovvero quella ottenuta eguagliando tutti i termini unsolvable. Il teorema che segue, come vedremo, generalizza il ben noto lemma di genericità.

Teorema 6.2.9 *Siano $M, N \in \Lambda$ due λ-termini tali che M sia unsolvable mentre N possieda una $\lambda\beta$-normal form. Allora, per ogni funzione $f : \Lambda \to \Lambda$ calcolabile e totale che rispetta $=_{\mathcal{H}}$, vale che*

$$f(M) =_\beta N \Rightarrow f(L) =_\beta N \text{ per ogni } L \in \Lambda.$$

Dim. Si osservi che l'insieme $\mathscr{U}$ contenente tutti i termini unsolvable è, per la Proposizione 6.2.5, un aperto della $\mathcal{RE}_{\mathcal{H}}$-topologia di Visser. Dal Corollario 6.2.8 segue che $\mathscr{U} \cap A \neq \emptyset$ per ogni aperto A di Visser. Questo implica che la chiusura di $\mathscr{U}$ è l'intero insieme Λ. Quindi, poiché N ha una forma β-normale, l'insieme $\{P \mid P =_{\mathcal{H}} N\}$ è un chiuso della base in quanto chiuso rispetto ad $=_{\mathcal{H}}$ ed è un insieme ricorsivamente enumerabile per [Barendregt, 1984, Teorema 16.1.9]. Poiché, per la Proposizione 6.2.3, la funzione f è continua rispetto alla $\mathcal{RE}_{\mathcal{H}}$-topologia di Visser, l'immagine inversa dell'insieme chiuso $\{P \mid P =_{\mathcal{H}} N\}$ è l'unico insieme $\mathcal{H}$-chiuso contenente $\mathscr{U}$, cioè Λ. Da ciò ne consegue che $f(L) =_{\mathcal{H}} N$ per ogni $L \in \Lambda$, quindi, per [Barendregt, 1984, Teorema 16.1.9], $f(L) =_\beta N$ per ogni $L \in \Lambda$.$\Box$

Corollario 6.2.10 *(Lemma di genericità) Siano $M, N \in \Lambda$ due λ-termini tali che M sia unsolvable mentre N possieda una head normal form. Allora, per ogni contesto $C[a]$ vale che*

$$C[M] =_\beta N \Rightarrow C[L] =_\beta N \text{ per ogni } L \in \Lambda.$$

Dim. Deriva direttamente dal Teorema 6.2.9 in quanto $C[\cdot]$ è una funzione calcolabile e totale che rispetta $=_{\mathcal{H}}$. $\square$

Teorema 6.2.11 *Sia $M \in \Lambda$ un termine unsolvable. Allora, per ogni funzione $f : \Lambda \to \Lambda$ calcolabile e totale che rispetta $=_{\mathcal{H}}$, vale che*

$$f(M) =_\beta N \Rightarrow N \sqsubseteq f(L) \textit{ per ogni } L \in \Lambda.$$

Dim. Per ogni λ-termine B con albero di Böhm finito, l'insieme $B_{\geq} = \{P|(P \in \Lambda) \wedge (B \sqsubseteq P)\}$ è ricorsivamente enumerabile. Inoltre, per ogni λ-termine C con albero di Böhm infinito vale $C_{\geq} = \{P|(P \in \Lambda) \wedge (C \sqsubseteq P)\} = \bigcap_{B \sqsubseteq C} B_{\geq}$. Allora $C_{\geq}$ è un insieme $\mathcal{B}$-chiuso e, poiché per la Proposizione 2.13.7 si ha che $\mathcal{H} \subseteq \mathcal{B}$, $C_{\geq}$ è anche un insieme $\mathcal{H}$-chiuso per ogni λ-termine C. La conclusione della dimostrazione del teorema è ora immediata, infatti l'antimmagine di $N_{\geq}$ attraverso f è l'unico insieme $\mathcal{H}$-chiuso contenente $\mathscr{U}$. $\square$

Corollario 6.2.12 *Sia $M \in \Lambda$ un termine unsolvable. Allora, per ogni contesto $C[a]$ vale che*

$$C[M] =_\beta N \Rightarrow N \sqsubseteq C[L] \textit{ per ogni } L \in \Lambda.$$

Dim. Segue banalmente dal Teorema 6.2.11 in quanto la funzione $C[\cdot]$ è calcolabile, totale e rispetta $=_{\mathcal{H}}$ e $\mathcal{B}$ è una teoria sensibile e quindi anche semi-sensibile. $\square$

Capitolo 7

Conclusioni

In questa tesi si è definito ed analizzato il formalismo computazionale del λ-calcolo non tipato partendo dalla nozione di pre-termine ed estendendola poi a quella di λ-termine, sono state introdotte le relazioni di riduzione più importanti descrivendone le relative semantiche e sono stati presentati alcuni risultati fondamentali in questo campo. Oltre alle nozioni di termine solvable e unsolvable, di albero di Böhm e di approssimazione di un λ-termine sono state studiate le λ-teorie e le proprietà ad esse collegate. Il Capitolo 3 si propone di essere un compendio *self contained* di topologia che non richiede preconoscenze di sorta al fine di garantire la comprensione di questo elaborato anche a chi si addentri per la prima volta nello studio di questa disciplina. Nel Capitolo 4 vengono introdotti e studiati i reticoli, gli ordinamenti parziali e vengono portati come esempi alcuni casi di studio di topologie particolarmente importanti per ottenere risultati di informatica teorica. Fra queste ricordiamo la topologia di Alexandroff, quella di Scott, quella di Visser, quella di Ershov e quella di Lawson. Nel Capitolo 5, dove vengono dimostrati i risultati più importanti di questa tesi, vengono definite la topologia di Scott e la topologia di Ershov sul cpo delle funzioni calcolabili parziali e su quello degli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere. Viene inoltre dimostrato che, in questi casi, le due topologie coincidono e viene fornita una dimostrazione topologica del teorema di Kleene per la teoria delle funzioni ricorsive parziali. Il teorema del parametro, il secondo e il terzo teorema di Rice e il teorema di Kleene vengono riformulati in una versione per gli alberi di Böhm e se ne fornisce anche una dimostrazione. Nel Capitolo 6 viene definita la topologia di Visser sull'insieme dei λ-termini e viene data una dimostrazione topologica del lemma di genericità per il λ-calcolo.

7.1 Problemi aperti

In questa sezione vogliamo proporre alcune problematiche che sono state focalizzate durante la scrittura di questa tesi ma che non sono state trattate per mancanza di tempo o in quanto troppo complesse da affrontare. Tali tematiche potrebbero, probabilmente, rivelarsi interessanti spunti iniziali per futuri lavori di ricerca sull'argomento.

Abbiamo visto come la topologia di Scott e la topologia di Ershov definite nel Capitolo 5 coincidano sia sul cpo delle funzioni parziali calcolabili, sia su quello degli alberi di Böhm ricorsivamente enumerabili con un numero finito di variabili libere. Probabilmente è possibile generalizzare questo risultato a qualsiasi cpo $(D, \leq)$ tale che valgano le seguenti proprietà:

- D è un dominio di Scott, ovvero l'estremo superiore di due elementi compatti compatibili è, a sua volta, un elemento compatto,
- esista una funzione $\Phi : \mathbb{N} \to D$ tale che l'antimmagine attraverso Φ di un elemento compatto sia un insieme non vuoto,
- gli elementi compatti di D possano essere codificati con i numeri naturali attraverso una qualche funzione,
- gli insiemi del tipo $\Phi^{-1}[\theta\!\uparrow]$ dove θ è un elemento compatto del cpo sono ricorsivamente enumerabili,
- valgano in modo "astratto" il secondo e il terzo teorema di Rice.

La parte, a nostro avviso, più complicata sembra essere la riformulazione astratta dei teoremi di Rice e la relativa dimostrazione in quanto gli originali sono molto legati a risultati difficilmente generalizzabili della teoria classica della computabilità.

Sarebbe, inoltre, interessante riuscire a dimostrare che la $\mathcal{RE}_{\mathcal{BT}}$-topologia di Visser, introdotta nella Sezione 5.2, coincide con la topologia upper interval, ovvero quella generata dalla base di aperti che ha come elementi insiemi del tipo $\mathcal{BT}^{re} \setminus \{t_1\uparrow \cup \ldots \cup t_n\uparrow \quad | \; n \in \mathbb{N}\}$ al variare di $t_1, \ldots t_n \in \mathcal{BT}^{re}$. Nel qual caso questo fosse vero, come noi supponiamo, l'unione topologica della $\mathcal{RE}_{\mathcal{BT}}$-topologia di Visser e della topologia di Scott sarebbe uguale alla *topologia di Lawson* e da ciò ne conseguirebbero interessanti proprietà.

Ringraziamenti

Vorrei, innanzitutto, ringraziare tutti coloro hanno permesso, o in qualche modo incoraggiato, sia la mia carriera scolastica che la stesura di questo elaborato; primo fra tutti, quindi, il mio relatore Antonino Salibra che mi ha iniziato al mondo della ricerca nel campo dell'informatica teorica e che mi ha supportato ancora una volta in veste di relatore.

I miei più sentiti ringraziamenti vanno, inoltre, a tutto il dipartimento di informatica dell'università di Amsterdam per avermi ospitato per cinque mesi e a quello di Udine di cui sono stato studente per un anno. In questi anni di studio credo di essere cresciuto molto sia umanamente che professionalmente e questo lo devo soprattutto a tutti quelli che mi hanno circondato, supportato e sopportato tra cui tutti i professori dell'università di Ca'Foscari, il professor Claudio Mirolo dell'ateneo di Udine, il gruppo di ricerca di informatica teorica dell'università Vrije (Mister Klop, in particolare), i miei colleghi e i miei studenti.

Nei confronti di Alvise Spanò e di Diego Dainese, che sono stati i miei migliori maestri, sento di essere infinitamente debitore. Un grosso grazie va a Giulio Marcon e Andrea Pavan per aver letto questa tesi e avermi dato un loro giudizio oltre che per essere stati degli splendidi amici con cui anche condivido l'amore per la ricerca matematica, alla mia collega e discepola Anna Sperotto per avermi consentito di farla diventare un baluardo di speranza per l'Informatica Teorica nel dipartimento di informatica di Ca'Foscari, a tutti coloro che mi hanno insegnato qualcosa o che hanno imparato qualcosa da me. Mi sento, inoltre, di dover ringraziare i Team che hanno sviluppato *OCaml* ed *Haskell* e che hanno in questo modo fornito un potente strumento per programmare a chiunque abbia, come me, una mentalità da lambda calcolista.

Tante volte durante il periodo passato ad Amsterdam, lontano da casa e dagli amici di una vita, mi sono trovato a ripensare a quanta strada abbiamo fatto da quando al liceo avevamo fondato il *demo-group* "Syntax Error", in quell'epoca della nostra gioventù in cui riuscivamo a mangiare in quattro un tiramisù evidentemente cucinato per un plotone dell'esercito e pensavamo che gli algoritmi fossero strani tipi di panini al salmone. Voglio, quindi, ringraziare PaN, Sir, Giop, Zeff, R-Boot, e gli altri per quel periodo spensierato.

Fra gli altri che il Fato ha voluto essere i miei compagni di erasmus Andrea Fontanelli, Lorenzo Cecchi, Giulia Caruso, Kathleen Reinhardt, Manuela Stillitano e Cristina Lopez-Herrera Sanchez hanno reso la mia esperienza nella terra dei tulipani e dei mulini a vento particolarmente positiva[1] aiutandomi peraltro a salvaguardare la mia sanità mentale in un paese i cui abitanti cenano alle sei del pomeriggio pasteggiando a latte e pancake.

Come potrei, inoltre, non citare gli amici e strumentisti passati e presenti che, negli anni, si sono alternati nella formazione del mio gruppo musicale, gli ybcos, parlo chiaramente di Checco, Laura, Giulio, Cicciobursiccio, Anna, Lau, Ruby, Tony, Claudia, Monica, Silvia.

Ma tra mille facezie il grazie più grande e sincero va, forse, ai miei genitori che mi hanno supportato durante questi lunghi anni di studio e a tutti i miei amici, *i fioi*, ovvero Giulio Marcon, Luca Nascimben, Erica Panigas, Andrea Pavan, Francesco Targhetta, Davide Vido, Domenico Zanata, che ormai da tempo mi circondano, con i quali ho avuto occasione di condividere tante esperienze formative essendo stati fedeli compagni nella tortuosa strada della vita. Grazie.

[1] Pur dimostrandosi perplessi nei confronti del mio splendido cappello da cowboy acquistato a Bruxelles.

Bibliografia

[Amadio and Curien, 1998] Amadio, R. and Curien, P.-L. (1998). *Domains and Lambda-Calculi.* Number 46 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

[Ambriola and Mirolo, 1994] Ambriola, V. and Mirolo, C. (1994). *Programmazione Razionale.* Jackson Libri, first edition.

[Barendregt, 1984] Barendregt, H. P. (1984). *The Lambda calculus - It's Syntax and Semantics.* North-Holland, Amsterdam.

[Bredon, 1993] Bredon, G. E. (1993). *Topology and Geometry*, volume 139 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, Heidelberg, Berlin.

[Burris and Sankappanavar, 1981] Burris, S. and Sankappanavar, H. (1981). *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics.* Springer, Berlin.

[Church and Rosser, 1936] Church, A. and Rosser, J. B. (1936). Some properties of conversion. 36(3):472–482.

[Condò, 2001] Condò, C. (2001). Types for modules. Master's thesis, Ca'Foscari, Venice.

[Cousineau G., 1998] Cousineau G., M. M. (1998). *The Functional Approach to Programming.* Cambridge University Press.

[Cutland, 1980] Cutland, N. J. (1980). *Computability: An Introduction to Recursive Function Theory.* Cambdridge University Press, first edition.

[Ershov, 1999] Ershov, Y. L. (1999). *Theory of Numberings, Handbook in Computability Theory*, volume 140 of *Studies in logic and the foundations of mathematics.* Elsevier Science B.V.

[Frink, 1942] Frink, O. (1942). Topology in lattices. *Trans. Americ. Math. Soc.*, 5:569–582.

[Gierz et al., 1980] Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M. W., and Scott, D. S. (1980). *A compendium of continuous lattices.* Springer-Verlag, Berlin.

[Gierz et al., 2002] Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M. W., and Scott, D. S. (2002). *Continuous Lattices and Domains*, volume 93 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press.

[Goldberg, 1996] Goldberg, M. (1996). Recursive application survival in the λ-calculus. Phd thesis, Computer Science Department, Indiana University, Bloomington, Indiana.

[Goldberg, 2000] Goldberg, M. (2000). Gödelization in the lambda calculus. 75(1–2):13–16.

[Hindley and Seldin, 1986] Hindley, R. and Seldin, J. (1986). *Introduction to Combinators and Lambda Calculus.* Number 1 in London Mathematical Society Student Texts. Cambridge University Press.

[Huth and Mislove, 1999] Huth, M. and Mislove, M. (1999). Linear fs-lattices and their characterization via function spaces.

[Kelley, 1955] Kelley, J. L. (1955). *General Topology.* Springer-Verlag, New York, NY, USA.

[Krivine, 1990] Krivine, J.-L. (1990). *Lambda-calcul Types et Modèles.* Etudes et Recherches en Informatique.

[Manzonetto, 2002] Manzonetto, G. (2002). Sugli approcci alla teoria astratta della computabilità . Bachelor's thesis, Ca'Foscari, Venice.

[Nerode, 1957] Nerode, A. (1957). General topology and partial recursive functionals. *Talks Cornell Summ. Inst. Symb. Log.*, pages 247–251.

[Odifreddi, 1989] Odifreddi, P. (1989). *Classical Recursion Theory: the Theory of Functions and Sets of Natural Numbers.* Number 125 in Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam.

[Revesz, 1988] Revesz, G. (1988). *Lambda-Calculus Combinators and Functional Programming.* Cambridge Tracts in Theoretical Comp Sci. Cambridge Univ Press, 1 edition.

[Schönfinkel, 1924] Schönfinkel, M. (1924). Uber die bausteine der mathematischen logik.

[Scott, 1972] Scott, D. (1972). *Continuous lattices, toposes, algebraic geometry and logic*, pages 97–136. Springer, Berlin.

[Selinger, 2001] Selinger, P. (2001). Course notes, mat 4162/5162 mathematical foundation of computation, department of mathematics, university of pennsylvania.

[Sòrensen and Urzyczyn, 1998] Sòrensen, M. H. and Urzyczyn, P. (1998). Lectures on the Curry-Howard isomorphism. Technical report 98/14 (= TOPPS note D-368), Univ. of Copenhagen.

[Stone, 1937] Stone, M. H. (1937). Applications of the theory of boolean rings to general topology. transactions of the american mathematical society. *Transactions of the American Mathematical Society*, 41 (3):375–481.

[Uspenskii, 1955] Uspenskii, V. (1955). On enumeration operators. *Dokl. Acad. Nauk*, 103:773–776.

[Winskel, 1993] Winskel, G. (1993). *The Formal Semantics of Programming Languages*. Foundations of Computing. The MIT Press. WIN g2 93:1 P-Ex.

Appendice A

Decidibilità dell'α-equivalenza

Nel Capitolo 2 di questo elaborato (Osservazione 2.3.8) affermiamo che la relazione di α-equivalenza fra λ-termini è decidibile; in questa appendice ci proponiamo di giustificare tale affermazione fornendo un programma che computi la funzione caratteristica di tale relazione. Il codice di tale programma è espresso in pseudo codice *ocaml-like*, chi non conoscesse tale linguaggio può consultare [Cousineau G., 1998] o il sito *http://caml.inria.fr/*.

Ricordiamo le definizioni di sostituzione e α-equivalenza:

Definizione A.0.1 *Siano* $M,\ N \in \Lambda$ *e* $x \in V$ *allora si dice* sostituzione *di* N *a* x *in* M *e si scrive* $M[x := N] \in \Lambda$ *l'operazione così definita induttivamente:*

$$\begin{array}{lcll} x[x := N] & = & N; & \\ y[x := N] & = & y & \mathit{con}\ x \neq y; \\ (M\ P)[x := N] & = & M[x := N]\ P[x := N]; & \\ (\lambda x.M)[x := N] & = & \lambda x.M; & \\ (\lambda y.M)[x := N] & = & \lambda y.M[x := N] & \mathit{se}\ y \notin FV(N)\ \mathit{o}\ x \notin FV(M); \\ (\lambda y.M)[x := N] & = & \lambda z.M[y := z][x := N] & \mathit{se}\ y \in FV(N)\ \mathit{e}\ x \in FV(M) \end{array}$$

con $z \in \{v_i \in V|\ v_i \notin FV(M) \cup FV(N)\}$.

Definizione A.0.2 *Sia detta* α-equivalenza *o* α-congruenza *e si scriva* $=_\alpha$ *la più piccola relazione su* Λ^-*, tale che:*

$$\begin{array}{ll} M =_\alpha M & \mathit{per\ ogni}\ M \in \Lambda^-; \\ \lambda x.M =_\alpha \lambda y.M[x := y] & \mathit{se}\ y \notin FV(M), \end{array}$$

ed inoltre sia chiusa rispetto alle seguenti regole:

$$
\begin{array}{lcl}
M =_\alpha M' & \Rightarrow & (\forall x \in V).(\lambda x.M =_\alpha \lambda x.M'); \\
M =_\alpha M' & \Rightarrow & (\forall Q \in \Lambda^-).(M\ Q =_\alpha M'\ Q); \\
M =_\alpha M' & \Rightarrow & (\forall Q \in \Lambda^-).(Q\ M =_\alpha Q\ M'); \\
M =_\alpha M' & \Rightarrow & M' =_\alpha M; \\
M =_\alpha M' \ \wedge \ M' =_\alpha M'' & \Rightarrow & M =_\alpha M''.
\end{array}
$$

Supponiamo di avere a disposizione l'operazione di sostituzione definita come sopra. Scrivere un programma che computi la funzione caratteristica della relazione di α-congruenza è semplice; intuitivamente il ragionamento da fare è il seguente:

- se i due termini considerati sono due variabili x e y essi sono α-equivalenti se e solo se $x \equiv y$;
- se i due termini sono due λ-astrazioni $\lambda x.M$ e $\lambda y.N$ essi sono α-equivalenti se e solo se sono equivalenti $M[x := y]$ e N, ovvero, dobbiamo applicare ricorsivamente quelle operazioni di rinomina delle variabili legate del primo termine che lo rendono sintatticamente uguale al secondo (sappiamo che esistono e sono in numero finito);
- se i due termini sono nella forma MN e $M'N'$ allora essi saranno α-equivalenti se sono equivalenti i sottotermini M ed M' ed N ed N';
- se i due termini hanno una struttura fra loro incompatibile (ad esempio $\lambda x.M$ e y) essi non saranno mai fra loro α-equivalenti.

A questo punto scrivere il codice è immediato:

$$
\begin{array}{llcl}
\multicolumn{2}{l}{\textit{let rec equivalence } M\ N} & = & \textit{match } (M,N) \textit{ with} \\
 & \mid \ (x,y) & \rightarrow & x == y \\
 & \mid \ (\lambda x.M, \lambda y.N) & \rightarrow & \textit{equivalence } M[x := y]\ N \\
 & \mid \ (MN, M'N') & \rightarrow & (\textit{equivalence } M\ M') \wedge (\textit{equivalence } N\ N') \\
 & \mid \ \_ & \rightarrow & \textit{false}
\end{array}
$$

con tipo: *val equivalence* $: \Lambda \rightarrow \Lambda \rightarrow bool$ dove

$$equivalence\ M\ N = true \iff M =_\alpha N.$$

È possibile dimostrare che la funzione calcolata da questo programma è totale per induzione sulla dimensione dei termini M ed N.

Appendice B

Una dimostrazione del teorema di Church-Rosser

In questa sezione dimostreremo il teorema di Church-Rosser per la $\beta\eta$-riduzione. La dimostrazione per la β-riduzione (senza η) è molto simile ed un po' più semplice, per cui in questa tesi verrà omessa.

B.1 Il Teorema di Church-Rosser

Per prima cosa enunciamo la formulazione estesa del teorema.

Teorema B.1.1 (Church Rosser) *Sia $\twoheadrightarrow$ il simbolo che utilizziamo per denotare*[1] *la relazione $\twoheadrightarrow_\beta$ o $\twoheadrightarrow_{\beta\eta}$. Siano M, N e P λ-termini tali che $M \twoheadrightarrow N$, e $M \twoheadrightarrow P$. Allora esiste un λ-termine Z tale che $N \twoheadrightarrow Z$ e $P \twoheadrightarrow Z$.*

La dimostrazione originale del teorema di Church-Rosser che si trova in [Church and Rosser, 1936] è molto lunga e complicata. Molte altre dimostrazioni e generalizzazioni sono state pubblicate negli ultimi cinquant'anni. La dimostrazione più corta, finora conosciuta, è stata fatta da P. Martin Löf e W. Tait e se ne trova una trattazione in

[1] In modo esclusivo.

[Hindley and Seldin, 1986], noi ne proponiamo una variante ispirata da [Revesz, 1988] e [Selinger, 2001].

B.1.1 Riduzione parallela ad un passo

Cominciamo definendo una nuova relazione binaria $M \rhd M'$ fra termini chiamata *riduzione parallela ad un passo*. Definiamo $\rhd$ come la più piccola relazione che soddisfa le seguenti regole:

$$(1) \qquad \frac{}{x \rhd x}$$

$$(2) \qquad \frac{P \rhd P' \qquad N \rhd N'}{PN \rhd P'N'}$$

$$(3) \qquad \frac{N \rhd N'}{\lambda x.N \rhd \lambda x.N'}$$

$$(4) \qquad \frac{Q \rhd Q' \qquad N \rhd N'}{(\lambda x.Q)N \rhd Q'[x := N']}$$

$$(5) \qquad \frac{P \rhd P', \text{ dove } x \notin FV(P)}{\lambda x.Px \rhd P'}$$

L'idea intuitiva di questo tipo di relazione di riduzione è che, in un solo passo, è possibile ridurre in parallelo più redex all'interno di un termine, cosa che non viene consentita dalla relazione $\rightarrow_\beta$.

Lemma B.1.2

(a) Per ogni $M, M' \in \Lambda$, se $M \rightarrow_{\beta\eta} M'$ allora $M \rhd M'$,

(b) Per ogni $M, M' \in \Lambda$, se $M \rhd M'$ allora $M \twoheadrightarrow_{\beta\eta} M'$,

(c) $\twoheadrightarrow_{\beta\eta}$ è la chiusura riflessiva e transitiva di $\rhd$.

Dim. ***(a)*** Per prima cosa notiamo che vale $P \rhd P$, per ogni λ-termine P (facilmente dimostrabile per induzione strutturale su P). Ora dimostriamo la prima parte del lemma per induzione sulla derivazione di

$$(\beta) \qquad \frac{}{(\lambda x.M)N \to_{\beta\eta} M[x := N]}$$

$$(\eta) \qquad \frac{}{M \to_{\beta\eta} \lambda x.Mx \ \mathit{con}\ x \notin FV(M)}$$

$$(cong_1) \qquad \frac{M \to_{\beta\eta} M'}{MN \to_{\beta\eta} M'N}$$

$$(cong_2) \qquad \frac{N \to_{\beta\eta} N'}{MN \to_{\beta\eta} MN'}$$

$$(\xi) \qquad \frac{M \to_{\beta\eta} M'}{\lambda x.M \to_{\beta\eta} \lambda x.M'}$$

Tabella B.1: Definizione della $\beta\eta$-riduzione ad un passo come la più piccola relazione che soddisfa queste proprietà.

$M \to_{\beta\eta} M'$ (è possibile dare una definizione succinta di $\to_{\beta\eta}$ come la più piccola relazione che soddisfa le regole enunciate in Tabella B.1).

Facciamo una distinzione per casi sull'ultima regola usata nella derivazione di $M \to_{\beta\eta} M'$.

- Se l'ultima regola era (β), allora $M \equiv (\lambda x.Q)N$ e $M' \equiv Q[x := N]$, per qualche Q ed N. Ma allora, per la (4), otteniamo:

  $$\frac{Q \rhd Q \qquad N \rhd N}{M \equiv (\lambda x.Q)N \rhd Q[x := N] \equiv M'}$$

  ovvero $M \rhd M'$;

- se l'ultima regola era (η), allora $M \equiv (\lambda x.Px)$ e $M' \equiv P$, per qualche P tale che $x \notin FV(P)$. Allora dalla regola (5) e usando $P \rhd P$ segue che $M \rhd M'$;

- se l'ultima regola usata era $(cong_1)$ allora $M \equiv PN$ e $M' \equiv P'N$, per un qualche P, P' e N dove $P \twoheadrightarrow_{\beta\eta} P'$. Per ipotesi induttiva,

$P \rhd P'$. Da ciò e dal fatto che $N \rhd N$ ne deriva immediatamente che $M \rhd M'$ per la (2);

- se l'ultima regola usata era ($cong_2$) procediamo in modo analogo all'ultimo caso considerato;
- se l'ultima regola era (ξ), allora $M \equiv \lambda x.N$ e $M' \equiv \lambda x.N'$ per un qualche N e N' tali che $N \twoheadrightarrow_{\beta\eta} N'$. Per ipotesi induttiva $N \rhd N'$ che implica $M \rhd M'$ per la (3).

(b) La dimostrazione viene eseguita attraverso un procedimento induttivo sulla derivazione di $M \rhd M'$. Distinguiamo vari casi, a seconda dell'ultima regola usata nella derivazione.

- Se l'ultima regola era (1), allora $M \equiv M' \equiv x$ ed è chiaro che $x \twoheadrightarrow_{\beta\eta} x$ per simmetria;
- se l'ultima regola era (2), allora $M \equiv PN$ e $M' \equiv P'N'$ per qualche P, P', N, N' con $P \rhd P'$ e $N \rhd N'$. Per ipotesi induttiva $P \twoheadrightarrow_{\beta\eta} P'$ e $N \twoheadrightarrow_{\beta\eta} N'$. Quindi $\twoheadrightarrow_{\beta\eta}$ soddisfa ($cong$), ne segue che $PN \twoheadrightarrow_{\beta\eta} P'N'$, dunque $M \twoheadrightarrow_{\beta\eta} M'$ come desiderato;
- se l'ultima regola era (3), allora $M \equiv \lambda x.N$ e $M' \equiv \lambda x.N'$ per qualche N, N'; con $N \rhd N'$. Per ipotesi induttiva $N \twoheadrightarrow_{\beta\eta} N'$, quindi $M \equiv \lambda x.N \twoheadrightarrow_{\beta\eta} \lambda x.N' \equiv M'$ per la (ξ);
- se l'ultima regola era (4), allora $M \equiv (\lambda x.Q)N$ e $M' \equiv Q'[x := N']$ per qualche Q, Q', N, N'; con $Q \rhd Q'$ e $N \rhd N'$. Per ipotesi induttiva $Q \twoheadrightarrow_{\beta\eta} Q'$ e $N \twoheadrightarrow_{\beta\eta} N'$, quindi $M \equiv (\lambda x.Q)N \twoheadrightarrow_{\beta\eta} (\lambda x.Q')N' \twoheadrightarrow_{\beta\eta} Q'[x := N'] \equiv M'$ come desiderato;
- se l'ultima regola era (5), allora $M \equiv (\lambda x.Px)$ e $M' \equiv P'$ per qualche P, P'; con $P \rhd P'$ e $x \notin FV(P)$. Per ipotesi induttiva $P \twoheadrightarrow_{\beta\eta} P'$ e $M \equiv (\lambda x.Px) \twoheadrightarrow_{\beta\eta} P \twoheadrightarrow_{\beta\eta} P'$, quindi $M \equiv \lambda x.Px \twoheadrightarrow_{\beta\eta} P \twoheadrightarrow_{\beta\eta} P' \equiv M'$ come desiderato.

(c) Questa proprietà segue direttamente da (a) e (b). Scriviamo R^* intendendo la chiusura riflessiva e transitiva della relazione R. Dalla

(a) otteniamo $\to_{\beta\eta} \subseteq \rhd$, quindi $\twoheadrightarrow_{\beta\eta} = \to^*_{\beta} \subseteq \rhd^*$. Per la ($b$) abbiamo che $\rhd \subseteq \twoheadrightarrow_{\beta\eta}$, quindi $\rhd^* \subseteq \twoheadrightarrow^*_{\beta\eta} = \twoheadrightarrow_{\beta\eta}$. Ne segue che $\rhd^* = \twoheadrightarrow_{\beta\eta}$. $\square$

B.2 Lemma di sostituzione

Ora dimostreremo che la relazione $\rhd$ soddisfa la proprietà del diamante. Si noti che, assieme al Lemma B.1.2 (c), questo implica direttamente che $\twoheadrightarrow_{\beta\eta}$ gode della proprietà di Church-Rosser.

Lemma B.2.1 *(di sostituzione). Se $M \rhd M'$ e $U \rhd U'$, allora*

$$M[y := U] \rhd M'[y := U']$$

Dim. Assumiamo, senza perdità di generalità , che ogni variabile legata di M sia differente da y e da ogni occorrenza di variabile libera in U. La proposizione verrà ora dimostrata per induzione sulle derivazioni di $M \rhd M'$. Distinguiamo vari casi, a seconda dell'ultima regola utilizzata nella derivazione.

- Se l'ultima regola era (1), allora $M \equiv M' \equiv x$, per una qualche variabile x. Se $x \equiv y$ allora $M[y := U] = U \rhd U' = M'[y := U']$. Se $x \neq y$ allora, per la (1), $M[y := U] = y \rhd y = M'[y := U']$;

- se l'ultima regola era (2), allora $M \equiv PN$ e $M' \equiv P'N'$ per qualche P, P', N, N'; con $P \rhd P'$ e $N \rhd N'$. Per ipotesi induttiva $P[y := U] \rhd P'[y := U']$ e $N[y := U] \rhd N'[y := U']$, quindi, per la (2), $M[y := U] = P[y := U]N[y := U] \rhd P'[y := U']N'[y := U'] = M'[y := U']$;

- se l'ultima regola era (3), allora $M \equiv \lambda x.N$ e $M' \equiv \lambda x.N'$ per qualche N, N'; con $N \rhd N'$. Per ipotesi induttiva $N[y := U] \rhd N'[y := U']$ quindi, per la (3), $M[y := U] = \lambda x.N[y := U] \rhd \lambda x.N'[y := U'] = M'[y := U']$;

- se l'ultima regola era (4), allora $M \equiv (\lambda x.Q)N$ e $M' \equiv Q'[x := N']$ per qualche Q, Q', N, N'; con $Q \rhd Q'$ e $N \rhd N'$. Per ipotesi induttiva $Q[y := U] \rhd Q'[y := U']$ e $N[y := U] \rhd N'[y := U']$, quindi, per la (4), $(\lambda x.Q[y := U])N[y := U] \rhd Q'[y := U'][x := N'[y := U']] = Q'[x := N'][y := U']$. Quindi $M[y := U] = M'[y := U']$;

- se l'ultima regola era (5), allora $M \equiv (\lambda x.Px)$ e $M' \equiv P'$ per qualche P, P'; con $P \rhd P'$ e $x \notin FV(P)$. Per ipotesi induttiva $P[y := U] \rhd P'[y := U]$, quindi, per la (5), $M[y := U] = \lambda x.P[y := U]x \rhd P'[y := U'] = M'[y := U']$. $\square$

Un modo più concettuale di vedere questa dimostrazione è il seguente: consideriamo ogni possibile derivazione di $M \rhd M'$ ottenuta attraverso gli assiomi $(1) - (5)$. In questa derivazione rimpiazziamo ogni assioma $y \rhd y$ con $U \rhd U'$ e propaghiamo i cambiamenti, ovvero sostituiamo ogni occorrenza di y con U nella parte sinistra e ogni occorrenza di U con U' nella parte destra di ogni $\rhd$. Il risultato è una derivazione di $M[y := U] \rhd M'[y := U']$.

B.3 Proprietà del diamante per la riduzione parallela ad un passo

Il nostro prossimo obiettivo è dimostrare che $\rhd$ gode della proprietà del diamante. Per poterlo fare dobbiamo definire preliminarmente la *riduzione massima in parallelo a un passo* di un termine M che indicheremo con $M^\star$. Al solito la definizione che ne daremo sarà induttiva sulla struttura del termine:

(i) $x^\star = x$,

(ii) $(PN)^\star = P^\star N^\star$, se PN non è un β-redex,

(iii) $((\lambda x.Q)N)^\star = Q^\star[x := N^\star]$,

(iv) $(\lambda x.N)^\star = \lambda x.N^\star$, se $\lambda x.N$ non è un η-redex,

(v) $(\lambda x.Px) = P^\star$ se $x \notin FV(P)$.

Si osservi che $M^\star$ dipende unicamente da M. Il lemma che segue implica la proprietà del diamante per $\rhd$.

Lemma B.3.1 *(di massima riduzione in parallelo a un passo) Se $M \rhd M'$, allora $M' \rhd M^\star$.*

Dim. Per induzione sulla grandezza di M. Distinguiamo cinque casi, a seconda dell'ultima regola utilizzata nella derivazione di $M \rhd M'$.

- se l'ultima regola usata era (1) allora $M \equiv M' \equiv x$ anche $M^\star = x$,
- se l'ultima regola era (2), allora $M \equiv PN$ e $M' \equiv P'N'$, dove $P \rhd P'$ e $N \rhd N'$. Per ipotesi induttiva $P' \rhd P^\star$ e $N' \rhd N^\star$. Ci sono due casi:
  - se PN non è un β-redex, allora $M^\star = P^\star N^\star$. Quindi $M' = P'N' \rhd P^\star N^\star = M^\star$ per la (2),
  - se PN è un β-redex allora $P \equiv \lambda x.Q$, allora $M^\star = Q^\star[x := N^\star]$. Distinguiamo due sottocasi, a seconda dell'ultima regola di derivazione utilizzata in $P \rhd P'$:
    * se l'ultima regola usata era la (3) allora $P' \equiv \lambda x.Q'$ con $Q \rhd Q'$. Per ipotesi induttiva sappiamo che $Q' \rhd Q^\star$, quindi $N' \rhd N^\star$ da cui ne deriva che $M' = (\lambda x.Q')N' \rhd Q^\star[x := N^\star] = M^\star$ per la (4),
    * se l'ultima regola era la (5) allora $P \equiv \lambda x.Rx$ e $P' \equiv R'$ con $x \notin FV(R)$ e $R \rhd R'$. Consideriamo il termine $Q \equiv Rx$. Possiamo dire che $Rx \rhd R'x$ e che Rx è un sottotermine di M. Allora, per ipotesi induttiva, $R'x \rhd (Rx)^\star$. Per il lemma di sostituzione $M' \equiv R'N' = (R'x)[x := N'] \rhd (Rx)^\star[x := N^\star] = M^\star$;
- se l'ultima regola usata era la (3), allora $M \equiv \lambda x.N$ e $M' \equiv \lambda x.N'$ dove $N \rhd N'$. A questo punto sono possibili due casi:

– se M non è un η-redex, allora $M^\star = \lambda x.N^\star$ e, per ipotesi induttiva, $N' \rhd N^\star$, quindi $M' \rhd M^\star$ per la (3),

– se M è un η-redex, allora $N \equiv Px$ con $x \notin FV(P)$. In questo caso $M^\star = P^\star$. Distinguiamo due sottocasi, a seconda della regola usata nella derivazione di $N \rhd N'$:

  * se l'ultima regola era la (2), allora $N' \equiv P'x$, con $P \rhd P'$. Per ipotesi induttiva sappiamo che $P' \rhd P^\star$, quindi $M' = \lambda x.P'x \rhd P^\star = M^\star$ per la (5),
  * se l'ultima regola era la (4), allora $P \equiv \lambda y.Q$ e $N' \equiv Q'[y := x]$ dove $Q \rhd Q'$, Allora $M' = \lambda x.Q'[y := x] = \lambda y.Q'$ (si osservi che $x \notin FV(Q')$). Ma $P \rhd \lambda y.Q'$ quindi, per ipotesi induttiva, otteniamo $\lambda y.Q' \rhd P^\star$ e per la (5) $P^\star = M^\star$;

- se l'ultima regola era la (4) allora $M \equiv (\lambda x.Q)N$ e $M' \equiv Q'[x := N']$ dove $Q \rhd Q'$ e $N \rhd N'$. Allora $M^\star = Q^\star[x := N^\star]$ e $M' \rhd M^\star$ per il lemma di sostituzione;

- se l'ultima regola era la (5) allora $M \equiv \lambda x.Px$ e $M' \equiv P'$ dove $P \rhd P'$ e $x \notin FV(P)$. Allora $M^\star = P^\star$ e, per ipotesi induttiva, $P' \rhd P^\star$. Quindi $M' \rhd M^\star$. $\Box$

Il lemma precedente implica direttamente che $\rhd$ gode della proprietà del diamante.

Lemma B.3.2 *(Proprietà del diamante per $\rhd$). Se $M \rhd N$ e $M \rhd P$, allora esiste un termine Z tale che $N \rhd Z$ e $P \rhd Z$.*

Dim. È sufficiente considerare $Z \equiv M^\star$. $\Box$

B.4 Una dimostrazione del teorema di Church-Rosser

Ora possiamo finalmente enunciare la dimostrazione del teorema di Church-Rosser.

Teorema B.4.1 (Church Rosser) *Sia* $\twoheadrightarrow$ *il simbolo che utilizziamo per denotare la relazione* $\twoheadrightarrow_\beta$ *o* $\twoheadrightarrow_{\beta\eta}$*. Siano* M, N *e* P λ*-termini tali che* $M \twoheadrightarrow N$*, e* $M \twoheadrightarrow P$*. Allora esiste un* λ*-termine* Z *tale che* $N \twoheadrightarrow Z$ *e* $P \twoheadrightarrow Z$*.*

Dim. Poiché $\rhd$ soddisfa la proprietà del diamante, ne segue che anche la sua chiusura riflessiva e transitiva $\rhd^*$ soddisfa la proprietà del diamante ma $\rhd^*$ è uguale a $\twoheadrightarrow_{\beta\eta}$, come visto nel Lemma B.1.2 (c), e il fatto che $\twoheadrightarrow_{\beta\eta}$ goda della proprietà del diamante coincide col dire che vale la proprietà di Church-Rosser per $\rightarrow_{\beta\eta}$. $\Box$

Appendice C

Potenza delle topologie

Riportiamo in questa appendice una dimostrazione topologica di un teorema piuttosto noto[1] nell'ambito della teoria elementare dei numeri per fornire un esempio di come, con poche definizioni di carattere topologico, si possano eseguire dimostrazioni estremamente formali ed eleganti.

C.1 Esistenza di infiniti numeri primi

In questa sezione introduciamo e dimostriamo il teorema sull'esistenza di infiniti numeri primi.

Teorema C.1.1 *Esiste un numero infinito di numeri primi.*

Dim. Introduciamo, per prima cosa, una topologia[2] $\mathbf{P}$ sull'insieme $\mathbb{Z}$ dei numeri interi ponendo:

$$U \in \mathbf{P} \iff \text{per ogni } a \in U \text{ esiste un } t \in \mathbb{Z} \setminus \{0\} \text{ con } a + t\mathbb{Z} \subseteq U$$

si noti che con $t\mathbb{Z}$ intendiamo l'insieme $t\mathbb{Z} = \{tx \mid x \in \mathbb{Z}\}$ ovvero l'insieme di tutti i multipli di t e con $a + t\mathbb{Z}$ intendiamo $\{(a + tx) \mid x \in \mathbb{Z}\}$.

[1] Un teorico dei numeri che leggesse questa tesi farebbe notare che "piuttosto noto" è da considerarsi un eufemismo in quanto tale teorema è conosciuto fin dai tempi di Pitagora.

[2] La cosiddetta *topologia profinita*.

Vogliamo, innanzitutto, essere sicuri che $\mathbf{P}$ sia una topologia su $\mathbb{Z}$. Per questo dobbiamo verificare le tre condizioni della topologia come enunciate in 3.1.1 a pagina 53. Ora, è evidente che $\emptyset, \mathbb{Z} \in \mathbf{P}$. Inoltre se $U_1, U_2 \in \mathbf{P}$ e se $a \in U_1 \cap U_2$ allora esistono $t_1, t_2 \in \mathbb{Z}$ con $a + t_1\mathbb{Z} \subseteq U_1, a + t_2\mathbb{Z} \subseteq U_2$. Da questo segue il fatto che $a + (t_1t_2)\mathbb{Z} \subseteq U_1 \cap U_2$ e quindi $U_1 \cap U_2 \in \mathbf{P}$. È, infine, ovvio che l'unione arbitraria di elementi di $\mathbf{P}$ è ancora contenuta in $\mathbf{P}$.

Continuiamo la dimostrazione osservando che $t\mathbb{Z}, \mathbb{Z} \setminus t\mathbb{Z} \in \mathbf{P}$ ovvero $t\mathbb{Z}$ è aperto e chiuso nella topologia profinita. Dal fatto che $t\mathbb{Z}$ è un aperto ne deriva che $(\forall b \in t\mathbb{Z})(\exists t_2 \in \mathbb{Z} \setminus \{0\}).(b + t_2\mathbb{Z} \subseteq t\mathbb{Z})$. Si osservi che, poiché b appartiene all'insieme $t\mathbb{Z}$, esso sarà nella forma $t \cdot a$; si noti, inoltre, che è possibile prendere come t_2 proprio t: da questo possiamo concludere che $(ta) + t\mathbb{Z} \subseteq t\mathbb{Z}$. Per vedere che $t\mathbb{Z}$ è un chiuso è sufficiente notare che $\mathbb{Z} \setminus t\mathbb{Z} = \bigcup_{s=1,\ldots,t-1} s + t\mathbb{Z}$, cioè che il suo complemento è unione di insiemi aperti e, di conseguenza, un aperto.

Siamo ora pronti per completare la dimostrazione. Assumiamo per assurdo che il sottoinsieme $Primes \subseteq \mathbb{Z}$ di tutti i numeri primi in $\mathbb{Z}$ sia finito. Allora l'unione $\bigcup_{p \in Primes} p\mathbb{Z}$ è un chiuso. Ciò implica che

$$\mathbb{Z} \setminus \bigcup_{p \in Primes} p\mathbb{Z}$$

è un aperto. Ma

$$\mathbb{Z} \setminus \bigcup_{p \in Primes} p\mathbb{Z} = \{1, -1\}$$

in quanto 1 e -1 non ammettono decomposizione in fattori primi, ma $\{1, -1\}$ non è un aperto della topologia profinita. Questo assurdo termina la dimostrazione. $\square$

Appendice D

Teoria Classica Della Computabilità

Facendo riferimento più volte in questo elaborato a tematiche e teoremi della teoria classica della computabilità sembra doveroso riportare in appendice quantomeno i risultati più importanti. Si osservi innanzitutto che, scelto un linguaggio di programmazione Turing equivalente, è possibile trovare una funzione di codifica effettiva, ovvero calcolabile e totale, che associ ad ogni numero naturale un programma inteso come stringa sintatticamente corretta espressa in tale linguaggio. Nel seguito della trattazione scriveremo P_n intendendo il programma associato tramite tale codifica al numero natuale n e Φ_n intendendo la funzione calcolata dal programma P_n. Scriveremo, inoltre, $f \downarrow x$ intendendo che la funzione f è definita sul naturale x mentre, con la notazione $f \uparrow x$, indicheremo che la funzione f è indefinita sul naturale x.

D.1 Il problema della fermata

Una funzione f viene detta calcolabile se esiste un programma P_f che la calcola, ovvero un programma tale che, per ogni input x, P_f non termini se $x \notin dom(f)$ o, in caso contrario, termini con output $f(x)$. Come vedremo è possibile dimostrare che esistono funzioni non calcolabili. Un esempio di funzione non calcolabile di rilevante importanza nella teoria classica della computabilità è quello conosciuto in letteratura con il nome di *problema della fermata*.

Teorema D.1.1 *(Problema della fermata) La funzione* $H : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *così definita:*

$$H(x, y) = \begin{cases} 1 & \textit{se } P_x \downarrow y \\ 0 & \textit{se } P_x \uparrow y \end{cases}$$

non è calcolabile.

Dim. Una dimostrazione di tale teorema può essere fatta ragionando per assurdo. Supponiamo, infatti, che la funzione H appena definita sia calcolabile. Allora anche la funzione

$$h(x) = \begin{cases} 1 & \text{se } P_x \downarrow x \\ 0 & \text{se } P_x \uparrow x \end{cases}$$

è calcolabile in quanto $h(x) = H(x, x)$ e H è, per costruzione, una funzione totale. Ma, se h è calcolabile, lo è anche la funzione g definita come segue:

$$g(x) = \begin{cases} \uparrow & \text{se } P_x \downarrow x \\ 0 & \text{se } P_x \uparrow x \end{cases}$$

Sia dunque $k \in \mathbb{N}$ l'indice del programma P_k che calcola la funzione $\Phi_k = g$. Consideriamo ora

$$g(k) = \Phi_k(k) = \begin{cases} \uparrow & \text{se } P_k \downarrow k \\ 0 & \text{se } P_k \uparrow k. \end{cases}$$

Supponiamo $P_k \downarrow k$ ovvero $\Phi_k \downarrow k$ ma allora, per definizione di g, dovremmo avere $\Phi_k \uparrow k$; supponiamo allora $P_k \uparrow k$ ovvero $\Phi_k(k) = 0$, questo implica che $\Phi_k \downarrow k$, siamo dunque arrivati ad una contraddizione. Da questo assurdo ne deriva che la funzione g non è calcolabile il che implica che neppure h e H sono calcolabili. $\square$

D.2 Il Teorema del parametro

In questa sezione enunceremo il *teorema del parametro*. Esso viene spesso chiamato in letteratura *teorema di parametrizzazione* in quanto mostra che un indice di una funzione computabile può essere trovato in modo effettivo a partire da un parametro da cui dipende.

Teorema D.2.1 *Teorema s-m-n (o del parametro, forma semplice).*
Sia data una funzione calcolabile $f : \mathbb{N} \times \mathbb{N} \rightharpoonup_p \mathbb{N}$ allora esiste una funzione s calcolabile e totale tale che per ogni $x, y \in \mathbb{N}$:

$$\Phi_{s(x)}(y) = f(x, y).$$

La dimostrazione viene qui omessa, chi fosse interessato può consultare [Cutland, 1980], Teorema 4.1. Una forma più generale del teorema s-m-n è la seguente.

Teorema D.2.2 *Teorema s-m-n (o del parametro). Sia data una funzione calcolabile $f : \mathbb{N}^{(n+m)} \rightharpoonup_p \mathbb{N}$ allora esiste, per ogni $n, m > 0$, una funzione $s : \mathbb{N}^n \to \mathbb{N}$ calcolabile e totale tale che per ogni $x, y \in \mathbb{N}$:*

$$\Phi_{s(x_0,\ldots,x_{n-1})}(y_0, \ldots, y_{m-1}) = f(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}).$$

D.3 Decidibilità, indecidibilità e semidecidibilità

La problematica centrale della teoria classica della computabilità è cercare di stabilire per quali problemi è possibile scrivere un programma che li risolve e per quali è impossibile. Intuitivamente tutti i problemi formulabili sono rappresentabili come funzioni parziali definite da $\mathbb{N}$ in $\mathbb{N}$, vogliamo, quindi, capire quali funzioni sono computabili algoritmicamente e quali no. In questa sezione introdurremo i concetti di *decidibilità*, *indecidibilità* e *semidecidibilità* che sono nozioni che in qualche modo determinano il grado di complessità dei problemi che vogliamo risolvere.

Definizione D.3.1 *Un insieme I viene detto* decidibile *se, dato un elemento x, è possibile decidere algoritmicamente la sua appartenenza ad I, ovvero se la funzione caratteristica di I*

$$f_I(x) = \begin{cases} 1 & \textit{se } x \in I, \\ 0 & \textit{altrimenti.} \end{cases}$$

è calcolabile.

Definizione D.3.2 *Un insieme $I \subseteq \mathbb{N}$ viene detto* semidecidibile *se la funzione semicaratteristica di I*

$$f_I(x) = \begin{cases} 1 & \textit{se } x \in I, \\ \uparrow & \textit{altrimenti.} \end{cases}$$

è calcolabile.

Definizione D.3.3 *Un insieme I viene detto* indecidibile *se non è semidecidibile.*

Un insieme semidecidibile particolarmente interessante in computabilità è l'insieme $K \subseteq \mathbb{N}$ definito come segue:

$$K = \{x \mid P_x \downarrow x\}.$$

Si noti che K^C (dove con K^C indichiamo il complemento dell'insieme K rispetto all'insieme universo che, in questo capitolo, si sottointende essere $\mathbb{N}$, ovvero $K^C = \mathbb{N} \setminus K$) è un insieme non semidecidibile in quanto semidecidere l'appartenenza di un elemento a K^C equivarrebbe a risolvere il problema della fermata.

Definizione D.3.4 *Siano dati due insiemi $A, B \subseteq \mathbb{N}$ allora diremo che l'insieme A è Turing-riducibile all'insieme B, e scriveremo $A \leq_T B$ se e solo se esiste una funzione calcolabile e totale $s : \mathbb{N} \to \mathbb{N}$ tale che*

- $x \in A \Rightarrow s(x) \in B$,
- $x \in A^C \Rightarrow s(x) \in B^C$.

Intuitivamente l'operazione di Turing-riducibilità ci permette di *trasferire*, nel senso che vedremo ora, la proprietà di essere decidibile (o semidecidibile) da un insieme ad un altro. Vale, infatti, la seguente proposizione.

Teorema D.3.5 *Siano dati due insiemi $A \subseteq \mathbb{N}$ e $B \subseteq \mathbb{N}$ tali che $A \leq_T B$. Allora:*

(i) se B è decidibile allora A è decidibile,

(ii) se B è semidecidibile allora A è semidecidibile,

(iii) se A non è decidibile allora anche B non è decidibile,

(iv) se A non è semidecidibile allora anche B non è semidecidibile.

Definizione D.3.6 *Un insieme $A \subseteq \mathbb{N}$ viene detto* ricorsivamente enumerabile, *in breve r.e., se esiste un algoritmo che enumera gli elementi dell'insieme A, ovvero se esiste una funzione g calcolabile e totale tale che $A = cod(g)$.*

La definizione di insieme ricorsivamente enumerabile può essere estesa anche ad un insieme A arbitrario in questo senso.

Definizione D.3.7 *Sia A un insieme arbitrario e $f : \mathbb{N} \rightarrow A$ una funzione totale e suriettiva. Un sottoinsieme $B \subseteq A$ viene detto* ricorsivamente enumerabile, *in breve r.e., se l'insieme $f^{-1}[B]$ è ricorsivamente enumerabile.*

Si noti che il concetto di insieme ricorsivamente enumerabile è equivalente a quello di insieme semidecidibile. Vale, quindi, il seguente teorema.

Teorema D.3.8 *$A \subseteq \mathbb{N}$ è un insieme ricorsivamente enumerabile se e solo se A è semidecidibile.*

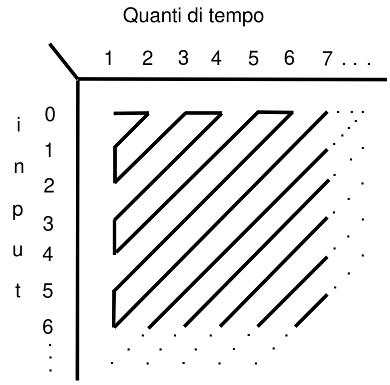


Figura D.1: Rappresentazione del metodo per spazzare il piano Tempo/Input. Si suppone il tempo diviso in quanti e si esegue un programma P sull'input x per n quanti di tempo dove la coppia (n, x) varia nel piano Tempo/Input seguendo la linea tracciata in figura.

Dim. ($\Leftarrow$) Supponiamo A semidecidibile, allora, dal fatto che la funzione semicaratteristica di A sia calcolabile, ne deriva che anche la funzione

$$f(x) = \begin{cases} x & \text{se } x \in A, \\ \uparrow & \text{altrimenti.} \end{cases}$$

è calcolabile. Chiamiamo P_f il programma che calcola f e consideriamo il programma che esegue P_f su input n per m quanti di tempo dove la coppia (m, n) varia nel piano Tempo/Input seguendo la linea tracciata in Figura D.1 e, quando P_f termina con risultato k, enumera k. Questo programma enumera con ripetizione l'intero insieme A che, quindi, è ricorsivamente enumerabile.
($\Rightarrow$) Supponiamo A ricorsivamente enumerabile, allora esiste una funzione g, calcolabile e totale, con $dom(g) = A$. Chiamiamo P_g un programma che calcola g. Per dimostrare che A è semidecidibile è necessario trovare un algoritmo che calcoli la funzione semicaratteristica di A. L'algoritmo che cerchiamo è il seguente: si considera una variabile n inizializzata a 0, si esegue il programma P_g sull'input n e si confronta l'output con x. In caso siano uguali si restituisce 1, se invece sono di-

versi si incrementa n e si ripete il test. Se $x \in A$ esiste un k tale che il programma P_g, eseguito su input k, restituisca x e quindi l'algoritmo descritto termina con output 1, non termina se $x \notin A$. Il programma descritto calcola, quindi, la funzione semicaratteristica di A. $\Box$

Un altro risultato che viene spesso utilizzato in questo elaborato è il seguente.

Teorema D.3.9 *Siano $A \subseteq \mathbb{N}$ e $B \subseteq \mathbb{N}$ due insiemi ricorsivamente enumerabili, allora anche $A \cap B$ e $A \cup B$ sono ricorsivamente enumerabili.*

Dim. Poiché A, B sono ricorsivamente enumerabili le funzioni semicaratteristiche di questi due insiemi, che denoteremo con f_A e f_B rispettivamente, sono calcolabili. Allora possiamo costruire

$$f_{A\cap B}(x) = \begin{cases} 1 & \text{se } (f_A \downarrow x) \wedge (f_B \downarrow x) \\ \uparrow & \text{altrimenti} \end{cases}$$

e

$$f_{A\cup B}(x) = \begin{cases} 1 & \text{se } (f_A \downarrow x) \vee (f_B \downarrow x) \\ \uparrow & \text{altrimenti} \end{cases}$$

che sono le funzioni caratteristiche di $A \cap B$ e $A \cup B$ e sono calcolabili. $\Box$

Definizione D.3.10 *Un insieme $A \subseteq \mathbb{N}$ viene detto* co-ricorsivamente enumerabile*, in breve co-r.e., se l'insieme A^C è ricorsivamente enumerabile.*

D.4 I teoremi di Rice

Si noti che è possibile definire una relazione $\sim \subseteq \mathbb{N} \times \mathbb{N}$ tale che

$$n \sim m \iff \Phi_n = \Phi_m$$

e che $\sim$ è una relazione di equivalenza che induce una partizione di $\mathbb{N}$. Intuitivamente una classe di equivalenza contiene tutti gli indici dei programmi che calcolano la stessa funzione. Diremo che un insieme $I \subseteq \mathbb{N}$

rispetta le funzioni se è possibile costruirlo tramite l'unione di queste classi di equivalenza ovvero se vale $(\Phi_x = \Phi_y) \Rightarrow ((x, y \in I) \vee (x, y \in I^C))$.

Teorema D.4.1 *(Primo teorema di Rice) Sia $I \subseteq \mathbb{N}$ tale che $I \neq \emptyset, \mathbb{N}$ e I rispetta le funzioni, allora l'insieme I è non decidibile e, se gli indici dei programmi che calcolano la funzione vuota appartengono a I (risp. I^C), allora I (risp. I^C) è non semidecidibile.*

Dim. Poiché l'insieme I rispetta le funzioni tutti gli indici dei programmi che calcolano la funzione vuota saranno elementi di I o di I^C. Supponiamo, quindi, senza perdita di generalità, che $\{ x \mid \Phi_x = f_\emptyset\} \subseteq I$ e, fissato un elemento $z \in I^C$, costruiamo la funzione f come segue

$$f(x, y) = \begin{cases} \Phi_z(y) & \text{se } x \in K, \\ \uparrow & x \in K^C. \end{cases}$$

Si noti che, essendo f computabile, è possibile applicare il teorema del parametro; esisterà dunque una funzione s calcolabile e totale tale che $\Phi_{s(x)}(y) = f(x, y)$. Ciò ci permette di affermare che

- $x \in K \Rightarrow \Phi_{s(x)}(y) = \Phi_z(y) \Rightarrow s(x) \in I^C$,
- $x \in K^C \Rightarrow \Phi_{s(x)}(y) = f_\emptyset \Rightarrow s(x) \in I$,

ovvero vale $K \leq_T I^C$ e $K^C \leq_T I$, quindi I è un insieme non semidecidibile. □

Teorema D.4.2 *(Secondo teorema di Rice) Sia $I \subseteq \mathbb{N}$ tale che $I \neq \emptyset, \mathbb{N}$ e I rispetta le funzioni. Supponiamo che esistano due funzioni calcolabili f e g tali che:*

(i) gli indici dei programmi che calcolano f sono contenuti in I ovvero $\{x \mid \Phi_x = f\} \subseteq I$,

(ii) gli indici dei programmi che calcolano g sono contenuti in I^C ovvero $\{x \mid \Phi_x = g\} \subseteq I^C$,

(iii) $f \leq g$, dove $\leq$ è la relazione introdotta nella Definizione 4.1.1 a pagina 74,

allora I non è semidecidibile.

Dim. Definiamo la funzione

$$t(x,y) = \begin{cases} g(x) & \text{se } x \in K, \\ f(y) & \text{se } x \notin K. \end{cases}$$

Si noti che la funzione t è calcolabile nel seguente modo: si esegue in parallelo il programma che semidecide l'appartenenza di x a K e il programma che calcola la funzione f sull'input y (esso esiste in quanto f calcolabile per ipotesi). Si possono verificare tre casi:

- termina il programma che semidecide l'appartenenza di x all'insieme K. Allora $x \in K$ e si può eseguire il programma che calcola la funzione g su input y;
- termina prima il programma che calcola f su input y con, come risultato, $f(y)$. Poiché la funzione g estende la funzione f non è necessario attendere la fine della computazione del programma che calcola g sull'input y in quanto il risultato dev'essere il medesimo. Possiamo quindi fornire come output $f(y)$;
- nessuno dei due programmi termina. Allora $x \notin K$ ed anche il programma che calcola t, come richiesto, non termina.

Essendo, dunque t calcolabile possiamo applicare il Teorema del parametro; esisterà dunque una funzione s calcolabile e totale tale che $\Phi_{s(x)}(y) = t(x,y)$. Possiamo dunque affermare che

- $x \in K \Rightarrow \Phi_{s(x)} = g \Rightarrow s(x) \in I^C$,
- $x \in K^C \Rightarrow \Phi_{s(x)} = f \Rightarrow s(x) \in I$,

ovvero vale $K^C \leq_T I$, questo ci permette di concludere che I non è semidecidibile. $\square$

Teorema D.4.3 *(Terzo teorema di Rice) Sia $I \subseteq \mathbb{N}$ tale che $I \neq \emptyset, \mathbb{N}$ e I rispetta le funzioni. Supponiamo che esista una funzione f calcolabile tale che:*

(i) gli indici dei programmi che calcolano f sono contenuti in I ovvero $\{x \mid \Phi_x = f\} \subseteq I$,

(ii) gli indici dei programmi che calcolano un'approssimazione finita di f sono contenuti in I^C ovvero per ogni funzione θ con dominio finito tale che $\theta \leq f$ vale $\{x \mid \Phi_x = \theta\} \subseteq I^C$,

allora I non è semidecidibile.

Dim. Si definisca la funzione

$$t(x,y) = \begin{cases} f(y) & \text{se } P_x \uparrow x \text{ in meno di } y \text{ passi} \\ & \text{di computazione,} \\ \uparrow & \text{altrimenti.} \end{cases}$$

Tale funzione è calcolabile, possiamo dunque applicare il teorema del parametro. Esiste quindi una funzione s calcolabile e totale tale che

$$\Phi_{s(x)}(y) = t(x,y) = \begin{cases} f(y) & \text{se } P_x \uparrow x \text{ in meno di } y \text{ passi di} \\ & \text{computazione,} \\ \uparrow & \text{altrimenti.} \end{cases}$$

Ciò ci permette di affermare che

(i) $x \in K \Rightarrow s(x)$ è l'indice di un programma che calcola un'approssimazione finita di $f \Rightarrow s(x) \in I^C$,

(ii) $x \in K^C \Rightarrow s(x)$ è l'indice di un programma che calcola la funzione $f \Rightarrow s(x) \in I$,

ovvero vale $K^C \leq_T I$, questo ci permette di concludere che I non è semidecidibile. □

D.5 I teoremi di Ricorsione

In questo paragrafo termineremo lo studio delle funzioni ricorsive enunciando e dimostrando i teoremi di ricorsione. È necessario, però, fornire alcune definizioni preliminari e i teoremi che utilizzeremo in seguito.

Definizione D.5.1

- *Una funzione h viene detta* estensionale *se*

$$\phi_x = \phi_y \;\Rightarrow\; \phi_{h(x)} = \phi_{h(y)} \quad \textit{per ogni } x, y \in \mathbb{N},$$

- *se h è una funzione calcolabile, totale ed estensionale il funzionale ρ definito come $\rho(\Phi_x) = \Phi_{h(x)}$ viene detto* funzionale ricorsivo*,*

- *una funzione g tale che $\rho(g) = g$ viene chiamata* punto fisso *di ρ.*

Teorema D.5.2 *Dato un insieme $I \neq \mathbb{N}, \emptyset$ che rispetta le funzioni se I è semidecidibile allora*

$$x \in I \Rightarrow \textit{esiste } g, \textit{ funzione finita, tale che } (g \leq \Phi_x) \wedge (\{y \mid \Phi_y = g\}).$$

Dim. Consideriamo un arbitrario $x \in I$ che sarà l'indice di un programma che computa una qualche funzione f, allora, poiché I rispetta le funzioni, si ha che $\{y \mid \Phi_z = g\}$. Supponiamo, ora, per assurdo che non esista alcuna funzione finita $g \leq \Phi_x$ tale che $\{y \mid \Phi_z = g\} \subseteq I$, allora $\{y \mid \Phi_z = g\} \subseteq I^C$. In questo caso, però, varrebbero le ipotesi del terzo teorema di Rice in quanto gli indici dei programmi che calcolano la funzione g appartengono ad I e tutti gli indici dei programmi che calcolano le restrizioni finite di g appartengono al complementare, l'insieme I sarebbe, dunque, non semidecidibile. Questo ci porta ad un assurdo. □

Prima di enunciare i teoremi di ricorsione introduciamo e dimostriamo il seguente lemma che si rivelerà di cruciale importanza nelle dimostrazioni che seguiranno.

Lemma D.5.3 *Sia h una funzione calcolabile, totale ed estensionale, allora $\rho(f)(x) = y$ se e solo se esiste una funzione θ, restrizione finita di f, tale che $\rho(\theta)(x) = y$.*

Dim. La funzione f è, per ipotesi, calcolabile, esiste dunque un indice z tale che il programma P_z calcola f e $\Phi_z = f$. Essendo h calcolabile, totale ed estensionale ρ è un funzionale ricorsivo e, per definizione di funzionale ricorsivo, vale

$$\rho(\Phi_x) = \Phi_{h(z)}.$$

Consideriamo ora l'insieme $A = \{z \mid \rho(\Phi_z)(x) = y\}$, si verifica banalmente che tale insieme rispetta le funzioni poiché h è estensionale, è diverso da $\emptyset$ in quanto contiene almeno gli indici dei programmi che calcolano f ed è diverso da $\mathbb{N}$ perché gli indici dei programmi che calcolano la funzione $f_\emptyset$ sono contenuti in A^C. È possibile pertanto applicare il primo teorema di Rice e affermare che A non è decidibile e A^C è non semidecidibile. A è, però, semidecidibile in quanto esiste un algoritmo che semidecide l'appartenenza di un naturale z ad A. Si può costruire, infatti, un programma che prende in input z e calcola $h(z)$, lo decodifica ottenendo il programma $P_{h(z)}$ e lo esegue facendo variare input e tempo di esecuzione spazzando il piano come in Figura D.1. A questo punto confronta gli output ottenuti con il numero y. Tale programma termina con output 1 se esiste un n tale che $P_{h(z)}$ calcolato in n termina con output y, non termina altrimenti. Poiché $A \neq \emptyset, \mathbb{N}$ ed è semidecidibile esiste, per il Teorema D.5.2, una funzione θ, restrizione finita di f, tale che $\{y \mid \Phi_y = \theta\} \subseteq A$, e ciò equivale a dire che $\rho(\theta)(x) = y$. $\square$

D.5.1 Il Primo Teorema di Ricorsione

Il primo teorema di ricorsione, altresì detto teorema di Kleene, è un teorema del punto fisso per i funzionali ricorsivi. In questo paragrafo lo enuncieremo e ne forniremo la dimostrazione classica.

Teorema D.5.4 *(Teorema di Kleene o* primo teorema di ricorsione*) Sia $h : \mathbb{N} \to \mathbb{N}$ una funzione computabile e totale che soddisfi la seguente proprietà di estensionalità:*

$$\phi_x = \phi_y \ \Rightarrow\ \phi_{h(x)} = \phi_{h(y)} \quad \textit{per ogni } x, y \in \mathbb{N}.$$

Allora il funzionale ρ, definito da:

$$\rho(\phi_x) = \phi_{h(x)} \quad \textit{per ogni } x \in \mathbb{N},$$

ammette un minimo punto fisso computabile, ovvero esiste una funzione calcolabile $fix : \mathbb{N} \to \mathbb{N}$ *tale che*

(i) $\rho(fix) = fix$*, cioè* $(\forall z \in \mathbb{N}).(\Phi_z = fix \Rightarrow \Phi_z = \Phi_{h(z)})$*,*

(ii) se $\rho(g) = g$ *allora* $fix \leq g$*, ovvero* fix *è il più piccolo punto fisso dell'equazione ricorsiva,*

(iii) esiste un metodo effettivo per calcolare fix*.*

Dim. (iii) Costruisco una successione di funzioni $f_0, f_1, \ldots$ tale che

$$\begin{array}{l} f_0 = f_\emptyset, \\ f_1 = \rho(f_0), \\ f_2 = \rho(f_1), \\ \vdots \\ f_{n+1} = \rho(f_n), \\ \vdots \end{array}$$

Mostriamo, per induzione, che la successione $\{f_n\}_{n\in\mathbb{N}}$ è crescente, ovvero che

$$f_0 \leq f_1 \leq \ldots \leq f_n \leq \ldots$$

Caso base: se $n = 0$ allora $f_0 \leq f_1$ in quanto $f_0 = f_\emptyset$ che è minore di ogni altra funzione.
Passo induttivo: se $n > 0$ allora, per ipotesi induttiva, possiamo supporre $f_0 \leq f_1 \leq \ldots \leq f_n$ e mostrare che vale anche $f_n \leq f_{n+1}$. Supponiamo esista un naturale x tale che $f_n(x) = y \Rightarrow \rho(f_{n-1})(x) = y$, allora, per il Lemma D.5.3, esiste una funzione θ, restrizione finita di f_{n-1}, tale che $\rho(\theta)(x) = y$. Poiché $\theta \leq f_{n-1}$ e $f_{n-1} \leq f_n$ si ha che vale $\theta \leq f_n$. Pertanto θ è una restrizione finita di f_n dunque, applicando il Lemma D.5.3 in verso opposto, si ottiene

$$\rho(f_n)(x) = y$$

e questo implica $f_{n+1}(x) = y$ ovvero $f_n \leq f_{n+1}$. Definiamo, dunque, la funzione $fix = \bigcup_{i=0}^{\infty} f_i$ e mostriamo che fix è calcolabile. Definiamo per ricorsione primitiva una funzione e tale che, per ogni $n \in \mathbb{N}$, $e(n)$ sia l'indice di un programma che calcola f_n, ovvero:

$$e(n) = \begin{cases} k & \text{se } n = 0, \\ h(e(n-1)) & \text{se } n > 0. \end{cases}$$

dove k è l'indice del programma che calcola la funzione vuota $f_\emptyset$. Allora, per come è costruita fix avremo che $fix(x) = y$ se e solo se esiste un $n \in \mathbb{N}$ tale che $\Phi_{e(n)}(x) = y$. Per dire che fix è computabile dobbiamo poter calcolare, per ogni n, $\Phi_{e(n)}(x)$. Ciò è possibile in quanto è possibile scrivere un programma che continui ad eseguire in parallelo $P_{e(n)}(x)$ al variare di $x, n \in \mathbb{N}$. Se esiste un indice ℓ tale che $\Phi_{e(\ell)}(x) = y$ il programma che stiamo descrivendo restituisce y, altrimenti non termina. Questo algoritmo, dunque, calcola fix.
(i) Mostriamo, per prima cosa, che $\rho(fix) \leq fix$. Supponiamo che $\rho(fix)(x) = y$, allora per il Lemma D.5.3 esiste una funzione θ, restrizione finita di fix, tale che $\rho(\theta)(x) = y$. Esiste, allora, un naturale n tale che $\theta \leq f_n$, quindi θ è una restrizione di questa f_n e $\rho(\theta)(x) = y \Rightarrow \rho(f_n)(x) = f_{n+1}(x) = y$. Poiché, però, fix estende ogni funzione f_n vale $f_{n+1} \leq fix$, perciò $fix(x) = y$ e, visto che $\rho(fix)(x) = y$, $\rho(fix) \leq fix$. Dimostriamo ora che $fix \leq \rho(fix)$. Supponiamo $fix(x) = y$, allora esiste un indice n tale che $f_n(x) = y \Rightarrow \rho(f_{n-1})(x) = y$. Esiste dunque, per il Lemma D.5.3 una funzione θ, restrizione finita di f_{n-1} tale che $\rho(\theta)(x) = y$. Poiché $f_{n-1} \leq fix$ si ha che $\theta \leq fix$ pertanto, per il Lemma D.5.3, si ottiene $\rho(fix)(x) = y$ che implica $fix \leq \rho(fix)$. Da ciò si conclude che $\rho(fix) = fix$.
(ii) Dobbiamo mostrare che, se g è un'altra soluzione dell'equazione ricorsiva, ovvero vale $\rho(g) = g$, allora $fix \leq g$. Per far questo procediamo per induzione sull'indice $n \in \mathbb{N}$ delle funzioni f_n.
Caso base: $(n = 0)$, provare che $f_0 \leq g$ è banale in quanto $f_0 = f_\emptyset$.
Passo induttivo: $n > 0$, supponiamo per ipotesi induttiva che $f_{n-1} \leq g$. Se per un qualche x si ha che $f_n(x) = y$, allora, per definizione delle funzioni f_n, si ha che $\rho(f_{n-1}(x)) = f_n(x) = y$. Applicando il Lemma D.5.3 otteniamo che esiste una certa funzione finita θ, restrizione finita di f_{n-1}, tale che $\rho(\theta)(x) = y$. Per ipotesi induttiva sappiamo che $f_{n-1} \leq g$ per cui possiamo concludere $\theta \leq g$. Quindi, dal momento che $\rho(\theta)(x) = y$

avremo anche che $\rho(g)(x) = y$ e, dal momento che $\rho(g) = g$ per ipotesi, possiamo concludere $g(x) = y$. $\square$

D.5.2 Il Secondo Teorema di Ricorsione

Il secondo teorema di ricorsione può sembrare un po' strano, ad una prima lettura, ciò nonostante esso gioca un ruolo importante in molti risultati della teoria classica della computabilità.

Teorema D.5.5 *(Secondo teorema di ricorsione) Sia $f : \mathbb{N} \to \mathbb{N}$ una funzione totale e computabile, allora esiste un naturale n tale che*

$$\Phi_{f(n)} = \Phi_n.$$

Dim. Per il teorema del parametro esiste una funzione calcolabile e totale s tale che, per ogni $x \in \mathbb{N}$,

$$\Phi_{f(\Phi_x(x))}(y) \simeq \Phi_{s(x)}(y). \tag{D.1}$$

Consideriamo ora un qualunque numero m tale che $s = \Phi_m$ ed effettuiamo la sostituzione di s con Φ_m in D.1, ne deriva che:

$$\Phi_{f(\Phi_x(x))}(y) \simeq \Phi_{\Phi_m(x)}(y).$$

Se consideriamo, ora, $x = m$ e prendiamo $n = \Phi_m$ (che è definito poiché Φ_m è totale) abbiamo che vale:

$$\Phi_{f(n)} = \Phi_n$$

come richiesto. $\square$

Indice analitico