Un breve corso di logica classica

Silvio Valentini
Dipartimento di Matematica Pura ed Applicata
Università di Padova
via G. Belzoni n.7, I–35131 Padova, Italy
silvio@math.unipd.it

February 6, 2003

Contents

1 Semantica e sintassi

Uno degli scopi della logica é quello di vedere quali proprietà delle strutture matematiche possono essere descritte usando un linguaggio formale. Vediamo quindi di precisare un concetto di struttura che sia al tempo stesso sufficientemente generale e vicino alle usuali costruzioni matematiche ma anche non troppo complicato in modo che sia possibile utilizzare un linguaggio formale abbastanza semplice. Ricordiamo in ogni caso che i risultati che illustreremo dipendono in modo essenziale dalle caratteristiche del linguaggio che utilizzeremo.

Definizione 1.1 (Struttura) *Una* struttura $\mathcal{U} \equiv (U, \mathcal{R}, \mathcal{F}, \mathcal{C})$ *é costituita dalle seguenti parti:*

- *Un insieme non vuoto U detto l'*universo *della struttura;*
- *Un insieme non vuoto $\mathcal{R}$ di relazioni tra elementi di U;*
- *Un insieme, eventualmente vuoto, $\mathcal{F}$ di* operazioni*, vale a dire funzioni totali, su elementi di U e ad immagine in U;*
- *Un sottoinsieme, eventualmente vuoto, $\mathcal{C}$ di U di* costanti*, vale a dire elementi particolari e designati di U.*

Ad esempio, un matematico che voglia parlare di proprietà dei numeri naturali potrebbe essere interessato a lavorare nella struttura

$$\mathcal{N} \equiv (\omega, \{=, \geq\}, \{+, \times\}, \{0, 1\})$$

che ha l'insieme dei numeri naturali per universo e analizza le relazioni di uguaglianza e d'ordine tra numeri naturali, le operazioni di somma e prodotto tra numeri naturali e utilizza le sole costanti 0 e 1.

Supponendo di avere una struttura di cui vogliamo parlare, il problema che dobbiamo affrontare é quello di determinare un linguaggio per esprimere le affermazioni che su tale struttura vogliamo fare; ad esempio, il nostro matematico vorrà probabilmente fare affermazioni quali "$3+2 = 5$" o "se $x \geq 3$ allora $2 \times x \geq 6$" o ancora "*la somma é associativa*".

Il primo requisito che il linguaggio che stiamo cercando di definire deve soddisfare é allora che ci siano abbastanza simboli per denotare tutti gli oggetti che appaiono nella struttura che é chiamato a descrivere. Continuando con il nostro esempio pretenderemo che nel linguaggio per parlare della struttura $\mathcal{N}$ compaiano il simbolo Eq per denotare l'uguaglianza, il simbolo Gt per denotare la relazione d'ordine, il simbolo sum per denotare l'operazione di somma, il simbolo prod per denotare l'operazione di prodotto, il simbolo zero per denotare il numero naturale 0 ed infine il simbolo uno per denotare il numero naturale 1. Si noti che, per evitare di fare confusione tra gli elementi del linguaggio che stiamo definendo e gli oggetti della struttura che tali elementi devono denotare, siamo stati estremamante attenti a non usare nel linguaggi i simboli usuali. Naturalmente nella pratica matematica non si segue questa via e quindi é facile fare confusione.

Quanto abbiamo fatto non é sufficiente per esprimere generiche proprietà di una struttura, proprietà che valgano cioè per ogni elemento della struttura. Quel che ci serve é la possibilità di indicare un generico elemento della struttura, vale a dire che abbiamo bisogno delle *variabili.* Introduciamo quindi nel linguaggio un insieme numerabile (é bene averne sempre di riserva!) di simboli per variabili $v_0, v_1, ...$ oltre ad un simbolo predicativo P_i^n, di arietà n, per ogni relazione $R_i \in \mathcal{R}$ ad n posti, un simbolo funzionale f_j^n, di arietà n, per ogni operazione $F_j^n \in \mathcal{F}$ ad n argomenti, ed un simbolo c_k per ogni costatnte designata appartenente al sottoinsieme $\mathcal{C}$ delle costanti. In questo modo ci sarà nel linguaggio un simbolo per denotare ogni relazione, ogni operazione ed ogni costante designata della struttura che ci interessa.

Usando i simboli finora introdotti si possono costruire induttivamente delle espressioni, che chiameremo *termini*, nel modo che segue.

Definizione 1.2 (Termine) *Le seguenti espressioni sono* termini

- *v, se v é una variabile;*
- *c, se c é un simbolo per una costante;*
- *$f_j(t_1, ..., t_n)$, se t_1, ..., t_n sono termini e f_j é un simbolo per una funzione di arietà n.*

Niente altro é un termine.

Possiamo considerare i termini, almeno quelli che non contengono variabili, come i nomi che si possono usare nel linguaggio per denotare elementi dell'universo della struttura. Infatti

- un simbolo per costante é il nome nel linguaggio della costante che denota;
- un termine complesso, come $f_j(t_1, ..., t_n)$, é il nome nel linguaggio dell'elemento dell'universo della struttura i cui valore é il risultato della funzione denotata da f_j quando applicata agli elementi i cui nomi sono t_1,...,t_n.

Nel caso in cui un termine contenga delle variabili non possiamo più considerarlo come un nome per un preciso elemento del universo della struttura, ma dobbiamo piuttosto considerarlo come un nome per un generico elemento del universo della struttura che può cambiare, al variare delle possibili interpretazioni delle variabili che il termine stesso contiene.

Tornando al nostro esempio con la struttura $\mathcal{N}$ possiamo ad esempio considerare il termine $\mathsf{sum}(\mathsf{uno}, \mathsf{uno})$ come un nome nel linguaggio per il numero due, mentre il termine $\mathsf{sum}(\mathsf{uno}, v)$ denoterà il nome del numero naturale successivo al numero in cui v é stato interpretato. Vale la pena di notare che é ben possibile che lo stesso elemento della struttura abbia più nomi nel linguaggio. Ad esempio uno and $\mathsf{sum}(\mathsf{uno}, \mathsf{zero})$ sono due nomi diversi per il numero naturale 1. Tuttavia utilizzando solamente i termini non possiamo esprimere affermazioni, e in particolare non possamo dire quando due termini sono due nomi diversi per lo stesso oggetto dell'universo. Questo é il motivo per cui abbiamo introdotto nel linguaggio anche i simboli predicativi. Utilizzandoli possiamo infatti costruire delle nuove espressioni, che chiameremo *formule*, che denoteranno affermazioni sulla struttura. Possiamo allora comiciare dal caso più semplice.

Definizione 1.3 (Formula atomica) *Le seguenti espressioni sono formule atomiche:*

- $P_i(t_1, ..., t_n)$, *se* t_1,...,t_n *sono termini e* P_i *é un simbolo predicativo di arietà* n.

Se abbiamo avuto cura di introdurre un nome nel linguaggio per la relazione di uguaglianza tra gli elementi della struttura, possiamo, utilizzando una formula atomica, esprimere il fatto che due nomi distinti denotano lo stesso elemento dell'universo della struttura. Continuando con il nostro esempio sulla descrizione della struttura $\mathcal{N}$, abbiamo che la formula atomica $\mathsf{Eq}(\mathsf{uno}, \mathsf{sum}(\mathsf{uno}, \mathsf{zero}))$ può essere utilizzata proprio a questo scopo.

Naturalmente le formule atomiche non esauriscono il tipo di affermazioni che un matematico può desiderare di fare sulla struttura: egli sarà sicuramente interessato a combinare tra loro le formule atomiche per affermare, ad esempio, che una certa relazione tra i numeri naturali é una conseguenza di un'altra. Già con una analisi superficiale delle affermazione cui un matematico può essere interessato si vede che risulteranno indispensabili le *congiunzioni* per espremire affermazioni come "é vero sia A che B" e le *disgiunzioni* per esprimere la verità di fasi come "o é vero A o é vero B". Per il nostro matematico avranno inoltre un ruolo determinante le *implicazioni* per esprimere affermazioni come "se é vero A allora é vero B", ma anche le *negazioni* come "non é vero A". Ma ci saranno anche altre affermazioni che risulteranno particolarmente importanti. Tra queste spiccano sicuramente le affermazioni universali come "A vale per ogni elemento dell'universo della struttura" e le affermazioni esistenziali come "esiste almeno un elemeno della struttura per cui A vale". Naturalmente é possibilissimo che ci siano anche altre affermazioni che potrebbero essere utili, come ad esempio "é quasi sempre vero che vale A", ma per il momento noi ci fermiamo alle precedenti visto che esse sono già sufficientemente espressive per esprimere fatti rilevanti su una struttura.

Un aspetto particolarmente interessante delle operazioni linguistiche che abbiamo messo in evidenza sta nel fatto che esse in realtà non dipendono dalla particolare struttura che stiamo studiando ma hanno un carattere più generale di natura puramente formale, vale a dire legata solo alla forma delle espressioni coinvolte. Ad esempio non é difficile riconoscere la correttezza della seguente argomentazione

1. Se ogni pricchio é uno slorpio
2. Mirpo é un pricchio
3. quindi Mirpo é uno slorpio

anche senza avere la più pallida idea del significato delle parole "pricchio" e "slorpio". Esiste cioè una parte delle affermazioni ottenute per composizione tra affermazioni elementari che non dipende dalla particolare struttura che si sta analizzando, ma ha un carattere più generale ed astratto. Questa é la parte che siamo interessati ad analizzare in quetso corso.

Supponiamo allora di arrricchire il nostro linguaggio con un simbolo per ogni connettivo che il nostro ipotetico matematico può essere ineteressato ad usare: $\wedge$, di arietà $0 \to (0 \to 0)$, sarà utilizzato per la congiunzione, $\vee$, di nuovo di arietà $0 \to (0 \to 0)$, per la disgiunzione, $\to$, anche esso di arietà $0 \to (0 \to 0)$, per la implicazione e infine $\neg$, di arietà $0 \to 0$ per la negazione. L'arietà é naturalmente una guida immediata per l'utilizzo di questi simboli per connettivi; ad esempio la congiunzione si aspetta due espressioni per fornire una nuova espressione, mentre la negazione si aspetta una sola espressione.

Aggiungeremo inoltre un simbolo per ogni quantificatore: $\forall$, di arietà $(0 \to 0) \to 0$ per la quantificazione universale e $\exists$, della stessa arietà, per quella esistenziale. Notate che questi simboli si aspettano quindi una espressione funzionale per fornire una nuova espressione. Questo é naturalmente in accordo che il significato che intendiamo assegnare loro.

Dobbiamo ora dare una definizione induttiva che ci permetta di costruire formule più complicate a partire dalle formule atomiche. Per farlo ci lascieremo naturalmente guidare dal significato che intendiamo attribuire ai simboli logici che abbiamo introdotto.

Definizione 1.4 (Formula) *Le seguenti espressioni sono formule:*

- *ogni formula atomica;*
- $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \to \beta$, $\not\alpha$, *se* α *e* β *sono formule;*
- $(\forall v_i)\alpha$, $(\exists v_i)\alpha$, *se* α *é una formula e* v_i *é una variabile*

Niente altro é una formula.

1.1 Interpretazione delle formule

Abbiamo già fornito una interpretazione intuitiva dei vari connettivi e quantificatori; tuttavia se vogliamo poter usufruire dei vari strumenti che la matematica ci mette a disposizione nella nostra analisi delle possibilità espressive del linguaggio, dobbiamo ora fornire una *semantica matematica*, vale a dire che dobbiamo fissare una volta per tutte il significato dei connettivi e dei quantificatori. Per fare questo dobbiamo sostanzialmente dire in che modo i connetivi e i quantificatori *trasportano* la verità delle affermazioni dalla verità di affermazioni semplici a quella di affermazioni composte. Lo faremo utilizzando, fin dove possibile, quanto stabilito dall'uso nel linguaggio naturale e quando questo non fosse possibile accetteremo l'uso che dei connettivi e dei quantificatori si fa in matematica.

Il primo passo consiste naturalmente nel vedere come attribuire un valore di verità alle formule atomiche e, prima ancora, come interpretare i termini. Appena si cerca di formalizzare le spiegazioni intuitive che abbiamo utilizzato finora ci si imbatte in un grave problema: come interpretare termini e formule che contengono variabili libere? Ad esempio, tornando al caso della struttura $\mathcal{N}$ consideriamo la formula atomica $\mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{uno}, \mathsf{uno}))$ in cui compare la variabile v_1. Ora, si può decidere subito di interpretare il simbolo predicativo Gt nella relazione $\geq$, il simbolo funzionale sum nella funzione $+$, il simbolo di costante uno nel numero naturale 1 visto che abbiamo introdotto apposta nel linguaggio tutti questi simboli. Tuttavia é chiaro che non possiamo ancora sapere come interpretare la formula $\mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{uno}, \mathsf{uno}))$ finchè non ci decidiamo ad interpretare la variabile v_1 in un qualche numero naturale. Infatti se v_1 é chiaro che se v_1 viene interpretata nel numero naturale 0 o 1 abbiamo a che fare con una formula falsa, mentre in tutti gli altri casi otteniamo una formula vera. D'altra parte il termine v_1 deve pure essere interpretato da qualche parte visto che le strutture non sanno nulla di variabili!

Una volta chiarito dove sta il problema, diamo la definizione formale di interpretazione delle formule atomiche di un linguaggio $\mathcal{L}$ in una struttura $\mathcal{U}$.

Definizione 1.5 (Interpretazione) *Una* interpretazione *I delle formule del linguaggio $\mathcal{L}$ nella struttura $\mathcal{U} \equiv (U, \mathcal{R}, \mathcal{F}, \mathcal{C})$ per $\mathcal{L}$ consiste di*

- *una mappa che assegna ad ogni simbolo di costante c di $\mathcal{L}$ un elemento c_I di U tra quelli in $\mathcal{C}$;*
- *una mappa che assegna ad ogni simbolo funzionale f di $\mathcal{L}$ una operazione f_I su U tra quelle di $\mathcal{F}$;*
- *una mappa che assegna ad ogni simbolo predicativo P di $\mathcal{L}$ una relazione P_I tra elementi di U tra quelle di $\mathcal{R}$;*
- *una valutazione σ, cioè una mappa che assegna ad ogni variabile v_i di $\mathcal{L}$ un elemento di U.*

É allora chiaro che la generalità delle formule, espressa nel linguaggio dalla presenza delle variabili e non esistente nelle strutture viene resa dalle possibili scelte delle mappe di intrepretazione σ.

Data una interpretazione del linguaggio $\mathcal{L}$ nella struttura $\mathcal{U}$ possiamo facilmente decidere quali formule di $\mathcal{L}$ saranno interpretate in vero, vale a dire, saranno vere in $\mathcal{U}$ con l'interpretazione I, e quali saranno interpretate in falso, i.e., saranno false in $\mathcal{U}$ con l'interpretazione I. Faremo inoltre in modo che la nostra interpretazione decida su ogni formula, vale a dire che ogni formula sarà intrepretata o in vero o in falso e non ci saranno formule senza interpretazione.

Da punto di vista pratico la dafinizione della funzione di interpretazione si svilupperà in due passi: prima vedremo come interpretare i termini in elementi della struttura e poi come interpretare le formule in vero o falso.

Definizione 1.6 (Interpretazione dei termini) *Sia t un termine del linguaggio $\mathcal{L}$. Allora l'interpretazione di t secondo I, basata sulla valutazione σ (notazione t^σ o $\sigma(t)$) si definisce induttivamente come segue:*

- *se t é la variabile x allora t^σ é $\sigma(x)$*
- *se t é la costante c allora t^σ é c_I*
- *se t é il termine $f(t_1, ..., t_n)$ allora t^σ é $f_I(t_1^\sigma, ..., t_n^\sigma)$*

Poichè i termini sono definiti induttivamente secondo lo stesso schema é chiaro che con questa definzione ogni termine viene interpretato in un elemento dell'universo della struttura.

Possiamo passare allora all'interpretazione delle formule. Abbiamo tuttavia bisogno di introdurre prima un po' di notazioni che utilizzaremo nel seguito.

- indicheremo con $\sigma(x/u)$ l'interpretazione che coincide completamente con l'interpretazione I, basata su σ per quanto riguarda l'interpretazione dei simboli predicativi, dei simboli funzionali, dei simboli di costante e sulla valutazione di tutte le variabili eccetto la variabile x che non viene più interpretata secondo la valutazione σ ma invece nell'elemento $u \in U$.
- abbrevieremo inoltre vero con $\top$ e falso con $\bot$.

Definizione 1.7 (Semantica delle formule) *Sia α una formula del linguaggio $\mathcal{L}$. Allora l'interpretazione di α secondo I, basata sulla valutazione σ delle variabili, (notazione α^σ) é definita induttivamente come segue:*

$$
\begin{array}{lll}
se\ \alpha \equiv P(t_1, ..., t_n) & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \langle t_1^\sigma, ..., t_n^\sigma \rangle \in P_I \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv \mathsf{Eq}(s, t) & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ s^\sigma = t^\sigma \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv \neg\beta & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \beta^\sigma = \bot \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv \beta \wedge \gamma & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \beta^\sigma = \top\ e\ \gamma^\sigma = \top \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv \beta \vee \gamma & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \beta^\sigma = \top\ o\ \gamma^\sigma = \top \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv \beta \to \gamma & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \beta^\sigma = \bot\ o\ \gamma^\sigma = \top \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv (\forall v_i)\ \beta & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \beta^{\sigma(x/u)} = \top\ per\ ogni\ u \in U \\ \bot & altrimenti \end{cases} \\
se\ \alpha \equiv (\exists v_i)\ \beta & allora & \alpha^\sigma \equiv \begin{cases} \top & se\ \beta^{\sigma(x/u)} = \top\ per\ qualche\ u \in U \\ \bot & altrimenti \end{cases}
\end{array}
$$

Anche in questo caso, visto che le formule sono indittivamente definite secondo lo stesso schema é chiaro che con la precedente definizione ad ogni formula verrà assegnato un valore di verità.

Possiamo ora tornare alla nostra struttura $\mathcal{N}$ con l'interpretazione naturale I, basata sulla valutazione σ, per vedere come interpretare alla luce di questa definizione la formula $(\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero}))$. Allora, ricordiamo che $\mathsf{Gt}_I \equiv \geq$, $\mathsf{sum}_I \equiv \lambda x.\lambda y.\ x + y$, $\mathsf{prod}_I \equiv \lambda x.\lambda y.\ x \times y$, $\mathsf{zero}_I \equiv 0$ e $\mathsf{uno}_I \equiv 1$.

Otteniamo quindi

$$
\begin{array}{lll}
((\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero})))^\sigma = \top & \text{sse} & (\mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero})))^{\sigma(v_1/u)} = \top \\
& & \text{per ogni } u \in \omega \\
& \text{sse} & \langle v_1^{\sigma(v_1/u)}, \mathsf{sum}(\mathsf{zero}, \mathsf{zero})^{\sigma(v_1/u)} \rangle \in \mathsf{Gt}_I \\
& & \text{per ogni } u \in \omega \\
& \text{sse} & v_1^{\sigma(v_1/u)} \geq \mathsf{sum}_I(\mathsf{zero}^{\sigma(v_1/u)}, \mathsf{zero}^{\sigma(v_1/u)}) \\
& & \text{per ogni } u \in \omega \\
& \text{sse} & u \geq 0 + 0 \\
& & \text{per ogni } u \in \omega
\end{array}
$$

L'ultima condizione é chiaramente vera nella struttura che stiamo considerando, vale a dire che tutti i numeri naturali sono maggiori o uguali a 0, e quindi la formula $(\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero}))$ sarà valutata in $\top$.

É interessante notare che l'interpretazione non dipende in alcun modo dalla valutazione σ delle variabili. In realtà, guardando con attenzione la definizione di valutazione di una formula ci si rende conto senza troppe difficoltà che la valutazione di una formula dipende solo dalla valutazione delle variabili che vi occorrono libere.

É importante sottolineare il fatto che mentre una struttura determina il linguaggio formale da usare per parlarne, una volta fissato tale linguaggio i simboli che in esso campaiono si possono interpretare anche in strutture del tutto differenti. Si consideri ad esempio la struttura $\mathcal{U} \equiv (\text{Esseri umani}, \{=, \text{genitore di}\}, \{\lambda x.\lambda y.\ x, \lambda x.\lambda y.\ y\}, \{\text{Enrico}, \text{Silvio}\})$.

Il linguaggio $\mathcal{L}$ che abbiamo definito per parlare della struttura $\mathcal{N}$ si interpretare anche in $\mathcal{U}$; nel seguito diremo che $\mathcal{U}$ e $\mathcal{N}$ sono strutture dello stesso *tipo* perchè hanno lo stesso numero di relazioni di corrispondente arietà, lo stesso numero di operazioni di corrispondente ariuetà e lo stesso numero di costanti designate; $\mathcal{L}$ si potrà quindi interpretare in ogni struttura dello stesso tipo di $\mathcal{U}$ e $\mathcal{N}$. Supponiamo ora di voler interpretare la stessa formula $(\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero}))$ che abbiamo considerato prima ma di utilizzare questa volta la seguente interpretazione: $\mathsf{Gt}_I \equiv$ genitore di, $\mathsf{sum}_I \equiv \lambda x.\lambda y.\ x$, $\mathsf{prod}_I \equiv \lambda x.\lambda y.\ y$, $\mathsf{zero}_I \equiv$ Enrico e $\mathsf{uno}_I \equiv$ Silvio.

Otteniamo quindi

$$
\begin{array}{lll}
((\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero})))^\sigma = \top & \text{sse} & (\mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero})))^{\sigma(v_1/u)} = \top \\
& & \text{per ogni essere umano } u \\
& \text{sse} & \langle v_1^{\sigma(v_1/u)}, \mathsf{sum}(\mathsf{zero}, \mathsf{zero})^{\sigma(v_1/u)} \rangle \in \mathsf{Gt}_I \\
& & \text{per ogni essere umano } u \\
& \text{sse} & v_1^{\sigma(v_1/u)} \text{ é genitore di } \mathsf{sum}_I(\text{Enrico}, \text{Silvio}) \\
& & \text{per ogni essere umano } u \\
& \text{sse} & u \text{ é genitore di Enrico} \\
& & \text{per ogni essere umano } u
\end{array}
$$

Questa volta l'ultima affermazione é falsa (ci sarà pure un essere umano che non é genitore di Enrico!) e quindi la formula $(\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero}))$ in questa struttura e con questa interpretazione sarà valutata in $\perp$.

Abbiamo quindi visto che la stessa formula può essere interpretata in modi diversi in strutture diverse. In un certo senso la formula $(\forall v_1)\ \mathsf{Gt}(v_1, \mathsf{sum}(\mathsf{zero}, \mathsf{zero}))$ permette di discriminare tra la struttura $\mathcal{U}$ e la struttura $\mathcal{N}$, anche se queste due strutture sono dello stesso tipo. Questa osservazione é di estremo interesse in quanto suggerisce un possibile uso del linguaggio per descrivere le strutture: un insieme di formule Φ determina le strutture che rendono vere tutte le sue formule per ogni interpretazione delle variabili. A questo punto diventa interessante scoprire, tramite strumenti puramente sintattici, che considerano cioè solamente la manipolazione simbolica, quali sono le formule α che in tali strutture sono vere, la cui verità é cioè conseguenza di quella delle formule in Φ.

Naturalmente, cosí come esistono formule la cui valutazione dipende dall'interpretazione, ce ne sono alcune, ad esempio $(\forall v_1)\ \mathsf{Eq}(v_1, v_1)$, la cui valutazione é indipendente dall'interpretazione considerata. É chiaro che sono anche queste un interessante oggetto di studio. In realtà le due questioni sono strettamente correlate: infatti le formule vere in ogni interpretazione saranno quelle che sono conseguenza di un insieme vuoto di formule, perchè evidentemente tutte le formule nell'insieme vuoto di formule sono soddisfatte in ogni struttura. Vedremo inoltre che, se l'insieme Φ di formule é finito allora il problema della conseguenza logica si può ridurre al problema della verità in ogni struttura.

Allo scopo di lavorare un po' di meno nel seguito conviene osservare subito che la semantica che abbiamo imposto sulle formule rende superfluo l'uso di tutti i connettivi e di tutti i quantificatori. In realtà possiamo scrivere formule che hanno per ogni interpretazione lo stesso valore di verità di quelle esprimibili con il linguaggio fin qui proposto anche se ci limitiamo ad usare un linguaggio che contenga i soli connettivi $\neg$ e $\rightarrow$ e il solo quantificatore $\forall$. Infatti possiamo definire, matenedone il significato, i connettivi e il quantificatore mancante nel modo che segue:

$$\begin{array}{lcl} \beta \wedge \gamma & \equiv & \neg(\beta \rightarrow \neg\gamma) \\ \beta \vee \gamma & \equiv & \neg\beta \rightarrow \gamma \\ (\exists v_i)\ \alpha & \equiv & \neg(\forall v_i)\ \neg\alpha \end{array}$$

Per questo motivo d'ora in avanti nelle dimostrazioni svilupperemo solamente i passi relativi a tali connettivi e al quantificatore universale.

1.1.1 Esercizi

1. Siano β e γ due formule. Dimostrare che per ogni interpretazione I in ogni struttura le seguenti coppie di formule hanno la stessa interpretazione

   (a) $\beta \wedge \gamma$ e $\neg(\beta \rightarrow \neg\gamma)$

   (b) $\beta \vee \gamma$ e $\neg\beta \rightarrow \gamma$

   (c) $(\exists v_i)\ \alpha$ e $\neg(\forall v_i)\ \neg\alpha$

2 La logica proposizionale

Cminceremo lo studio dei problemi che abbiamo introdotto alla fine del paragrafo precedente limitandoci ad analizzare un caso più semplice che se considerassimo la situazione generale.

2.1 Formule proposizionali e tautologie

In questa parte del corso ci limiteremo ad analizzare le formula costruite con i soli connettivi e ci fermeremo non appena arriveremo ad analizzare una formula quantificata.

Definizione 2.1 (Formula prima) *La formula α si dice* formula prima *se α é una formula atomica o é una formula quantificata universalmente.*

La parte della logica che si interessa delle formule che si possono costruire a partire dalle formule prime tramite l'uso dei soli connettivi si dice *logica proposizionale*. Per il momento ci limiteremo quindi studiare la logica proposizionale. Vale la pena di notare immediatamente che il limitarsi allo studio della logica proposizionale ci impedirà di renderci conto che alcune formule, come ad esempio $(\forall v_i)\ \mathsf{Eq}(v_i, v_i)$, sono vere in ogni struttura.

Definizione 2.2 (Formula proposizionale) *Le seguenti formule sono* formule proposizionali

- *una formula prima é una formula proposizionale;*
- *se α e β sono formule proposizionali allora $\alpha \rightarrow \beta$ e $\neg\alpha$ sono formule proposizionali.*

Niente altro é una formula proposizionale.

Il motivo per cui ci siamo limitati alle formule proposizionali é che in questo caso la semantica di una formula può essere molto semplificata: non si tratta più di vedere come interpretare una formula in ogni struttura del tipo adatto al linguaggio in esame ma solamente di scoprire come una data valutazione di verità sulle formule prime, che non andiamo più ad analizzare direttamente, si trasporta ad una qualsiasi formula proposizionale. Se rileggiamo allora la definizione di semantica di una formula limitandoci al caso delle sole formule proposizionali otteniamo la definizione che ci interessa.

Definizione 2.3 (Valutazione proposizionale) *Una valutazione proposizionale σ é una assegnazione di verità a tutte le formule proposizionali α tali che:*

$$\begin{array}{lll} \textit{se } \alpha \equiv \neg\beta & \textit{allora} & \alpha^\sigma \equiv \begin{cases} \top & \textit{se } \beta^\sigma = \bot \\ \bot & \textit{altrimenti} \end{cases} \\ \textit{se } \alpha \equiv \beta \rightarrow \gamma & \textit{allora} & \alpha^\sigma \equiv \begin{cases} \top & \textit{se } \beta^\sigma = \bot \textit{ o } \gamma^\sigma = \top \\ \bot & \textit{altrimenti} \end{cases} \end{array}$$

Fissata una valutazione proposizionale sulle formule prime le condizioni poste fissano una valutazione di verità su ogni formula proposizionale.

Si noti ora che se una formula risulta vera per ogni interpretazione proposizionale essa risulta ovviamente vera anche per ogni interpretazione in una qualsiasi struttura visto che c'é una diversa valutazione proposizionale per ogni assegnazione di verità alle formule prime, e quindi che le valutazioni proposizionale comprendono sicuramente ogni possibile assegnazione di verità in una interpretazione generale; inoltre le valutazioni proposizionali si comportano esattamente come le interpretazioni generali per quanto riguarda i connettivi. Come abbiamo già osservato, non vale invece il viceversa; ad esempio la formula prima $(\forall v_i)\ \mathsf{Eq}(v_i, v_i)$ può venire valutata da una valutazione proposizionale sia in vero che in falso, in quanto si tratta di una formula prima, mentre una interpretazione generale la valuta comunque in vero.

Definizione 2.4 (Tautologia) *Una formula valutata in vero da ogni valutazione proposizionale si dice* tautologia*.*

Scoprire le tautologie é quindi, in prima approssimazione, una risposta al nostro problema di scoprire le formule vere per ogni interpretazione: é vero che le formule vere in ogni interpretazione saranno di più che le sole tautologia, ma queste costituiscono comunque una buona base di partenza!

Il problema di scoprire se una data formula α é una tautologia viene semplificato se ricordiamo che la valutazione proposizionale di una formula dipende solo dalle formule prime che in essa compaiono; quindi per scoprire se α é una tautologia non ci serve provare a vedere se α viene valutata in vero per ognuna delle infinite valutazioni proposizionali ma ci basta limitarci a considerare quelle che differiscono sulle formule prime che appaiono in α. Ora un facile calcolo dimostra che se in α appaiono n formule prime allora abbiamo bisogno di analizzare solo 2^n diverse valutazioni proposizionali che differiscono sulla valutazione delle formule prime che appaiono in α. Per essere sicuri di non dimenticarsene nessuna vale lapena di organizzare il lavoro in quelle che si chiamano *tabelline di verità*: queste non sono altro che una diversa presentazione della semantica delle formule proposizionali.

Lo scopo é di fornire la valutazione di una formula proposizionale α a partire dalle sue componenti prime e le tabelline di verità insegnano come ottenere il valore di verità delle varie sottoformule di α in funzione del valore di verità delle loro componenti.

Definizione 2.5 (Tabelline di verità) *Siano β eγ due formule proposizionali, allora i valori di verità di $\neg\beta$, $\beta \wedge \gamma$, $\beta \vee \gamma$, $\beta \to \gamma$ nella valutazione σ sono funzioni dei valori di verità di β e γ nella stessa valutazione proposizionale σ secondo la seguente tabella*

$\sigma(\beta)$	$\sigma(\gamma)$	$\sigma(\neg\beta)$	$\sigma(\beta \wedge \gamma)$	$\sigma(\beta \vee \gamma)$	$\sigma(\beta \to \gamma)$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$

Ad esempio, trascurando di scrivere ogni volta che si tratta dell'applicazione della funzione di valutazione σ alla formula, per vedere se la formula $\alpha \equiv (\forall v_1)\ \neg\mathsf{Eq}(v_1, v_1) \rightarrow \neg(\exists v_2)(\forall v_3)\ \mathsf{Eq}(v_2, v_3)$ é una tautologia possiamo usare una tabellina di verità ed ottenere

$(\forall v_1)\ \neg\mathsf{Eq}(v_1, v_1)$	$(\exists v_2)(\forall v_3)\ \mathsf{Eq}(v_2, v_3)$	$\neg(\exists v_2)(\forall v_3)\ \mathsf{Eq}(v_2, v_3)$	α
$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$

La formula α non risulta essere quindi una tautologia: ogni valutazione σ che valuti sia $(\forall v_1)\ \neg\mathsf{Eq}(v_1, v_1)$ che $(\exists v_2)(\forall v_3)\ \mathsf{Eq}(v_2, v_3)$ in vero rende infatti falsa la formula α; vale la pena di notare che, mentre ci sono valutazioni proposizionali che si comportano così, non ci sono interpretazioni siffatte.

Anche il problema della conseguenza di una formula da un insieme di formule può avere una chiara definizione semantica.

Definizione 2.6 (Conseguenza tautologica) *Sia Φ un insieme di formule proposizionali ed α una formula proposizionale. Allora α si dirà essere una* conseguenza tautologica *di Φ (notazione $\Phi \vDash \alpha$) se ogni valutazione che rende vere tutte le formule di Φ rende vera anche α.*

Come abbiamo già osservato, nel caso l'insieme Φ sia vuoto allora α é una conseguenza tautologica di Φ se e solo se ogni valutazione proPosizionale rende vera α, cioè se e solo se α é una tautologia. Quindi, la nozione di tautologia altro non é che un caso speciale della nozione di conseguenza tautologica; quindi useremo la notazione $\vDash \alpha$ per indicare che α é una tautologia. Inoltre, in virtù di quest'ultima osservazione, d'ora in avanti l'argomento centrale del nostro studio sarà la nozione di conseguenza tautologica invece che quella di tautologia. D'altra parte, se l'insieme Φ é finito, in seguente teorema riduce la nozione di conseguenza tautologica a quella di tautologia e rende quindi possibile utilizzare le tabelline di verità per scoprire se una formula proposizionale é conseguenza tautologica di un insieme finito di formule proposizionali.

Teorema 2.7 (Teorema di deduzione semantico) *Sia Φ un insieme di formule proposizionali e α e β siano due formule proposizionali. Allora*

$$\Phi, \alpha \vDash \beta \textit{ se e solo se } \Phi \vDash \alpha \rightarrow \beta$$

Dimostrazione. Supponiamo che $\Phi, \alpha \vDash \beta$ e sia σ una valutazione che renda vere tutte le formule di Φ. Possono allora verificarsi due casi:

- $\sigma(\alpha) = \top$. In questo caso, l'ipotesi che β sia una conseguenza tautologica di Φ, α porta subito a dire che $\sigma(\beta) = \top$ e quindi, per la definizione di valutazione proposizionale, $\sigma(\alpha \rightarrow \beta) = \top$.
- $\sigma(\alpha) = \bot$. In questo caso la definizione di valutazione proposizionale fornisce immediatamente la conclusione che $\sigma(\alpha \rightarrow \beta) = \top$.

Abbiamo quindi dimostrato che $\Phi \models \alpha \to \beta$.

Supponiamo, d'altro canto, che $\Phi \models \alpha \to \beta$ e consideriamo una qualsiasi valutazione σ che renda vere tutte le formule di Φ e α. Allora, dall'ipotesi che $\alpha \to \beta$ sia una conseguenza tautologica di Φ otteniamo $\sigma(\alpha \to \beta) = \top$ e quindi $\sigma(\alpha) = \top$ ci obbliga a concludere che $\sigma(\beta) = \top$. Abbiamo cioè dimostrato che $\Phi, \alpha \models \beta$.

Questo teorema é particolarmente interessante perchè esso collega due nozioni di carattere semantico (la validità di certe conseguenze tautologiche) sfruttando una nozione sintattica (l'implicazione tra due formule). Esso fa vedere cioè che il segno di implicazione é la controparte sintattica della nozione semantica di conseguenza tautologica.

Adesso é facile vedere che se $\Phi \equiv \{\phi_1, \ldots, \phi_n\}$ é un insieme finito di formule proposizionali allora la formula proposizionale α é conseguenza tautologica di Φ se e solo se $\phi_1 \to (\ldots \to (\phi_n \to \alpha) \ldots)$ é una tautologia.

2.1.1 Esercizi

1. Determinare le formule che permettono di caratterizzare il simbolo predicativo Gt come una relazione d'ordine.

2. Determinare le formule che permettono di caratterizzare il simbolo funzionale sum come la funzione somma $\lambda x.\lambda y.\ x + y$ tra numeri naturali.

3. Determinare le formule che permettano di caratterizzare la struttura $\mathcal{N}$ come una struttura che contiene un numero infinito di elementi.

4. Dimostrare che, fissata una valutazione proposizionale sulle formule prime, le condizioni poste nella definizione di valutazione proposizionale fissano una valutazione di verità su ogni formula proposizionale.

5. Dimostrare che esistono una quantità più che numerabile di valutazioni proposizionali.

6. Dimostrare che se nella formula proposizionale α appaiono n formule prime allora per scoprire se α é una tautologia basta analizzare 2^n valutazioni proposizionali.

7. Determinare quali tra le seguenti formule sono tautologie:

   -
   -

8. Dimostrare che $\alpha \to \beta, \alpha, \neg\beta \models \gamma$

9. Dimostrare che esiste una quantità più che numerabile di valutazioni proposizionali.

2.2 Gli alberi proposizionali

Ci poniamo ora il problema di verificare se una data formula proposizionale sia una tautologia senza dover ricorrere alla definizione (che rende la decisione in linea di principio impossibile perchè richiede di andare avedere se tutte le infinite valutazioni proposizionali rendono vera la formula) o alle tabelline di verità (che in genere richiedono troppi calcoli). L'idea é quella di analizzare la verità di una formula analizzando la verità delle sue componenti, scomponendola un po' per volta. Tuttavia piuttosto che verificare se una certa formula proposizionale é una tautologia (che richiederebbe di nuovo di verificare tutte le valutazioni proposizionali) é in generale più facile vedere se esiste una valutazione proposizionale che la renda vera, cioè che la *soddisfi*: ci si accorge subito che una formula é una tautologia se e solo se la sua negazione non é mai soddisfatta. Per avere una immegine mentale del processo di analisi si può considerare una formula come una storia, allora quel che si cerca di fare é analizzare tale storia per vedere se c'é sempre un modo per continuarla in modo coerente ed esente da contraddizioni. Cominciamo subito con un esempio: per vedere se la formula[1] $\alpha \to (\alpha \wedge \neg\beta)$ é una tautologia possiamo provare a vedere se $\neg(\alpha \to (\alpha \wedge \neg\beta))$ é soddisfacibile:

$$
\begin{array}{cccl}
 & \neg(\alpha \to (\alpha \wedge \neg\beta)) & & \text{é vera se } \alpha \text{ é vera e } \alpha \wedge \neg\beta \text{ é falsa} \\
 & \alpha & & \\
 & \neg(\alpha \wedge \neg\beta) & & \text{é vera se } \alpha \text{ é falso o se } \neg\beta \text{ é falso} \\
\neg\alpha & & \neg\neg\beta & \text{é vera se } \beta \text{ é vero} \\
\times & & \beta & \\
 & & \times &
\end{array}
$$

Ora nel ramo di sinistra abbiamo chiaramente ottenuto una storia che non può continuare in modo coerente perchè vorrei che fosse vera contemporaneamente sia α che la sua negazione mentre il ramo di destra rappresenta una storia plausibile a patto che sia α che β siano vere. Abbiamo cioè fatto vedere che la formula $\neg(\alpha \to (\alpha \wedge \neg\beta))$ é soddisfacibile utilizzando una qualsiasi valutazione σ tale che $\sigma(\alpha) = \top$ e $\sigma(\beta) = \top$ e quindi la formula $\alpha \to (\alpha \wedge \neg\beta)$ non può essere una tautologia visto che risulta falsa nella valutazione σ.

Possiamo formalizzare il metodo usato nel precedente esempio iniziando dalla definizione di *albero proposizionale*. Come abbiamo visto nell'esempio precedente, anche se lo scopo iniziale era quello di analizzare la soddisfacibilità di *una* formula, già al secondo passo nello sviluppo della storia ci si ritrova a chiedersi se un insieme di formule é soddisfacibile. Per questo motivo, invece che definere gli alberi proposizionali per una formula daremo direttamente la definzione nel caso più generale considerando direttamente un qualunque insieme di formule. L'idea di fondo rimane comunque la stessa: la storia raccontata da un insieme

[1] D'ora in avanti useremo sempre le lettere greche per indicare le componenti prime.

di formule é coerente se tutte le formule dell'insieme, e quelle da loro ottenute, sono tra loro coerenti. Questo approccio ci permette inoltre di affrontare, insieme alla ricerca delle tautologie, anche la ricerca delle conseguenze tautologiche; infatti la formula α é conseguenza tautologica dell'insieme di formule Φ se e solo se l'insieme di formule $\Phi \cup \{\neg\alpha\}$ é non soddisfacibile: potremo quindi scoprire se $\Phi \models \alpha$ guardando se tutte le storie raccontate da $\Phi \cup \{\neg\alpha\}$ sono non coerenti.

Definizione 2.8 (Albero proposizionale) *Un albero proposizionale per l'insieme di formule Φ é un albero i cui nodi sono insiemi di formule tali che*

- *Φ é un albero proposizionale per Φ (costituito dalla sola radice)*
- *Se T é un albero proposizionale per Φ, tale che l'insieme di formule Γ sia una foglia e R sia il ramo che unisce Γ alla radice, allora anche l'albero T', ottenuto in uno dei seguenti modi, é un albero proposizionale per Φ:*
  - *se $\neg\neg\alpha$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\alpha\}$;*
  - *se $\alpha \to \beta$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con due nodi, immediati successori di Γ, $\Gamma' \equiv \{\neg\alpha\}$ e $\Gamma'' \equiv \{\beta\}$;*
  - *se $\neg(\alpha \to \beta)$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\alpha, \neg\beta\}$*

Niente altro é un albero proposizionale.

Anche se abbiamo deciso di limitare la nostra analisi ai soli connettivi $\neg$ e $\to$, in quanto sufficienti per esprimere tutto ciò che é esprimibile trami tutti i connettivi, possiamo, per completezza, facilmente definire come estendere i rami di un albero proposizionale anche relativamente ai connettivi $\wedge$ e $\vee$:

- se $\alpha \wedge \beta$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\alpha, \beta\}$;
- se $\neg(\alpha \to \beta)$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con due nodi, immediati successori di Γ, $\Gamma' \equiv \{\neg\alpha\}$ e $\Gamma'' \equiv \{\neg\beta\}$;
- se $\alpha \vee \beta$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con due nodi, immediati successori di Γ, $\Gamma' \equiv \{\alpha\}$ e $\Gamma'' \equiv \{\beta\}$;
- se $\neg(\alpha \vee \beta)$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\neg\alpha, \neg\beta\}$.

Si noti che le formule prime e le loro negazioni non permettono alcuno sviluppo dell'albero proposizionale: questo si accorda con la nostra idea di analizzare una formula nelle sue componenti, infatti le formule prime e le loro negazioni non hanno componenti (almeno dal punto di vista proposizionale).

Se l'insieme delle formule che compaiono nei nodi di un ramo non costituiscono uno sviluppo coerente della storia descritta da Φ, vale a dire, se nei nodi di un ramo appaiono sia una formula prima che la sua negazione, allora quel ramo non merita di essere ulteriormente analizzato.

Definizione 2.9 (Ramo chiuso) *Un ramo proposizionale si dice chiuso se in esso occorre sia una formula prima che la sua negazione.*

Quindi nel nostro precedente esempio abbiamo costruito un albero proposizionale per l'insieme di formule $\{\neg(\alpha \to (\alpha \wedge \neg\beta))\}$ e il ramo che parte dalla radice e si conclude in $\neg\alpha$ é un ramo chiuso.

Se ogni ramo di un albero proposizional e per Φ é chiuso significa che tutte le storie che Φ descrive sono incoerenti.

Definizione 2.10 (Albero di confutazione) *Un albero proposizionale per l'insieme di formule Φ é di confutazione se ogni suo ramo é chiuso.*

Si può allora stabilire un primo legame tra l'esistenza di un albero di confutazione per un insieme di formule Φ e la posibilità di trovare una valutazione che renda vere tutte le formule di Φ.

Teorema 2.11 (Correttezza del calcolo degli alberi proposizionali) *Se l'insieme di formule Φ é soddisfacibile allora non esiste un albero di confutazione per Φ.*

Prima di dimostrare questo teorema vediamone il significato. Intanto é il primo legame che abbiamo tra una nozione semantica, che ha cioè a che fare con la nozione di verità, quale é la nozione di *soddisfacibilità* di un insieme di formule, e una nozione di carattere sintattico, quale la nozione di *confutazione* di un albero proposizionale che ha a che fare solamente con la manipolazione di formule. Inoltre il legame stabilito ci assicura che gli alberi di confutazione non sono mai menzonieri, vale a dire che se troviamo che un certo insieme di formule ha un albero di confutazione é proprio perchè tale insieme non é soddisfacibile.

Vediamo adesso di dimostrare questo teorema. Iniziamo facendo vedere che, se tutte le formule di un ramo R di un albero proposizionale T sono soddisfatte da una valutazione proposizionale σ allora, se T' é un albero che estebde T, esiste in T' un ramo le cui formule sono tutte soddisfacibili, e sono addirittura soddisfatte dalla stessa valutazione proposizionale σ. Analizziamo dunque i vari modi in cui T' può essere stato ottenuto a partire da T:

- se T' é stato ottenuto da T estendendo un ramo diverso da R allora la tesi é banale: tutte le formule che appaio nei nodi del ramo R dell'albero T', che é identico al ramo R dell'albero T, sono soddisfatte da σ.

- Se T' é stato ottenuto estendendo il ramo R di T aggiungendo un nuovo nodo $\{\alpha\}$ perchè un qualche nodo del ramo R conteneva la formula $\neg\neg\alpha$ allora, per ipotesi $\sigma(\neg\neg\alpha) = \top$ e quindi $\sigma(\neg\alpha) = \bot$ da cui si ottiene $\sigma(\alpha) = \top$; quindi tutte le formule nel nuovo ramo sono soddisfatte da σ.

- Se T' é stato ottenuto estendendo il ramo R di T aggiungendo un nuovo nodo $\{\alpha, \neg\beta\}$ perchè un qualche nodo del ramo R conteneva la formula $\neg(\alpha \to \beta)$ allora, per ipotesi $\sigma(\neg(\alpha \to \beta)) = \top$ e quindi $\sigma(\alpha) = \top$ e $\sigma(\beta) = \bot$, da cui si ottiene $\sigma(\neg\beta) = \top$; quindi tutte le formule nel nuovo ramo sono soddisfatte da σ.

- Se T' é stato ottenuto estendendo il ramo R di T aggiungendo i due nuovi nodi $\{\neg\alpha\}$ e $\{\beta\}$ perchè un qualche nodo del ramo R conteneva la formula $\alpha \to \beta$ allora, per ipotesi $\sigma(\alpha \to \beta) = \top$ e quindi $\sigma(\alpha) = \bot$, e quindi $\sigma(\neg\alpha) = \top$, oppure $\sigma(\beta) = \top$; quindi le formule in almeno uno dei nuovi rami sono tutte soddisfatte da σ.

La conclusione é ora immediata: infatti, se Φ é un insieme soddisfacibile di formule allora, per quanto abbiamo appena visto, ogni ramo di un albero proposizionale per Φ deve avere almeno un ramo le cui formule sono tutte soddisfacibili (dimostrazione per induzione sulla costruzione dell'albero proposizionale per Φ). Quindi ogni albero proposizionale per Φ deve avere almeno un ramo che non é chiuso perchè non é possibile che una valutazione proposizionale possa rendere vera sia una formula prima che la sua negazione. Perciò non ci può essere alcun albero di confutazione per Φ.

Abbiamo allora definito un metodo che può aiutarci a scoprire se un insieme Φ di formule é non soddisfacibile: se Φ ha albero di confutazione allora Φ non é soddisfacibile. La prossima domanda é allora: questo metodo é sufficiente per scoprire tutti gli insiemi di formule non soddisfacibili? Vedremo che un uso accurato degli alberi proposizionali, almeno nel caso di insiemi finiti di formule, permette di rispondere positivamente a questa domanda e quindi di risolvere sintatticamente il problema di scoprire le tautologie.

É infatti chiaro che le condizioni date per costruire gli alberi proposizionali non costituiscono un vero e proprio algoritmo per generare un albero proposizionale a partire da un dato insieme di formule; infatti esse lasciano la possibilità di scegliere liberamente ad ogni passo dello sviluppo di un albero di confutazione su quale ramo e su quale formula agire. Esse costituiscono piuttosto delle condizioni per riconoscere, dato un albero di formule, se esso é oppure no un albero proposizionale per un dato insieme di formule Φ. Se vogliamo risolvere il nostro problema dobbiamo piuttosto cercare un algoritmo per costruire l'albero proposizionale per Φ allo scopo di scoprire se Φ é soddisfacibile oppure no. La prima cosa che si nota é che se si continua ad espandere un albero usando sempre la stessa formula sio spreca solamente tempo senza ottenere nuova conoscenza: della storia che quella formula narra viene continuamente ripetuto un frammento come un disco rotto ma non si sa mai come va avanti! Quindi vediamo prima di tutto di chiarire quando é inutile continuare ad usare una certa formula per sviluppare un ramo.

Definizione 2.12 (Formula usata in un ramo) *Una formula α si dice* usata in un ramo R *di un albero proposizionale se*

- *α é una formule prima o la negazione di una formula prima;*

- $\alpha \equiv \neg\neg\beta$ *e* β *appare in uno dei nodi del ramo* R*;*
- $\alpha \equiv \neg(\beta \to \gamma)$ *e* β *e* $\neg\gamma$ *appaiono tra le formule nei nodi del ramo* R*;*
- $\alpha \equiv \beta \to \gamma$ *e* $\neg\beta$ *o* γ *appaiono in uno dei nodi del ramo* R*.*

Ovviamente, se una formula é usata in ogni ramo dell'albero che passa per il nodo che la contiene non ha più senso usarla per sviluppare i rami.

Definizione 2.13 (Formula usata in un albero) *La formula* α *si dice* formula usata nell'albero T *se é usata in ogni ramo di* T *che passa per il nodo che la contiene.*

É chiaro che quando in un albero T tutte le formule sono usate non ha più molto senso espanderlo: tutto ciò che si otterrebbe sarebbe una ripetizione degli insiemi di formule che già vi appaiono.

Definizione 2.14 (Albero completo di formule) *Un albero di formule le cui formule siano tutte usate si dirà* albero completo di formule.

Dimostriamo allora che gli alberi completi di formule sono il nostro miglior tentativo per scoprire sintatticamente tutte le tautologie: essi infatti rappresentano lo svolgimento di tutte le storie possibili narrate da un insieme finito di formule.

Teorema 2.15 (Completezza proposizionale debole) *Se un albero completo* T *per un insieme di formule* Φ *non é di confutazione allora* Φ *é un insieme soddisfacibile di formule.*

La dimostrazione di questo teorema consiste nello scoprire una storia coerente tra tutte quelle che l'insieme di formule Φ può narrare: la storia andrà cercata nel ramo aperto R dell'albero T, che deve esserci visto che per ipotesi l'albero T non é un albero di confutazione. Il nostro scopo sarà quindi di trovare una valutazione proposizionale σ che renda vere tutte le formule che appaiono nel ramo R. Questo é infatti sufficiente per far vedere che Φ é soddisfacibile poichè Φ é uno dei nodi di qualunque ramo R (essendo la radice dell'albero di confutazione) e quindi tutte le formule di Φ saranno vere nella valutazione σ.

Sia allora $\mathcal{R}$ l'unione di tutti gli insiemi di formule che appaiono nei nodi del ramo aperto R e definiamo la valutazione σ sulle formule prime π tramite la seguente posizione:

$$\begin{array}{lll} \sigma(\pi) = \top & \text{se} & \pi \in \mathcal{R} \\ \sigma(\pi) = \bot & \text{se} & \pi \notin \mathcal{R} \end{array}$$

In questo modo é evidente che riusciamo a soddisfare, per quanto riguarda le formule prime, la richiesta che le formule che appaiono nel ramo R siano valutate da σ in vero. Ricordiamo ora che una valutazione proposizionale viene completa determinata dal modo in cui sono valutate le formule prime. Dobbiamo quindi dimostrare che, per ogni formula α, la valutazione σ che abbiamo definito soddisfa la condizione da noi richiesta. La dimostrazione ssi svolgerà per induzione sulla complessità della costruzione della formula α, ma invece di dimostrare solo che

- se $\alpha \in \mathcal{R}$ allora $\sigma(\alpha) = \top$

é più facile dimostrare contemporaneamente anche che

- se $\neg\alpha \in \mathcal{R}$ allora $\sigma(\alpha) = \bot$

Analizziamo infatti i tre casi che si possono presentare:

- α sia una formula prima; allora
  - se $\alpha \in \mathcal{R}$ allora $\sigma(\alpha) = \top$ per definizione;
  - se $\neg\alpha \in \mathcal{R}$ allora, poichè R é un ramo aperto, $\alpha \notin \mathcal{R}$ e quindi, per definizione, $\sigma(\alpha) = \bot$.
- $\alpha \equiv \neg\beta$
  - se $\alpha \in \mathcal{R}$, vale a dire $\neg\beta \in \mathcal{R}$, allora, per ipotesi induttiva, $\sigma(\beta) = \bot$ e quindi $\sigma(\alpha) = \sigma(\neg\beta) = \top$
  - se $\neg\alpha \in \mathcal{R}$, vale a dire $\neg\neg\beta \in \mathcal{R}$, allora, poichè per ipotesi T é un albero completo, $\beta \in \mathcal{R}$ e quindi, per ipotesi induttiva, $\sigma(\beta) = \top$ che implica che $\sigma(\alpha) = \sigma(\neg\beta) = \bot$
- $\alpha \equiv \beta \to \gamma$
  - se $\alpha \in \mathcal{R}$, vale a dire $\beta \to \gamma \in \mathcal{R}$, allora, poichè per ipotesi T é un albero completo, $\neg\beta \in \mathcal{R}$ oppure $\gamma \in \mathcal{R}$ e quindi, per ipotesi induttiva, $\sigma(\beta) = \bot$ oppure $\sigma(\gamma) = \top$; quindi otteniamo in ogni caso $\sigma(\alpha) = \sigma(\beta \to \gamma) = \top$
  - se $\neg\alpha \in \mathcal{R}$, vale a dire $\neg(\beta \to \gamma) \in \mathcal{R}$, allora, poichè per ipotesi T é un albero completo, $\beta \in \mathcal{R}$ e $\neg\gamma \in \mathcal{R}$; quindi, per ipotesi induttiva, $\sigma(\beta) = \top$ e $\sigma(\gamma) = \bot$ che implicano che $\sigma(\alpha) = \sigma(\beta \to \gamma) = \bot$

Abbiamo visto cosí che un qualunque albero completo per un insieme di formule proposizionali Φ é di confutazione se e solo se Φ non é soddisfacibile. Ma in quali casi un insieme di formule Φ ammette un albero completo? Chairamente se Φ non é un insieme finito non cé speranza di costruire, secondo la definizione di albero proposizionale che abbiamo fornito, un albero completo per Φ poichè ci saranno sempre delle formule non (ancora) usate dentro Φ. Tuttavia se Φ é un insieme finito é un insieme fnito di formule e abbiamo l'accortezza di usare in ogni passo dello sviluppo dell'albero solamente formule non usate prima o poi otterremo un albero completo per Φ. Infatti, anche se é vero che in un passo dell'espansione possiamo ottenere un albero in cui compaiono pi‘u formule non usate che nel passo precedente (ad esempio, se sviluppiamo un ramo in cui compare la formula $\neg(\alpha \to \beta)$ essa viene usata ma aggiungiamo le due formule non usate α e $\neg\beta$) queste nuove formule saranno più semplici di quella di partenza e quindi prima o poi si arriva alle formule prime o alle loro negazioni e queste sono formule usate per definizione.

Abbiamo quindi dimostrato il seguente teorema.

Teorema 2.16 *Se Φ é un insieme finito di formule proposizionali ed α é una formula proposizionale allora le seguenti affermazioni sono equivalenti:*

- $\Phi \models \alpha$
- $\Phi \cup \{\neg\alpha\}$ *non é soddisfacibile*
- *esiste un albero completo di confutazione per* $\Phi \cup \{\neg\alpha\}$
- *ogni albero completo per* $\Phi \cup \{\neg\alpha\}$ *é di confutazione*

Il teorema precedente risolve quindi completamente il problema di riconoscere sintatticamente le tautologie. Ricordando infatti che quando vogliamo sapere se la formula α é una tautologia ci basta scoprire se l'insieme $\{\neg\alpha\}$ é non soddisfacibile e che $\{\neg\alpha\}$ é non soddisfacibile se é confutabile, allora ci troviamo nella situazione di applicare il teorema precedente visto che stiamo chiaramente trattando con un insieme finito. Per decidere se α é una tautologia quel che dobbiamo fare é allora trovare l'albero completo per $\{\neg\alpha\}$ e vedere se esso é di confutazione: se esso é di confutazione allora $\{\neg\alpha\}$ non é soddisfacibile e quindi α é una tautologia, altrimenti $\neg\alpha$ é una formula soddisfacibile e quindi α non può essere una tautologia.

2.2.1 Esercizi

1. Dimostrare che una formula proposizionale é una tautologia se e solo se la sua negazione non é mai soddisfatta.

2. Sia Φ un qualsiasi insieme di formule proposizionali e α sia una formula proposizionale. Dimostrare che α é conseguenza tautologica dell'insieme di formule Φ se e solo se l'insieme di formule $\Phi \cup \{\neg\alpha\}$ é non soddisfacibile.

3. Utilizzando le definizioni di $\wedge$ e $\vee$ in termini di $\neg$ e $\rightarrow$, verificare che le istruzioni per estendere un albero proposizionale nel caso di $\wedge$ e $\vee$ coincidono con quelle che si otterrebbero estendendo tale albero seguende le regole per $\neg$ e $\rightarrow$ quando le formule che utilizzano $\wedge$ e $\vee$ vengono sostituite con quelle ottenute tramite le loro definizioni.

4. Sia Φ un insieme soddisfacibile di formule. Dimostrare che ogni ramo di un albero proposizionale per Φ deve avere almeno un ramo le cui formule sono tutte soddisfacibili (sugg.: usare una prova per induzione sulla costruzione dell'albero proposizionale per Φ).

5. (Difficile) Fornire una completa dimostrazione formale del fatto che dato un insieme *finito* di formule proposizionali esso ammette un albero proposizionale completo.

6. Dimostrare che i seguenti insiemi di formule proposizionali non sono soddisfacibili o fornire una valutazione che renda vere tutte le loro formule:

   -

-
-

2.3 Il calcolo proposizionale

Un approccio di carattere più matematico per affrontare il problema di scoprire sintatticamente le atutologie é quello adottato con il calcolo proposizionale. In questo approccio si determina un insieme di formule scegliendone dapprima alcune da cui partire e fornendo poi delle regole che permettono di inserirne di nuove, a partire dalle formule che già si sono riconosciute appartenere all'insieme desiderato. Se si scelgono come formule di partenza delle tautologie e le regole sono fatte in modo da far ottenere tautologie quando sono applicate a tautologie é chiaro che in questo modo definiremo un insieme di formule che contiene solamente tautologie: il problema é naturalmente di scegliere opportunamente le formule di partenza e le regole in modo da ottenere tutte le tautologie!

Usando le lettere greche come variabili che stanno per formule qualsiasi, scegliamo come formule di partenza le seguenti:

- $\alpha \rightarrow (\beta \rightarrow \alpha)$
- $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
- $(\neg\alpha \rightarrow \beta) \rightarrow ((\neg\alpha \rightarrow \neg\beta) \rightarrow \alpha)$

D'ora in avanti chiameremo *assiomi* tutte le formule che appartengono all'insieme di partenza. É importante notare che qui sopra non abbiamo scelto tre assiomi ma, per il fatto che abbiamo usato le variabili per formule α, β e γ abbiamo introdotto nel nostro insieme di partenza infinite formule, ua per ogni scelta di una formula da sostituire al posto della variabile α, di una formula da sostituire al posto della variabile β e di una formula da sostituire al posto della variabile γ. Si verifica inoltre facilmente che tutte queste infinite formule sono tautologie.

Si vede comunque subito che, ad esempio, la tautologia $\alpha \rightarrow \alpha$ non é uno dei nostri assiomi; infatti gli assiomi sono facilmente, vale a dire meccanicamente, riconoscibili da tutte le altre formule[2] e si vede subito che $\alpha \rightarrow \alpha$ non é uno di essi. Le infinite formule che abbiamo scelto di introdurre quali nostro insieme di partenza non esauriscono quindi tutte le tautologie: se vogliamo ottenerle tutte dobbiamo aggiungere ancora assiomi o introdurre qualche regola. Scegliamo questa seconda alternativa ed introduciamo l'unica regola del nostro calcolo. Vale la pena di notare che questo approccio é sprecato nel caso del calcolo proposizionale; si potrebbero infatti introdurre quali assiomi tutte le tautologie visto che esse sono in realtà meccanicamente riconoscibile tramite il metodo delle tabelline di verità. Tuttavia questo metodo apre delle possibilità per i

[2] Questa é una richiesta che vogliamo sempre che sia soddisfatta da un insieme di assiomi, altrimenti l'approccio sintattico sarebbe inutile visto che il suo intento é quello di ridurre l'infinito che appare nelle nozioni semantiche al finito o almeno al meccanicamente verificabile.

successivi sviluppi di un calcolo logico per l'intero linguaggio predicativo visto che per esso non esiste comunque una possibilità analoga alle tabelline di verità.

$$\text{(Modus Ponens)} \quad \frac{\alpha \qquad \alpha \to \beta}{\beta}$$

Le verifica che tramite questa regola si ottengono tautologia quando essa venga utilizzata su delle tautologie é pressichè immediata: supponiamo infatti che una generica valutazione proposizionale σ valuti in vero sia α che $\alpha \to \beta$, vale a dire che $\sigma(\alpha) = \top$ and $\sigma(\alpha \to \beta) = \top$, allora deve chiaramente valere $\sigma(\beta) = \top$.

Ci troviamo ora di fronte ad un difficile problema: basteranno gli assiomi che abbiamo scelto e il *modus ponens* per ottenere tutte le tautologie? Per rispondere a questa domanda cominciamo definendo formalmente l'insieme delle formule che possiamo ottenere a partire dagli assiomi utilizzando il *modus ponens*, vale a dire l'insieme dei *teoremi*. Possiamo risparmiarci un po' di lavoro successivo se diamo subito una definizione generale abbastanza da non considerare solamente le tautologie ma che ci fornisca direttamente la controparte sintattica della nozione di conseguenza tautologica: sappiamo già che le tautologie altro non sono che un caso speciale di quest'ultima.

Definizione 2.17 (Dimostrazioni e Teoremi) *Una* derivazione *della formula α dall'insieme di formule Φ detto l'insieme delle ipotesi, notazione $\Phi \vdash \alpha$, é una sequenza $\phi_1, \ldots, \phi_n$ di formule, tale che $\phi_n \equiv \alpha$ e, per ogni i compreso tra 1 e n,*

- *ϕ_i é un assioma, oppure*
- *ϕ_i é una delle ipotesi in Φ, oppure*
- *ϕ_i é ottenuto per* modus ponens *da due formule che precedono ϕ_i, vale a dire che esistono nella sequenza due numeri j e k, minori di i, tali che $\phi_k \equiv \phi_j \to \phi_i$.*

Nel caso la derivazione della formula α si ottenga a partire dall'insieme vuoto di ipotesi si derà che tale derivazione é una dimostrazione *di α e, in questo caso, si dirà che α é un teorema, notazione $\vdash \alpha$.*

Rileggendo la precedente definizione si scopre subito che se α é derivabile da Φ allora essa é una conseguenza tautologica di Φ e, come caso speciale, che ogni teorema é una tautologia: il nostro problema si allora riformulare chiedendoci se qualora α sia una conseguenza tautologica di Φ allora α é derivabile da Φ.

Vediamo intanto che adesso riusciamo almeno a dimostrare la tautologia $\alpha \to \alpha$ che non appariva tra gli assiomi:

1.	Ass.2	$(\alpha \to ((\alpha \to \alpha) \to \alpha)) \to ((\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha))$
2.	Ass.1	$(\alpha \to ((\alpha \to \alpha) \to \alpha))$
3.	MP(2,1)	$(\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha)$
4.	Ass.1	$\alpha \to (\alpha \to \alpha)$
5.	MP(4,3)	$\alpha \to \alpha$

Come si vede lo scoprire un teorema é un'operazione molto più complicata che costruire un albero di confutazione o una tabellina di verità! Converrà quindi elaborare qualche strumento che renda più facile scoprire le derivazioni. Premettiamo una piccola osservazione che useremo nel seguito senza menzione: se la formula γ é derivabile a partire da un insieme Π di ipotesi allora essa é derivabile anche a partire da un qualunque insieme di ipotesi Σ che estenda Π visto che possiamo ritrovare in Σ tutte le formule di Π che possiamo aver usato nella derivazione di γ.

Teorema 2.18 (Teorema di deduzione sintattico) *Sia Φ un insieme di formule proposizionali e siano α e β due formule proposizionali. Allora*

$$\Phi, \alpha \vdash \beta \textit{ se e solo se } \Phi \vdash \alpha \to \beta$$

L'enunciato di questo teorema altro non é che la trascrizione sintattica del teorema 2.7 di deduzione semantico che avevamo visto nel precedente paragrafo. Questo non dovrebbe stupirci troppo: se vogliamo sperare che il calcolo proposizionale sia sufficiente per scoprire tutte le conseguenze tautologiche bisogna che le relazioni semantiche, come quelle espresse nel teorema di deduzione semantico, abbiano una loro controparte sintattica. Il nostro attuale interesse in questo teorema consiste nel fatto che esso rende più facile trovare le dimostrazioni; infatti é più facile dimostrare $\Phi, \alpha \vdash \beta$ che $\Phi \vdash \alpha \to \beta$ poichè nel primo caso possiamo sfruttare più ipotesi per dimostrare una formula più semplice.

Come al solito, divideremo in due parti la dimostrazione del "se e solo se":

- Supponiamo che $\Phi, \alpha \vdash \beta$. Allora la dimostrazione che $\Phi \vdash \alpha \to \beta$ si ottiene per induzione completa sulla lunghezza della derivazione di β dalle ipotesi in Φ e α.
  - supponiamo che la derivazione sia di lunghezza 1; allora si possono presentare tre casi:
    - Casi 1 e 2: supponiamo che β sia un assioma oppure che β sia una ipotesi in Φ; in entrambi i casi si può ottenere una derivazione, da Φ di $\alpha \to \beta$ usando *modus ponens* tra β e l'istanza $\beta \to (\alpha \to \beta)$ del primo schema di assiomi;
    - Caso 3: β sia α; in questo caso $\beta \to \alpha$ coincide con $\alpha \to \alpha$ e quindi abbiamo già visto come si può costruire una sua derivazione
  - supponiamo di avere una derivazione di lunghezza n maggiore di 1; possono allora presentarsi quattro casi, tuttavia i primi tre coincidono con quelli già visti nel punto precedente; vediamo allora solo il caso in cui β sia stato ottenuto per *modus ponens* a partire dalle due formule ϕ_j e $\phi_k \equiv \phi_j \to \beta$ che si derivano dalle ipotesi Φ e α con una prova di lunghezza minore di n; per ipotesi induttiva questo significa che possiamo ottenere una derivazione di $\alpha \to \phi_j$ ed una derivazione di $\alpha \to (\phi_j \to \beta)$ a partire da Φ; allora una derivazione di $\alpha \to \beta$ si

può ottenere tramite due applicazioni della regola di *modus ponens* all'istanza $(\alpha \to (\phi_j \to \beta)) \to ((\alpha \to \phi_j) \to (\alpha \to \beta))$ del secondo schema di assiomi

- Supponiamo di avere una derivazione di $\Phi \vdash \alpha \to \beta$. Allora anche $\Phi, \alpha \vdash \alpha \to \beta$ é derivabile, ma naturalmente anche $\Phi, \alpha \vdash \alpha$ é derivabile e quindi $\Phi, \alpha \vdash \beta$ si ottiene subito per *modus ponens*.

Come applicazione del teorema di deduzione sintattica che abbiamo appena dimostrato vediamo di dimostrare un'altra formula proposizionale. Vogliamo vedere che la formula $\neg\neg\alpha \to \alpha$ é un teorema. Prima di tutto si verifica facilmente che questa formula é una tautologia pur non essendo sicuramente uno dei nostri assiomi. Per far vedere che ha una dimostrazione possiamo allora utilizzare il teorema di deduzione e dimostrare che $\neg\neg\alpha \vdash \alpha$ invece che $\vdash \neg\neg\alpha \to \alpha$. Questa conseguenza logica risulta infatti più facile da dimostrare che la formula originale.

		... dimostrazione di $\neg\alpha \to \neg\alpha$
5.		$\neg\alpha \to \neg\alpha$
6.	Ip	$\neg\neg\alpha$
7.	Ass.1	$\neg\neg\alpha \to (\neg\alpha \to \neg\neg\alpha)$
8.	MP(6,7)	$\neg\alpha \to \neg\neg\alpha$
9.	Ass.3	$(\neg\alpha \to \neg\alpha) \to ((\neg\alpha \to \neg\neg\alpha) \to \alpha)$
10.	MP(5,9)	$(\neg\alpha \to \neg\neg\alpha) \to \alpha$
11.	MP(8,10)	α

Quindi anche questa tautologia viene recuperata tramite l'uso della regola di *modus ponens*. Per vedere che in realtà gli assiomi che abbiamo fornito e il *modus ponens* sono sufficienti per ottenere tutte le tautologie ci basta far vedere che se l'insieme $\{\alpha\}$ é confutabile allora α é un teorema del calcolo proposizionale. Infatti α é una tautologia se e solo se $\neg\alpha$ non é soddisfacibile che per il teorema di completezza debole vale se e solo se $\{\alpha\}$ é confutabile. Per ottenere questa prova introduciamo un concetto che, nel contesto del calcolo proposizionale, é l'analogo della nozione di insieme confutabile di formule.

Definizione 2.19 (Insieme inconsistente di formule) *L'insieme do formule proposizionali Φ é* inconsistente *se da esso si pò dedurre ogni formula.*

É facile far vedere che in realtà affinchè un insieme di formule Φ sia inconsistente basta che da esso sia derivabile sia una formula β che la sua negazione. Infatti in questo caso possiamo ottenere la seguente derivazione di una generica formula proposizionale γ:

	$\ldots$	derivazione di β da Φ
	β	
Ass.1	$\beta \to (\neg\gamma \to \beta)$	
MP	$\neg\gamma \to \beta$	
	$\ldots$	derivazione di $\neg\beta$ da Φ
	$\neg\beta$	
Ass.1	$\neg\beta \to (\neg\gamma \to \neg\beta)$	
MP	$\neg\gamma \to \neg\beta$	
Ass.3	$(\neg\gamma \to \beta) \to ((\neg\gamma \to \neg\beta) \to \gamma)$	
MP	$(\neg\gamma \to \neg\beta) \to \gamma$	
MP	γ	

Un ulteriore conferma che la nozione di insieme inconsistente é importante per trovare relazioni tra la sintassi del calcolo proposizionale e la sua semantica viene dal fatto che gli insiemi inconsistenti di formule sono correlati alla noziene di derivazione nello stesso modo in cui gli insiemi non soddisfacibili di formule sono correlati alla nozione di conseguenza tautologica.

Teorema 2.20 *Sia Φ un insieme di formule proposizionali e sia α una formula proposizionale. Allora*

1. $\Phi \vdash \alpha$ *se e solo se* $\Phi \cup \{\neg\alpha\}$ *é inconsistente*
2. $\Phi \vdash \neg\alpha$ *se e solo se* $\Phi \cup \{\alpha\}$ *é inconsistente*

Dimostrazione.

1.
   - Supponiamo che $\Phi \vdash \alpha$ e vediamo come dimostrare che $\Phi \cup \{\neg\alpha\}$ é inconsistente. Prima di tutto, é evidente che se α si può derivare da Φ essa si può derivare anche da $\Phi \cup \{\neg\alpha\}$. Ma da $\Phi \cup \{\neg\alpha\}$ si deriva subito anche $\neg\alpha$ e quindi $\Phi \cup \{\neg\alpha\}$ é inconsistente.
   - Supponiamo che $\Phi \cup \{\neg\alpha\}$ sia inconsistente. Allora esiste una formula β tale che sia $\Phi, \neg\alpha \vdash \beta$ che $\Phi, \neg\alpha \vdash \neg\beta$ valgono. Allora, per il teorema di deduzione, sono derivabili dall'insieme Φ di ipotesi sia $\neg\alpha \to \beta$ che $\neg\alpha \to \neg\beta$. Quindi possiamo derivare α tramite due applicazioni della regola di *modus ponens* all'istanza $(\neg\alpha \to \beta) \to ((\neg\alpha \to \neg\beta) \to \alpha)$ del terzo assioma.
2.
   - La prova che $\Phi \vdash \neg\alpha$ implica che $\Phi \cup \{\alpha\}$ é inconsistente é completamente analoga a quella fatta nel primo punto qui sopra.
   - Vogliamo ora dimostrare che se $\Phi \cup \{\alpha\}$ é inconsistente allora $\Phi \vdash \neg\alpha$. Tuttavia per ottenere questo risultato non serve fare una prova *ad hoc* in quanto lo possiamo ottenere a partire dalla analoga prova del punto precedente se possiamo far vedere che se $\Phi \cup \{\alpha\}$ é inconsistente allora lo é anche $\Phi \cup \{\neg\neg\alpha\}$. Vediamo allora di dimostrare questo

risultato. Supponiamo che β sia una formula tale che $\Phi \cup \{\alpha\} \vdash \beta$ e $\Phi \cup \{\alpha\} \vdash \neg\beta$. Possiamo allora ottenere una derivazione di β a partire dalle ipotesi in $\Phi \cup \{\neg\neg\alpha\}$ modificando la derivazione di β da $\Phi \cup \{\alpha\}$ nel modo che segue: sostituiamo ogni occorrenza della formula α nella derivazione con una derivazione di α da $\neg\neg\alpha$ che abbiamo già visto come costruire. Naturalmente possiamo ottenere in modo analogo una dimostrazione di $\neg\beta$ a partire dalle ipotesi $\Phi \cup \{\neg\neg\alpha\}$ modificando la derivazione di $\neg\beta$ da $\Phi \cup \{\alpha\}$.

Il prossimo passo é ora quello di trovare un sistema per simulare, all'interno del calcolo proposizionale, ciò che succede nello sviluppo di un albero di confutazione. Questo é lo scopo dei prossimi lemmi. L'dea di fondo é quella di ripercorrere i vari passi che ci hanno portato alla dimostrazione del teorema di completezza debole della sezione precedente sostituiendo la nozione di insieme confutabile con quella di insieme inconsistente e dimostrando in particolare che se é inconsistente un insieme di formule ottenuto prolungando un ramo di un albero proposizionale allora é inconsistente anche l'insieme di formule che si aveva prima di tale estensione.

Lemma 2.21 *Sia Φ un insieme di formule proposizionali e siano α e β due furmule proposizionali. Allora*

1. *se $\Phi \cup \{\alpha\}$ é inconsistente e $\neg\neg\alpha \in \Phi$ allora Φ é inconsistente;*
2. *se $\Phi \cup \{\alpha, \neg\beta\}$ é inconsistente e $\neg(\alpha \to \beta) \in \Phi$ allora Φ é inconsistente;*
3. *se $\Phi \cup \{\neg\alpha\}$ e $\Phi \cup \{\beta\}$ sono inconsistenti e $\alpha \to \beta \in \Phi$ allora Φ é inconsistente.*

Dimostrazione.

1. Se $\Phi \cup \{\alpha\}$ é inconsistente allora $\Phi \vdash \neg\alpha$. Ma per ipotesi $\Phi \vdash \neg\neg\alpha$, visto che $\neg\neg\alpha \in \Phi$, e quindi Φ é inconsistente poichè da questo insieme di ipotesi sono derivabili sia $\neg\alpha$ che la sua negazione.
2. Se $\Phi \cup \{\alpha, \neg\beta\}$ é inconsistente allora $\Phi, \alpha \vdash \neg\beta$ e quindi $\Phi \vdash \alpha \to \beta$ segue dal teorema di deduzione sintattico. Ma per ipotesi $\Phi \vdash \neg(\alpha \to \beta)$, visto che $\neg(\alpha \to \beta) \in \Phi$, e quindi Φ é inconsistente.
3. Se $\Phi \cup \{\neg\alpha\}$ é inconsistente allora $\Phi \vdash \alpha$. D'altra parte, per ipotesi $\alpha \to \beta \in \Phi$ e quindi $\Phi \vdash \alpha \to \beta$ che, insieme $\Phi \vdash \alpha$, implica $\Phi \vdash \beta$ per *modus ponens*. Ma allora Φ é inconsistente perchè per ipotesi $\Phi \cup \{\beta\}$ é inconsistente e quindi $\Phi \vdash \neg\beta$.

Siamo ora in grado di concludere la dimostrazione del teorema che stabilisce che ogni tautologia é un teorema.

Teorema 2.22 (Confutabilità implica inconsistenza) *Sia Φ un insieme confutabile di formule. Allora Φ é inconsistente.*

Questo teorema é sufficiente per ottenere il risultato desiderato. Infatti se Φ é un insieme finito di formule, allora $\Phi \models \alpha$ implica che l'insieme $\Phi \cup \{\neg\alpha\}$ é non soddisfacibile e quindi, per quanto visto nella sezione precedente, $\Phi \cup \{\neg\alpha\}$ é un insieme confutabile (almeno se supponiamo che Φ sia un insieme finito) e quindi, in virtù del teorema qui sopra, $\Phi \cup \{\neg\alpha\}$ é un insieme inconsistente, ma per quanto abbiamo visto questo implica che $\Phi \vdash \alpha$; ora, nel caso speciale in cui l'insieme Φ sia vuoto, e sia quindi in particolare finito, quanto appena detto significa appunto che se α é una tautologia allora α é un teorema del calcolo proposizionale.

Naturalmente, la questione é ora quella di dimostrare il teorema. Possiamo costruire la dimostrazione ragionando per induzione sulla profondità dell'albero di confutazione per Φ, vale a dire sulla lunghezza del suo ramo più lungo; infatti, l'ipotesi del teorema é che Φ é un insieme confutabile ed esiste quindi un albero proposizionale per Φ i cui rami risultano tutti chiusi dopo un numero finito di passi e hanno quindi tutti una lunghezza finita.

- $n = 0$. In questo caso, affinchè Φ sia confutabile bisogna che esista una formula prima tale che sia lei che la sua negazione appartengono a Φ, ma allora esse sono derivabili da Φ e quindi Φ é inconsistente.

- $n > 0$ In questo caso l'albero di confutazione per Φ deve essere stato ottenuto tramite una delle regole che permettono di espandere l'albero. Quindi al primo passo dell'espansione deve essersi presentato uno dei seguenti casi.

  – Abbiamo espanso $\neg\neg\alpha$ che era una delle formule in Φ, vale a dire,

  $$\Phi$$

  $$\{\alpha\}$$

  $$\ldots T \ldots$$

  Ma allora

  $$\Phi \cup \{\alpha\}$$

  $$\ldots T \ldots$$

  é un albero di confutazione per $\Phi \cup \{\alpha\}$ di profondità $n - 1$. Quindi, per ipotesi indutttiva, $\Phi \cup \{\alpha\}$ é incosistente. Ma allora, visto che stiamo supponendo che $\neg\neg\alpha \in \Phi$, possiamo usare il lemma precedente per stabilire che Φ é inconsistente.

  – Abbiamo espanso $\neg(\alpha \to \beta)$ che era una delle formule in Φ, vale a dire,

$$\begin{array}{c}\Phi\\ \\ \{\alpha,\neg\beta\}\\ \\ \ldots T\ldots\end{array}$$

Ma allora

$$\begin{array}{c}\Phi\cup\{\alpha,\neg\beta\}\\ \\ \ldots T\ldots\end{array}$$

é un albero di confutazione per $\Phi \cup \{\alpha, \neg\beta\}$ di profondità $n-1$. Quindi, per ipotesi indutttiva, $\Phi \cup \{\alpha, \neg\beta\}$ é incosistente. Ma allora, visto che stiamo supponendo che $\neg(\alpha \to \beta) \in \Phi$, possiamo usare il lemma precedente per stabilire che Φ é inconsistente.

– Abbiamo espanso $\alpha \to \beta$ che era una delle formule in Φ, vale a dire,

$$\begin{array}{ccc} & \Phi & \\ & & \\ \{\neg\alpha\} & & \{\beta\}\\ & & \\ \ldots T_1\ldots & & \ldots T_2\ldots\end{array}$$

Ma allora

$$\begin{array}{c}\Phi\cup\{\neg\alpha\}\\ \\ \ldots T_1\ldots\end{array} \quad \text{e} \quad \begin{array}{c}\Phi\cup\{\beta\}\\ \\ \ldots T_2\ldots\end{array}$$

sono due alberi di confutazione rispettivamente per $\Phi\cup\{\neg\alpha\}$ e $\Phi\cup\{\beta\}$ di profondità $n-1$. Quindi, per ipotesi indutttiva, $\Phi\cup\{\neg\alpha\}$ e $\Phi\cup\{\beta\}$ sono insiemi inconsistenti di formule. Ma allora, visto che stiamo supponendo che $\alpha \to \beta \in \Phi$, possiamo usare il lemma precedente per stabilire che Φ é inconsistente.

2.3.1 Esercizi

- Verificare con il metodo delle tabelline di verità o con quello degli alberi proposizionali che gli assiomi del calcolo proposizionale sono tautologie.

- Verificare con il metodo degli alberi proposizionali che $\neg\neg\alpha \to \alpha$ é una tautologia.

- (Teorema di validità) Sia Φ un insieme di formule proposizionali e α una formula proposizionale. Dimostrare per induzione sulla lunghezza della derivazione che se $\Phi \vdash \alpha$ allora $\Phi \models \alpha$, vale a dire che se α é derivabile da Φ allora essa é una conseguenza tautologica di Φ.

- Dimostrare i seguenti teoremi:
  -
  -

3 La logica predicativa

Abbiamo quindi esaurito la questione di ridurre la nozione semantica di conseguenza tautologica ad una nozione puramente sintattica, almeno nel caso che stiamo trattando con insiemi finiti di formule. Sappiamo tuttavia che gli strumenti proposizionali non sono sufficienti per catturare la nozione generale di conseguenza anche se qualora α sia conseguenza tautologica di Φ ne é pure conseguenza. É quindi giunto il momento di rivolgerci allo studio delle formule che si possono scrivere tramite l'uso dell'intero linguaggio, vale a dire di volgere la nostra attenzione a quella che si chiama *logica predicativa*.

La prima osservazione da fare é allora che la via semantica per decidere sulla verità di una formula é divenuta assolutamente impercorribile: se nel caso proposizionali eravamo riusciti a ridurre l'analisi del valore di verità di una formula all'analisi di un numero finito di casi, funzione del numero di formule prime che in tale formula compaiono, tale trucco adesso non funziona più. Infatti anche solo per analizzare semanticamente come una valutazione agisce su una formula ci troviamo facilmemte a dover considerare infiniti casi: si consideri ad esempio il problema di valutare il valore di verità di una formula quantificata universalmente in un universo, quale ad esempio i numeri naturali, con infiniti elementi. Il passaggio dalla semantica alla sintassi, che nel caso proposizionale era un lusso per appagare la pigrizia nel fare i conti, é ora diventato una vera necessità. Sarà quindi necessario trovare un sistema per ottenere, in analogia agli alberi proposizionali, anche *alberi predicativi*, ed eventualmente anche un *calcolo predicativo*.

In questo tentativo ci si scontra però subito con alcuni problemi di carattere notazionale. Il primo di questi é legato alla necessità di introdurre una operazione di sostituzione sintattica: sono infatti apparse le variabili ed una delle caratteristiche che le variabili hanno é proprio quella di poter essere sostituite! Si consideri ad esempio la formula $((\forall v_1)\ P(v_1)) \to P(t)$ dove P é un simbolo predicativo del linguaggio che stiamo considerando e t é un qualsiasi termine; é facile convincersi che si tratta di una formula vera in ogni interpretazione visto che é una implicazione dove l'antecedente é vero solo se ogni elemento dell'universo appartiene all'interpretazione di P e il termine t deve comunque essere interpretato in un qualche elemento dell'universo. Ma ciò che ci interessa in questo momento é che per scrivere tale formula abbiamo compiuto una operazione sintattica di sostituzione: abbiamo sostituito nella formula $P(v_1)$ il termine t al posto della variabile v_1. Dovrebbe essere immediatamente chiaro che la pura sostituzione testuale, anche se funziona nell'esempio appena visto, na fa, in generale, al caso nostro; si consideri ad esempio la formula $(\forall v_1)\ \mathsf{Eq}(v_1, v_2)$: essa é vera solo in una struttura con un solo elemento. Se a questo punto vogliamo sostituire la variabile v_2 proprio con il termine v_1 ed utilizziamo la semplice sostituzione testuale otteniamo $(\forall v_1)\ \mathsf{Eq}(v_1, v_1)$ che, come sappiamo, risulta vera in ogni struttura. Tuttavia é chiaro che non era questo quello che si voleva ottenere operando la sostituzione di v_2 con il termine v_1; dalla sostituzione si voleva chiaramente che la variabile v_1 svolgesse nella nuova formula il ruolo svolto precedentemente dalla variabile v_2, vale a dire, che quel che si voleva era asserire che comunque si interpreti la variabile v_1 la sua interpretazione coincide con quella di un qualunque elemento dell'universo. Quello che non funziona in questa sostituzione testuale é che la variabile v_1 viene *catturata* dal quantificatore una volta che abbiamo operato la sostituzione. Se vogliamo evitare tale eventualità possiamo usare un piccolo trucco: se c'é pericolo di cattura di una variabile operando una sostituzione, cambiamo il nome della variabile quantificata con un nome *nuovo* che non appaia né nella formula dove vogliamo fare la sostituzione né nel termine che vogliamo sostituire; questa operazione é semanticamente giustificata dal fatto che, come abbiamo già osservato, il valore di verità di una formula non dipende dall'interpretazione delle variabili quantificate e quindi il loro nome non é rilevante. Diamo quindi la seguente definizione dell'operazione di sostituzione operando per induzione sulla complessità della formula in cui vogliamo operare la sostituzione.

Definizione 3.1 (Sostituzione di una variabile con un termine) *Sia x una variabile e t un termine.*

- *Sia s un termine. Allora la* sostituzione *della variabile x con il temine t nel termine s é definita induttivamente come segue*
  - *se $s \equiv x$ allora $s[x := t] \equiv t$;*
  - *se $s \equiv y$ allora $s[x := t] \equiv y$, dove stiamo supponendo che y sia una variabile diversa da x;*
  - *se $s \equiv f(t_1, \ldots, t_n)$ allora $s[x := t] \equiv f(t_1[x := t], \ldots, t_n[x := t])$.*
- *Sia α una formula. Allora la* sostituzione *della variabile x con il temine t nella formula α é definita induttivamente come segue*
  - *se $\alpha \equiv P(t_1, \ldots, t_n)$ allora $\alpha[x := t] \equiv P(t_1[x := t], \ldots, t_n[x := t])$;*
  - *se $\alpha \equiv \mathsf{Eq}(t_1, t_2)$ allora $\alpha[x := t] \equiv \mathsf{Eq}(t_1[x := t], t_2[x := t])$;*
  - *se $\alpha \equiv \neg\beta$ allora $\alpha[x := t] \equiv \neg(\beta[x := t])$;*
  - *se $\alpha \equiv \beta \to \gamma$ allora $\alpha[x := t] \equiv \beta[x := t] \to \gamma[x := t]$;*
  - *se $\alpha \equiv (\forall x)\ \beta$ allora $\alpha[x := t] \equiv \alpha$;*
  - *se $\alpha \equiv (\forall y)\ \beta$ allora $\alpha[x := t] \equiv (\forall z)\ \beta[y := z][x := t]$, dove stiamo supponendo che y sia una variabile diversa da x e che la variabile z non appaia né in β né in t.*

Per vedere fino in fondo che la definizione che abbiamo dato dell'algoritmo di sostituzione fa proprio quello che vogliamo é conveniente essere un po' più precisi riguardo alle occorrenze delle variabili nelle formule.

Definizione 3.2 (Occorrenza libera di una variabile in una formula) *Sia α una formula e x una variabile che in tale formula appare. Allora una occorrenza di x in α é libera se*

- *se $\alpha \equiv P(t_1, \ldots, t_n)$;*
- *se $\alpha \equiv \neg\beta$ e tale occorrenza di x é libera in β;*
- *se $\alpha \equiv \beta \to \gamma$ e tale occorrenza di x é libera in β oppure in γ;*
- *se $\alpha \equiv (\forall v_i)\ \beta$ e x é diversa da v_i e l'occorrenza di x considerata é libera in β.*

Se una occorrenza di una variabile in una formula non é libera si dice che é una occorrenza legata della variabile nella formula. Naturalamente una variabile può avere sia occorrenze libere che legate in una formula: ad esempio la variabile v_1 ha nella formula $((\forall v_1)\ \mathsf{Eq}(v_1, 0)) \to \mathsf{Eq}(v_2, v_1)$ una occorrenza legata, la prima, ed una occorrenza libera, la seconda, mentre l'unica occorrenza della variabile v_2 é libera. Diremo ora che una variabile x occorre libera nella formula α se in α almeno una delle occorrenze x é libera. Si può allora dimostrare i seguente lemma.

Lemma 3.3 *La valutazione della formula α dipende solamente dalla valutazione delle variabili che in α occorrono libere, oltre che dalla valutazione delle costanti extralogiche.*

Lasciamo la sua dimostrazione come esercizio.

Possiamo finalmente dimostrare che la sostituzione che abbiamo introdotto nella definizione qui sopra si comporta bene rispetto alle valutazioni, vale a dire che la valutazione della formula $\alpha[x := t]$ dipende dalla valutazione del termine t nello stesso modo in cui la valutazione di α dipendeva dalla valutazione di x.

Teorema 3.4 (Correttezza della sostituzione) *Sia α una formula, x una variabile, t un termine e σ una valutazione. Allora $\alpha[x := t]^\sigma = \alpha^{\sigma(x/\sigma(t))}$.*

La dimostrazione si ottiene per induzione sulla costruzione della formula α ma, come al solito, per risolvere il caso in cui α é una formula atomica ci serve un risultato analogo sui termini.

Lemma 3.5 *Sia s un termine, x una variabile, t un termine e σ una valutazione. Allora $s[x := t]^\sigma = s^{\sigma(x/\sigma(t))}$.*

Dimostrazione. La dimostrazione si ottiene per induzione sulla complessità del termine s. Dobbiamo tuttavia anche considerare i due diversi casi dovuti alla definizione di sostituzione.

- $s \equiv x$. Allora

$$\begin{array}{rcll} s[x := t]^\sigma & \equiv & t^\sigma & \text{definizione di sostituzione} \\ & \equiv & x^{\sigma[x/\sigma(t)]} & \text{definizione di valutazione} \\ & \equiv & s^{\sigma[x/\sigma(t)]} & \text{perchè } s \equiv x \end{array}$$

- $s \equiv y$. Allora

$$\begin{array}{rcll} s[x := t]^\sigma & \equiv & y^\sigma & \text{definizione di sostituzione} \\ & \equiv & y^{\sigma[x/\sigma(t)]} & \text{definizione di valutazione} \\ & \equiv & s^{\sigma[x/\sigma(t)]} & \text{perchè } s \equiv y \end{array}$$

- $s \equiv f(t_1, \ldots, t_n)$. Allora

$$\begin{array}{rcll} s[x := t]^\sigma & \equiv & f(t_1[x := t], \ldots, t_n[x := t])^\sigma & \text{definizione di sostituzione} \\ & \equiv & f_I(t_1[x := t]^\sigma, \ldots, t_n[x := t]^\sigma) & \text{definizione di valutazione} \\ & \equiv & f_I(t_1^{\sigma[x/\sigma(t)]}, \ldots, t_n^{\sigma[x/\sigma(t)]}) & \text{ipotesi induttiva} \\ & \equiv & f(t_1, \ldots, t_n)^{\sigma[x/\sigma(t)]} & \text{definizione di valutazione} \\ & \equiv & s^{\sigma[x/\sigma(t)]} & \text{perchè } s \equiv f(t_1, \ldots, t_n) \end{array}$$

Con questo abbiamo finito la dimostrazione del lemma e siamo pronti ad affrontare la prova del teorema che avevamo lasciato in sospeso.

- $\alpha \equiv P(t_1, \ldots, t_n)$. Allora

$$\begin{array}{rcll} \alpha[x := t]^\sigma & \equiv & P(t_1[x := t], \ldots, t_n[x := t])^\sigma & \text{definizione di sostituzione} \\ & \equiv & \langle t_1[x := t]^\sigma, \ldots, t_n[x := t]^\sigma \rangle \in P_I & \text{definizione di valutazione} \\ & \equiv & \langle t_1^{\sigma[x/\sigma(t)]}, \ldots, t_n^{\sigma[x/\sigma(t)]} \rangle \in P_I & \text{lemma precedente} \\ & \equiv & P(t_1, \ldots, t_n)^{\sigma[x/\sigma(t)]} & \text{definizione di valutazione} \\ & \equiv & \alpha^{\sigma[x/\sigma(t)]} & \text{perchè } \alpha \equiv P(t_1, \ldots, t_n) \end{array}$$

- $\alpha \equiv \neg\beta$ o $\alpha \equiv \beta \rightarrow \gamma$. Questi casi sono abbastanza semplici e vengono lasciati al lettore come esercizio.

- $\alpha \equiv (\forall x)\ \beta$. Allora

$$\begin{array}{rcll} \alpha[x := t]^\sigma & \equiv & ((\forall x)\ \beta)^\sigma & \text{definizione di sostituzione} \\ & \equiv & ((\forall x)\ \beta)^{\sigma[x/\sigma(t)]} & \text{la valutazione di una formu-} \\ & & & \text{la dipende solo dalle varia-} \\ & & & \text{bili che appaiono libere nella} \\ & & & \text{formula ed } x \text{ non appare} \\ & & & \text{libera in } (\forall x)\ \beta \\ & \equiv & \alpha^{\sigma[x/\sigma(t)]} & \text{perchè } \alpha \equiv (\forall x)\ \beta \end{array}$$

- $\alpha \equiv (\forall y)\ \beta$. Allora

$$
\begin{array}{rcll}
\alpha[x := t]^\sigma & \equiv & ((\forall z)\ \beta[y := z][x := t])^\sigma & \text{definizione di sostituzione} \\
& \equiv & \text{per ogni } u \in U & \text{definizione di valutazione} \\
& & \beta[y := z][x := t]^{\sigma[z/u]} & \\
& \equiv & \text{per ogni } u \in U & \text{ipotesi induttiva} \\
& & \beta[y := z]^{\sigma[z/u][x/\sigma[z/u](t)]} & \\
& \equiv & \text{per ogni } u \in U & z \text{ non appare in } t \\
& & \beta[y := z]^{\sigma[z/u][x/\sigma(t)]} & \\
& \equiv & \text{per ogni } u \in U & \text{ipotesi induttiva} \\
& & \beta^{\sigma[z/u][x/\sigma(t)][y/\sigma[z/u][x/\sigma(t)](z)]} & \\
& \equiv & \text{per ogni } u \in U & \sigma[z/u][x/\sigma(t)](z) = u \\
& & \beta^{\sigma[z/u][x/\sigma(t)][y/u]} & \\
& \equiv & \text{per ogni } u \in U & z \text{ non appare in } \beta \\
& & \beta^{\sigma[x/\sigma(t)][y/u]} & \\
& \equiv & ((\forall y)\ \beta)^{\sigma[x/\sigma(t)]} & \text{definizione di valutazione} \\
& \equiv & \alpha^{\sigma[x/\sigma(t)]} & \text{perchè } \alpha \equiv (\forall y)\ \beta
\end{array}
$$

Questo lemma é tutto quello che ci serve per poter introdurre gli alberi di formule predicativi.

3.0.2 Esercizi

1. Verificare che la formula $(\forall v_1)\ \mathsf{Eq}(v_1, v_2)$ vale in una struttura $\mathcal{U}$ se e solo se il suo universo ha un solo elemento mentre la formula $(\forall v_1)\ \mathsf{Eq}(v_1, v_1)$ vale in ogni struttura.
2. Sia t un termine e I una interpretazione basata sulla valutazione σ delle variabili. Dimostrare per induzione sulla costruzione del termine t che la valutazione di t dipende solo dalla valutazione delle variabili che appaiono in t, vale a dire che due valutazioni che coincidono sulle variabili che appaiono in t valutano t nello stesso modo.
3. Utilizzando il risultato dell'esercizio precedente dimostrare che la valutazione della formula α dipende solamente dalla valutazione delle variabili che in α occorrono libere.
4. Siano α e β due formule, x una variabile, t un termine e σ una valutazione delle variabili. Dimostrare che $\neg\alpha[x := t]^\sigma = \neg\alpha^{\sigma(x/\sigma(t))}$ e che $\alpha \to \beta[x := t]^\sigma = \alpha \to \beta^{\sigma(x/\sigma(t))}$.

3.1 Gli alberi predicativi

La definizione degli alberi predicativi di formule ricalca completamente quella già data per gli alberi proposizionali eccetto che ora l'analisi delle formule non si ferma più al caso delle formule prime ma prosegue analizzando anche le formule atomiche e quelle quantificate universalmente. Dobbiamo quindi estendere la definizione di albero proposizionale in modo da considerare questi due casi.

Definizione 3.6 (Albero predicativo di formule) *Un albero predicativo per l'insieme di formule Φ é un albero i cui nodi sono insiemi di formule tali che*

- *Φ é un albero proposizionale per Φ (costituito dalla sola radice)*
- *Se T é un albero proposizionale per Φ, tale che l'insieme di formule Γ sia una foglia e R sia il ramo che unisce Γ alla radice, allora anche l'albero T', ottenuto in uno dei seguenti modi, é un albero proposizionale per Φ:*
  - *se $\neg\neg\alpha$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\alpha\}$;*
  - *se $\alpha \to \beta$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con due nodi, immediati successori di Γ, $\Gamma' \equiv \{\neg\alpha\}$ e $\Gamma'' \equiv \{\beta\}$;*
  - *se $\neg(\alpha \to \beta)$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\alpha, \neg\beta\}$;*
  - *se $(\forall x)\ \alpha$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\alpha[x := t]\}$ dove t é un termine qualsiasi;*
  - *se $\neg(\forall x)\ \alpha$ occorre in uno dei nodi del ramo R e T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\neg\alpha[x := y]\}$ dove y é una variabile che non compare libera in nessuna delle formule che appartengono agli insiemi di formule che formano il ramo R.*

Qualora il linguaggio che stiamo considerando preveda anche l'uso di un predicato di uguaglianza Eq *alle condizioni precedenti vanno anche aggiunte le seguenti:*

- *se T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\mathsf{Eq}(t,t)\}$ dove t é un termine qualsiasi;*
- *se T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\mathsf{Eq}(t_1, s_1) \to (\ldots(\mathsf{Eq}(t_n, s_n) \to \mathsf{Eq}(f(t_1, \ldots, t_n), f(s_1, \ldots, s_n)))\ldots)\}$ dove $t_1, \ldots, t_n$ e $s_1, \ldots, s_n$ sono termini qualsiasi e f é un simbolo per funzione di arietà n;*
- *se T' coincide con T eccetto che il ramo R é stato esteso con il nodo $\Gamma' \equiv \{\mathsf{Eq}(t_1, s_1) \to (\ldots(\mathsf{Eq}(t_n, s_n) \to (P(t_1, \ldots, t_n) \to P(s_1, \ldots, s_n)))\ldots)\}$ dove $t_1, \ldots, t_n$ e $s_1, \ldots, s_n$ sono termini qualsiasi e P é un simbolo predicativo di arietà n.*

Niente altro é un albero predicativo.

Possiamo ora estendere in modo naturale al caso predicativo tutte le definizioni che abbiamo già dato nel caso proposizionale. Otteniamo infatti la definizione di *ramo chiuso* e di *albero di confutazione per un insieme di formule* semplicemente sostituendo alla condizione che un ramo é chiuso se in esso appare una formula prima e la sua negazione la nuova condizione che esso é chiuso se in esso appare una formula atomica e la sua negazione.

Il prossimo passo consiste nel dimostrare che le regole che abbiamo fornito sono corrette Ricordando quel che abbiamo fatto nel caso proposizionale si vede subito che dobbiamo dimostrare che se T é un albero predicativo che contiene un ramo le cui formule sono tutte soddisfacibili e T' é un albero ottenuto da T rispettando le regole che abbiamo fornito allora anche T' contiene un ramo le cui formule sono tutte soddisfacibili. Questa condizione é infatti sufficiente per assicurarci che se un insieme di formule Φ é confutabile allora esso non é soddisfacibile. Per ottenere la prova di questo risultato sarà sufficiente analizzare solamente le nuove condizioni, visto che per le espansioni dovute alla presenza di formule del tipo $\neg\neg\alpha$, $\alpha \to \beta$ e $\neg(\alpha \to \beta)$ continua a valere la dimostrazione già fatta.

- Supponiamo che T' sia stato ottenuto da T aggiungendo al ramo R di formule soddisfacibili il nuovo nodo $\{\alpha[x := t]\}$, con t termine qualsiasi, perchè la formula $(\forall x)\ \alpha$ compariva in R. Sia allora σ la valutazione che rende vere tutte le formule di R, allora $((\forall x)\ \alpha)^\sigma = \top$ e quindi $\alpha^{\sigma[x/u]} = \top$ per ogni elemento u dell'universo della struttira su cui σ é basata. Ma allora dal precedente lemma sulla correttezza dell'operazione di sostituzione noi sappiamo che $\alpha[x := t]^\sigma = \alpha^{\sigma[x/\sigma(t)]}$ e quindi $\alpha[x := t]^\sigma = \top$ visto che $\sigma(t)$ é chiaramente un elemento della struttura.

- Supponiamo che T' sia stato ottenuto da T aggiungendo al ramo R di formule soddisfacibili il nuovo nodo $\{\neg\alpha[x := y]\}$, dove y é una variabile che non appare libera in nessuna delle formule del ramo R, perchè la formula $\neg(\forall x)\ \alpha$ compariva in R. Sia allora σ la valutazione che rende vere tutte le formule in R. Allora $(\neg(\forall x)\ \alpha)^\sigma = \top$ e quindi $((\forall x)\ \alpha)^\sigma = \bot$. Esiste perciò un elemento u^* dell'universo U della struttura su cui σ é basata tale che $\alpha^{\sigma[x/u^*]} = \bot$. Consideriamo ora il valore di verità della formula $\alpha[x := y]$ secondo la valutazione $\sigma[y/u^*]$:

  $$\begin{array}{rcll} \alpha[x := y]^{\sigma[y/u^*]} & = & \alpha^{\sigma[y/u^*][x/\sigma[y/u^*](y)} & \text{per il lemma sulla sostituzione} \\ & = & \alpha^{\sigma[y/u^*][x/u^*]} & \text{perchè } \sigma[y/u^*](y) = u^* \\ & = & \alpha^{\sigma[x/u^*]} & y \text{ non appare libera in } \alpha \\ & = & \bot & \end{array}$$

  Quindi $\neg\alpha[x := y]^{\sigma[y/u^*]} = \top$. D'altra parte tutte le formule che appaiono nel ramo R sono ancora valutate in $\top$ dalla valutazione $\sigma[y/u^*]$ visto che per ipotesi y non appare libera in nessuna delle formule di R e che sappiamo che la valutazione di una formula dipende solo dalla valutazione delle variabili che in essa appaiono libere. Ma allora la valutazione $\sigma[y/u^*]$ rende vere tutte le formule che appaiono nel ramo esteso.

- É immediato verificare le tre formule che si aggiungono al ramo secondo le ultime tre condizioni relative al predicato Eq sono soddisfatte da ogni valutazione e non pongono quendi nessun problema all'estensioe di un ramo.

Abbiamo quindi dimostrato il seguente teorema.

Teorema 3.7 *Sia Φ un insieme soddisfacibile di formule. Allora non esiste un albero di confutazione per Φ.*

Siamo ora nuovamente di fronte, ma nel caso predicativo, al problema della completezza: basteranno le regole che abbiamo introdotto a confutare un qualsiasi insieme di formule qualora esso non sia soddisfacibile?

É chiaro che la situazione si presenta più difficile che nel caso proposizionale. Infatti non possiamo più sfruttare l'idea di usare completamente una formula perchè evidentemente una formula quale $(\forall x)\ \alpha$ non sarà mai completamente usata e darà sempre luogo a nuove espansioni, al variare del termine t da sostituire per ottenere la formula $\alpha[x := t]$. D'altra parte si può comunque sperare che se un insieme di formule risulta confutabile questo dipenda dal fatto che alcune, e comunque un numero finito, di istanze, opportunamente scelte, di una formula rendano la storia che l'insieme racconta non soddisfacibile. Il problema diventa quindi quello di trovare le istanze adatte, vale a dire che il problema diviene quello di trovare una corretta strategia nello sviluppo dell'albero che ci assicuri che se l'insieme di formule Φ non é soddisfacibile prima o poi, ma in un nemeri finito di passi, possiamo accorgercene. Si tratta perciò di sviluppare l'albero usando una stategia di sviluppo che non trascuri nessuna formula che possa servire per chiudere i suoi rami. Dobbiamo quindi riuscire a procurarci una opportuna lista di tutte le formule necessarie in modo da sapere ad ogni passo dello sviluppo con quale formula dobbiamo provare ad espandere l'albero.

Se supponiamo di lavorare con un insieme al più numerabile Φ di formule, possiamo procurarci la lista richiesta nel seguente modo. Prima di tutto costruiamo una lista che contenga *tutte* le formule che il linguaggio *minimo* $\mathcal{L}_\Phi$ per scrivere le formule di Φ ci permette di scrivere. Useremo poi le formule così ottenute per farci guidare nello sviluppo dell'albero per Φ in modo da essere sicuri di non trascurare nessuna formula. Poiché per nostra ipotesi Φ é un insieme numerabile di formule, in $\mathcal{L}_\Phi$ compariranno al più una quantità numerabile di simboli predicativi, una quantità numerabile di simboli funzionali ed una quantità numerabile di costanti; ci saranno poi i soliti connettivi e quantificatori ed una quantità numerabile di variabili. Perció usando il linguaggio $\mathcal{L}_\Phi$ sarà possibile costruire solo una quantità numerabile di formule. Infatti le formule sono solo alcune tra le stringhe di lunghezza finita che posso costruire usando una quantità numerabile di simboli e sono quindi al più tante quante sono i sottoinsiemi finiti dei numeri naturali, cioè sono una quantità numerabile. D'altra parte poichè esiste comunque una infinità numerabile di variabili ci saranno comunque almeno una quantità numerabile di formule visto che, per ipotesi, in un linguaggio almeno un segno predicativo é comunque sempre presente. Ora, dire che esiste una quantità numerabile di formule vuole dire che possiamo costruirne una lista. Tuttavia questa particolare lista non é ancora sufficiente per guidarci nello sviluppo dell'albero. Infatti nello sviluppo dell'albero per un certo insieme di formule Φ può succedere di aver bisogno di usare la stessa formula un qualsiasi numero di volte; ad esempio, può essere necessario usare una formula quantificata universalmente un arbitrario numero di volte scegliendo di volta in volta un termine diverso oppure può succedere che nel momento in cui la lista ci suggerisce di usare una certa formula questa non sia utilizzabile per espandere in modo corretto l'albero mentre sviluppi successivi dei rami dell'albero la rendano usabile in modo legale. Conviene allora modificare la lista ϕ_1, ϕ_2, ϕ_3, ... di formule che abbiamo costruito in modo da costruire una nuova lista in cui ogni formula appaia un numero infinito di formule. Approfittando di una delle stranezze dell'infinito possiamo riuscirci utilizzando ad esempio la nuova lista ϕ_1, ϕ_2, ϕ_1, ϕ_2, ϕ_3, ϕ_1, ϕ_2, ϕ_3, ϕ_4, ... dove evidentemente ogni formula appare un numero infinito di volte.

In modo completamente analogo possiamo costruire una lista numerabile che comprenda tutti i termini che con il linguaggio $\mathcal{L}_\Phi$ si possono scrivere. La useremo per essere sicuri non stiamo trascurando alcun termine t nel momento in cui espanderemo una formula quantificata universalmente $(\forall x)\ \alpha$ aggiungendo un nuovo nodo $\alpha[x := t]$.

Se il linguaggio ha tra i suoi simboli anche il predicato di uguaglianza Eq, costruiremo anche una lista numerabile di tutte le formule del tipo $\mathsf{Eq}(t,t)$ al variare del termine t, $\mathsf{Eq}(t_1, s_1) \to (\dots(\mathsf{Eq}(t_n, s_n) \to \mathsf{Eq}(f(t_1, \dots, t_n), f(s_1, \dots, s_n)))\dots)$, dove t_1, ..., t_n e s_1, ..., s_n sono termini qualsiasi e f é un simbolo per funzione di arietà n, e $\mathsf{Eq}(t_1, s_1) \to (\dots(\mathsf{Eq}(t_n, s_n) \to (P(t_1, \dots, t_n) \to P(s_1, \dots, s_n)))\dots)$ dove t_1, ..., t_n e s_1, ..., s_n sono termini qualsiasi e P é un simbolo predicativo di arietà n. Si tratterà evidentemente di una sottolista della precedente lista di tutte le formule che si possono scrivere usando il linguaggio $\mathcal{L}_\Phi$.

Adesso siamo pronti per costruire un albero che sia il più generale possibile che non trascuri nessuno sviluppo legale ottenibile dall'insieme di formule Φ di partenza. Procediamo dunque induttivamente in questo modo.

- Sia T_0 l'albero costituito dalla sola radice formata dall'insieme di formule Φ.

- Supponiamo di aver costruito l'albero T_n. La costruzione dell'albero T_{n+1} verrà allora effettuata in più passi che dipendono dalla presenza o meno nel linguaggio $\mathcal{L}_\Phi$ del simbolo Eq. Prima di tutto sia T_n^* l'albero definito in funzione della formula ϕ di posto n nella nostra lista di formule nel modo che segue:

    1. se ϕ é una formula del tipo $\neg\neg\alpha$, $\alpha \to \beta$, $\neg(\alpha \to \beta)$ o $\neg(\forall x)\ \alpha$ che appare in T_n allora T_n^* si ottiene allungando tutti i rami che passano per l'insieme di formule che contiene ϕ nel modo opportuno nel rispetto della definizione di albero predicativo;

    2. se ϕ é una formula del tipo $(\forall x)\ \alpha$ che appare in T_n allora T_n^* si ottiene aggiungendo ad ogni ramo che passa per l'insieme di formule che contiene ϕ uno dopo l'altro i nodi $\{\alpha[x := t_0]\}$, ..., $\{\alpha[x := t_n]\}$, dove t_1, ..., t_n sono i primi n termini della lista di termini che ci siamo costruiti;

3. se ϕ é una formula atomica, una negazione di formula atomica oppure proprio non appare in T_n allora non facciamo nulla e poniamo semplicemente $T_n^* \equiv T_n$.

Ora, se $\mathcal{L}_\Phi$ non contiene il simbolo di uguaglianza Eq allora abbiamo finito e poniamo $T_{n+1} \equiv T_n^*$. Altrimenti per ottenere T_{n+1} aggiungiamo ad ogni foglia di T_n^* la formula di posto n nella lista delle formule di che considerano l'uguaglianza che ci eravamo costruiti.

Naturalmente seguendo questa ricetta costruiamo un albero di formule corretto per Φ, per cui se in un qualche passo della costruzione ci troviamo di fronte ad un albero i cui rami sono tutti chiusi significherà che abbiamo ottenuto un albero di confutazione per Φ e quindi che tale insieme di formule non é soddisfacibile come abbiamo già dimostrato nel precedente teorema di correttezza. Il resto di questo paragrafo consisterà quindi nel dimostrare che se l'albero cosí ottenuto non é di confutazione allora esso contiene informazioni sufficienti per dimostrare che l'insieme di formule Φ che compare nella sua radice é soddisfacibile.

É importante sottolineare a questo punto che una volta fatta tale dimostrazione avremo ottenuto un metodo che applicato ad un insieme Φ di formule ci fornisce sicuramente una dimostrazione della sua insoddisfacibilità quando Φ é davvero non soddisfacibile ma che non é in generale in grado di dirci nulla in modo effettivo sulla soddisfacibilità di Φ: infatti in ogni passo noi sappiamo solo se siamo riusciti a chiudere l'albero ma, fintanto che non ci siamo riusciti non sappiamo se esso si chiuderà al prossimo passo, tra cento passi o mai. Come vedremo, questo non é un difetto dovuto al nostro algoritmo particolare per espandere un albero, o al particolare metodo degli alberi di formule che stiamo utilizzando. Infatti qualunque procedura di decisione che cerchi di scoprire se una certo formula é una coseguenza tautologica di un dato insieme di formule é sottoposta ad un analogo vincolo; talvolta questo fatto si esprime dicendo che la logica predicativa é semi-decidibile visto che abbiamo un metodo per scoprire la insoddisfacibilità di un insieme di formule ma non ne abbiamo nessuno per dimostrare la sua soddisfacibilità. Naturalmente per poter dimostrare un risultato cosí generale sarà necessario analizzare meglio le nozioni di metodo, algoritmo e chiarire meglio cosa intendiamo per algoritmo effettivo: é chiaro che nella mente di dio non esistono problemi di decisione per la logica predicativa, ma gli esseri umani hanno qualche vincolo in più.

Naturalmente di fronte a questo risultato teorico non bisogna pensare che un qualunque metodo sia equivalente a qualsiasi altro, nel senso che abbiamo detto che non esiste alcun metodo che possa risolvere completamente il problema della decisione della soddisfacibilità della logica predicativa. Infatti si innestano a quasto punto anche considerazioni sull'efficenza di un metodo rispetto ad un altro che chiaramente bisogna tenere in considerazione qualora si sia interessati a sviluppare un vero dimostratore di teoremi per la logica predicativa. Ad esempio l'algoritmo che abbiamo proposto brilla probabilmente per la sua semplicità espositiva ma non ha sicuramente molti altri pregi per quanto riguarda l'efficenza e non é certamente conveniente fondare su di esso un qualsiasi tentativo d implementazione.

Vediamo comunque di completare la dimostrazione del teorema di completezza. Supponiamo quindi che per quanto avanti andiamo nello sviluppo dell'albero per l'insieme di formule Φ seguendo il metodo che abbiamo descritto non riusciamo mai ad ottenere un albero di confutazione, che resti cioè sempre almeno un ramo R aperto. continuando in tal modo arriveremo a costruire un ramo di lunghezza infinita. Consideriamo allora l'insieme Σ di tutte le formule che stanno su tale ramo. É allora facile verificare che l'insieme di formule Σ gode delle seguenti proprietà:

- per ogni formula atomica π non é possibile che sia π che $\neg\pi$ appartengano a Σ altrimenti il ramo R sarebbe chiuso;
- se $\neg\neg\alpha \in \Sigma$ allora $\alpha \in \Sigma$; infatti la formula α appare infinite volte nella lista delle formule e quindi essa sicuramente appare quale formula sotto analisi in un momento nello sviluppo dell'albero in cui era possibile aggiungere il nodo $\{\alpha\}$ al ramo; per lo stesso motivo
- se $\alpha \to \beta \in \Sigma$ allora $\neg\alpha \in \Sigma$ o $\beta \in \Sigma$
- se $\neg(\alpha \to \beta) \in \Sigma$ allora $\alpha \in \Sigma$ e $\neg\beta \in \Sigma$
- se $\neg(\forall x)\ \alpha \in \Sigma$ allora $\neg\alpha[x := y] \in \Sigma$ per una opportuna scelta della variabile y
- se $(\forall x)\ \alpha \in \Sigma$ allora $\alpha[x := t] \in \Sigma$ per ogni termine t

e se il simbolo di eguaglianza Eq appartiene al linguaggio $\mathcal{L}_\Phi$ anche tutte le formule seguenti appartengono al insieme Σ

- $\mathsf{Eq}(t, t)$ per ogni scelta del termine t,
- $\mathsf{Eq}(t_1, s_1) \to (\ldots(\mathsf{Eq}(t_n, s_n) \to \mathsf{Eq}(f(t_1, \ldots, t_n), f(s_1, \ldots, s_n)))\ldots)$, dove $t_1, \ldots, t_n$ e $s_1, \ldots, s_n$ sono termini qualsiasi e f é un simbolo per funzione di arietà n, e
- $\mathsf{Eq}(t_1, s_1) \to (\ldots(\mathsf{Eq}(t_n, s_n) \to (P(t_1, \ldots, t_n) \to P(s_1, \ldots, s_n)))\ldots)$ dove $t_1, \ldots, t_n$ e $s_1, \ldots, s_n$ sono termini qualsiasi e P é un simbolo predicativo di arietà n compreso naturalmente lo stesso simbolo di uguaglianza Eq.

Il problema della dimostrazione del teorema di completezza si é quindi trasformato in quello di dimostrare che un insieme di formule che gode delle proprietà di cui l'insieme Σ é in realtà soddisfacibile, vale a dire che esiste una struttura ed una valutazione su tale struttura che riesce a rendere vera ogni formula che appartiene a Σ e quindi, in particolare, quelle dell'insieme Φ che appaiono ovviamente tutte in Σ visto che l'insieme, essendo il nodo alla radice dell'albero é comunque un nodo presente in un qualsiasi ramo dell'albero e quindi in particolare nel ramo R le cui formule compaiono in Σ.

Naturalmente il primo problema che dobbiamo affrontare é quello di procurarci la struttura adatta. Vedremo che una volta trovata tale struttura la scelta della valutazione opportuna risulterà ovvia.

Quindi, prima di tutto dobbiamo trovare un insieme poi le funzioni, le relazioni e alcuni elementi particolari su cui interpretare i simboli funzionali, i simboli predicativi e le costanti del linguaggio $\mathcal{L}_\Phi$. Poichè l'unica cosa che abbiamo a disposizione in questo momento sono i termini e le formule del linguaggio $\mathcal{L}_\Phi$ converrà cercare qui l'insieme da utilizzare quale universo della nostra struttura. Perciò il nostro insieme di partenza sarà, in prima approssimazione, costituito da stringhe, per la precisione le stringhe con cui sono scritti i termini di $\mathcal{L}_\Phi$, visto che proprio i termini dovranno poi essere interpretati in elementi di questo universo. D'ora in avanti chiameremo $\mathsf{Str}_{\mathcal{L}_\Phi}$ tale insieme.

Ammesso che si possa realizzare, l'ipotesi più semplice é quindi quella di interpretare un termine t del linguaggio $\mathcal{L}_\Phi$ nella stringa di $\mathsf{Str}_{\mathcal{L}_\Phi}$, che indicheremo con $\mathbf{t}$, con cui tale termine viene scritto.

Se in $\mathcal{L}_\Phi$ non c'é il simbolo di uguaglianza questa scelta é del tutto ragionevole e ci possiamo fermare qui per quanto riguarda la scelta della collezione che costituisce l'universo della struttura che intendiamo costruire. Tuttavia se in $\mathcal{L}_\Phi$ compare il simbolo di uguaglianza Eq e per qualche termine t e s succede che la formula $\mathsf{Eq}(s,t)$ appartiene a Σ allora ci troviamo nella situazione di dover trovare una valitazione che renda vera la formula $\mathsf{Eq}(s,t)$ anche se le due stringhe $\mathbf{t}$ e $\mathbf{s}$ non sono identiche come é invece richiesto dalla nostra definizione della semantica per il simbolo predicativo Eq quando ci limitiamo a considerare un universo fatto proprio con le stringhe. La soluzione consiste allora nel definere una relazione E tra stringhe di $\mathsf{Str}_{\mathcal{L}_\Phi}$ che metta in relazione le stringhe $\mathbf{t}$ e $\mathbf{s}$ esattamente nel caso in cui il predicato $\mathsf{Eq}(s,t)$ appartenga all'insieme di formule Σ. Se poi risulterà che la relazione E é una relazione di equivalenza potremo allora interpretare i termini t e s non direttamente nelle stringhe $\mathbf{t}$ e $\mathbf{s}$ con cui tali termini vengono scritti ma piuttosto nelle classi di equivalenza

$$[\mathbf{s}]_E \equiv \{\mathbf{u} |\ \mathbf{u}\mathsf{E}\mathbf{s}\}$$

e

$$[\mathbf{t}]_E \equiv \{\mathbf{u} |\ \mathbf{u}\mathsf{E}\mathbf{t}\}$$

che la relazione di equivalenza E induce sull'insieme $\mathsf{Str}_{\mathcal{L}_\Phi}$ delle stringhe per termini. É chiaro ora che se l'insieme Σ *pensa* che i termini s e t sono uguali, vale a dire se $\mathsf{Eq}(s,t) \in \Sigma$, allora $[\mathbf{s}]_E = [\mathbf{t}]_E$ e quindi s e t vengono interpretati nello stesso elemento dell'insieme quoziente.

Lemma 3.8 *La relazione* E *tra stringhe in* $\mathsf{Str}_{\mathcal{L}_\Phi}$ *é una relazione di equivalenza.*

Dimostrazione Dobbiamo dimostrare che E é una relazione *riflessiva*, *simmetrica* e *transitiva*.

- (riflessività) Visto che per ipotesi, per ogni termine t, $\mathsf{Eq}(t,t) \in \Sigma$ ne segue immediatamente che E é una relazione riflessiva.

- (simmetria) Supponiamo che **s**E**t**. Allora ne segue che $\mathsf{Eq}(s,t) \in \Sigma$. Visto che Eq é un predicato atomico, dall'ultima condizione sull'insieme di formule Σ otteniamo che $\mathsf{Eq}(s,t) \rightarrow (\mathsf{Eq}(s,s) \rightarrow (\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(t,s))) \in \Sigma$. Quindi, visto che abbisamo una implicazione che appartiene a Σ ne segue che $\neg\mathsf{Eq}(s,t) \in \Sigma$ o $\mathsf{Eq}(s,s) \rightarrow (\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(t,s)) \in \Sigma$. Ma noi sappiamo che $\mathsf{Eq}(s,t) \in \Sigma$ e quindi, per la prima delle condizioni sull'insieme Σ, non può essere che $\neg\mathsf{Eq}(s,t) \in \Sigma$. Quindi abbiamo dimostrato che $\mathsf{Eq}(s,s) \rightarrow (\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(t,s)) \in \Sigma$. Ma ora con analogo modo di ragionare si deduce che $\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(t,s) \in \Sigma$, visto che sicuramente $\mathsf{Eq}(s,s) \in \Sigma$, e quindi, per lo stesso motivo, che $\mathsf{Eq}(t,s) \in \Sigma$ da cui segue immediatamente che **t**E**s**.

- (transitività) Supponiamo che **rEs** e **s**E**t**. Allora per la simmetria di E, che abbiamo appena dimostrato, vale anche che **sEr**. Quindi, per la definzione di E, sappiamo che $\mathsf{Eq}(s,r) \in \Sigma$ e $\mathsf{Eq}(s,t) \in \Sigma$. Analogamente a quanto abbiamo fatto nella prova della simmetria, possiamo ora notare che $\mathsf{Eq}(s,r) \rightarrow (\mathsf{Eq}(s,t) \rightarrow (\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(r,t))) \in \Sigma$ e quindi, con un ragionamento simile, otteniamo dapprima che $\mathsf{Eq}(s,t) \rightarrow (\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(r,t)) \in \Sigma$, in quanto $\mathsf{Eq}(s,r) \in \Sigma$, poi che $\mathsf{Eq}(s,s) \rightarrow \mathsf{Eq}(r,t) \in \Sigma$, in quanto $\mathsf{Eq}(s,t) \in \Sigma$ ed infine che $\mathsf{Eq}(r,t) \in \Sigma$ in quanto $\mathsf{Eq}(s,s) \in \Sigma$. Ora naturalmente possiamo concludere **rEt** utilizzando la definzione di E.

Con questa dimostrazione abbiamo quindi determinato quale deva essere l'universo della nostra struttura anche nel caso in cui il simbolo predicativo Eq appaia tra i simboli del linguaggio $\mathcal{L}_\Phi$. D'ora in avanti, per evitare di dover distingure tra il caso in cui $\mathcal{L}_\Phi$ contenga il simbolo di uguaglianza e quello in cui esso non vi sia, svilupperemo le dimostrazioni solamente per il primo caso, visto che esso é il più complesso; per ottenere le dimostrazioni per il caso in cui tale simbolo predicativo non dovesse apparire in $\mathcal{L}_\Phi$ basterà cancellare tutte le occorrenze delle classi di equivalenza e saltare le prove che sono dovute alla loro presenza.

Adesso che abbiamo definito quale é l'universo della struttura che vogliamo costruire possiamo vedere dove dobbiamo interpretare i simboli funzionali che appaiono nel linguaggio $\mathcal{L}_\Phi$. L'ipotesi più semplice é, di nuovo, quella di definire una funzione $\underline{f}$, di arietà n, che operi sugli elementi dell'insieme $\mathsf{Str}_{\mathcal{L}_\Phi}$ in corrispondenza di ogni simbolo funzionale di arietà n che appare in $\mathcal{L}_\Phi$ su cui poter poi interpretare il simbolo f stesso, vale a dire defiinire l'interpretazione I ponendo $f_I \equiv \underline{f}$. Il primo modo che viene in mente é naturalmente quello di porre

$$\underline{f}([\mathbf{t_1}]_E, \ldots, [\mathbf{t_n}]_E) \equiv [\mathbf{f}(\mathbf{t_1}, \ldots, \mathbf{t_n})]_E$$

che, come vedremo, é proprio il modo che funziona. Tuttavia sorge subito il problema di dimostrare che si tratta di una buona definizione, vale a dire che essa non dipende dalla scelta dei rappresentanti delle classi di equivalenza. Dobbiamo cioè dimostrare che la relazione di equivalenza E é in realtà una congruenza sulle funzioni così definite, vale a dire che ci serve il seguente lemma.

Lemma 3.9 *Se $t_1,\ldots,t_n$ e $s_1,\ldots,s_n$ sono termini del linguaggio $\mathcal{L}_\Phi$ tali che, per ogni $1 \leq i \leq n$, $[\mathbf{t_i}]_E = [\mathbf{s_i}]_E$ e f é un simbolo funzionale di arietà n allora $[\mathbf{f}(\mathbf{t_1},\ldots,\mathbf{t_n})]_E = [\mathbf{f}(\mathbf{s_1},\ldots,\mathbf{s_n})]_E$.*

Dimostrazione. Supponiamo che per ogni $1 \leq i \leq n$, $[\mathbf{t_i}]_E = [\mathbf{s_i}]_E$. Allora $\mathsf{Eq}(t_i, s_i)$. Ma, per ipotesi,

$$\mathsf{Eq}(t_1, s_1) \rightarrow (\ldots(\mathsf{Eq}(t_n, s_n) \rightarrow \mathsf{Eq}(f(t_1,\ldots,t_n), f(s_1,\ldots,s_n)))\ldots) \in \Sigma$$

e quindi é facile vedere che $\mathsf{Eq}(f(t_1,\ldots,t_n), f(s_1,\ldots,s_n)) \in \Sigma$. Ma naturalmente questo significa proprio che $[\mathbf{f}(\mathbf{t_1},\ldots,\mathbf{t_n})]_E = [\mathbf{f}(\mathbf{s_1},\ldots,\mathbf{s_n})]_E$.

Possiamo allora verificare che i termini di $\mathcal{L}_\Phi$ sono davvero interpretati in quella classe di equivalenza di $\mathsf{Str}_{\mathcal{L}_\Phi}$ che ha esattamente quel termine come rappresentante. Prima di poter verificare questo risultato dobbiamo però definire l'interpretazione delle variabili e delle costanti; naturalmente c'é una sola scelta possibile se vogliamo sperare di ottenere il risultato desiderato

$$\begin{array}{rcl} \sigma(x) & \equiv & [\mathbf{x}]_E \\ c_I & \equiv & [\mathbf{c}]_E \end{array}$$

Adesso possiamo finalmente dimostrare il seguente lemma.

Lemma 3.10 (Interpretazione dei termini) *Sia t un termine del linguaggio $\mathcal{L}$. Allora*

$$t^\sigma = [\mathbf{t}]_E$$

Dimostrazione. Dopo che abbiamo preparato tutto con lo scopo di provare questo lemma la dimostrazione si ottiene facilmente per induzione sulla complessità del termine t. Infatti

$$\begin{array}{llcll}
(t \equiv x) & t^\sigma & = & \sigma(x) & \text{per definizione} \\
 & & = & [\mathbf{x}]_E & \\
 & & = & [\mathbf{t}]_E & \\
 & & & & \\
(t \equiv c) & t^\sigma & = & c_I & \text{per definizione} \\
 & & = & [\mathbf{c}]_E & \\
 & & = & [\mathbf{t}]_E & \\
 & & & & \\
(t \equiv f(t_1,\ldots,t_n)) & t^\sigma & = & f(t_1,\ldots,t_n)^\sigma & \text{per definizione} \\
 & & = & f_I(t_1^\sigma,\ldots,t_n^\sigma) & \text{ip. induttiva} \\
 & & = & \underline{f}([\mathbf{t_1}]_E,\ldots,[\mathbf{t_n}]_E) & \text{per definizione} \\
 & & = & [\mathbf{f}(\mathbf{t_1},\ldots,\mathbf{t_n})]_E & \\
 & & = & [\mathbf{t}]_E &
\end{array}$$

Dopo aver completamente risolto la questione della interpretazione dei termini possiamo passare alla interpretazione delle formule. Il primo passo consiste naturalmente nel definire le relazioni su cui interpretare i simboli predicativi del linguaggio $\mathcal{L}_\Phi$. Ci faremo naturalmente guidare dal linguaggio stesso, vale a dire che per ogni simbolo predicativo P di arietà n definiremo una relazione $\underline{P}$ ad n posti e poi porremo $P_I \equiv \underline{P}$. Poichè vogliamo che le formule che appartengono a Σ vengano interpretate in $\top$ la soluzione più semplice é quella di porre

$$\langle [\mathbf{t_1}]_E, \ldots, [\mathbf{t_n}]_E \rangle \in \underline{P} \text{ se e solo se } P(t_1, \ldots, t_n) \in \Sigma$$

Come si vede la definzione fa uso dei rappresentanti delle classi di equivalenza e bisogna quindi dimostrare che essa non dipende da essi.

Lemma 3.11 *Se $t_1,\ldots,t_n$ e $s_1,\ldots,s_n$ sono termini del linguaggio $\mathcal{L}_\Phi$ tali che, per ogni $1 \leq i \leq n$, $[\mathbf{t_i}]_E = [\mathbf{s_i}]_E$ e P é un simbolo predicativo di arietà n allora $P(t_1, \ldots, t_n) \in \Sigma$ se e solo se $P(s_1, \ldots, s_n) \in \Sigma$.*

Dimostrazione. La dimostrazione é molto simile a quella che abbiamo fornito sopra per far vedere che abbiamo dato una buona definizione delle funzioni. Infatti, le formule

$$\mathsf{Eq}(t_1, s_1) \to (\ldots (\mathsf{Eq}(t_n, s_n) \to (P(t_1, \ldots, t_n) \to P(s_1, \ldots, s_n))) \ldots)$$

e

$$\mathsf{Eq}(s_1, t_1) \to (\ldots (\mathsf{Eq}(s_n, t_n) \to (P(s_1, \ldots, s_n) \to P(t_1, \ldots, t_n))) \ldots)$$

appartengono entrambe all'insieme Σ e quindi la condizione che, per ogni $1 \leq i \leq n$, $[\mathbf{t_i}]_E = [\mathbf{s_i}]_E$ implica sia che $P(t_1, \ldots, t_n) \to P(s_1, \ldots, s_n) \in \Sigma$ e che $P(s_1, \ldots, s_n) \to P(t_1, \ldots, t_n) \in \Sigma$ che forniscono immediatamente il risultato richiesto.

Possiamo finalmente finire la nostra dimostrazione che il metodo degli alberi predicativi é completo.

Teorema 3.12 (Completezza del calcolo degli alberi predicativi) *Sia α una formula appartenente all'insieme Σ. Allora $\alpha^\sigma = \top$.*

Dimostrazione. La dimostrazione si ottiene naturalmente per induzione sulla costruzione della formula α. Tuttavia per poter superare agevolmente il caso in cui α sia una formula del tipo $\neg\beta$ conviene dimostrare un risultato apparentemente più forte. Infatti dimostreremo che

1. Se $\alpha \in \Sigma$ allora $\alpha^\sigma = \top$
2. Se $\neg\alpha \in \Sigma$ allora $\alpha^\sigma = \bot$

Analiziamo quindi i vari casi:

- $(\alpha \equiv P(t_1, \ldots, t_n))$ Allora

$$\begin{array}{rcll}
P(t_1, \ldots, t_n) \in \Sigma & \Rightarrow & \langle [\mathbf{t_1}]_E, \ldots, [\mathbf{t_n}]_E \rangle \in \underline{P} & \text{def. } \sigma \\
& \Rightarrow & \langle t_1^\sigma, \ldots, t_n^\sigma \rangle \in \underline{P} & \text{lemma 3.10} \\
& \Rightarrow & P(t_1, \ldots, t_n)^\sigma = \top & \text{def. semantica}
\end{array}$$

$$\begin{array}{rcll}
\neg P(t_1, \ldots, t_n) \in \Sigma & \Rightarrow & P(t_1, \ldots, t_n) \notin \Sigma & \text{condizione su } \Sigma \\
& \Rightarrow & \langle [\mathbf{t_1}]_E, \ldots, [\mathbf{t_n}]_E \rangle \notin \underline{P} & \text{def. } \sigma \\
& \Rightarrow & \langle t_1^\sigma, \ldots, t_n^\sigma \rangle \notin \underline{P} & \text{lemma 3.10} \\
& \Rightarrow & P(t_1, \ldots, t_n)^\sigma = \bot & \text{def. semantica}
\end{array}$$

- ($\alpha \equiv \mathsf{Eq}(s,t)$) Allora

$$\begin{array}{rcll}
\mathsf{Eq}(s,t) \in \Sigma & \Rightarrow & [\mathbf{s}]_E = [\mathbf{t}]_E & \text{def. relazione } E \\
& \Rightarrow & s^\sigma = t^\sigma & \text{lemma 3.10} \\
& \Rightarrow & \mathsf{Eq}(s,t)^\sigma = \top & \text{def. semantica}
\end{array}$$

$$\begin{array}{rcll}
\neg\mathsf{Eq}(s,t) \in \Sigma & \Rightarrow & \mathsf{Eq}(s,t) \notin \Sigma & \text{condizione su } \Sigma \\
& \Rightarrow & [\mathbf{s}]_E \neq [\mathbf{t}]_E & \text{def. relazione } E \\
& \Rightarrow & s^\sigma \neq t^\sigma & \text{lemma 3.10} \\
& \Rightarrow & \mathsf{Eq}(s,t)^\sigma = \bot & \text{def. semantica}
\end{array}$$

- ($\alpha \equiv \neg\beta$) Allora

$$\begin{array}{rcll}
\neg\beta \in \Sigma & \Rightarrow & \beta^\sigma = \bot & \text{ip. induttiva} \\
& \Rightarrow & \neg\beta^\sigma = \top & \text{def. semantica} \\
& \Rightarrow & \alpha^\sigma = \top &
\end{array}$$

$$\begin{array}{rcll}
\neg\neg\beta \in \Sigma & \Rightarrow & \beta \in \Sigma & \text{condizione su } \Sigma \\
& \Rightarrow & \beta^\sigma = \top & \text{ip. induttiva} \\
& \Rightarrow & \neg\beta^\sigma = \bot & \text{def. semantica} \\
& \Rightarrow & \alpha^\sigma = \bot &
\end{array}$$

- ($\alpha \equiv \beta \to \gamma$) Allora

$$\begin{array}{rcll}
\beta \to \gamma \in \Sigma & \Rightarrow & \neg\beta \in \Sigma \text{ o } \gamma \in \Sigma & \text{condizione su } \Sigma \\
& \Rightarrow & \beta^\sigma = \bot o \gamma^\sigma = \top & \text{ip. induttiva} \\
& \Rightarrow & (\beta \to \gamma)^\sigma = \top & \text{def. semantica} \\
& \Rightarrow & \alpha^\sigma = \top &
\end{array}$$

$$\begin{array}{rcll}
\neg(\beta \to \gamma) \in \Sigma & \Rightarrow & \beta \in \Sigma \text{ e } \neg\gamma \in \Sigma & \text{condizione su } \Sigma \\
& \Rightarrow & \beta^\sigma = \top \text{ e } \gamma^\sigma = \bot & \text{ip. induttiva} \\
& \Rightarrow & (\beta \to \gamma)^\sigma = \bot & \text{def. semantica} \\
& \Rightarrow & \alpha^\sigma = \bot &
\end{array}$$

- $(\alpha \equiv (\forall x)\ \beta)$ Allora

$$
\begin{array}{rcll}
(\forall x)\ \beta \in \Sigma & \Rightarrow & \beta[x := t] \in \Sigma \text{ per ogni termine } t & \text{condizione su } \Sigma \\
& \Rightarrow & \beta[x := t]^\sigma = \top \text{ per ogni termine } t & \text{ip. induttiva} \\
& \Rightarrow & \beta^{\sigma(x/\sigma(t))} = \top \text{ per ogni termine } t & \text{teorema 3.4} \\
& \Rightarrow & \beta^{\sigma(x/[\mathbf{t}]_E)} = \top \text{ per ogni termine } t & \text{lemma 3.10} \\
& \Rightarrow & \beta^{\sigma(x/u)} = \top \text{ per ogni } u \in \mathsf{Str}_{\mathcal{L}_\Phi} & \text{costruzione di } \mathsf{Str}_{\mathcal{L}_\Phi} \\
& \Rightarrow & ((\forall x)\ \beta)^\sigma = \top & \text{def. semantica} \\
& \Rightarrow & \alpha^\sigma = \top &
\end{array}
$$

$$
\begin{array}{rcll}
\neg((\forall x)\ \beta) \in \Sigma & \Rightarrow & \neg\beta[x := y] \in \Sigma \text{ per } y \text{ opportuna} & \text{condizione su } \Sigma \\
& \Rightarrow & \beta[x := y]^\sigma = \bot \text{ per } y \text{ opportuna} & \text{ip. induttiva} \\
& \Rightarrow & \beta^{\sigma(x/\sigma(y))} = \bot \text{ per } y \text{ opportuna} & \text{teorema 3.4} \\
& \Rightarrow & \beta^{\sigma(x/[\mathbf{y}]_E)} = \bot \text{ per } y \text{ opportuna} & \text{lemma 3.10} \\
& \Rightarrow & \beta^{\sigma(x/u)} = \bot \text{ per un } u \in \mathsf{Str}_{\mathcal{L}_\Phi} & \text{costruzione di } \mathsf{Str}_{\mathcal{L}_\Phi} \\
& \Rightarrow & ((\forall x)\ \beta)^\sigma = \bot & \text{def. semantica} \\
& \Rightarrow & \alpha^\sigma = \top &
\end{array}
$$

Con questo termina la dimostrazione del teorema di completezza. Vale la pena di notare che non abbiamo solo costruito una struttura che soddisfa tutte le formule di Σ, e quindi tutte le formule dell'insieme Φ di partenza per il quale non eravamo riusciti a costruire un albero di confutazione, ma che la struttura che lo siddisfa é addirettura numerabile.

3.1.1 Esercizi

1. Determinare le condizioni per estendere un albero predicativo nel caso di un linguaggio che usi anche il quantificatore esistenziale e verificare che esse mantengono la soddisfacibilità.

2. Verificare che le seguenti formule sono soddisfatte da ogni valutazione

   - $\mathsf{Eq}(t, t)$, dove t é un arbitrario termine del linguaggio;
   - $\mathsf{Eq}(t_1, s_1) \rightarrow (\ldots(\mathsf{Eq}(t_n, s_n) \rightarrow \mathsf{Eq}(f(t_1, \ldots, t_n), f(s_1, \ldots, s_n)))\ldots)$, dove $t_1, \ldots t_n$ e $s_1, \ldots s_n$ sono arbitrari termini del linguaggio e f é un simbolo funzionale di arietà n;
   - $\mathsf{Eq}(t_1, s_1) \rightarrow (\ldots(\mathsf{Eq}(t_n, s_n) \rightarrow (P(t_1, \ldots, t_n) \rightarrow P(s_1, \ldots, s_n)))\ldots)$, dove $t_1, \ldots t_n$ e $s_1, \ldots s_n$ sono arbitrari termini del linguaggio e P é un simbolo predicativo di arietà n.

3. (Difficile) Dimostrare che gli insiemi finiti dei numeri naturali sono una quantità numerabile.

4. (Difficile) Dato un linguaggio con una quantità numerabile di simboli extralogici, descrivere una procedura per elencare tutte le sue formule.

5. Si supponga di essere il proprietario di un albergo con un numero numerabile di stanze. Una sera, in occasione di un congresso, arrivano una quantità numerabile di ospiti che dovranno essere sistemati nelle varie stanze. Naturalmente potete facilmente soddisfare tale richiesta. Tuttavia, qualche tempo dopo per ciascuno dei congressisti arrivano i familiari. Supponendo che ogni congressista abbia una famiglia con una quantità numerabile di persone , come trovare una stanza per ciascuno di loro?

3.2 Il calcolo predicativo

???

Tutto da fare, ma ne vale la pena? Non mi pare ci siano speranze di farlo durante il corso

???

3.2.1 Esercizi

4 Limiti espressivi del linguaggio

Il risultato fin qui raggiunto con il teorema di completezza é estremamente interessante. Infatti abbiamo fatto vedere che il sistema degli alberi predicativi permette di dimostrare la insoddisfacibilità di un insieme di formule ogniqualvolta tale insieme sia davvero insoddisfacibile. Tuttavia proprio tale risultato positivo dovrebbe fare sorgere qualche sospetto: é mai possibile che si possa davvero dimostrare tutto quel che é vero? In realtà il risultato é limitato in almeno due direzioni: quello che abbiamo dimostrato é infatti che é dimostrabile a partire dall'insime di formule Φ tutto quello che si riesce ad *esprimere* utilizzando il linguaggio del primo ordine e che é vero in *tutte* le strutture che soddisfano Φ. É quindi essenziale cercare di capire cosa si può esprimere con il linguaggio che abbiamo a disposizione e quanto possiamo restringere la classe dei modelli che soddisfano un certo insieme di formule in modo da parlare *precisamente* delle strutture che ci interessano.

Teorema 4.1 (Teorema di Compattezza) *Un insieme Φ di formule é soddisfacibile se e solo se ogni sottoinsieme finito di Φ é soddisfacibile.*

Dimostrazione. Un verso del teorema é completamente banale: sia Φ in insieme soddisfacibile di formule e sia Φ_0 un suo sottoinsieme qualsiasi (in particolare finito); é allora ovvio che ogni struttura che rende vere tutte le formule in Φ rende vere anche tutte le formule in Φ_0.

Per dimostrare l'altro verso faciamo uso della dimostrazione del teorema di completezza. Supponiamo infatti che l'insieme di formule Φ sia insoddisfacibile Allora esiste un albero predicativo chiuso per Φ. Naturalmente tale albero é finito e quindi solo una quantità finita di formule di Φ é stata utilizzata durante la costruzione dell'albero. Ma allora esiste un sottoinsieme finito di Φ che non é soddisfacibile, vale a dire quello delle formule usate per chiudere l'albero per Φ.

Naturalmente il risultato stabilito nel teorema di completezza é interessante solo quando l'insieme di formule Φ é infinito. Tuttavia in questo caso esso non sembra poi un gran risultato. Dice infatti che per vedere se un certo insieme infinito di formule é soddisfacibile ci basta andare a vedere se sono soddisfacibili tutti i suoi infiniti sottoinsieme finiti: sembra che verificare tale evenienza sia ancora più difficile che vedere direttamente se l'insieme di partenza non é soddisfacibile. In realtà le cose non sono cosí brutte perchè spesso basta la stessa struttura per soddisfare tutti i sottoinsiemi finiti di formule. Tuttavia le conseguenze possono essere inaspettate.

4.1 Quel che non si può dire

In questa sezione faremo vedere che limitandoci all'uso di un linguaggio del primo ordine non riusciamo ad esprimere concetti che sono d'uso comune in matematica. Iniziamo con la nozione di insieme finito. Dovrebbe essere per tutti ovvio cosa si deve intendere per insieme finito. Tuttavia non appena si tenta di esprimere tale nozione in modo formale é chiaro che cominciano i problemi. Vediamo infatti come potremo procedere. Innanzitutto é facile dire formalmente che un insieme ha un certo numero n di elementi. Infatti possiamo facilmente dire che un insieme ha almeno n elementi tramite la formula

$$\begin{array}{rcl}\mathsf{Almeno}_n & \equiv & (\exists x_1)(\exists x_2)\ldots(\exists x_n)\ \neg\mathsf{Eq}(x_1,x_2)\wedge\neg\mathsf{Eq}(x_1,x_3)\wedge\ldots\wedge\neg\mathsf{Eq}(x_1,x_n)\\ & & \hfill \wedge\neg\mathsf{Eq}(x_2,x_3)\wedge\ldots\wedge\neg\mathsf{Eq}(x_2,x_n)\\ & & \hfill \ldots\wedge\neg\mathsf{Eq}(x_{n-1},x_n)\end{array}$$

In modo completamente analogo possiamo anche esprimere la proprietà che una struttura abbia non più di n elementi tramite la formula

$$\begin{array}{rcl}\mathsf{Alpiu}_n & \equiv & (\forall x_1)(\forall x_2)\ldots(\forall x_n)(\forall x_{n+1})\ \mathsf{Eq}(x_1,x_2)\vee\mathsf{Eq}(x_1,x_3)\vee\ldots\vee\mathsf{Eq}(x_1,x_{n+1})\\ & & \hfill \vee\mathsf{Eq}(x_2,x_3)\vee\ldots\vee\mathsf{Eq}(x_2,x_{n+1})\\ & & \hfill \ldots\vee\mathsf{Eq}(x_n,x_{n+1})\end{array}$$

É allora ovvio che possiamo caratterizzare le strutture che hanno esattamente n elementi tramite la formula

$$\mathsf{Elem}_n \equiv \mathsf{Almeno}_n\ \wedge\ \mathsf{Alpiu}_n$$

Tornando al nostro problema di caratterizzare le strutture con un numero finito di elementi potremo ora pensare di usare la formula (?)

$$\mathsf{Finito} \equiv \mathsf{Elem}_1 \vee \mathsf{Elem}_2 \vee \mathsf{Elem}_3 \ldots$$

Tuttavia é chiaro che questo metodo non funziona! Infatti mentre i puntini "…" che abbiamo utilizzato per scrivere le formule Almeno_n e Alpiu_n sono il risultato della pigrizia nel senso che dato un n qualsiasi, e abbastanza carta, possiamo comunque fare sparire tali puntini e scrivere completamente una formula, questa soluzione non potrà assolutamente funzionare nel caso della formula Finito in quanto i puntini che abbiamo utilizzato in questo caso sono ineliminabili. Similmente non funziona molto meglio l'idea di scrivere

$$\mathsf{Finito} \equiv (\exists n \in \omega)\ \mathsf{Elem}_n$$

Infatti in questo caso abbiamo addirittura peggiorato le cose visto che il quantificatore esistenziale in questa formula vuole operare sui numeri naturali e quindi ci servirebbe una teoria dei numeri naturali; ma anche nel caso possedessimo tale teoria ci sarebbe sempre il problema di scrivere la generica formula Elem_n: in questo caso infatti non si tratta più di scriverla per un n dato ma per un n variabile sui numeri naturali. Le cose sono quindi più difficili del previsto: si tratta ora di capire se il problema sta nella nostra incapacità di trovare la formula adatta o se tale formula proprio non c'é. Per risolvere tale problema possiamo farci aiutare dal teorema di compattezza.

L'essere finito non si può esprimere ???

La più bella teoria possibile dei numeri naturali descrive strutture che contengono elementi che non sono numeri naturali ???

4.1.1 Esercizi

1. Verificare che la formula tramite la formula

$$\begin{aligned}\mathsf{Almeno}_n \quad \equiv \quad (\exists x_1)(\exists x_2)\ldots(\exists x_n)\ \neg\mathsf{Eq}(x_1,x_2) \wedge \neg\mathsf{Eq}(x_1,x_3) \wedge \ldots \wedge \neg\mathsf{Eq}(x_1,x_n) \\ \wedge\neg\mathsf{Eq}(x_2,x_3) \wedge \ldots \wedge \neg\mathsf{Eq}(x_2,x_n) \\ \ldots \wedge \neg\mathsf{Eq}(x_{n-1},x_n)\end{aligned}$$

   é vera in tutte e sole le strutture che hanno almeno n elementi.

2. Verificare che la formula

$$\begin{aligned}\mathsf{Alpiu}_n \quad \equiv \quad (\forall x_1)(\forall x_2)\ldots(\forall x_n)(\forall x_{n+1})\ \mathsf{Eq}(x_1,x_2) \vee \mathsf{Eq}(x_1,x_3) \vee \ldots \vee \mathsf{Eq}(x_1,x_{n+1}) \\ \vee\mathsf{Eq}(x_2,x_3) \vee \ldots \vee \mathsf{Eq}(x_2,x_{n+1}) \\ \ldots \vee \mathsf{Eq}(x_n,x_{n+1})\end{aligned}$$

   vale in tutte e sole le strutture che hanno al più n elementi.