Teorema Secondo di Ricorsione: Kleene 2 (K2)

December 4, 2013

Abstract

Dimostrazione e qualche esempio

by MLS

Contents

1 Teorema secondo di ricorsione (K2)

Nella teoria della computabilità, i teoremi di ricorsione di Kleene sono due fondamentali risultati riguardanti l'applicazione di funzioni calcolabili a loro stesse. I teoremi furono dimostrati per la prima volta da Stephen Kleene[1] nel 1938. I due teoremi di ricorsione possono essere applicati per costruire punti fissi di certe operazioni su funzioni calcolabili, per generare quine[2], e per costruire funzioni definite mediante definizione ricorsiva.

Il secondo teorema di ricorsione dice che, presa una qualsiasi funzione totale h calcolabile, esiste sempre un programma n che, tramite la funzione h, viene trasformato in un programma $m = h(n)$ equivalente: $\Phi_n = \Phi_{h(n)}$ a prescindere dal tipo di codifica adottata per associare programmi a numeri naturali.

1.1 Enunciato

- Ipotesi

  Sia h una qualsiasi funzione calcolabile e totale.

- Tesi

  allora esiste un programma n tale che: $\Phi_n = \Phi_{h(n)}$

1.2 Dimostrazione

Consideriamo la seguente funzione: $f(x,y) = \Phi_{h(\Phi_x(x))}(y) = \Phi(h(\Phi(x,x),y)$. Brutta, vero? Ma cerchiamo di analizzarla con calma. $\Phi_x(x)$ è il risultato dell'elaborazione di un programa su sè stesso. Se $x \in K$ questo risultato sarà un numero, altrimenti no. Nel primo caso $h(\Phi_x(x))$ sarà a sua volta un numero, poichè h è calcolabile totale. Di conseguenza $\Phi_{h(\Phi_x(x))}$ sarà la funzione calcolata dal programma $h(\Phi_x(x))$. L'ultimo membro non è nient'altro che la riscrittura del secondo utilizzando una funzione a due variabili Sappiamo che esistono programmi che appartengono a K quindi la funzione $f(x,y)$ non avrà dominio vuoto, anche se per qualche x non sarà definita; $f(x,y)$ è quindi calcolabile e possiamo applicare il teorema del parametro. Esisterà quindi una funzione $S(x)$, calcolabile e totale, per cui

$$\Phi_{S(x)}(y) = f(x,y) = \Phi_{h(\Phi_x(x))}(y) \tag{1}$$

Sia m un programma che calcola $S(x)$ e se applichiamo $S(x)$ ad m abbiamo: $S(m) = \Phi_m(m) = n$.

$$S(m) = \Phi_m(m) = n \tag{2}$$

Ora al posto di x mettiamo m nella espressione (1). Otteniamo:

$$\Phi_{S(m)}(y) = f(m,y) = \Phi_{h(\Phi_m(m))}(y) \tag{3}$$

[1] Stephen Cole Kleene (Hartford, 5 gennaio 1909 – Madison, 25 gennaio 1994) è stato un matematico statunitense che lavorò all' università di Wisconsin-Madison dove predispose le fondamenta dell'informatica teorica. Kleene fu ancor meglio conosciuto per la fondazione del ramo della logica matematica conosciuta come teoria della ricorsione insieme con Alonzo Church, Kurt Gödel, Alan Turing ed altri, e per l'aver inventato le espressioni regolari. Fornendo metodi per determinare quali problemi sono risolvibili, il lavoro di Kleene portò allo studio di quali funzioni fossero calcolabili. Tra le altre cose, l'algebra di Kleene, la star di Kleene, il teorema di ricorsione di Kleene ed il teorema di punto fisso di Kleene sono stati chiamati così in suo onore. Ha anche contribuito all'intuizionismo matematico fondato da Luitzen Egbertus Jan Brouwer.

[2] In informatica, un quine è un algoritmo che riproduce il suo stesso codice sorgente senza usare funzioni di I/O (aprire il file sorgente e stampare il suo contenuto è considerato "barare"). Inserito per la curiosità dell'autore. :P

Infine inserendo il valore n trovato nella (2):

$$\Phi_n(y) = \Phi_{h(n)}(y) \tag{4}$$

che è ciò che si voleva dimostrare.□

2 Considerazioni ed esempi

Consideriamo ad esempio la funzione successore: $f(n) = n + 1$. Questa è chiaramente una funzione totale e calcolabile; possiamo quindi applicare il K2. Esisterà quindi un valore n tale che il programma da cui è codificato e il programma da cui è codificato $n + 1$ calcolano la stessa cosa, cioè: $P_n = P_{n+1}$ e quindi anche: $\Phi_n = \Phi_{n+1}$. Naturalmente non sappiamo cosa faranno i programmi n e $n+1$, potrebbero entrambi divergere, ma il K2 ci garantisce che esiste almeno un n per cui è vero quanto detto.

2.1 K non rispetta le funzioni

Questo ce lo eravamo già detto e dimostrato con considerazioni fatte sulla codifica dei programmi e sul numero di istruzioni.

Vediamo quest'affermazione alla luce di K2.

Cosideriamo la seguente funzione a due variabili: $f(x,y) = \begin{cases} 1 & se\, x = y \\ \uparrow & altrimenti \end{cases}$. Questa funzione è chiaramente calcolabile in quanto si può facilmente decidere se $x = y$.

Applichiamo quindi il teorema del parametro: esiste una funzione calcolabile e totale $S(x)$ tale che $\Phi_{S(x)}(y) = f(x,y)$. Per il secondo teorema di ricorsione, utilizzando come funzione calcolabile totale proprio $S(x)$, esisterà un certo numero n per cui: $\Phi_{S(n)}(y) = \Phi_n(y) = f(n,y)$. Per come abbiamo costruito $f(x,y)$ questa darà un risultato se, e solo se, $y = n$; quindi Φ_n converge solo su se stessa quindi $\Phi_n \in K$. Prendiamo un altro qualsiasi programma r equivalente ad n. Poichè deve risultare: $\Phi_r = \Phi_n$ il programma codificato da r convergerà sempre e solo su n e non su r , ovvero su se stesso. Quindi r non appartiene a K e di conseguenza K non rispetta le funzioni.

Programmi che calcolano funzioni come Φ_n si dicono programmi che si *autoriconoscono*: infatti Φ_n restituisce 1 solo se riceve come input se stesso ovvero n.

2.2 Dimostrazione di R1 tramite K2

Sia I un insieme che rispetta le funzioni e che, inoltre, sia: $I \neq \emptyset$ e $I \neq \mathbb{N}$. Insomma che rispetti le ipotesi di R1.

Consideriamo la seguente funzione: $g(x) = \begin{cases} C_0 & se\, x \in I\ con\, C_0 \in \bar{I} \\ C_1 & se\, x \notin I\ con\, C_1 \in I \end{cases}$. Tale funzione ha senso solo se I è decidibile.

Supponiamo che I sia decidibile.

In questo caso $g(x)$ è, ovviamente, calcolabile e totale quindi possiamo applicare il K2. Esisterà quindi (almeno) un numero naturale n per cui: $\Phi_n = \Phi_{g(n)}$.

Possono succedere due cose:

$n \in I$ questo implica che $\Phi_n = \Phi_{g(n)} = \Phi_{C_0}$. Poichè $n \in I$ e $C_0 \notin I$ troviamo una prima contraddizione.

$n \notin I$ questo implica che $\Phi_n = \Phi_{g(n)} = \Phi_{C_1}$. Poichè $n \notin I$ e $C_1 \in I$ troviamo una seconda contraddizione.

Quindi, l'ipotesi che I sia decidibile è assurda. Abbiamo dimostrato la prima parte di R1 tramite l'applicazione di K2.□

2.3 Costruzione di un programma che genera se stesso (quine)

Cosideriamo la segeuente funzione a due variabili:

$$f(x,y) = x \tag{5}$$

Questa funzione è ovviamente calcolabie. Per cui, applicando il teorema del parametro esisterà una funzione $S(x)$ tale che $\Phi_{S(x)}(y) = f(x,y)$. Applichiamo K2 ad S e avremo che esisterà un numero naturale n per cui: $\Phi_n(y) = \Phi_{S(n)}(y) = n$ per la (5). Allora il programma codificato dal numero n restituisce la codifica di sè stesso per qualsisi input.□

2.4 Dimostrazione che K non è decidibile

Supponiamo che K sia decidibile. Allora prendiamo la seguente funzione:

$$g(x) = \begin{cases} a & se\, x \in K\ con\, a\, programma\, per\, la\, funzione\, vuota \\ b & se\, x \notin K\ con\, b\, programma\, per\, la\, funzione\, identica \end{cases} \tag{6}$$

Poichè abbiamo considerato K decidibile allora $g(x)$ è, ovviamente, calcolabile e totale e quindi possiamo applicare K2. Esisterà quindi (almeno) un numero naturale n per cui: $\Phi_n = \Phi_{g(n)}$. Allora abbiamo le solite due possibilità:

$n \in I$ questo implica che $\Phi_n = \Phi_{g(n)} = \Phi_a$ assurdo poichè a è la codifica della funzione vuota quindi $\Phi_a \notin K$

$n \notin I$ questo implica che $\Phi_n = \Phi_{g(n)} = \Phi_{b_1}$ assurdo poichè b è la codifica di un programma che realizza la funzione identica, la quale terminando (anche) su sè stessa appartiene a K. Avendo ottenuto contraddizione possiamo concludere che K non è decidibile.□

2.5 Dimostrazione che esistono infiniti valori di n per cui K2 è verificato

K2 afferma che per ogni funzione calcolabile totale $h(x)$ esiste un numero naturale n per cui: $\Phi_n = \Phi_{h(n)}$

Cosideriamo la seguente funzione:

$$h'(x) = \begin{cases} C_o & se\, 0 \le x \le n\ con\, C_0\, diverso\, da\, programmi\, codificati\, da\, numeri\, compresi\, tra\, 0\, ed\, n \\ h(x) & se\, x > n \end{cases} \tag{7}$$

Questa funzione è calcolabile e totale: esisterà quindi un numero naturale n' per cui: $\Phi_{n'} = \Phi_{h'(n')}$.

Possono quindi verificarsi le solite due condizioni:

$0 \le n' \le n$ allora $\Phi_{n'} = \Phi_{h'(n)} = \Phi_{C_0}$ ma ciò non può essere perchè n' è minore o uguale ad n, mentre C_0 è maggiore di n

$n' > n$ allora $\Phi_{n'} = \Phi_{h'(n')} = \Phi_{h(n)} = \Phi_n$ e questa è l'unica situazione accettabile e che può verificarsi.

Abbiamo quindi trovato un nuovo numero naturale $n' \neq n$ per cui vale $\Phi_{n'} = \Phi_{h(n')}$. Ovviamente, poichè questo ragionamento può ripetersi infinite volte, partendo dal nuovo numero trovato, abbiamo dimostrato che esistono infiniti numeri naturali che soddisfano la tesi K2.□