Modelli di calcolo, Tesi di Church-Turing

Lezione 25 di Fondamenti di informatica

Docenti: Marina Madonia & Giuseppe Scollo

Università di Catania
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica, I livello, AA 2008-09

Indice

algoritmi e modelli di calcolo

cos'è un **modello di calcolo?**

astrazione di una classe di agenti di calcolo (esecutori di algoritmi)

ingredienti tipici di un modello di calcolo:

interazione ingresso/uscita con l'ambiente

capacità di **memoria** (illimitata)

nucleo essenziale di **primitive di calcolo**

transizioni di stato dipendenti dall'ingresso e dallo stato

N.B. per semplicità, la classe degli algoritmi in considerazione è limitata al calcolo di **funzioni sui numeri naturali**, cioè di tipo $\mathbf{N^m \to N}$

il calcolo di funzioni di altro tipo può ridursi a questo mediante codifica

a che servono i modelli di calcolo?

fondamentalmente: a stabilire cosa **non** è calcolabile (!)

a prefigurare nuovi sistemi di calcolo e paradigmi di programmazione

macchine a registri

un modello di calcolo concettualmente fedele all'architettura Von Neumann:

memoria: una sequenza illimitata di **registri** $R_1, R_2, \ldots, R_n, \ldots$

da cui il nome del modello: ***Unlimited Register Machine*** **(URM)**

istruzioni primitive:

- $Z(n)$ azzera R_n
- $S(n)$ incrementa R_n
- $T(m,n)$ copia R_m in R_n
- $J(m,n,q)$ salta alla q-sima istruzione se R_m e R_n hanno uguale contenuto, altrimenti procedi in sequenza

interazione I/O: mediante (alcuni dei) registri

la classe delle funzioni URM-calcolabili coincide con la classe delle funzioni calcolabili da una Macchina di Turing

calcolabilità e decidibilità

non tutte le funzioni sui numeri naturali sono URM-calcolabili:

l'insieme delle funzioni sui naturali ha la cardinalità del continuo

l'insieme delle URM è numerabile

ogni URM calcola esattamente una funzione (parziale) sui naturali

per la stessa ragione, non miglior sorte tocca alla sottoclasse delle funzioni in questione costituita dalle **funzioni caratteristiche** di appartenenza a un insieme (sottoinsieme di $\mathbf{N}^m$), le quali hanno immagine in {0, 1}

ovvero: non tutti i predicati sui naturali sono URM-decidibili

decidibilità di un insieme = calcolabilità della sua funzione caratteristica

un insieme è **semidecidibile**, ovvero **ricorsivamente enumerabile**, se è calcolabile la sua **funzione caratteristica parziale**, data dalla co-restrizione della funzione caratteristica all'immagine {1}:

i.e. coincide con la caratteristica se questa vale 1, altrimenti è indefinita

macchina di Turing universale

una Macchina di Turing (MdT), modello già noto dalle lezioni precedenti, è essenzialmente determinata dal suo programma, e calcola dunque una (sola) funzione, tuttavia ...

esiste una MdT in grado di calcolare ogni funzione calcolata da qualsiasi MdT, e detta pertanto **MdT Universale (MdTU)**

come è possibile? non c'è qui una palese contraddizione?

il "trucco" sta nell'ingresso della MdTU, la quale **emula** una data MdT: una codifica di questa (ovvero del suo programma) e il suo input costituiscono l'input della MdTU

geniale intuizione del concetto di macchina da calcolo a programma memorizzato, ovvero di calcolatore ***general purpose***

epperò anche la MdTU ha i suoi limiti, non diversi da quelli delle URM

il caso che segue li rivela, in connessione profonda con il paradosso di Russell, il teorema di incompletezza di Gödel e la diagonalizzazione di Cantor

problema dell'arresto delle MdT

problema: decidere, per qualsiasi MdT ***M*** e input ***I***, se l'esecuzione di ***M*** con input ***I*** giunga o meno all'arresto

ipotizziamo per assurdo la decidibilità del problema: esiste allora una MdT ***P*** che, con input ***M*bI***, dà output 1 quando ***M*** con input ***I*** si arresta, 0 in caso contrario, dove ***M**** è la codifica di ***M*** e *b* è il simbolo blank che li separa nel nastro

P, dunque, si arresta in ogni caso

l'arresto di una MdT avviene per mancanza di istruzioni applicabili, possiamo quindi aggiungere opportune istruzioni a ***P*** per ottenere una MdT ***Q*** che

non si arresta mai quando ***M*** con input ***I*** si arresta

si arresta con output 0 quando ***M*** con input ***I*** non si arresta

esiste poi una MdT ***S*** che duplica il contenuto del nastro e poi esegue ***Q*** su tale input

che succede se si esegue ***S*** con input la sua stessa codifica ***S**** ?

dopo la duplicazione di ***S**** sul nastro, ***S*** esegue ***Q*** con input ***S*bS*...*** si traggono due conclusioni alternative, che ricordano quelle del paradosso di Russell:

S con input ***S**** non si arresta quando ***S*** con input ***S**** si arresta con output 0

S con input ***S**** si arresta con output 0 quando ***S*** con input ***S**** non si arresta

qui però non siamo in presenza di un paradosso, bensì di una dimostrazione dell'assurdità dell'ipotesi di decidibilità del problema dell'arresto della MdTU

λ-calcolo

il λ-calcolo, fondamento della programmazione funzionale, fu ideato da A. Church per fondare la matematica sul concetto (intensionale) di **funzione** quale **regola di calcolo**

sintassi dei λ-termini: $E ::= V \mid \lambda V.E \mid (EE)$

dove il non-terminale V produce un insieme infinito di variabili e l'operatore di astrazione lega una variabile nell'ambito di un λ-termine

la giustapposizione designa l'applicazione funzionale; la semantica del λ-calcolo va dunque cercata in un opportuno spazio di funzioni (di ordine superiore), che permetta fra l'altro l'autoapplicazione

analogamente a quanto accade con i quantificatori nella logica predicativa, si definiscono le variabili libere di un λ-termine, la libera sostituibilità di un λ-termine per una variabile in un λ-termine, e si stipula l'α-convertibilità di λ-termini

ovvero l'invarianza semantica per ridenominazione di variabili vincolate

le **regole** del λ-calcolo formalizzano la relazione (binaria) di **β-riduzione** "→" fra λ-termini, che mutua il nome dalla regola:

(β) $(\lambda x.E)M \rightarrow E[x/M]$

dove E[x/M] è il λ-termine ottenuto da E per sostituzione delle occorrenze libere di x con M (dopo l'eventuale ridenominazione di variabili vincolate in E, per garantire la libera sostituibilità)

β-riduzione e combinatori

le altre regole del λ-calcolo formalizzano le proprietà di **riflessività, transitività e chiusura per contesti** della β-riduzione, nonché le proprietà della relazione di equivalenza (invero, congruenza) da essa generata, detta **β-convertibilità**

per l'applicabilità in contesto della β-regola, ossia a sottotermini propri di un dato λ-termine, questo può avere riduzioni distinte; il **Teorema di Church-Rosser** garantisce che ciò non danneggia la semantica funzionale della β-riduzione:

confluenza: se $E \to M$ e $E \to N$, allora esiste F tale che $M \to F$ e $N \to F$

se M e N sono β-convertibili, allora hanno un comune β-ridotto

un **combinatore** è un λ-termine privo di variabili libere; eccone esempi significativi:

$\Omega = (\lambda x.xx)(\lambda x.xx)$

$I = \lambda x.x$, $K = \lambda x.\lambda y.x$, $S = \lambda x.\lambda y \lambda z.(xz)(yz)$

Ω mostra l'esistenza di β-riduzioni infinite; la β-regola in questo caso dà solo $\Omega \to \Omega$

le riduzioni dei combinatori I, K, S, possono essere formalizzate senza ricorso esplicito alla λ-astrazione, ottenendo così un calcolo applicativo **senza variabili vincolate**, detto **logica combinatoria:** $Ix \to x$, $Kxy \to x$, $Sxyz \to (xz)(yz)$

N.B. dove si conviene che l'applicazione associ a sinistra: EMN = (EM)N

λ-calcolabilità

un λ-termine è una **forma normale** se non è suscettibile di β-riduzione

il termine **x** è una forma normale

un λ-termine **ha** una forma normale se è β-riducibile ad una forma normale

il combinatore Ω non ha forma normale

il fatto che un termine abbia forma normale non garantisce che ogni sua β-riduzione termini, tuttavia vale l'**unicità della forma normale:**

se un λ-termine ha una forma normale, allora questa è unica

il calcolo di funzioni sui numeri naturali nel λ-calcolo si rappresenta mediante una rappresentazione dei numeri, e di operazioni elementari quali: successore, condizionale, coppia ordinata e proiezioni, etc., quali **combinatori**

Church (1941) rappresenta i numeri naturali mediante i seguenti combinatori (tutti forme normali): $\lambda f \lambda x.f^n x$, dove $f^0 x = x$ e $f^{n+1} x = f(f^n x)$

una funzione sui naturali è λ-calcolabile se esiste un combinatore che, applicato alle rappresentazioni degli argomenti, β-riduce alla rappresentazione del risultato ogniqualvolta la funzione sia definita

Tesi di Church-Turing

la Tesi asserisce l'equivalenza di **effettiva calcolabilità** di una funzione ed **esistenza di un programma** che la calcola (in uno qualsiasi dei modelli di calcolo visti sopra)

- il concetto di calcolabilità è basato su una nozione inerentemente **informale** di algoritmo, dunque la Tesi non è dimostrabile
- la Tesi è tuttavia generalmente accettata... perché?

la Tesi è corroborata dalle dimostrazioni di **equivalenza dei modelli di calcolo** considerati, e di numerosi altri modelli (v. appresso) proposti in seguito a quelli di Church e Turing

- la prima di tali dimostrazioni è dovuta ad Alonzo Church, ideatore del λ-calcolo, e mostra per l'appunto l'equivalenza di λ-calcolabilità e Turing-calcolabilità

altri modelli di calcolo

schemi di funzioni ricorsive (Gödel, Kleene)

- ricorsione **primitiva**
- ricorsione **generale**
  - funzione di **Ackermann:** primo esempio di funzione ricorsiva (generale) non definibile per ricorsione primitiva

sistemi di Post

- sistemi **canonici**
- sistemi **monogenici**

sistemi di Markov

- variante dei sistemi di Post

modelli di calcolo non convenzionali

- modelli naturali di calcolo, modelli quantistici, ...

esercizi e approfondimenti

1. ideare un programma URM per il calcolo della somma di due numeri
2. se un predicato è semidecidibile, e lo è anche la sua negazione, allora il predicato è decidibile; perché?
3. la composizione di funzioni λ-calcolabili è λ-calcolabile; perché?
4. reperire in rete la definizione della funzione di Ackermann e farsi un'idea della sua velocità di crescita; cosa si può dire in proposito?
5. ciascuno dei modelli di calcolo indicati alla fine della lezione costituisce un tema suscettibile di vari approfondimenti, quali: caratteristiche del modello, rappresentazione del calcolo nel modello, relazione con altri modelli, paradigmi correlati, etc. (né questa lista di tipi di approfondimento, né la lista di modelli proposta, sono esaustive)