Dimostrazioni e programmi come oggetti geometrici

Simone Martini

Dipartimento di Scienze dell'Informazione

Alma Mater Studiorum

Università di Bologna

Dimostrazioni “alla Hilbert”

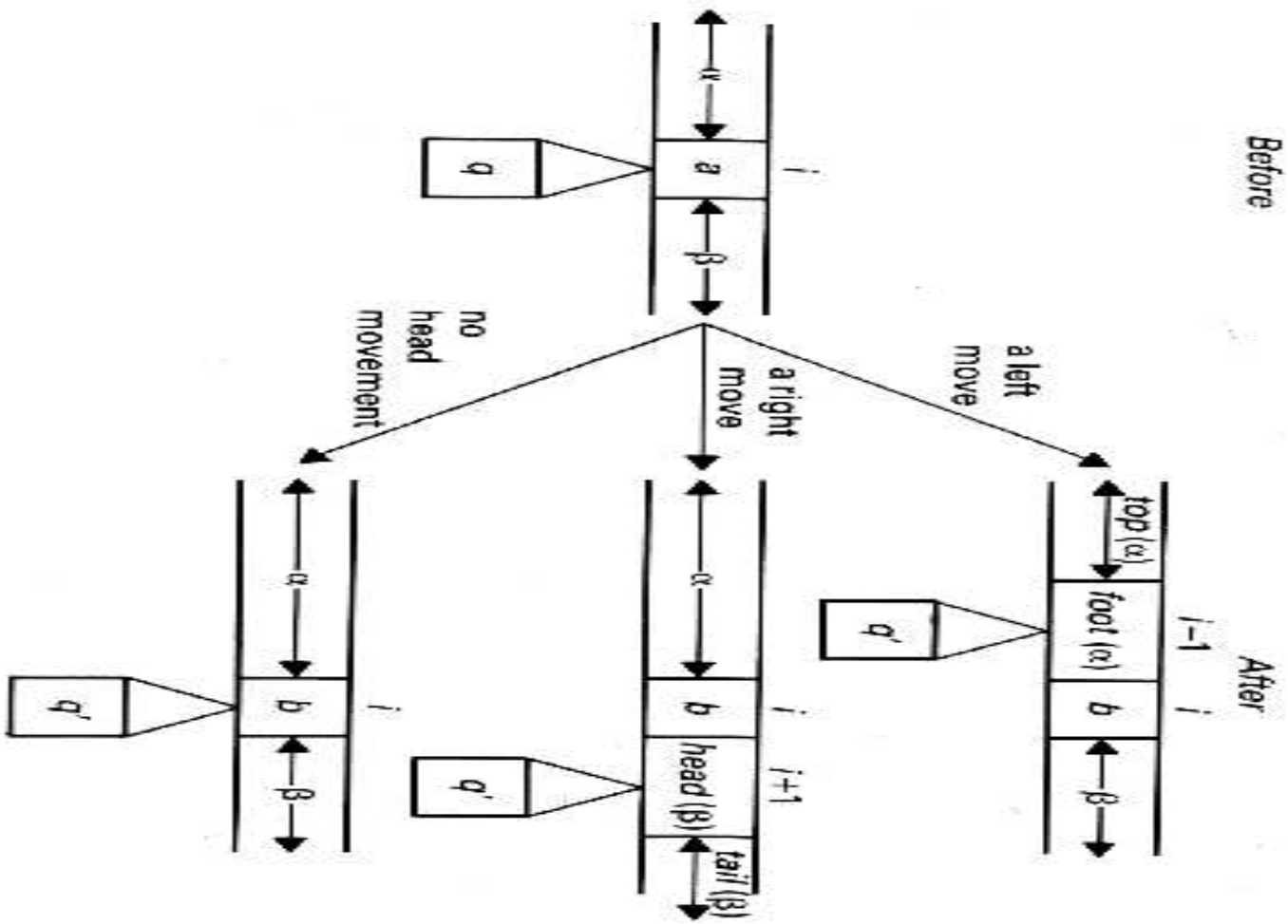
- Assiomi, che caratterizzano la logica, o il modello
- Due sole regole di inferenza
  - modus ponens
  - generalizzazione

$$\frac{\vdash A \to B \qquad \vdash A}{\vdash B} \qquad\qquad \frac{\vdash A}{\vdash \Box x.A}$$

- Dimostrazione
  - sequenza non strutturata di formule dedotte dagli assiomi

- La struttura del modello (o del particolare sistema logico) descritta senza struttura, negli assiomi
- Ogni assiomatizzazione equivalente va bene

La macchina di Turing



Computazioni “alla Turing”

- Macchina di Turing
  - ingegnerizzata nella macchina di Von Neumann
  - Programma non strutturato: quintuple
  - Computazione: sequenza di passi elementari
- La struttura del problema e della soluzione è perduta
- Caratterizzazione di calcolabilità e complessità modulo ogni codifica accettabile

Computazioni

- Molta strada nella descrizione delle computazioni
  - linguaggi di programmazione raffinati
  - espressibilità non riconducibile a Turing-completezza

  - solo negli anni 80 modelli di PTIME non appiattiti sulla MdT
    - Bellantoni & Cook; Leivant

Computazioni “alla Church”

- Modelli computazionali con struttura
  - sintassi:

    $x \mid (M\ N) \mid \lambda x.M$
  - computazione:

    $(\lambda x.M)N \rightarrow M\{N/x\}$

    $M\{N/x\}$ è la sostituzione di N al posto di x in M

- Turing-completo modulo codifica, ma ricco di proprietà non preservate per codifica

- Gli alberi di Böhm
  - topologia
  - tecniche raffinate per ottenere risultati sulla riduzione…

Dimostrazioni “alla Gentzen”

- Controparte logica del calcolo di Church
- La struttura logica dei connettivi è parte della sintassi

$$x:A \vdash x:A$$

$$\frac{E \vdash M:A \to B \qquad E \vdash N:A}{E \vdash (MN) : B}$$

$$\frac{E, x:A \vdash M:B}{E \vdash \lambda x:A.M:A \to B}$$

Potente impatto

- Il λ-calcolo ha ispirato la programmazione funzionale

- La nozione di computazione del λ-calcolo
  - ambienti annidati
  - riscrittura di variabili
  - iterazione e ricorsione invece di salti
  - ….

  ha guidato il progetto di trent'anni di linguaggi di programmazione !

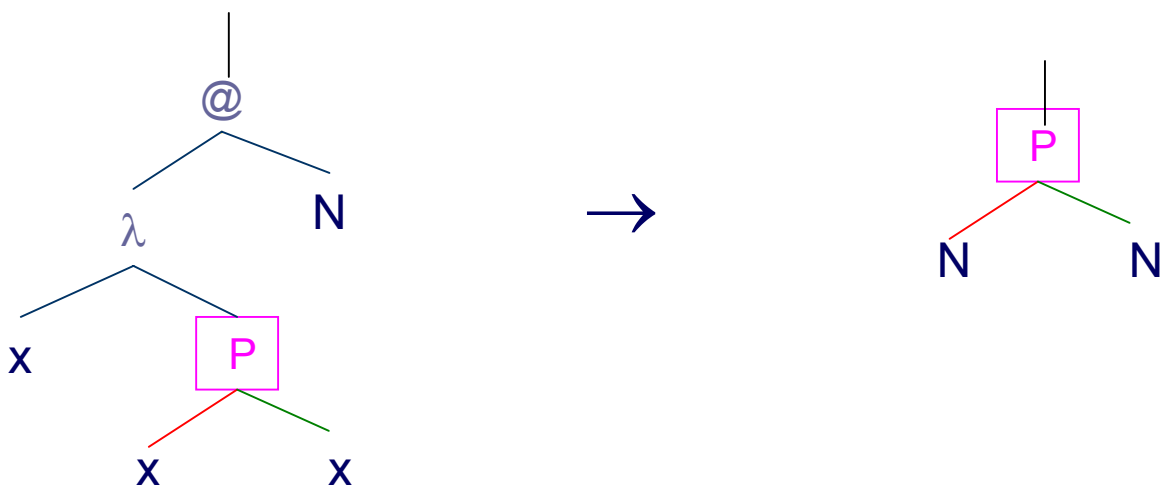
E tuttavia…

- … la comprensione di aspetti importanti della computazione è legata alla MdT

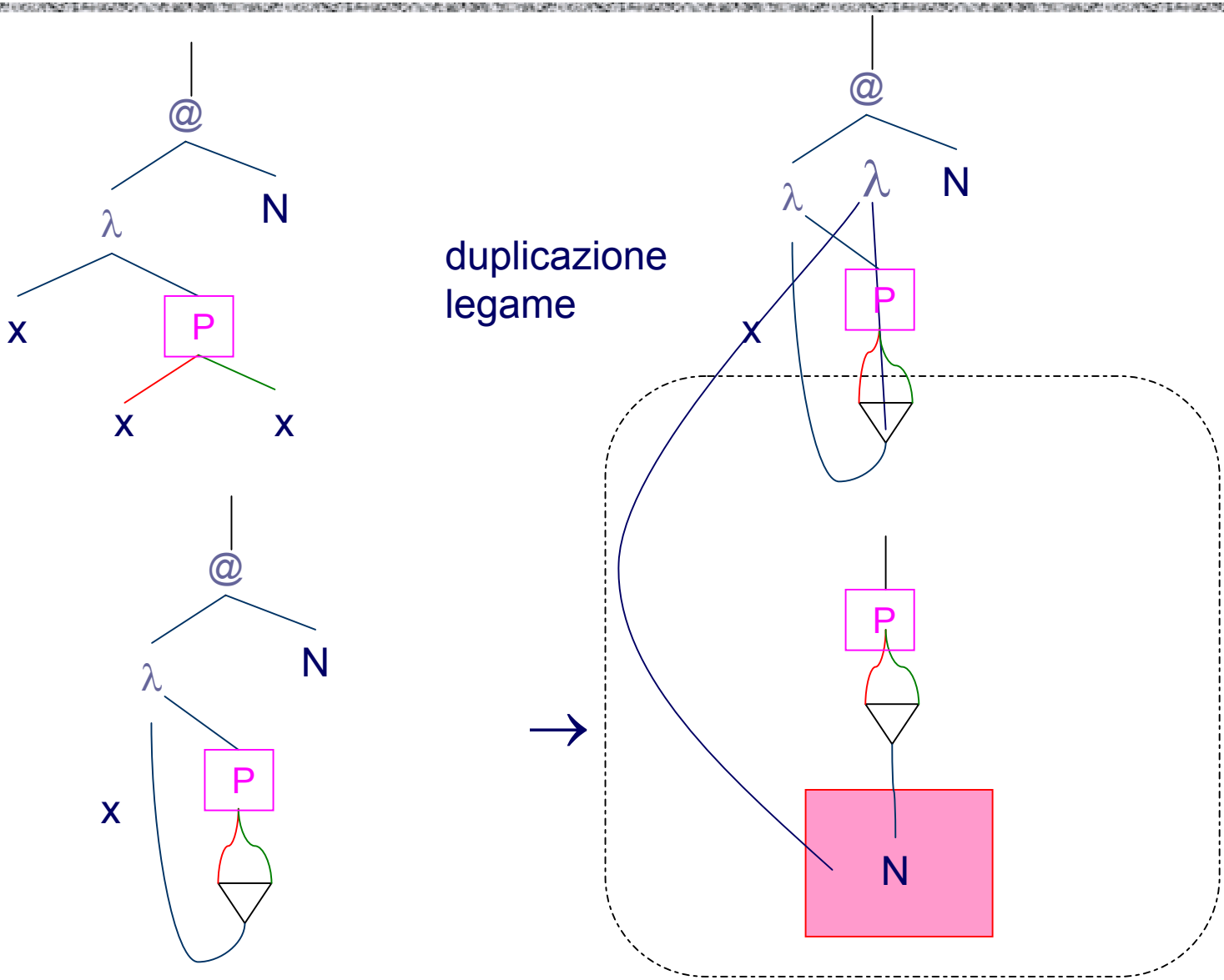
- complessità, anche dei programmi funzionali

$(\lambda x.M)N \rightarrow M\{x \leftarrow N\}$

$(\lambda x.\ P[x, x])N \rightarrow P[N, N]$



Duplicazione interna al calcolo



Logica Lineare

- Funzioni lineari –o, par
- Duplicazione esplicita contrazione tra !
- Applicata a sottotermini espliciti introduzione di !

I grafi sintattici della riduzione tra λ-termini

[Lévy 79, Lamping 89]

sono esattamente le proof-nets di Girard, 1987

[Gonthier et al., 91]

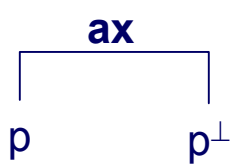
Computazione decomposta in strutture elementari

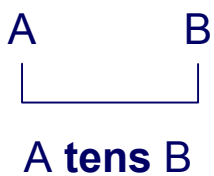
Logica Lineare: lessico

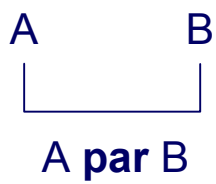
- Formule
  - $A , B ::= p \mid p^{\perp} \mid A \text{ tens } B \mid A \text{ par } B \mid ?A \mid !A$
  - $A \multimap B = A^{\perp} \text{ par } B$
  - $A \rightarrow B = !A \multimap B$
- Duali
  - $p^{\perp\perp} = p$
  - $(A \text{ tens } B)^{\perp} = A^{\perp} \text{ par } B^{\perp}$ e $(A \text{ par } B)^{\perp} = A^{\perp} \text{ tens } B^{\perp}$
  - $(!A)^{\perp} = ?A^{\perp}$ e $(?A)^{\perp} = !A^{\perp}$

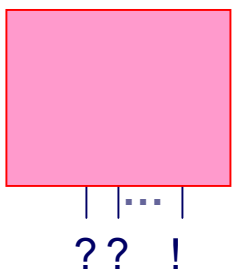
Logica Lineare: reti di dimostrazione (*proof-nets*)

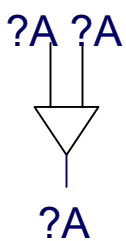
- Grafi , dimostrazioni
- Criteri combinatori caratterizzano quei grafi che corrispondono a dimostrazioni

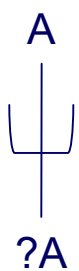






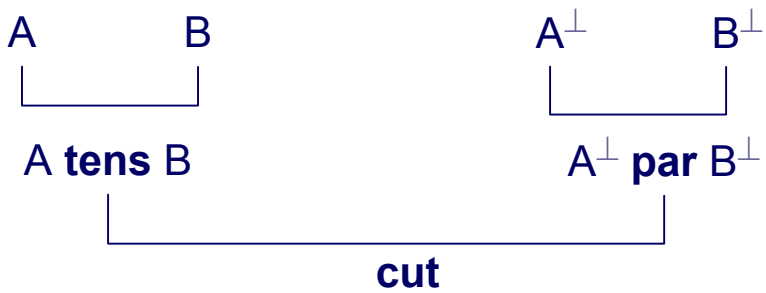






Logica Lineare: computazione

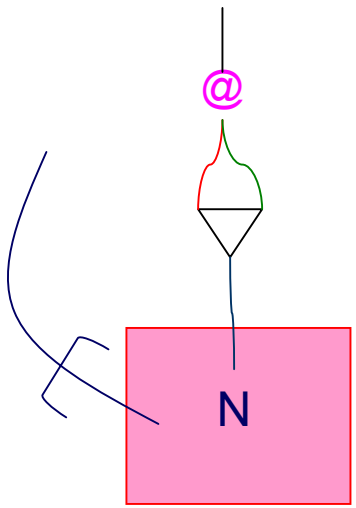
- Computazione (normalizzazione)
  - riscrittura (locale…) di grafi



viene riscritto in

La struttura delle reti

- La decomposizione dell'applicazione



- permette di studiare la complessità della riduzione in modo indipendente dalla sua codifica nella MdT

Asperti, Mairson, Lawall, Coppola, M.

La struttura delle reti

- La struttura delle reti permette di “calcolare la forma normale” attraverso un’analisi dei cammini nel grafo:

  “geometria dell’interazione” (Girard)

  - computazione = flusso di tokens nel grafo
    (e non riscrittura)

  - la forma normale è pienamente caratterizzata da opportune permutazioni degli assiomi

- Interpretazione come semantica dei giochi
  - dimostrazione come interazione (Abramsky)

Impatto ?

- Osservavamo:

  La nozione di computazione del λ-calcolo ha guidato il progetto di trent'anni di linguaggi di programmazione !

- Ma recentemente non è più così:
  - agenti
  - information flow
  - mobile, global computing
  - …

- Cosa calcola Internet?
  - certo non una funzione Turing-calcolabile…
  - …ma neppure la normalizzazione di un λ-termine !

Promesse ?

- Ingredienti
  - interazione, controllo, spazialità
- Ma
  - promesse non mantenute appieno…
- Indizi
  - semantica a giochi di linguaggi (di programmazione) complessi
  - basata sulla nozione di LL di interazione
  - …
- Prospettive
  - “processi computazionali” senza modello esplicito di macchina
  - “processi efficienti” (cioè poly) senza mod. espl. di macchina
  - flussi informativi in una rete di agenti