Aniello Murano

Ordinamenti parziali completi, funzioni continue e minimi punti fissi

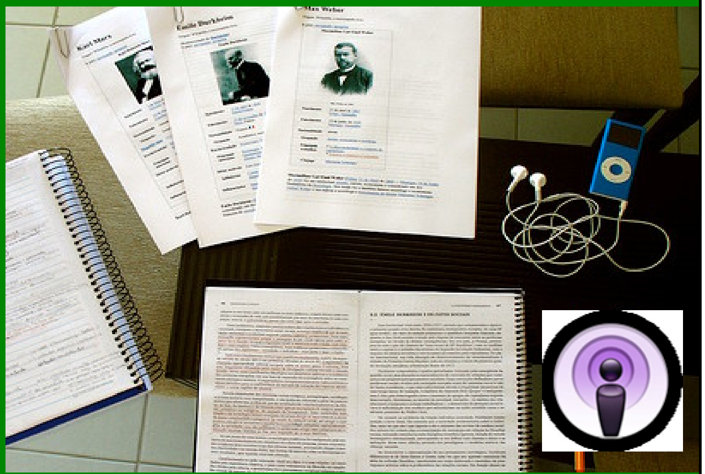
Lezione n. 6
Parole chiave:
Punto fisso

Corso di Laurea:
Informatica

Codice:

Email Docente:
murano@ na.infn.it

A.A. 2008-2009



Riassunto delle lezioni precedenti

- Prima Lezione: Introduzione e motivazioni del corso; Sintassi e semantica di ARITHM
- Seconda lezione: Sintassi e semantica operazionale del linguaggio imperativo IMP
- Terza lezione: Tecniche di prova per induzione (su IMP).
- Quinta lezione: Definizione induttiva di domini (per IMP).
- Occupiamoci adesso di semantica denotazionale.
- Prima introduciamo alcuni concetti matematici basilari:
  - Ordinamenti parziali completi, funzioni continue e minimi punti fissi
- Nella prossima lezione, mostreremo una semantica denotazionale per il linguaggio IMP e la sua equivalenza con la semantica operazionale.

Introduzione

- La semantica operazionale mostra come possono essere utilizzati (eseguiti) i vari costrutti di linguaggio.
- Essa interpreta i programmi come diagrammi di transizione, con azioni che permettono di muoversi tra i vari stati.
- Queste strutture descrivono il comportamento di un programma, ma non permettono di definire completamente il suo significato (il significato è trasferito in un altro linguaggio).
- Potremmo definire una semantica operazionale come una formalizzazione matematica di qualche strategia di implementazione del programma.

Semantica denotazionale

- La semantica denotazionale cerca di catturare il significato interno di un programma piuttosto che la strategia di implementazione. Per questo motivo è più astratta di quella operazionale ed è indipendente dalla macchina su cui si lavora.
- Questa semantica mappa un linguaggio in un modello matematico astratto (invece di utilizzare regole operazionali), in modo tale che il valore di un programma composto è determinabile direttamente dai valori delle sue singole parti.
- In pratica, ad ogni frase del linguaggio viene associata una denotazione (significato) come funzione delle denotazioni delle sue sottofrasi.
- Ne consegue che in alcuni casi la semantica di una frase è il minimo punto fisso di una funzione opportunamente definita.
- Tuttavia, e' fondamentale verificare che le specifiche denotazionali siano implementabili, in altre parole che la semantica denotazionale corrisponda sempre ad una semantica operazionale. Questo verra' mostrato nelle prossime lezioni

Strumenti matematici opportuni

- Lo spazio matematico solitamente utilizzato dalla teoria denotazionale della semantica dei linguaggi di programmazione è quello degli insiemi parzialmente ordinati (**domini**).
- Esempi di tali oggetti sono le **funzioni parziali**, unitamente ad un ordinamento parziale che permette di definire un concetto di approssimazione o di contenimento di informazione tra gli oggetti. Per esempio f · g può indicare che g si comporta come f su tutti i valori in cui quest'ultima è definita.
- Esistono elementi del linguaggi la cui denotazione dipende dalla soluzione di equazioni ricorsive (come per il comando while di IMP). Vedremo che una soluzione per tali equazioni è data dal **minimo punto fisso** di un **insieme completo parzialmente ordinato**.

Ordinamento Parziale

- Un ordinamento parziale (p.o.) è una coppia (P, v), dove P è un insieme di elementi e v è una relazione binaria su P che sia:
- **Riflessiva**: per ogni p in P, p v p
- **Transitiva**: Per ogni p, q, r in P, se p v q Æq v r allora p v r
- **Antisimmetrica**: Per ogni p, q in P, se p v q Æq v p allora p = q
- Remark:
  - "p v q significa che p e meno (o al più uguale) significativo di q"
  - Parziale significa che ci sono elementi in P che non sono confrontabili tra loro
- Esempi di insiemi parzialmente ordinati:
  - **(Z, ·)**, dove Z è l'insieme dei numeri interi e · è il solito ordinamento
  - **(2^S, µ)**, dove **2^S** denota l'insieme delle parti di S, (spesso scritto come P**(S)**, o P**ow(S)**
- Esempi di insiemi non parzialmente ordinati:
  - **(Z, <)** non è un p.o. perché **…**
  - **(Z, v)**, dove **m** v **n** , **|m|** · **|n|**, non è un p.o. perché **…**

Ordinamento Parziale Completo

- Dato un ordinamento parziale (P, v) e un sottoinsieme X µ P, allora p è un **upper bound** (maggiorante) di X sse 8 q 2X, q v p.
- Si dice che p è un **least upper bound** (minimo maggiorante, **lub**) di X sse
  - p è un upper bound di X
  - Per ogni upper bound q di X, p v q

1. Un lub di un sottoinsieme **X** di un p.o. è anche indicato con t **X**

- Una **catena** di un p.o. (P, v), è una sequenza d_0 v d_1 v ... v d_n, il cui lub è d_n (un insieme finito totalmente ordinato è una catena).
- Una ω**- catena** di un p.o. è una catena infinita. Anch'essa può avere un lub e sarà indicato con $\bigsqcup_{n\in w} d_n$ (intuitivamente, l'elemento limite della catena
- Un p.o. (P, v) è un **ordinamento parziale completo** (c.p.o.) se tutte le sue ω-catene hanno un lub
- Un c.p.o. (P, v) è con **elemento minimo** se esiste un elemento ? 2 P tale che 8 q 2 P, ? v q

Esempi di c.p.o.

- **(2^S, µ)** è un c.p.o., ed il minimo upper bound (lub) per la catena di tutti gli elementi di 2^S è proprio S (U = t).
- **(N, ·)** non è un c.p.o. perchè N non ha un lub. Se invece consideriamo (N [1 , ·), questo è un c.p.o dove il lub è 1 .
- **([0,1], ·)** è un c.p.o. dove [0,1] è l'intervallo continuo chiuso di reali e 1 è un lub per catene infinite. Si noti come ([0,1), ·) non è un c.p.o.. Infatti, non esiste un lub per la catena infinita, 1/2, 2/3, 3/4... che ha per limite 1 ∉ [0,1)
- Se (P, v) è un c.p.o. lo è anche (P, w)?
- No, perché (P, w) potrebbe non avere un lub. Per esempio, ((0,1], ·) è un c.p.o. con lub 1, mentre ((0,1], ¸) non ha un lub. Infatti, non esiste un lub per la catena infinita, 1/2, 1/3, 1/4... che ha per limite 0 ∉ (0,1]

Funzioni monotone e continue

- Siano (D, v) e (E, v) due c.p.o. Una funzione f : D ! E si dice **monotona** sse preserva il seguente ordine:

  8 d,d' 2 D. d v d') f(d) v f(d')

- Siano (D, v) e (E, v) due c.p.o. Una funzione f: D ! E si dice **(Scott-) continua** se f e' monotona e per ogni catena d_0 v d_1 v … v d_n v … in D vale

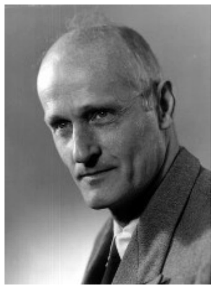
$$\bigsqcup_{n\in\omega} f(d_n) = f(\bigsqcup_{n\in\omega} d_n)$$

- Si noti come non tutte le funzioni monotone sono continue!
- Come vedremo, questa formulazione è molto utile per calcolare il minimo punto fisso di una funzione

Punto fisso

- Sia (D, v) un c.p.o. e f : D ! D una funzione monotona. Un **punto fisso** di f è un elemento d 2 D tale che f(d) =d.
  - La funzione polinomiale sui numeri reali f(x) = x^2 – 3x + 4 ha punto fisso e precisamente f(2) = 2.
  - La funzione f(x)=x ha infiniti punti fissi
  - La funzione f(x)=x+1 non ha punto fisso sui reali perché x non è mai uguale a x+1, per qualsiasi numero reale x.
- Un **pre- punto** fisso di f è un elemento d 2 D tale che f(d) v d.
- Una funzione può avere più punti fissi
- **Thm[Kleene]: Una funzione continua su un c.p.o. ha sempre un punto fisso.**
- Per essere precisi, anche una funzione **monotona** su un c.p.o ammette punto fisso, ma se la funzione è anche continua allora abbiamo un algoritmo per calcolare il punto fisso, come vedremo nelle prossime diapositive.

Intermezzo: Stephen Cole Kleene [1909-1994]



- Kleene è un matematico americano.
- Nato a Hartford, Connecticut(USA), si laureò nel 1930.
- Dal 1930 al 1935, fu assistente ricercatore all'università di Princeton, dove ricevette il dottorato in matematica nel 1934, supervisionato da Alonzo Church.
- Da 1935 lavorò all' università di Winsconsin-Madison dove predispose le fondamenta dell'informatica teorica.
- Kleene è noto per la fondazione del ramo della logica matematica conosciuta come **teoria della ricorsione**, insieme con Alonzo Church, Kurt Godel, Alan Turing ed altri, e per aver inventato le **espressioni regolari**.
- Fornendo metodi per determinare quali problemi sono risolvibili, il suo lavoro portò allo studio di quali funzioni fossero **calcolabili**.
- Tra le altre cose, l'algebre 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.

Terema del Punto fisso (1)

- Sia (D, v) un c.p.o. con minimo ? e f : D ! D una funzione continua. Allora si ha che:
  - ? v **f(**?) perché ? v x per ogni elemento x 2 D (e di f(D))
  - **f(**?) v **f(f(**?)) perché **f** è monotona e vale la riga precedente.
  - **f(f(**?)) v **f(f(f(**?))) per lo stesso motivo di prima e così via
- Sia f^{n+1}(?) = f (f^{n}(?)).
- ? v **f(**?**)** v **f²(**?**)** v **f³(**?**)**... è una catena in D
- Sia $fix(f) = \bigsqcup_{n\in\omega} f^n(\perp)$ allora
- **Thm[Kleene] (cont.): Una funzione continua su un c.p.o. ha sempre un minimo punto fisso ed e' uguale al least upper bound della catena su f.**
- In altre parole fix(f) è un punto fisso di f ed è il suo minimo pre-punto fisso.
  - In simboli: **(1) f(fix(f))=fix(f) e (2) se f(d)** v **d allora fix(f)** v **d**

Teorema di punto fisso (2)

- Proviamo il primo punto!
- Ricordiamo che $fix(f) = \bigsqcup_{n\in\omega} f^n(\bot)$
- Applicando **f** abbiamo che

$$f(fix(f)) = f(\bigsqcup_{n\in\omega} f^n(\bot))$$

- Per la continuità di f abbiamo che

$$\begin{aligned} &= (\bigsqcup_{n\in\omega} f(f^n(\bot))) \\ &= (\bigsqcup_{n\in\omega} f^{n+1}(\bot)) \\ &= (\bigsqcup_{n\in\omega} f^{n+1}(\bot)) \sqcup \{\bot\} \\ &= (\bigsqcup_{n\in\omega} f^n(\bot)) = fix(f) \end{aligned}$$

È la catena senza il primo elemento

Se aggiungo bottom, il risultato non cambia e ottengo la stessa catena di partenza

Dunque **fix(f)** è un punto fisso

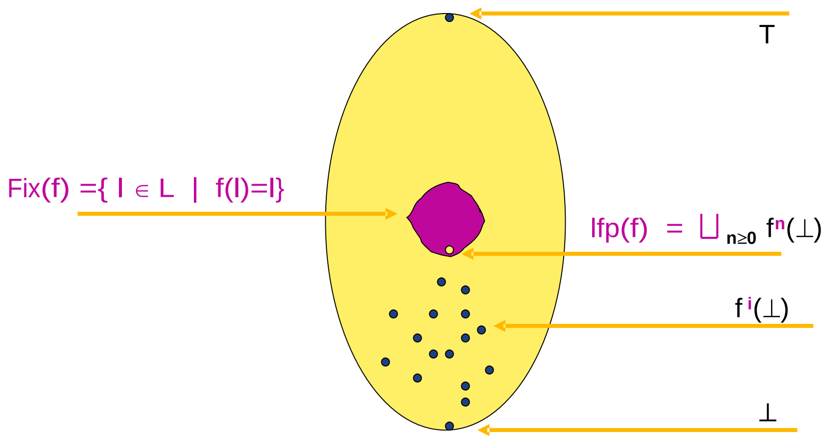
Teorema di punto fisso (3)

- Proviamo il secondo punto!
- Sia d un pre-punto fisso. Allora ? v d (per definizione di bottom).
- Per la monotonicità di f, f(?) v f(d)
- Siccome d è un pre-punto fisso, per definizione f(d) v d. Dunque f(?) v d.
- Per induzione f(f(?)) v f(d) v d.... Dunque f ⁿ(?) v d.
- Ma allora

$$fix(f) = (\bigsqcup_{n\in\omega} f^n(\bot)) \sqsubseteq d$$

- **Dunque fix(f) è il minimo punto fisso di f**

Punti fissi sui CPO



Osservazioni

- A questo punto abbiamo acquisito gli strumenti matematici di base per discutere della semantica denotazionale dei linguaggi di programmazione.
- In che modo questi strumenti vengono utilizzati?
- Gli ordinamenti parziali completi corrispondono ai tipi di dati (sia in input che output) di una computazione
- Le funzioni calcolabili sono rappresentate da funzioni continue fra c.p.o.
- Gli elementi di un c.p.o sono punti di informazione
- x v y può essere interpretato come "x approssima y" (oppure "x ha meno informazioni di y"). Di conseguenza ? è il punto con minima informazione.