APA Modulo 1 Lezione 2

Elena Zucca

13 marzo 2020

Correttezza di algoritmi ricorsivi

- principio di induzione (esempio noto: induzione aritmetica)
- si applica a a qualunque insieme definito induttivamente

Definizione induttiva

- R insieme di *regole* $\dfrac{Pr}{c}$
- Pr insieme delle *premesse*, c *conseguenza*
- l'insieme *definito induttivamente* $\mathcal{I}$ è
  il più piccolo tra gli insiemi X chiusi rispetto a R
- ossia tali che, per ogni regola $\dfrac{Pr}{c}$, se $Pr \subseteq X$ allora $c \in X$
- gli elementi dell'insieme sono *tutti e soli* quelli che si ottengono applicando ripetutamente un insieme di regole, a partire da quelle con premessa vuota che costituiscono la base della definizione induttiva

Esempio

L'insieme dei numeri pari è il più piccolo insieme tale che (oppure: è l'insieme definito induttivamente da):

- 0 è un numero pari,
- se n è un numero pari, allora $n + 2$ è un numero pari.
- insieme (infinito) di regole:

$$\frac{\emptyset}{0} \qquad \frac{n}{n+2} \; n \in \mathbb{N}$$

Quale di questi insiemi è chiuso?

$$\frac{\emptyset}{0} \qquad \frac{n}{n+2}$$

- insieme vuoto
- tutti i numeri naturali
- tutti i pari più il numero 3
- tutti i pari da 10 in poi
- quali sono tutti gli insiemi chiusi?

Soluzione

- tutti gli insiemi che contengono tutti i pari e i dispari da un certo punto in poi
  $\{\text{pari}\} \cup \{n \mid n \geq k\}$
  per esempio $\{\text{pari}\} \cup \{n \mid n \geq 35\}$
- Il più piccolo tra questi è l'insieme dei numeri pari
- è l'insieme di *tutti e soli* gli elementi che devono necessariamente esserci per soddisfare le regole

Altri esempi tipici di insiemi definiti induttivamente

- i numeri naturali: $0 \in \mathbb{N}$; se $n \in \mathbb{N}$ allora $n + 1 \in \mathbb{N}$
- le liste: ϵ è una lista; se l è una lista, $x : l$ con x elemento è una lista
- gli alberi binari: l'albero vuoto è un albero binario; se L e R sono alberi binari, (L, x, R) è un albero binario

Se un insieme è definito induttivamente possiamo . . .

- definire induttivamente funzioni
  (ossia seguendo la definizione induttiva dell'insieme)
- provarne proprietà con il principio di induzione

Principio di induzione

Sia $\mathcal{I}$ definito induttivamente dalle regole R
P un predicato su U con $\mathcal{I} \subseteq U$. Se

Base per ogni $\frac{\emptyset}{c} \in R$ vale $P(c)$

Passo induttivo per ogni $\frac{Pr}{c} \in R, Pr \neq \emptyset$
($P(d)$ vale per ogni $d \in Pr$) implica che valga $P(c)$

allora $P(d)$ vale per ogni $d \in \mathcal{I}$.

Dimostrazione.

Sia $C = \{d \mid P(d) \text{ vero}\}$
le condizioni sopra equivalgono a: $Pr \subseteq C$ implica $c \in C$
quindi C è chiuso, quindi $\mathcal{I} \subseteq C$, quindi $P(d)$ vale per ogni $d \in \mathcal{I}$ □

Principio di induzione aritmetica

Sia P un predicato sui numeri naturali tale che

Base vale $P(0)$

Passo induttivo se vale $P(n)$ allora vale $P(n+1)$

allora $P(n)$ vale per ogni $n \in \mathbb{N}$.

Dimostrazione.

$\mathbb{N}$ può essere visto come l'insieme definito induttivamente da

- $0 \in \mathbb{N}$
- se $n \in \mathbb{N}$ allora $n+1 \in \mathbb{N}$

Quindi la proposizione è un caso particolare del principio di induzione. $\square$

Esempio di prova: serie aritmetica

Proviamo per induzione aritmetica che $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, per tutti gli $n \in \mathbb{N}$.

Base $P(0)$: $0 = 0$

Passo induttivo Assumiamo $P(n)$ e dimostriamo $P(n+1)$:

$$\begin{aligned}
\sum_{i=1}^{n+1} i = & \sum_{i=1}^{n} i + (n+1) \\
= & \frac{n(n+1)}{2} + (n+1) \quad \text{(per ipotesi induttiva)} \\
= & \frac{(n+1)(n+2)}{2}
\end{aligned}$$

Esercizio per voi: serie geometrica

Provare per induzione aritmetica che $\sum_{i=0}^{n} q^i = \frac{q^{n+1}-1}{q-1}$, per tutti gli $n \in \mathbb{N}$.

Principio di induzione aritmetica completa (o forte)

Sia P un predicato sui numeri naturali tale che

Base vale $P(0)$

Passo induttivo se vale $P(m)$ per ogni $m < n$ allora vale $P(n)$

allora $P(n)$ vale per ogni $n \in \mathbb{N}$.

Dimostrazione.

$\mathbb{N}$ può essere visto come l'insieme definito induttivamente da

- $0 \in \mathbb{N}$
- se $\{m \mid m < n\} \subseteq \mathbb{N}$ allora $n \in \mathbb{N}$.

Quindi la proposizione è un caso particolare del principio di induzione. $\square$

Definizione induttiva degli alberi binari (con nodi in N)

- l'albero vuoto è un albero binario
- se L e R sono alberi binari e $x \in N$
  allora (L, x, R) è un albero binario.

Il principio di induzione prende la seguente forma:

> *Sia P un predicato sugli alberi binari tale che*
>
> - *P vale sull'albero vuoto,*
> - *se P vale su L e R, allora vale su (L, x, R),*
>
> *allora P vale su tutti gli alberi binari.*

Esercizio

- Definire induttivamente l'altezza di un albero binario.
- Se L ha altezza h_L e R ha altezza h_R, quale è l'altezza di (L, x, R)?
- Risposta: $\max(h_R, h_L) + 1$
- Caso base: altezza dell'albero vuoto?
- Risposta: deve essere -1
  infatti $(\text{vuoto}, x, \text{vuoto})$ ha altezza 0

Esercizio

- Provare per induzione che se h altezza e n numero nodi albero binario $h + 1 \leq n \leq 2^{h+1} - 1$
- Base: albero vuoto ha altezza -1 e 0 nodi: $-1 + 1 \leq 0 \leq 2^{-1+1} - 1$

Passo induttivo

- sia (L, x, R) con h altezza e n numero nodi
- Ipotesi induttiva: $h_L + 1 \leq n_L \leq 2^{h_L+1} - 1 \quad h_R + 1 \leq n_R \leq 2^{h_R+1} - 1$
- Tesi: $h + 1 \leq n \leq 2^{h+1} - 1$
- supponiamo per esempio $h_L \geq h_R$, allora $h = h^L + 1$
- quindi devo provare: $h^L + 2 \leq n_L + n_R + 1 \leq 2^{h_L+2} - 1$
- per ipotesi induttiva
  $h^L + 2 \leq n_L + 1 \leq n_L + n_R + 1$
  $n_L + n_R + 1 \leq 2^{h_L+1} - 1 + 2^{h_R+1} - 1 + 1$
  $\leq 2^{h_L+1} - 1 + 2^{h_L+1} \leq 2 \cdot 2^{h_L+1} - 1 = 2 \cdot 2^h - 1$

Correttezza di algoritmi ricorsivi per induzione

- input in insieme definito induttivamente
- algoritmo definito a sua volta in modo induttivo (ossia ricalcando la definizione induttiva dell'insieme)
- è facile provare la correttezza dell'algoritmo con il principio di induzione

Per esempio nel caso di un algoritmo sui numeri naturali:

- se l'algoritmo è corretto per 0
- e assumendo che sia corretto per n, è corretto per $n + 1$,

allora è corretto su qualunque numero naturale.
analogamente si può applicare il principio di induzione arimetica forte o il principio di induzione per alberi, liste, etc.

Induzione come guida per la progettazione

- principio di induzione non solo per provare a posteriori la correttezza di un algoritmo
- fornisce anche una guida per il suo sviluppo

Per esempio, nel caso dell'induzione aritmetica:

- si scrive l'algoritmo corretto per 0
- se l'argomento è $n > 0$, si assume che l'algoritmo sappia operare correttamente su $n - 1$, e fidandosi di tale assunzione si scrive l'algoritmo corretto per operare su n.

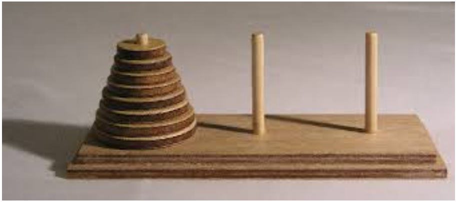
Esempio: correttezza di ricerca binaria ricorsiva

```
binary_search(x,a)// a[0..n-1]
        return binary_search(x,a,0,n-1)

binary_search(x,a,inf,sup)
  if (inf <= sup)
    mid = (inf + sup)/2
    if (x < a[mid]) return binary_search(x,a,inf,mid-1)
    else if (x > a[mid])
      return binary_search(x,a,mid+1,sup)
    else return true
  return false
```

correttezza per induzione aritmetica forte, in quanto le chiamate ricorsive sono effettuate su sequenze di dimensione strettamente minore di quella iniziale.

Esempio di progettazione e analisi di un algoritmi ricorsivo: torri di Hanoi



Scopo del gioco: spostare tutti i dischi su un altro piolo, eventualmente utilizzando il terzo, senza violare le seguenti regole:

- si può spostare un solo disco per volta (quello più in alto)
- non si può mettere un disco sopra un altro di diametro inferiore

- Problema: il gioco è risolubile per qualunque numero n di dischi?
- E se lo è, quale è per ogni n la sequenza di mosse che lo risolve?

Il gioco è risolubile per qualsiasi numero n di dischi.
Per induzione aritmetica su n.

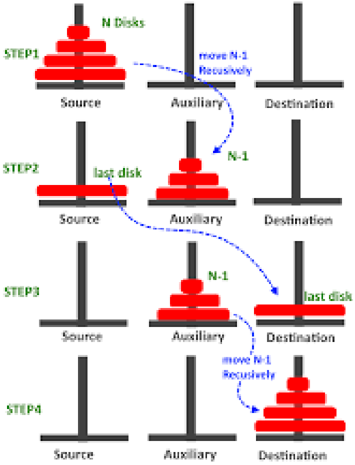
Base Il gioco è risolubile per $n = 1$. Ovvio, perché un solo disco può essere spostato con una sola mossa da un piolo a un altro.

Passo induttivo Assumiamo (ipotesi induttiva) che il problema sia risolubile per $n-1$, proviamo che è risolubile per n.

Per ipotesi induttiva possiamo spostare gli $n-1$ dischi superiori dal piolo di partenza (*source*) al piolo ausiliario (*aux*).

A questo punto è possibile spostare il disco più grande da *source* al piolo di arrivo (*dest*).

Infine, sempre per ipotesi induttiva è possibile spostare gli $n-1$ dischi superiori da *aux* a *dest*.


N Disks
STEP1
move N-1
Recusively
Source
Auxiliary
Destination
STEP2
last disk
N-1
Source
Auxiliary
Destination
STEP3
N-1
last disk
Source
Auxiliary
Destination
move N-1
Recusively
STEP4
Source
Auxiliary
Destination

La dimostrazione fornisce direttamente un algoritmo ricorsivo

```
hanoi(n, source, aux, dest)
      if (n = 1) move(source, dest)
      else
        hanoi(n-1, source, dest, aux)
        move(source, dest)
        hanoi(n-1, aux, source, dest)
```

Analisi della complessità

Contiamo il numero di mosse.

- Per $n = 1$ si ha una mossa, quindi $T(1) = 1$.
- Per $n > 1$ occorrono:
  - $T(n-1)$ mosse per spostare gli $n-1$ dischi da *source* ad *aux*
  - 1 mossa per spostare il disco più grande da *source* a *dest*
  - $T(n-1)$ mosse per spostare gli $n-1$ dischi da *aux* a *dest*.

Relazioni di ricorrenza

$T(1) = 1$
$T(n) = 1 + 2T(n-1)$, per $n > 1$.

Come le risolviamo?

Metodo delle sostituzioni successive

$T(n) = 1 + 2T(n-1) = 1 + 2(1 + 2T(n-2)) =$
$2^0 + 2^1 + 2^2 T(n-2) = \ldots =$
$2^0 + \ldots + 2^i T(n-i) = \ldots =$ ultimo termine si ha per $i = n-1$
$2^0 + \ldots + 2^{n-1} T(n-(n-1)) =$
$2^0 + \ldots + 2^{n-1} =$ (serie geometrica) $2^n - 1$

Modo alternativo: albero delle chiamate ricorsive

livello 0: $1 = 2^0$ mossa
livello 1: $2 = 2^1$ mosse
livello 2: 2^2 mosse
...
livello i: 2^i mosse
...
(come prima, l'ultimo livello che fornisce un contributo è quello $n - 1$)
livello $n - 1$: 2^{n-1} mosse

sommando il contributo dei vari livelli si ottiene la sommatoria precedente

Verifica rigorosa per induzione aritmetica

$$
\begin{aligned}
\text{Base} \quad & T(1) = 2^1 - 1 \text{ vero} \\
\text{Passo induttivo} \quad & T(n+1) = 2T(n) + 1 \\
& = (\text{hp. ind}) \ 2(2^n - 1) + 1 = 2^{n+1} - 1
\end{aligned}
$$

quindi complessità temporale $T(n) = \Theta(2^n)$
complessità in spazio = altezza dell'albero di ricorsione = massima profondità dello stack
quindi è $S(n) = \Theta(n)$

Si può fare di meglio?

- NO: questo numero di mosse è minimo (come provarlo?)
- quindi il gioco delle Torri di Hanoi è intrattabile: il numero di mosse cresce esponenzialmente al crescere del numero dei dischi
- (difficile) si può dare un algoritmo iterativo che produce la stessa sequenza di mosse

Esercizio

Per la ricerca binaria ricorsiva, dare le relazioni di ricorrenza e risolverle con il metodo delle sostituzioni successive.