UNIVERSITÀ DEGLI STUDI DI ROMA TRE
FACOLTÀ DI SCIENZE M.F.N.

Tesi di Laurea in Matematica
presentata da

Paola Stoico

La logica lineare: Proof-net e Semantica Denotazionale

(SINTESI)

Relatore

Prof. V. Michele Abrusci

Il Candidato Il Relatore

ANNO ACCADEMICO 2001 - 2002
Febbraio 2003

Capitolo 1

Sintesi

La teoria della dimostrazione è alla base dello studio della logica matematica. L'ingrediente essenziale di una dimostrazione, nella pratica matematica, è l'uso dei lemmi, ovvero, di risultati intermedi che permettono di semplificare la dimostrazione dei teoremi.
Per studiare le dimostrazioni, parte della logica matematica si è occupata di formalizzarle. Tra i sistemi logici più vicini al modo di "ragionare di un matematico", vi sono il calcolo dei sequenti e la deduzione naturale, entrambi introdotti da Gentzen nel 1935 ("Untersuchungen uber das logische Schliessen", vedi [22]):

- Il calcolo dei sequenti è uno strumento di base per lo studio delle leggi della logica; in esso, l'utilizzazione dei lemmi corrisponde ad una figura fondamentale del ragionamento: la regola del taglio.
- La deduzione naturale è un sistema in cui le regole logiche applicate assumono un ruolo di primaria importanza nella rappresentazione delle dimostrazioni: queste diventano "alberi" (grafi aciclici e connessi), i cui nodi sono le formule e le cui diramazioni sono le regole logiche.

Tra i risultati principali di Gentzen, vi è il **"teorema di eliminazione del taglio"**: in logica pura del primo ordine, una qualsiasi dimostrazione π di una formula A può essere trasformata in una dimostrazione π' di A che non

contiene alcuna occorrenza della regola di taglio. La dimostrazione del teorema suddetto oltre a fornire una dimostrazione cut-free π' di A, indica una strategia di riduzione (tramite delle regole di trasformazione) che a partire da π conduce a tale π'. Intuitivamente questo significa che è sempre possibie (almeno in teoria) dimostrare un teorema senza fare mai uso dei lemmi.

Oltre ad interassare i logici matematici, lo studio del processo di eliminazione del taglio ha riscosso un notevole interesse nel mondo informatico(teorico). Vediamo di capirne il motivo.

In maniera completamente indipendente dalla logica, negli anni '30 Church introdusse un sistema di termini detto λ-calcolo puro (di cui non parleremo in questa tesi); questo condusse a notevoli altri sviluppi, che successivamente portarono alla definizione del λ-calcolo tipato semplice. I termini che si definiscono in tale sistema hanno una caratteristica particolarmente interessante: essi possono essere visti come programmi e il tipo di tali termini corrisponde ad una specificazione di quello che il programma fa in modo astratto.

La scoperta della corrisondenza Curry-Howard, che è un isomorfismo tra dimostrazioni della logica intuizionista (minimale per l'esattezza) ristretta al frammento $(\wedge, \rightarrow)$ e termini del λ-calcolo tipato semplice consente, quindi, di stabilire un nesso tra la logica e l'informatica.

In quest'ambito, una dimostrazione può essere vista come un programma la cui esecuzione corrisponde in un senso molto preciso ad eliminare i tagli dalle dimostrazioni. Il processo di eliminazione del taglio consiste nel trasformare una dimostrazione in un'altra senza tagli mediante una successione di regole di riduzioni, come già osservato in precedenza; queste regole di riduzione rappresentano una sorta di calcolo. A questo punto è possibile porsi la seguente domanda: cosa si preserva sotto le regole di questo calcolo?

A tal proposito, viene introdotta la semantica denotazionale che costituirà uno degli argomenti portanti trattati in questa tesi.

Semantica denotazionale

Il tipo di semantica che consideriamo, in questa tesi, è concreto: ad ogni dimostrazione π, associamo un insieme, π^*, che è la *traccia* di una funzione. Questa associazione può essere vista come un modo di definire un'equivalenza, che denoteremo con $\approx$, tra dimostrazioni ($\pi \approx \pi'$ sse $\pi^* = (\pi')^*$) delle stesse formule (o degli stessi sequenti), e che soddisfa le seguenti condizioni:

1. se π' è la dimostrazione ottenuta da π, mediante il processo di eliminazione del taglio, allora $\pi \approx \pi'$;
2. la relazione $\approx$ è non degenere: esiste una formula con almeno due dimostrazioni non equivalenti;
3. la relazione $\approx$ è una congruenza: se π e π' sono state ottenute da λ e λ', mediante la stessa regola logica, e se $\lambda \approx \lambda'$, allora $\pi \approx \pi'$.

La semantica denotazionale che presentiamo è quella degli spazi coerenti, introdotta da Girard in [10], che viene prima descritta per la logica intuizionista e poi si vedrà come proprio la struttura di spazio coerente condurrà in modo naturale alla nascita della Logica Lineare. Per far questo ripercorriamo la strada che ha portato Girard all'introduzione della Logica Lineare, utilizzando la definizione di *Spazio Coerente* e di *funzione stabile*.

Definizione 1.0.1.

- X è uno **spazio coerente** sse X è un grafo non orientato e riflessivo, ossia, sse X = (|X|, $\frown$ [mod X]), dove |X| è un insieme e $\frown$ [mod X] è una relazione di coerenza binaria riflessiva e simmetrica su |X|.
- Gli elementi dello spazio coerente X sono chiamati cricche: a $\sqsubseteq$ X ("a è una cricca di X") sse a $\subseteq$ |X| $\wedge \forall$ x $\forall$ y(x $\in$ a $\wedge$ y $\in$ a $\rightarrow$ x $\frown$ y [mod X])

Su tali spazi coerenti è possibile definire anche tre tipi di operazioni: moltiplicative, additive ed esponenziali.

Ora introduciamo la nozione di funzione stabile tra spazi coerenti:

Definizione 1.0.2. Siano X e Y due spazi coerenti.

- F è una funzione **stabile** da X in Y quando F è una funzione dall'insieme delle cricche di X all'insieme delle cricche di Y tale che:
- se $a \subseteq b$ allora $\mathrm{F}(a) \subseteq F(b)$ (Monotonia)
- se (I, $\leq$) è un insieme diretto, e $(a_i)_{i \in I}$ è una famiglia di cricche di X, allora

$$F(\bigcup_{i \in I} a_i) = \bigcup_{i \in I} F(a_i) \text{ (continuità)}$$

- se per ogni cricca a di X e ogni cricca b di X

$$a \cup b \sqsubseteq X \rightarrow F(a \cap b) = F(a) \cap F(b)(stabilit)$$

A questo punto andiamo a definire la nozione di traccia per le funzioni stabili, cioè:

Definizione 1.0.3. Sia F una funzione stabile. Tr(F) (la traccia di F) è definita da:

$$Tr(F) = [(b, z)/b \sqsubseteq X finita, z \in |Y|, b \quad minimale \quad per \quad z \quad sotto \quad F]$$

Tramite la nozione di traccia si dimostra l'importante equazione

$$X \Rightarrow Y = !X \multimap Y$$

Dimostreremo (dopo avere definito lo spazio coerente $X \Rightarrow Y$) infatti le due proposizioni seguenti:

Proposizione 1.0.4. *$a \sqsubseteq X \Rightarrow Y$ se e solamente se a=Tr(F) per qualche funzione stabile da X in Y.*

Proposizione 1.0.5. *$a \sqsubseteq !X \multimap Y$ se e solamente se a=Tr(F) per qualche funzione stabile da X in Y.*

Questa equazione mostra una decomposizione matematica dell'implicazione intuizionista in due operazioni: la prima è l'esclamazione dello spazio coerente X, la seconda è l'implicazione lineare tra lo spazio coerente esclamato e Y. Tale decomposizione può essere internalizzata, cioè, assumere un significato logico.

Così come ! e $\multimap$ anche tutte le altre operazioni definite sugli spazi coerenti hanno un significato logico: sono i connettivi della Logica Lineare.

Dagli spazi coerenti per la logica intuizionista nasce naturalmente la Logica Lineare.

Per tale Logica Lineare andiamo a definire un alfabeto, l'insieme delle regole del calcolo dei sequenti e descriviamo una semantica coerente, seguendo quanto fatto da Girard in [13]. Lo scopo è quello di interpretare una dimostrazione π di un sequente $\vdash X_1, \cdots, X_n$ nel calcolo dei sequenti lineare in modo tale che l'interpretazione sia invariante rispetto al processo di eliminazione del taglio. Andiamo pertanto ad associare ad ogni formula della logica lineare uno spazio coerente (in modo molto naturale) e associamo al sequente $\vdash X_1, \cdots, X_n$ lo spazio coerente così definito:

Definizione 1.0.6. Al sequente $\vdash X_1, ..., X_n$ associamo lo spazio coerente, denotato con $\Vdash X_1^*, \cdots, X_n^*$: il cui supporto è dato da: $|\vdash X_1, ..., X_n| = |X_1| \times .... \times |X_n|$ e la relazione di compatibilità è definita da:
$(x_1, ..., x_n) \frown (y_1, ..., y_n)$ [mod $\Vdash X_1^* \cdots X_n^*$]
sse $x_i \frown y_i$ [mod X_i], per qualche i.

Il passo successivo è quello di interpretare in modo coerente le dimostrazioni. Mediante la definizione precedente possiamo interpretare una dimostrazione π di $\vdash X_1, \cdots, X_n$ come una cricca $\pi^* \sqsubseteq \Vdash X_1^*, \cdots, X_n^*$.

Andiamo, poi, a definire una procedura di eliminazione del taglio per il frammento moltiplicativo della logica lineare (indicato con MLL) e, per tale frammento, mostriamo che l'interpretazione coerente delle dimostrazioni della logica lineare è un invariante per il calcolo.

L'introduzione della logica lineare ha apportato notevoli innovazioni nella teoria della dimostrazione; una delle più significative è certamente l'intro-

duzione dei *proof-net* (o "reti di dimostrazione"), data da Girard in [10] che forniscono, per la prima volta, una rappresentazione geometrica delle dimostrazioni. Ad ogni dimostrazione del calcolo dei sequenti lineare si può associare un grafo i cui nodi sono chiamati legami e rappresentano le regole ed i cui archi sono etichettati da formule. Il fatto sorprendente è che la classe dei grafi, corrispondenti a dimostrazioni, è caratterizzata da alcune proprietà *puramente geometriche* che non fanno alcun riferimento al calcolo dei sequenti. Chiameremo, nel corso del lavoro, una tale proprietà "criterio di correttezza".
In questa parte della tesi, lavoriamo con il frammento moltiplicativo della logica lineare, MLL; riportiamo la definizione di proof-net moltiplicativo introdotta da Danos in [4] e ci occupiamo della studio del criterio aciclico connesso su cui si basa tale definizione. Il criterio di cui parliamo caratterizza i proof-net come le sole strutture di dimostrazione i cui grafi di correttezza sono tutti aciclici e connessi.

Prima di poter dire che cosa è un proof-net è necessario definire una struttura di dimostrazione e dare altre definizioni essenziali:

Definizione 1.0.7. (**Struttura di dimostrazione**) Una struttura di dimostrazione è un grafo orientato i cui nodi sono chiamati legami ed in cui gli archi sono etichettati da formule della Logica Lineare.
Ogni legame ha un certo numero di archi incidenti, chiamati premesse del legame, ed un certo numero di archi emergenti, chiamati conclusioni del legame; le premesse del legame sono chiamate le formule attive del legame.
I legami sono:

- un legame **assioma** ha due conclusioni etichettate da due formule duali, e nessuna premessa.
- Un legame **cut** ha due premesse etichettate da due formule duali e nessuna conclusione.
- Un legame **tensore** ($\otimes$) ha due premesse e una conclusione. Se la premessa sinistra del legame è etichettata dalla formula A e la premessa

destra è etichettata dalla formula B, allora la conclusione del legame sarà etichettata dalla formula A $\otimes$ B. (Questo legame non è simmetrico, ovvero, il legame che ha come conclusione A $\otimes$ B è diverso da quello che ha come conclusione B $\otimes$ A).

- Un legame **par** ($\wp$) ha due premesse e una conclusione. Se la premessa sinistra del legame è etichettata dalla formula A e la premessa a destra è etichettata dalla formula B, allora, la conclusione del legame è etichettata dalla formula A $\wp$ B.(Questo legame non è simmetrico, ovvero, il legame che ha come conclusione A $\wp$ B è diverso da quello che ha come conclusione B $\wp$ A).

- Un legame **ipotesi** (H) Ha n ≥ 1 conclusioni, ciascuna etichettata da una formula e nessuna premessa.

Un grafo orientato G, ottenuto utilizzando i legami appena enunciati, è una struttura di dimostrazione se sono verificate queste due condizioni:

1. ogni occorrenza di formula è premessa di al più un legame;

2. ogni occorrenza di formula è conclusione di esattamente un legame.

Definizione 1.0.8. (**Grafo con coppie**) Si chiama Grafo con coppie, un grafo orientato e finito S, costituito da un insieme C(S) di coppie a due a due disgiunte di archi coincidenti.
Si definisce **base** di una coppia un vertice dove coincidono i suoi due archi (nel caso in cui i due archi della coppia siano coicidenti in entrambi i vertici, la coppia avrà due basi).
Si definisce, inoltre, **estremità** di una coppia i vertici che non sono base della coppia.

Definizione 1.0.9. Sia S un grafo con coppie (una struttura di dimostrazione è un caso particolare di grafo con coppie). Chiameremo **Grafo di correttezza** di S, un grafo non orientato ottenuto a partire da S cancellando esattamente uno dei due archi di ogni coppia di S.

Definizione 1.0.10. Uno **switching** per una struttura di dimostrazione è la scelta di uno dei due archi per ogni legame $\wp$; ogni switching induce un grafo di correttezza.

Sulla definizione di grafo di correttezza di una struttura di dimostrazione si sviluppa gran parte del lavoro svolto in questa parte della tesi.

Ora, possiamo dare la definizione di proof-net:

Definizione 1.0.11. Sia R una struttura di dimostrazione. Si dice che R è corretta o che è un **proof-net** (con ipotesi), sse tutti i suoi grafi di correttezza sono aciclici e connessi.

Riguardo ai proof-net riportiamo alcuni risultati di cui ci siamo occupati in questa tesi:

Teorema 1.0.12. *(**Teorema di sequenzializzazione**) Sia R un proof-net di conclusione* Γ*, allora esiste una dimostrazione* π *nel calcolo dei sequenti della logica lineare di conclusione* Γ *e tale che* $R = \pi^{-}$*.*

Il teorema menzionato permette di distinguere, quali tra le strutture di dimostrazione, corrispondono a prove nel calcolo dei sequenti e, quindi, sono vere dimostrazioni. La sua dimostrazione necessita di un teorema altrattanto fondamentale, il teorema della sezione di [4], che rappresenta uno dei risultati più interessanti del lavoro svolto:

Teorema 1.0.13. *(**Teorema della sezione**). Sia S un grafo con coppie tale che ogni suo grafo di correttezza è aciclico, e tale che C(S) non sia vuoto; allora una delle coppie di S è una sezione.*

Grazie all'introduzione dei proof-net, il riferirsi a determinate proprietà geometriche, fa si, come vedremo, che il processo di eliminazione del taglio per essi diventa più semplice se paraganato a quello descritto precedentemente per il calcolo dei sequenti.
Dopo aver descritto l'eliminazione del taglio per i proof-net di MLL, dimostriamo un importante risultato noto sotto il nome di **preservazione della correttezza** il cui enunciato è il seguente:

Proposizione 1.0.14. *Se R è un proof-net, e R' è ottenuta a partire da R, applicando uno dei passi elementari di riduzione precedentemente descritti, allora R' è ancora un proof-net.*

Successivamente ci siamo soffermati sullo studio di un altro criterio di correttezza che si basa su due regole di retrazione definite da Danos in [4]. Questo criterio permette di caratterizzare i proof-net come le sole strutture di dimostrazione che si retraggono in un vertice. Grazie al criterio delle retrazioni, la verifica della correttezza di una struttura di dimostrazione è ancora più agevole di prima: si richiede, soltanto di verificare quali strutture di dimostrazione si retraggono in un vertice. La definizione di retrazione è la seguente:

Definizione 1.0.15. (**Retrazioni**). Si definiscono due regole di riscrittura che noi chiameremo "**retrazioni**", sui grafi con coppie nel seguente modo:

Con la condizione seguente sul grafo con coppie S:

1. per la 1-retrazione, l'arco non appartenga a nessuna coppia di C(S) e che le sue estremità siano distinte.

2. per la 2-retrazione, i due archi appartengano ad una medesima coppia di C(S) e le loro due estremità in comune siano distinte.

Proseguiamo con i due teoremi più significativi di questa parte del lavoro: il teorema che esprime l'equivalenza tra questo criterio e quello aciclico e

connesso, e il teorema di sequenzializzazione che utilizza la definizione di struttura di prova sequenzializzabile, entrambi questi risultati sono stati dimostrati in [4]-[6].

Teorema 1.0.16. *(**Equivalenza**). Sia S un grafo con coppie: S si retrae in un solo vertice sse tutti i suoi grafi di correttezza sono aciclici e connessi.*

Teorema 1.0.17. *(**Torema di sequenzializzazione**). Sia S una struttura di dimostrazione. S è una struttura di dimostrazione sequenzializzabile sse S si ritrae in un solo vertice.*

La semantica delle dimostrazioni che abbiamo definito precedentemente induce una semantica dei corrispondenti proof-net. In realtà, quello che ci proponiamo di fare è di lavorare direttamente sui proof-net. Quindi occorre interpretare un proof-net in modo coerente. Questo viene fatto dando la nozione di esperienza introdotta da Girard in [10].

Definizione 1.0.18. Sia R una struttura di dimostrazione. Un'**Esperienza** e di R è un'applicazione che associa ad ogni arco a, etichettato da A, di R un insieme {x}, con x $\in |A^*|$.

Definizione 1.0.19. Sia R una struttura di dimostrazione e $a_1, ..., a_n$ di tipo $A_1, ..., A_n$, le conclusioni di R, e sia e un'esperienza di R. Sia e$(a_i) = \{x_i\}$, $\forall$ i $\in$ {1,...,n}.
Si dice che $(x_1 \cdots x_n) \in |A_1^* \wp \cdots \wp A_n^*|$ è la conclusione, (o risultato), dell'esperienza e di R. Si può definire l'interpretazione di un proof-net R di conclusioni $A_1 \cdots A_n$, come l'insieme:
$[R] = \{\gamma \in |A_1 \wp \cdots \wp A_n|$, esiste un'esperienza e di R di risultato $\gamma\}$

In questa parte della trattazione, seguendo quanto svolto da Girard nel suo lavoro, si è dimostrato che un'esperienza è univocamente determinata dal suo risultato, quindi, è ben posta. Da questo scaturisce che l'insieme dei risultati di tutte le esperienze di un proof-net è una cricca dello spazio coerente associato al $\wp$ delle conclusioni di R, e quindi, è la traccia di una funzione lineare.

Infine, abbiamo affrontato la questione dell'invarianza dell'interpretazione rispetto al processo di eliminazione del taglio; rivisitando la dimostrazione vista in [10] è stato trattato quest'ultimo risultato:

Teorema 1.0.20. *Se R un proof-net e R si riduce in R' applicando uno dei passi elementari di riduzione precedentemente descritti, allora $R^* = R'^*$*

Bibliografia

[1] V.M. Abrusci. Logica classica. Dispense del corso di logica matematica.

[2] C. Barcaglioni. Le dimostrazioni logiche come costruzioni: spazi coerenti ed esperienze, Febbraio 2001.

[3] T. Brauner. Introduction to linear logic. *BRICS*, 1996.

[4] V. Danos. *La logique linéaire appliquée à l'étitude de diver processus de normalisation(principalment du lambda-calcul).* PhD thesis, Paris 7, 1990.

[5] V. Danos, J. Joinet, and H. Schellinx. A new deconstructive logic: linear logic. *Journal of Symbolic Logic*, 1997.

[6] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 1989.

[7] L. Tortora de Falco. Coherent obsessional experiments for linear logic proof-net. *Mathematical Structures in Computer Science.*

[8] L. Tortora de Falco. *Réseaux cohérence et expérience obsessionnelles.* PhD thesis, Paris 7-matematica, 2000.

[9] E. Duquesne and J. Van de Wiele. Modé le cohérent des reséaux de preuve. *Archive for Mathematical Logic*, 1994.

[10] J.Y. Girard. Linear logic. *Theoretical Computer Science*, 1987.

[11] J.Y. Girard. A new constructive logic: Classical logic. *Mathematical Structures in Computer Science*, 1991.

[12] J.Y. Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 1992.

[13] J.Y. Girard. Linear logic: its sintax and semantics. *Advances in linear logic*, 1995.

[14] J.Y. Girard. *Proof-nets: the parallel syntax for proof-theory.* Logic and Algebra, New-York, 1995.

[15] J.Y. Girard. On the meaning of logical rules 2: multiplicatives and additives. *Institut de Mathe matiques de Luminy, Marseille*, 1998.

[16] J.Y. Girard. *On the meaning of logical rules 1: syntax vs. semantics.* U. Berger and H. Schwichtenberg, editors, Computational Logic, Heidelberg, 1999.

[17] J.Y. Girard, Y. Lafont, and P. Taylor. *Proof and types.* Cambridge University Press, 1989.

[18] S. Guerrini. Correctness of multiplicative proof-nets is linear. *Annual Symposium on Logic in Computer Science*, 1999.

[19] J. L. Krivine and M. Parigot. Programming with proofs. *Journal of Information Processing and Cybernetics EIK (formerly Elektronische Informationsverarbeitung und Kybernetik)*, 1990.

[20] O. Laurent, M. Quatrini, and L. Tortora de Falco. Logique linéaire polariseé et logique classique. *Annals of Pure and Applied Logic*, 1999.

[21] L. Di Renzo. Le dimostrazioni logiche come grafi: i proof-net, 1999-2000.

[22] M.E. Szabo. Collected papers of gehrard gentzen. *North-Holland Publishing Company, Amsterdam-London*, 1969.