FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

GE460 - TEORIA DEI GRAFI

PROOF-NET

Elettra Cesareo
matricola 554337

Anno Accademico 2019-2020
Dipartimento di Matematica

Introduzione

In questo breve studio mi occupererò di introdurvi al mondo della logica che come vedremo, spesso si collega alla teoria dei grafi.

Ogni logica ha le sue particolarità. La maggior parte delle definizioni in logica vengono date per induzione, è quindi importante capire com'è strutturata una tale definizione. Solitamente, per definire una logica, si parte definendo per induzione le formule che ne costituiscono il linguaggio formale, si introduce il concetto di negazione e poi si parla di regole di inferenza, nel nostro caso nel calcolo dei sequenti, e di regole di trasformazione.

Nel primo capitolo vi farò vedere brevemente com'è definita la logica classica, per poi accennare alla logica intuizionista e lineare. Per concludere il capitolo introdurrò il concetto di *semantica* e *sintassi* di una logica in modo da potervi parlare finalmente, nel secondo capitolo, dei *proof-net*. Dopo aver definito il concetto di proof-net ed aver mostrato alcune osservazioni ed esempi, ci concentreremo sull'enunciato e la dimostrazione del teorema di sequenzializzazione, che sarà il risultato più importante di questo studio e ci permetterà di identificare il concetto di dimostrazione logica con un oggetto puramente geometrico: un grafo.

Nel primo capitolo, ogni volta che utilizzeremo il termine "logica" (a meno che non sia stato esplicitato diversamente) ci riferiremo alla logica classica del primo ordine.

Ovviamente per essere il più possibile sintetici non ci addentreremo troppo nello specifico della materia, anche perché per studiare a fondo certi argomenti ci sarebbe bisogno di un corso intero. Quindi vi invito a non considerare completo tutto quello che dirò poiché, per comprimere il tutto in poche pagine, potrei aver omesso concetti molto importanti per una comprensione profonda.

Spero che questa introduzione vi possa avvicinare al meraviglioso mondo della logica.

Indice

1 Cenni di logica

1.1 Definizioni preliminari

L'alfabeto della logica classica del primo ordine contiene le unità logiche $VERO$ e $FALSO$, i connettivi $\wedge$ e $\vee$, i quantificatori $\forall$ e $\exists$, un insieme infinito numerabile di variabili per individui, un insieme di variabili per funzioni n-arie, un insieme di variabili per proposizioni, un insieme di variabili per predicati n-ari ed i simboli ausiliari.

Sull'insieme costituito dalle variabili per proposizioni e dalle variabili per predicati è definita una funzione $iniettiva$ e $suriettiva$, denotata con NON, tale che per ogni variabile A appartenente a questo insieme, $NON(A)$ è una variabile dello stesso tipo della variabile v e $NON(NON(A)) = A$.

Le formule della logica classica del primo ordine sono definite come segue, con una definizione induttiva.

Definizione 1.1.1.

- BASE INDUZIONE
  - le unità logiche sono formule;
  - le variabili per proposizioni sono formule;
  - se P è una variabile per predicato n-ario, e $t_1, ..., t_n$ sono termini individuali, allora $P(t_1, ..., t_n)$ è una formula;
- PASSO DI INDUZIONE
  - se A e B sono formule, allora $A \wedge B$ e $A \vee B$ sono formule;
  - se A è una formula e x è una variabile individuale e $\forall x$ e $\exists x$ non compaiono in A, allora $\forall x A$ e $\exists x A$ sono formule;

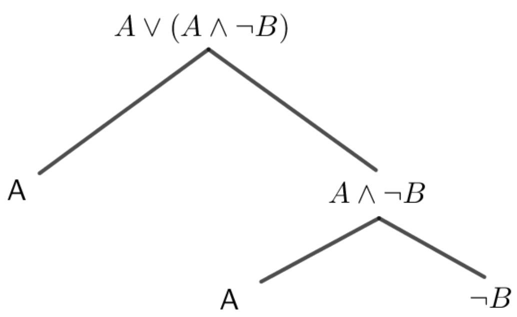
- CLAUSOLA FINALE
  - nient'altro è una formula.

DEFINIZIONE 1.1.2. Per ogni formula A, *la negazione* di A (denotata da $\neg A$) si definisce passando attraverso tutti i connettivi e i quantificatori, nel modo seguente:

- $\neg VERO = FALSO$ e $\neg FALSO = VERO$;
- se P è una variabile per proposizioni, allora $\neg P = NOT(P)$;
- se P è una variabile per predicato $n-$ario e $t_1, ..., t_n$ sono termini individuali, allora $\neg(P(t_1, ...t_n)) = NOT(P)(t_1, ..., t_n)$;
- $\neg(A \wedge B) = \neg A \vee \neg B$, $\neg(A \vee B) = \neg A \wedge \neg B$;
- $\neg\forall v A = \exists v \neg A$, $\neg\exists v A = \forall v \neg A$.

È facile mostrare che da questo segue che $\neg(\neg A) = A$.
A ciascuna formula A si associa il suo unico albero generativo, un albero di formule in cui la radice è A, le foglie sono formule atomiche e ciascun nodo è ottenuto dai nodi immediatamente precedenti con una delle operazioni previste nel passo di induzione.

ESEMPIO 1.1.1. Vediamo un esempio di come si rappresenta un albero generativo di una formula. Supponiamo di avere la formula $A \vee (A \wedge \neg B)$, allora l'albero generativo associato a tale formula è il seguente:



DEFINIZIONE 1.1.3. Una formula B è $sottoformula$ di una formula A se

e soltanto se B è un nodo dell'albero generativo di A.

DEFINIZIONE 1.1.4. Un *sequente* è un multinsieme finito di formule, e viene spesso rappresentato da Γ o Δ.

1.2 Regole nel calcolo dei sequenti e derivazioni

Le regole nel calcolo dei sequenti sono molte e per definirle e comprendere il significato di tutte servirebbe troppo tempo, quindi noi ci limiteremo ad elencarle tutte e definirne solo alcune. Se qualcuno fosse interessato ad uno studio più approfondito lo invito a leggere "Logica: Volume 1 - Dimostrazioni e modelli al primo ordine".
Le regole del calcolo dei sequenti si dividono in:

- REGOLE DI BASE:
  - AX (Axiom o Assioma);
  - CUT
- REGOLE STRUTTURALI:
  - W (Weakening o regola di indebolimento);
  - C (Contraction o regola di contrazione)
- REGOLE SULLE UNITA' LOGICHE:
  - 1 (Vero moltiplicativo)
  - $\perp$ (Falso moltiplicativo)
  - T (Vero additivo)
- REGOLE SUI CONNETTIVI:
  - $\otimes$ ($\wedge$ moltiplicativa)
  - $\parr$ ($\vee$ moltiplicativa)
  - $\oplus$ ($\vee$ additiva)

- $\&$ ($\wedge$ additiva)

- REGOLE SUI QUANTIFICATORI:
  - $\exists$ (Regola esiste);
  - $\forall$ (Regola per ogni)

Andiamo a vedere come sono definite alcune di queste regole.

- AX:

$$\frac{}{\vdash \neg A, A}\ (AX)$$

- CUT:

$$\frac{\vdash \Gamma, A \qquad\qquad \vdash \Delta, \neg A}{\vdash \Gamma, \Delta}\ (CUT)$$

- $\otimes$:

$$\frac{\vdash \Gamma, A \qquad\qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \wedge B}\ (\otimes)$$

- $\parr$:

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B}\ (\parr)$$

Le parti superiori alla linea di inferenza di una regola si chiamano *premesse* e quella inferiore è detta *conclusione*. Se una regola non ha premesse si dice che quella regola è $0-$aria, se ha una sola regola allora è $1-$aria, ecc. Le regole nel calcolo dei sequenti della logica classica sono tutte al massimo binarie.

Andiamo ora, finealmente, a definire cos'è una derivazione. Anche in questo caso si utilizza il concetto di albero.

DEFINIZIONE 1.2.1. $Un'analisi$ di un sequente $\vdash \Gamma$ nel calcolo dei sequenti da un insieme M di sequenti è un albero di sequenti nel quale ciascun nodo:

- se è la radice (ossia, se non ha successori), allora è il sequente $\vdash \Gamma$;
- se è una foglia (ossia se non ha predecessori) è conclusione di una regola 0-aria del calcolo dei sequenti oppure è un sequente appartenente a M;
- se ha un solo predecessore, allora è conclusione di una regola $1-$aria la cui premessa è il sequente suo predecessore;
- se ha due soli predecessori, allora è conclusione di una regola binaria le cui premesse sono i due sequenti suoi predecessori.

DEFINIZIONE 1.2.2. Una derivazione di un sequente $\vdash \Gamma$ nel calcolo dei sequenti da un insieme M di sequenti è un'analisi finita di $\vdash \Gamma$ da M.

Vediamo un esempio di derivazione di un sequente.

ESEMPIO 1.2.1.
Sia $\Gamma = \{A \wedge A, \neg A \vee \neg A\}$, allora $\vdash \Gamma$ si può derivare nel seguente modo: Ovviamente una derivazione non è unica ma questo sarebbe molto più chiaro

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, \neg A}\,(AX) \qquad \dfrac{}{\vdash A, \neg A}\,(AX)}{\vdash A \wedge A, \neg A, \neg A}\,(\otimes) \qquad \dfrac{}{\vdash A, \neg A}\,(AX)}{\vdash A \wedge A, \neg A, \neg A}\,(\text{CUT})}{\vdash A \wedge A, \neg A \vee \neg A}\,(\parr)$$

andando a definire le regole di indebolimento (W) e di contrazione (C) che sostanzialmente permettono di inserire la copia di una formula già presente nel sequente e di toglierne una qualsiasi, è chiaro a questo punto che è possibile applicare le due formule continuamente e creare così un numero qualsiasi di derivazioni dello stesso sequente.

1.3 Regole di trasformazione e teorema di CUT-elimination

Uno dei più importanti risultati della logica è il teorema di $CUT-elimination$ che sostanzialmente afferma che qualsiasi derivazione di un sequente in cui è presente una o più applicazioni della regola CUT può essere trasformata in una derivazione $equivalente$ alla prima che non contiene alcuna applicazione della regola CUT.

L'enunciato è il seguente:

Teorema 1.3.1. Se π è una derivazione di Γ nel calcolo dei sequenti per la logica classica del primo ordine allora esiste una derivazione $cut - free\ \psi$ di $\vdash \Gamma$ (nel calcolo dei sequenti per la logica classica del primo ordine) tale che:

- $\pi \rightsquigarrow \psi$ (π si riduce a ψ).
- se π non ha regole strutturali, allore nemmeno ψ ha regole strutturali.

Idea della dimostrazione. La dimostrazione di questo teorema è molto lunga e consiste nello studiare caso per caso tutte le possibili combinazioni di regole che si possono presentare prima dell'applicazione di una regola CUT e nella sua trasformazione in una derivazione equivalente ma senza l'uso della regola CUT. Queste vengono chiamate, appunto, regole di trasformazione.

Per chiarire le idee, vediamo un esempio di una tale regola di trasformazione. Supponiamo di avere una derivazione in cui prima della regola CUT è presente una regola AX. La regola di trasformazione associata alla coppia di regole (CUT, AX) è la seguente:

$$\dfrac{\dfrac{}{\vdash A, \neg A}\,(AX) \qquad \begin{matrix}\pi \\ \vdots \\ \vdash \Gamma, \neg A\end{matrix}}{\vdash \Gamma, \neg A} \quad \rightsquigarrow \quad \begin{matrix}\pi \\ \vdots \\ \vdash \Gamma, \neg A\end{matrix}$$

L'importanza di questo teorema segue dal fatto che in una derivazione senza tagli di una formula A occorrono esclusivamente sottoformule di A. Questa è una banale conseguenza del fatto che ogni regola diversa dalla regola di taglio gode della seguente proprietà: qualunque formula occorra in una delle premesse della regola è sottoformula di una delle formule che occorrono nella conclusione della regola.

1.4 Cenni su logica intuizionista e logica lineare

La logica intuizionista si ottiene dalla logica classica mediante la rinuncia alla $dualità$ (negazione). Mentre la logica lineare si caratterizza per molti aspetti innovativi in confronto alla logica intuizionista. Nella logica lineare verranno aggiunti altri due operatori detti "$esponenziali$" e denotati "?" e "!" che hanno un comportamento per molti aspetti analogo ai quantificatori ed un nuovo concetto: $la\ sottolineatura$ di una formula. Come nella logica classica, si procede definendo cosa si intende per formula e poi andando a studiare la negazione che in logica intuizionista è definita nel seguente modo:

DEFINIZIONE 1.4.1. Se A è una formula intuizionista, allora la negazione intuizionista di A (denotata $\sim A$) è la formula definita come segue: $\sim A := A \to FALSO$.

OSSERVAZIONE 1.4.1. Notiamo che in questo caso la doppia negazione di una formula intuizionista A non è sempre equivalente ad A.

Dopodiché si definiscono le regole per il calcolo dei sequenti per la logica intuizionista e lineare e le rispettive regole di trasformazione.

Ma a questo punto vi starete chiedendo perché tutto questo? Perché andiamo a definire nuove logiche, nuovi operatori nuovi concetti? A cosa serve tutto questo?

Lo scopo della logica intuizionista e lineare è rimediare ad un fatto che nella logica classica abbiamo totalmente ignorato, ossia "il significato di una dimostrazione". In logica classica vale il seguente teorema:

TEOREMA 1.4.1. Sia A una formula, π_1 una derivazione di $\vdash A$ e π_2 una derivazione di $\vdash A$. Esiste una derivazione ψ di $\vdash A$ tale che $\psi \rightsquigarrow \pi_1$ e $\psi \rightsquigarrow \pi_2$. Pertanto, $||\pi_1|| = ||\pi_2||$ (il significato di π_1 è uguale al significato di π_2.

Ossia in logica classica siamo sempre obbligati a dichiarare uguali tutte le derivazioni di una stessa formula; in sostanza, ciò mostra che in logica classica quel che conta non sono le singole dimostrazioni bensì la dimostrabilità. Invece, grazie alle restrizioni della logica intuizionista e lineare, si riesce ad assegnare un significato distinto a diverse dimostrazione della stessa formula o proposizione. Questi due sistemi, specialmente la logica lineare, ruotano intorno al concetto di *semantica delle dimostrazioni*: si vuole interpretare ogni dimostrazione con un oggetto matematico, sviluppando rigorosamente l'idea intuitiva che una dimostrazione di B dall'ipotesi A sia una funzione che manda ogni dimostrazione di A in una dimostrazione di B.

A partire da queste ricerche si ritorna verso la sintassi, con l'idea di trovare delle *forme normali* tali che due dimostrazioni semanticamente identiche siano anche sintatticamente, cioè *visibilmente*, identiche. Questa forma normale viene identificata con il lambda-calcolo per la logica intuizionista, mentre per la logica lineare si sviluppano i proof-net: grafi che rappresentano dimostrazioni, eliminando dettagli inutili e permutazioni di regole. Uno dei maggiori punti di interesse di questa soluzione è che il criterio di correttezza delle dimostrazioni non è più induttivo, ma puramente geometrico: un grafo è una dimostrazione sse gode di determinate proprietà globali.

2 Proof-net

2.1 Cenni storici sulla logica lineare ed introduzione ai proof-net

La logica lineare LL fa il suo debutto ufficiale nel 1986 con la pubblicazione di Linear Logic (Girard). Dal momento della sua nascita si nota immediatamente che è una teoria che rispetta tutti i canoni tradizionali.
Prima dell'avvento della logica lineare il concetto geometrico di dimostrazione è stato sempre un qualcosa di subordinato rispetto alle regole che si usano in una dimostrazione. Questo perché fino a quel momento si è prediletto il concetto sequenziale di dimostrazione: ossia essa era sempre identificata con la sua costruzione formata da un insieme di regole le quali venivano eseguite una dopo l'altra (questo approccio era il più naturale). Seguendo le regole sicuramente otteniamo risultati corretti ma il viceversa non è vero, infatti per ottenere risultati corretti non si deve per forza aver rispettato quelle regole. La domanda che ci poniamo è, quindi, la seguente: quando faccio una dimostrazione dobbiamo sempre presentarla con le regole che la caratterizzano oppure possiamo caratterizzarla medianti aspetti intrinsechi ad essa?
È proprio per questo che si è cominciato a cercare il concetto geometrico di dimostrazione che si è concretizzato, nello specifico caso moltiplicativo della logica lineare (MLL), nell'uso di grafi aciclici connessi (alberi), i quali si rivelano perfetti per incarnare il concetto geometrico di dimostrazione, infatti non solo non c'è bisogno di introdurre alcuna regola ma inoltre non bisogna specificare neppure che tipo di linguaggio si stia utilizzando, rendendo questo approccio molto più intuitivo, naturale e generale possibile.
Attraverso la teoria dei proof-net, assisteremo all'irruzione nella logica di criteri puramente geometrici: procederemo cioè a considerare le derivazioni logiche nella loro forma geometrica globale. Questo ci permetterà di mettere esplicitamente a tema l'innovativa connessione tra un grafo e una dimostrazione logica emersa dalla discussione del fondamentale teorema di sequenzializzazione.

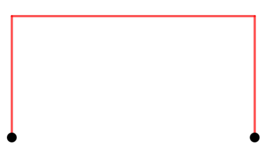
2.2 Proof-net e teorema di sequenzializzazione

Per comodità la logica lineare può essere suddivisa in più parti in base alle regole che si intende considerare. Con la sigla LL indichiamo la logica lineare nella sua interezza ossia in cui accettiamo di usare tutte le regole moltiplicative, additive ed esponenziali; con MLL indichiamo quella parte della logica lineare in cui accettiamo di usare soltanto le regole moltiplicative non considerando, dunque, quelle additive e gli esponenziali; infine indichiamo con $MALL$ quella parte della logica lineare in cui si accetta di usare le regole additive e moltiplicative, ignorando gli esponenziali. Per semplicità tratteremo la definizione di rete di prova in MLL. Inoltre eviteremo di trattare le unità logiche; questo perchè esse costituiscono una problematica ancora aperta, infatti non si è ancora trovato il modo di esprimere la loro essenza geometrica in maniera esplicita. Questo, probabilmente è dovuto al fatto di essere estremamente "elementari" e quindi di difficile trattazione (come ad esempio la nozione di punto nella geometria, il quale pur essendo estremamente intuitivo è addirittura di difficile definizione).

DEFINIZIONE 2.2.1. Una $proof - structure$ Π è un grafo finito (insieme finito di nodi alcuni legati fra di loro da un arco) il quale rispetta le seguenti proprietà:

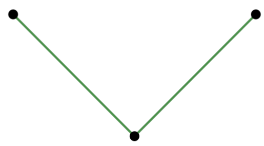
(1) Possono essere presenti solamente i seguenti legami fra due nodi:

- $Legame\ assioma\ (AX)$: esso è costituito da 2 punti che sono chiamati conclusione del legame ed un arco che li congiunge.

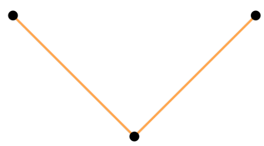


- *Legame cut* (CUT): esso è costituito da 2 punti che vengono chiamati premesse del legame ed un arco che li congiunge.

- *Legame tensore* ($\otimes$): esso è costituito da 3 punti di cui due sono le premesse del legame ed una è la conclusione. Sono presenti 2 archi che legano rispettivamente le premesse alla conclusione.



- *Legame par* ($\parr$): esso è costituito da 3 punti di cui due sono le premesse del legame ed una è la conclusione. Sono presenti 2 archi che legano rispettivamente le premesse alla conclusione.



(2) Ogni nodo è conclusione di esattamente un legame e premessa di al massimo un legame.

DEFINIZIONE 2.2.2. Un proof-net è una struttura di prova in cui per ogni eliminazione di uno solo dei due archi presenti nel legame ($\parr$) il grafo è connesso ed aciclico.

OSSERVAZIONE 2.2.1. La proprietà dei proof-net rende chiara la differenza, altrimenti inesistente, fra il legame tensore ed il legame par. La rimozione di uno dei due archi del legame par si chiama *switch*.

OSSERVAZIONE 2.2.2. Solo il legame assioma da solo è un proof-net in quanto tutti gli altri legami di base non rispettano da soli la proprietà 2.

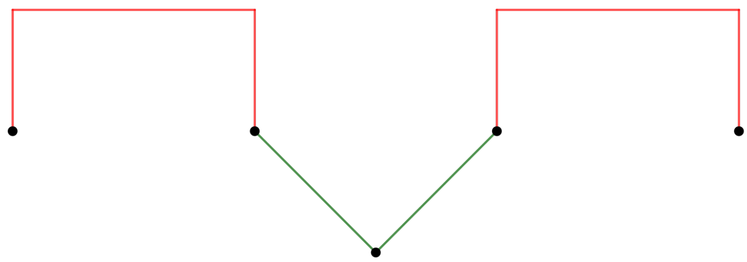
Inoltre il legame assioma è la rete di prova più semplice che esiste (di conseguenza, a questo legame, corrisponde la derivazione più semplice nel calcolo dei sequenti).

Osservazione 2.2.3. Viene richiesta l'aciclicità del grafo in quanto quando ad esso farò corrispondere una derivazione essa non dovrà ritornare mai sui suoi passi, reiterando regole già usate che non servono più.

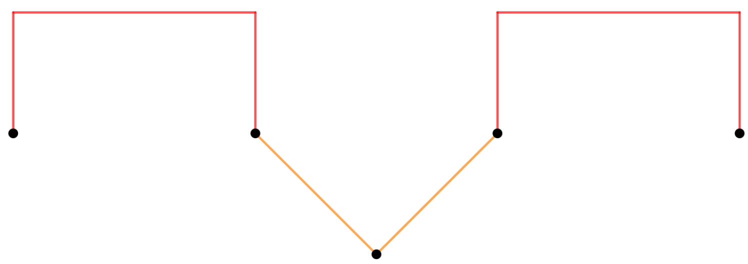
Osservazione 2.2.4. Viene richiesta la connessione perché essa mi garantisce che nella derivazione associata non ci siano parti scollegate e che tutto ciò che è presente prima o poi verrà utilizzato.

Osservazione 2.2.5. Se non sono presenti legami ($\parr$) l'aciclicità e la connessione sono garantiti dalle sole due proprietà delle proof-structure.

Esempio 2.2.1. Il seguente è un proof-net:



Esempio 2.2.2. Questo invece non è un proof-net in quanto qualunque arco del legame ($\parr$) io elimini il grafo non è più connesso dunque non soddisfo la proprietà 3; tuttavia è una proof-structure.



Definizione 2.2.3. *Un'etichettatura* di un proof-net Π è una funzione f che va dall'insieme dei nodi (che indicheremo con le lettere minuscole

$a, b, ...$) nell'insieme delle formule ($A, B, ...$)e che quando assegna etichette ai nodi dei legami rispetta queste proprietà:

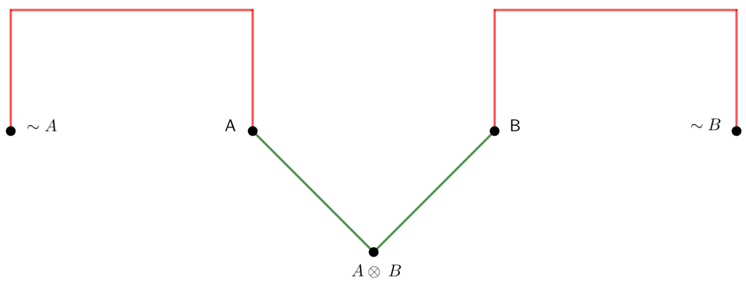
- Nel legame assioma etichetto i nodi con $f(a) = A$ e $\sim f(a) = \sim A$ in cui l'ordine non conta.

- Nel legame cut etichetto i nodi con $f(b) = B$ e $\sim f(b) = \sim B$ in cui l'ordine non conta.

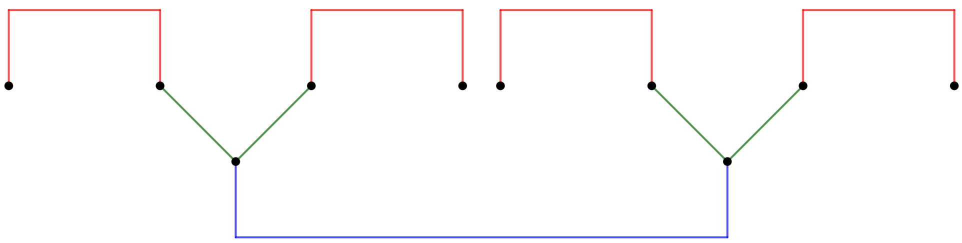
- Nel legame tensore se le premesse sono etichettate con le formule $f(a) = A$ e $f(b) = B$ allora la conclusione avrà come etichetta $f(a) \otimes f(b) = A \otimes B$.

- Nel legame par se le premesse sono etichettate con le formule $f(a) = A$ e $f(b) = B$ allora la conclusione avrà come etichetta $f(a) \parr f(b) = A \parr B$.

ESEMPIO 2.2.3. Il proof-net dell'esempio 2.2.1 è etichettabile nella seguente maniera.

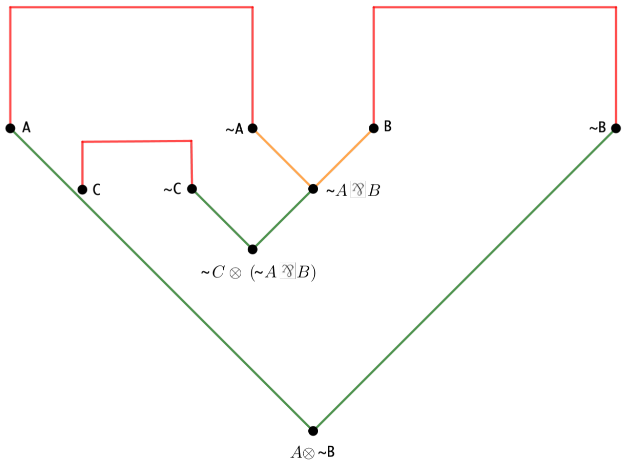


ESEMPIO 2.2.4. Il seguente proof-net, invece non è etichettabile poiché in qualsiasi modo si provi ad etichettare i nodi, il CUT tra i due tensori non potrà mai avere un nodo che è duale dell'altro, poiché la negazione di un $\wedge$ è un $\vee$.



DEFINIZIONE 2.2.4. I legami terminali di un proof-net sono l'insieme di tutti i nodi che sono conclusioni dei legami presenti nel proof-net e che non sono premessa di alcun legame.

ESEMPIO 2.2.5. Nel seguente proof-net sono legami terminali i nodi: $A \otimes \sim B, C, \sim C \otimes (\sim A \parr B)$.



OSSERVAZIONE 2.2.6. Un legame terminale CUT si comporta esattamente come un legame terminale $\otimes$, questo perché entrambi i legami hanno due premesse ed entrambi i legami non vengono influenzati dalla terza proprietà. Questa osservazione è utile poiché permette in molte dimostrazioni di trattare indistintamente questi due legami.

DEFINIZIONE 2.2.5. Un legame tensore si chiama *tensore splitting* se quando viene rimosso esso sconnette il grafo in tre componenti: le prime due sono a loro volta proof-net e l'ultima è il nodo di cui le due reti erano le premesse.

Enunciamo ora un teorema che sarà indispensabile per dimostrare il teorema di sequenzializzazzione.

LEMMA 2.2.1. (Lemma Fondamentale) Ogni proof-net Π che non ha legami terminali $\parr$ ammette almeno un $\otimes$ splitting (CUT splitting).

La dimostrazione del Lemma fondamentale necessita dell'introduzione di molte definizioni non banali che richiederebbero uno studio più lungo, quindi non la tratteremo.

Teorema 2.2.1. (Teorema di sequenzializzazione) Esiste una classe X di grafi (proof-net) tale che:

(1) Pero ogni derivazione $\mathcal{D}$ di $\vdash \Gamma$ nel calcolo dei sequenti esiste un $\Pi = \phi(\mathcal{D}) \in$ X ed esiste un'etichettatura f di Π tale che $f(\Pi)$ è un proof-net con conclusione Γ.

(2) Per ogni $\Pi \in$ X e per ogni etichettatura f di Π tale che $f(\Pi)$ è un proof-net di conclusione Γ allora esiste una derivazione $\mathcal{D} = \psi(\Pi)$ di $\vdash \Gamma$.

(3) $\psi(\phi(\mathcal{D})) = \mathcal{D}$ e $\phi(\psi(\Pi)) = \Pi$.

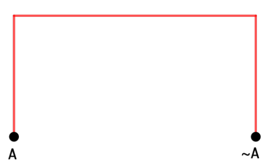
Questo teorema è molto importante in quanto mette in relazione le derivazioni in logica con i grafi; due oggetti che a prima vista possono essere diversi ma che come vedremo si prestano molto bene l'uno a capire meglio l'altro. La prima parte del teorema afferma che tutte le derivazioni in MLL possono essere viste in maniera geometrica; mentre la seconda parte afferma che ciò che è geometrico (come i grafi) può essere ottenuto mediante opportune regole logiche. La terza parte afferma che tale relazione è invertibile, ossia posso passare da proof-net a derivazione e viceversa e la dimostrazione sarà ovvia non appena si saranno dimostrati i punti (1) e (2).

Procediamo ora con la dimostrazione del teorema di sequenzializzazione.

Dimostrazione.

(1) La prima parte si dimostra per induzione sulla lunghezza della derivazione di $\vdash \Gamma$. Occorre dimostrare che comunque prendo una derivazione gli posso associare un grafo che è un proof-net. Siccome ci troviamo in MLL occorre mostrare come posso trasformare in proof-net (con le giuste etichettature) le 4 regole che posso usare.

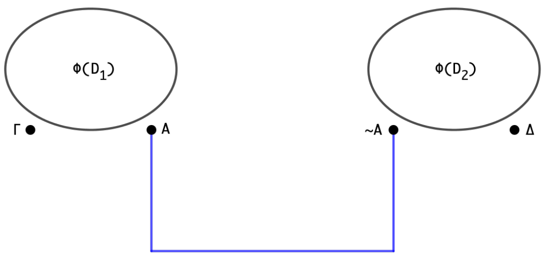
- AX: Se ho $\vdash A, \sim A$ allora il proof-net corrispondente è:



- CUT: Supponiamo di avere la seguente derivazione $\mathcal{D}$:

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots\\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ \vdash \Delta, \sim A\end{array}}{\vdash \Gamma, \Delta}$$

Sia $\mathcal{D}_1$ la derivazione di $\vdash \Gamma, A$ e $\mathcal{D}_2$ la derivazione di $\vdash \Delta, \sim A$ a cui, per ipotesi induttiva, sono associati 2 proof-net rispettivamente $\Pi_1 = \phi(\mathcal{D}_1)$ di conclusioni Γ, A e $\Pi_2 = \phi(\mathcal{D}_2)$ di conclusioni $\Delta, \sim A$ allora ci basta mostrare che il seguente grafo associato $\Pi = \phi(\mathcal{D})$ sia un proof-net.

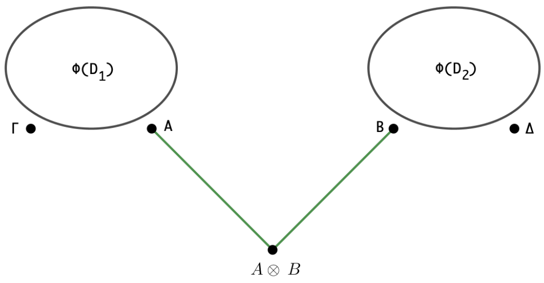


Sicuramente le prime 2 proprietà dei proof-net sono rispettate in quanto per ipotesi $\phi(\mathcal{D}_1)$ e $\phi(\mathcal{D}_2)$ sono proof-nets a cui ho semplicemente aggiunto un arco rappresentante la regola del CUT. Chiaramente anche la terza proprietà viene soddisfatta in quanto sto collegando due grafi che per ipotesi sono aciclici e connessi, dunque anche il grafo risultante è altrettanto.

- $Tensore$: Supponiamo di avere la seguente derivazione $\mathcal{D}$:

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots\\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ \vdash \Delta, B\end{array}}{\vdash \Gamma, \Delta, A \otimes B}$$

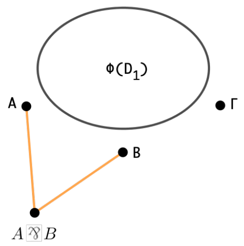
Sia $\mathcal{D}_1$ la derivazione di $\vdash \Gamma, A$ e $\mathcal{D}_2$ è la derivazione di $\vdash \Delta, B$ a cui, per ipotesi induttiva, sono associati 2 proof-net rispettivamente $\Pi_1 = \phi(\mathcal{D}_1)$ di conclusioni Γ, A e $\Pi_2 = \phi(\mathcal{D}_2)$ di conclusioni Δ, B, allora ci basta mostrare che il seguente grafo associato $\Pi = \phi(\mathcal{D})$ sia un proof-net.
In questo caso la discussione è analoga alla precedente.



- *Par* Supponiamo di avere la seguente derivazione $\mathcal{D}$:

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots\\ \vdash \Gamma, A, B\end{array}}{\vdash \Gamma, A \parr B}$$

  Sia $\mathcal{D}_1$ la derivazione di $\vdash \Gamma, A, B$ a cui, per ipotesi induttiva è associato il proof-net $\Pi_1 = \phi(\mathcal{D}_1)$ di conclusione Γ, A, B allora ci basta mostrare che il seguente grafo associato $\Pi = \phi(\mathcal{D})$ sia un proof-net.



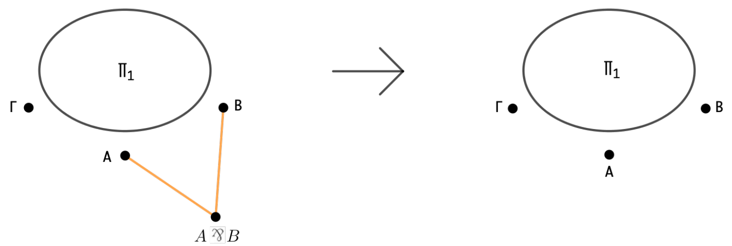
  Anche in questo caso la discussione è analoga alla precedente con l'accortezza che ciò si verifica non appena rimuovo indifferentemente uno ed uno solo fra i due archi del legame finale.

(2) La dimostrazione della seconda parte del teorema è più complessa, infatti quando consideriamo un proof-net è problematico associargli la derivazione corrispondente; questo è dovuto principalmente alla difficoltà di capire quale sia l'ultima regola utilizzata nella derivazione di cui noi possediamo il proof-net corrispondente. Guardando un proof-net non si riesce a stabilire l'ordine cronologico nel quale sono state utilizzate le regole. Per ovviare a questa problematica procederemo allora per induzione sul numero di legami presenti nel proof-net Π. In questo frangente dobbiamo dunque distinguere diversi casi:

- Π possiede un solo legame: in questo caso è facile procedere, infatti, come abbiamo visto nell'osservazione 2.2.2, l'unico proof-net che conosciamo con un solo legame è il legame AX. Dunque al proof-net Π possiamo facilmente associargli la derivazione $\mathcal{D} \vdash A, \sim A$.

- Π possiede $k+1$ legami: supponiamo per ipotesi induttiva di aver dimostrato il teorema per un proof-net con k legami e dimostriamolo per un proof-net con $k+1$ legami. Per prima cosa cominciamo a guardare i legami terminali, per l'osservazione 2.2.6 possiamo identificare i legami CUT e $\otimes$ come un unico legame, per cui consideriamo solo il legame $\otimes$. A questo punto dobbiamo studiare ulteriori due casi:

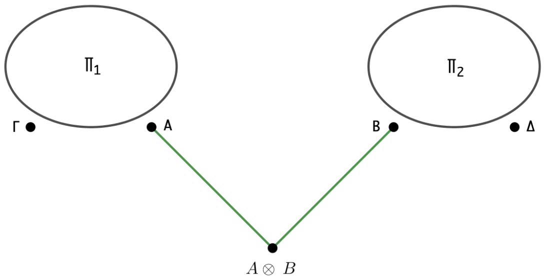
  - Fra tutti i legami terminali di Π è presente almeno un legame $⅋$ fra tutti i legami terminali di Π: in questo caso possiamo facilmente capire che l'ultima regola usata è il $⅋$ ed è come non averla. Questo perchè dimostrare $\vdash \Gamma, A, B$ è equivalente ad aver dimostrato $\vdash \Gamma, A ⅋ B$. Dunque possiamo considerare il grafo ottenuto da Π eliminando tale legame e questo nuovo grafo lo denotiamo Π_1.



Tale nuovo grafo ottenuto è sicuramente un proof-net perché per ipotesi Π lo è ed ho solo eliminato un legame $⅋$ terminale. A questo punto scatta l'ipotesi induttiva in quanto Π ha k legami e dunque posso associargli la derivazione $\mathcal{D}_1 =\vdash \Gamma, A, B$. Ma a questo punto conosco anche la derivazione corrispondente a Π infatti mi basta fare:

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots \\ \vdash \Gamma, A, B\end{array}}{\vdash \Gamma, A ⅋ B}$$

- Fra tutti i legami terminali di Π non è presente alcun legame $\parr$: questa è la parte più complicata di tutto il teorema. Sicuramente i legami terminali non possono essere tutti assiomi, altrimenti il grafo sarebbe sconnesso e non avremmo dunque un proof-net. Dunque non essendo tutti legami terminali assiomi e non essendoci neanche un legame $\parr$ necessariamente è presente almeno un legame terminale $\otimes$ (o CUT tanto sono la medesima cosa in questo frangente). A questo punto applico il lemma fondamentale che afferma che se si possiede un proof-net senza legami terminali $\parr$ allora esiste almeno un $\otimes$ splitting. Siccome il tensore splitting ha la proprietà di sconnettere il proof-net in 2 componenti che sono ancora proof-net, questo ci garantisce che tale tensore è l'ultima regola usata nella derivazione che cerchiamo. A questo punto



definendo Π_1 di conclusioni $\vdash \Gamma, A$ e Π_2 di conclusioni $\vdash \Delta, B$ rispettivamente le due componenti di Π ottenute eliminando il tensore splitting per ipotesi induttiva, possiamo associare rispettivamente le derivazioni $\mathcal{D}_\infty = \vdash \Gamma, A$ $\mathcal{D}_2 = \vdash B, \Delta$ a Π_2. Ora è facile capire che la derivazione $\mathcal{D}$ associata a Π è:

$$
\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots\\ \vdash \Gamma, A\end{array} \qquad\qquad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ \vdash \Delta, B\end{array}}{\vdash \Gamma, \Delta, A \otimes B}
$$

Bibliografia

[Fal14] Tortora de Falco L. Abrusci V.M. *Logica - Volume 1, Dimostrazioni e modelli al primo ordine*. Springer, 2014.

[Fal18] Tortora de Falco L. Abrusci V.M. *Logica - Volume 2, Incompletezza, teoria assiomatica degli insiemi*. Springer, 2018.

[VM18] Abrusci V.M. *Tipi e Logica Lineare, Proof-net*. (Dispense), 23 aprile 2018.