UNIVERSITÀ DEGLI STUDI DI MILANO

Dipartimento di Scienze dell'Informazione

Dispensa ad uso del corso di Logica Matematica

Introduzione alla Logica Elementare

Osservazione. Questi appunti sono destinati ad un uso didattico in corsi di base per studenti delle lauree triennali di classe Informatica. Rappresentano una versione rivista e semplificata della dispensa in formato elettronico *Corso Propedeutico di Logica* (a cura di M. Franchella, S. Ghilardi, L. Sacchetti) utilizzata fino all'anno accademico 2005-06.[1] A tale dispensa si rimanda per le dimostrazioni dei teoremi di validità e completezza (non obbligatorie per la preparazione dell'esame). Questi appunti trattano *esclusivamente la parte relativa ai tableaux per la logica elementare* e contengono già una sufficiente quantità di esercizi (svolti e non svolti); qualora li ritenesse insufficienti, lo studente potrà ritirare ulteriore materiale manoscritto presso la libreria CLUED di via Celoria (tale materiale manoscritto sarà pronto a fine corso).

La presente versione di questa dispensa porta la data data del **5 marzo 2007**.

[1] Gli studenti che intendano continuare a prepararsi su detta dispensa *possono liberamente farlo.* In particolare, *non c'è nessun obbligo* di utilizzare all'esame il calcolo presentato in questa dispensa a preferenza del calcolo dei sequenti presentato nella vecchia dispensa.

Indice

1 Preliminari insiemistici

Riassumiamo brevemente alcune nozioni di teoria ingenua degli insiemi, che dovrebbero essere note da corsi di base di materie matematiche.[2]

La nozione di **insieme** viene data per intuitiva e inanalizzata (in un corso specifico di teoria degli insiemi si vedrà eventualmente qualche sistema assiomatico che la tratta in maniera più rigorosa). Similmente, diamo per intuitiva la nozione di **funzione** $f : X \longrightarrow Y$ fra gli insiemi X e Y: la f viene genericamente definita come una 'legge' che ad ogni elemento di X (detto insieme dominio) fa corrispondere uno ed un solo elemento di Y (detto insieme codominio). Raddoppiare un numero per esempio è una funzione dall'insieme $\mathbb{N}$ dei numeri naturali in sè. Considerare la madre, è una funzione dall'insieme degli esseri umani nell'insieme degli esseri umani di sesso femminile. Invece, considerare i figli non è una funzione dall'insieme degli esseri umani verso l'insieme degli esseri umani per ben due motivi: non tutti gli esseri umani hanno figli e non tutti ne hanno uno solo, sicchè viene violato il tratto distintivo della nozione di funzione che, ripetiamo, consiste nell'essere una corrispondenza sempre definita e definita in modo univoco.

La nozione di **prodotto cartesiano** di n-insiemi $X_1, \dots, X_n$ è la seguente: il prodotto cartesiano, che scriviamo con

$$X_1 \times \cdots \times X_n$$

è l'insieme delle liste $\langle a_1, \dots, a_n \rangle$ di n elementi (dette n-ple) di elementi appartenenti, rispettivamente, ad $X_1, X_2, \dots, X_n$. Così ad esempio, l'insieme dei vestiti interi spezzati è il prodotto cartesiano dell'insieme delle giacche e dell'insieme dei pantaloni; un altro esempio è fornito dai punti del piano che si possono identificare con le coppie di punti presi uno dalla retta delle ascisse e uno dalla retta delle ordinate.

Quando $X_1, \dots, X_n$ concidono tutti con uno stesso insieme X, usiamo la notazione X^n per indicare il relativo prodotto cartesiano, detto anche **potenza** n-esima dell'insieme X. Per $n = 1$, X^1 è X stesso; per $n = 0$, è utile identificare la potenza 0-esima X^0 di X con l'insieme $\mathbf{1}$ definito come un qualsiasi insieme con un elemento solo ($\mathbf{1}$ è chiamato 'insieme terminale' e, se si preferisce, lo si può definire esplicitamente mediante $\mathbf{1} := \{*\}$). Si noti che una funzione

$$c : \mathbf{1} \longrightarrow X$$

[2] Questo riassunto é strettamente finalizzato ad introdurre nozioni che verranno utilizzate nei paragrafi seguenti.

è univocamente specificata una volta noto l'elemento $c(*) \in X$; quindi le funzioni $\mathbf{1} \longrightarrow X$ possono a buon diritto essere confuse con gli elementi di X stesso.

Un **sottoinsieme** di un insieme X è un insieme composto da alcuni (magari nessuno, magari tutti) degli elementi di X.

L'insieme delle **parti** $\mathcal{P}(X)$ di un insieme X è l'insieme di tutti i sottoinsiemi di X. Ad esempio, $\mathcal{P}(\mathbf{1})$ consta dei due elementi

$$\emptyset, \qquad \{*\}.$$

Ci tornerà utile identificare questi due sottoinsiemi con i due valori di verità, rispettivamente 'falso' e 'vero', che abbiamo già incontrato studiando la logica proposizionale.

Una **relazione** n-aria su un insieme X è un sottoinsieme della potenza n-esima di X (per $n = 1$, dunque, una relazione unaria è semplicemente un sottoinsieme di X e, per $n = 0$, una relazione n-aria su X è un valore di verità). Ad esempio, la relazione 'essere amico di' è la relazione che contiene le coppie di esseri umani che sono amici tra di loro; 'essere minore di' è l'ovvia relazione fra numeri naturali che tutti conosciamo (tale relazione conterrà ad esempio le coppie $\langle 2, 3\rangle, \langle 12, 45\rangle, \ldots$, ma non la coppia $\langle 3, 2\rangle$).

2 Linguaggi elementari

2.1 Motivazioni

Una delle applicazioni della logica è lo studio della correttezza delle inferenze (ossia degli 'schemi di ragionamento'). Ad esempio, l'inferenza

$$\begin{array}{l} \text{Se piove, prendo l'ombrello.} \\ \text{Non prendo l'ombrello.} \\ \hline \text{Quindi non piove.} \end{array}$$

è corretta. Per provarlo, associamo all'enunciato 'Piove' la lettera proposizionale p, all'enunciato 'Prendo l'ombrello' la lettera proposizionale q, sicchè l'inferenza viene schematizzata con

$$\begin{array}{l} p \to q \\ \neg q \\ \hline \neg p \end{array}$$

Per rilevare la correttezza di tale inferenza, basta ora osservare che la formula

$$(p \to q) \land \neg q \to \neg p \tag{1}$$

è una tautologia. La formula (1) è stata ottenuta considerando l'implicazione che ha come antecedente la congiunzione delle premesse dell'inferenza da testare e come conseguente la sua conclusione.

Tuttavia vi sono inferenze intuitivamente corrette che non possono essere studiate e giustificate mediante il calcolo proposizionale. Consideriamo ad esempio la seguente inferenza:

Tutti gli uomini sono mortali.
Socrate è un uomo.
―――――――――――――――
Quindi Socrate è mortale.

È facile vedere che questa inferenza, pur essendo intuitivamente corretta, non è giustificabile mediante il calcolo proposizionale. Infatti riscrivendo l'inferenza mediante il simbolismo della logica proposizionale otteniamo

$$\begin{array}{c} p \\ q \\ \hline r \end{array}$$

Condizione necessaria e sufficiente affinchè questa inferenza sia corretta è che la formula

$$p \land q \to r$$

sia una tautologia. Tuttavia è facile osservare l'assegnamento V tale che $V(p) = V(q) = 1$ e $V(r) = 0$ la falsifica. Dunque l'inferenza data, intuitivamente corretta, risulta formalmente scorretta se utilizziamo come strumento di indagine il calcolo proposizionale.

In effetti la correttezza delle inferenze non si basa solamente sulle relazioni vero funzionali tra le proposizioni di cui sono composte (cosa che invece succede nelle inferenze giustificabili mediante il solo calcolo proposizionale), ma può basarsi anche sulla struttura interna di queste proposizioni e sul significato di espressioni quali, ad esempio, "ogni", "tutti", ecc. In altre parole, per poter trattare inferenze come l'ultima che abbiamo visto è necessario considerare più a fondo la struttura interna delle proposizioni.

Occorre quindi introdurre un linguaggio più ricco e con maggiore capacità espressiva del calcolo proposizionale. Più precisamente, dovremo utilizzare un nuovo formalismo e introdurre poi delle regole che permettano di trattare formule contenenti i nuovi simboli introdotti.

Vediamo brevemente e in modo molto informale da che cosa è costituito un linguaggio del primo ordine.

Innanzitutto ampliamo l'insieme degli operatori logici. Ai connettivi $\wedge, \vee \rightarrow, \neg$ aggiungiamo due nuovi operatori $\forall$ e $\exists$. Questi due nuovi operatori sono di natura sensibilmente differente dai connettivi proposizionali, si dicono *quantificatori*. Il simbolo $\forall$ si chiama *quantificatore universale* e il suo significato intuitivo è "per ogni", Il simbolo $\exists$ si chiama *quantificatore esistenziale* e il suo significato intuitivo è "esiste".

Avremo poi, in particolare, due classi distinte di enti linguistici: le "costanti" e i "predicati". Le costanti rappresenteranno elementi del dominio del discorso (numeri, persone, oggetti, etc) mentre i "predicati" rappresenteranno le relazioni che possono intercorrere tra questi oggetti (esempi di predicati sono: "essere un uomo", "essere minore di", etc.). In altre parole, i predicati ci consentono di esprimere proprietà e relazioni su insiemi di oggetti.

Dunque se ad esempio $P(x)$ indica che x gode di una data proprietà P allora l'espressione $\forall x P(x)$ significherà "per ogni x vale la proprietà P" o, equivalentemente, "la proprietà P vale per ogni x", mentre l'espressione $\exists x P(x)$ significherà "esiste un x che gode della proprietà P".
Come esempio riconsideriamo l'inferenza

$$\begin{array}{l} \text{Tutti gli uomini sono mortali.} \\ \text{Socrate è un uomo.} \\ \hline \text{Quindi Socrate è mortale.} \end{array}$$

Supponiamo che

- U stia per il predicato "essere uomo"

- M stia per il predicato "essere mortale"

- la costante a stia per "Socrate"

Possiamo quindi formalizzare la nostra inferenza con

$$\begin{array}{l} \forall x(U(x) \rightarrow M(x)) \\ U(a) \\ \hline M(a) \end{array}$$

Abbiamo in questo modo formalizzato l'inferenza in maniera tale che sarà poi possibile, mediante opportune tecniche, analizzarla e studiarne la correttezza.

Più in generale, nel linguaggio del primo ordine avremo i seguenti simboli: i connettivi proposizionali e i due quantificatori, le *variabili individuali* $x_0, x_1 \ldots$, le *costanti individuali* $a_1, a_2, \ldots$, le *lettere predicative* $P, Q \ldots$ e le *lettere funzionali* $f, g \ldots$ (in realtà, consideremo le costanti individuali come simboli di funzione 0-arie).

Nel nostro esempio abbiamo utilizzato una sola costante a, una sola variabile individuale x, due predicati unari U e M, e nessuna lettera funzionale.

2.2 Segnature, Termini e Formule

Dunque un linguaggio del primo ordine (che chiameremo anche 'linguaggio elementare') è più ricco di un linguaggio proposizionale e consente di nominare individui, di costruire designazioni di individui a partire da altre, di parlare di proprietà di individui, di quantificare su di essi, ecc. Diamo ora la relativa definizione formale:

Definizione 1. *Un* **linguaggio elementare** *(o segnatura)* $\mathcal{L}$ *è una quadrupla*

$$\langle \mathcal{P}, \mathcal{F}, \alpha, \beta \rangle,$$

dove

- $\mathcal{P}$ *è un insieme (detto insieme dei simboli di predicato) e* $\alpha : \mathcal{P} \longrightarrow \mathbb{N}$ *è una funzione a valori nei numeri naturali (se* $\alpha(R) = n$ *diciamo che* R *è un* simbolo di predicato di arietà n*);*

- $\mathcal{F}$ *è un insieme (detto insieme dei simboli di funzione) e* $\beta : \mathcal{F} \longrightarrow \mathbb{N}$ *è una funzione a valori nei numeri naturali (se* $\beta(f) = n$ *diciamo che* f *è un* simbolo di funzione di arietà n*).*

I simboli di predicato di arietà 0 sono le vecchie lettere proposizionali: essi possono essere utilizzati per formalizzare frasi costituite da soli verbi impersonali (come 'Piove', 'Nevica'). I simboli di predicato di arietà 1 rappresentano proprietà di individui: essi possono essere utilizzati per modellizzare nomi comuni, aggettivi qualificativi, nonchè verbi intransitivi (con questo intendiamo dire che 'uomo' rappresenta la proprietà di essere uomo, 'bello' la proprietà di essere bello, 'dormire' la proprietà di essere addormentato, ecc). I simboli di predicato di arietà 2 rappresentano relazioni fra coppie di individui: essi possono essere utilizzati per modellizzare verbi transitivi (ad esempio, 'amare' rappresenta l'insieme delle coppie di esseri umani il cui primo componente è innamorato del secondo). I simboli di predicato di arietà 3 rappresentano relazioni fra terne di individui: essi possono essere utilizzati per modellizzare verbi dativi come 'dare', 'regalare', ecc.[3]

I simboli di funzione di arietà zero sono detti costanti individuali e possono essere utilizzati per modellizzare i nomi propri, nonchè costanti matematiche (come $0, 1, \pi, e, \dots$). I simboli di funzione di arietà 2 rappresentano operazioni binarie (come ad esempio le operazioni aritmetiche di somma e prodotto); naturalmente i simboli di funzione di arietà 1 rappresentano operazioni unarie (come il seno, l'esponenziale in una base fissata, il logaritmo, ecc.). Simboli di funzione di arietà 1 sono presenti anche nel linguaggio naturale, dove possono essere utilizzati per rappresentare le funzioni espresse da certe locuzioni come 'il padre di$\cdots$', 'il professore di$\cdots$'.

Simboli di funzione e di predicato di arietà elevata (maggiore di 3, per intenderci) sono raramente usati, ma li abbiamo inclusi nella definizione di segnatura per omogeneità di trattazione.

In aggiunta ai simboli di predicato e di funzione, per costruire le formule avremo a disposizione dei simboli universali (cioè comuni ad ogni linguaggio elementare), che sono, oltre ai simboli ausiliari (parentesi e vigole):

- l'insieme $\mathcal{V} = \{x_0, x_1, x_2, \dots, y_0, y_1, y_2, \dots\}$ detto insieme delle **variabili individuali**;

- i **connettivi proposizionali** $\wedge, \vee, \rightarrow, \neg$ che abbiamo incontrato nella parte del corso relativa alla logica proposizionale;

- i **quantificatori** $\forall$ (per ogni) ed $\exists$ (esiste).

Se $\mathcal{P}$ contiene il predicato $=$ di arietà 2, diciamo che $\mathcal{L}$ è un linguaggio con identità.

Le variabili individuali rappresentano individui indeterminati, la cui designazione può ad esempio essere fissata dal contesto (in tal senso, si può costruire un parallelo fra le variabili dei linguaggi formali e i pronomi dei linguaggi naturali).

Per riferirci ad individui, abbiamo ora a disposizione costanti e variabili: se applichiamo a queste ultime i simboli di funzione, possiamo costruire designatori più complessi. In tal

[3]Per un'analisi meno schematica e più approfondita di queste classificazioni occorre consultare un testo sulla semantica dei linguaggi naturali.

modo si costruiscono espressione matematiche come $\pi + log\,(x+y)$, ma anche locuzioni del linguaggio naturale come 'padre di Carlo', 'padre del padre di Carlo' (ossia 'nonno paterno di Carlo'), ecc. Tutto questo è riassunto nella definizione formale di termine:

Definizione 2. *Data una segnatura $\mathcal{L}$, l'insieme degli $\mathcal{L}$-termini (o più semplicemente* **termini***) è cosí definito:*

- *ogni $c \in \mathcal{F}$ con $\beta(c) = 0$ è un termine;*

- *ogni $x \in \mathcal{V}$ è un termine;*

- *se $f \in \mathcal{F}$, se $\beta(f) = n$ (per $n \geq 1$) e se $t_1, \ldots, t_n$ sono termini, $f(t_1, \ldots, t_n)$ è un termine.*[4]

Abbiamo visto che i termini aumentano le possibilità di un linguaggio elementare di riferirsi ad individui (possibilità inizialmente ristretta a costanti e variabili). In modo simile, la nozione di formula offre la possibilità di costruire relazioni e affermazioni complesse:

Definizione 3. *Data una segnatura $\mathcal{L}$, l'insieme delle $\mathcal{L}$-formule (o più semplicemente* **formule***) è cosí definito:*

- *se $R \in \mathcal{P}$, se $\alpha(R) = n$ e se $t_1, \ldots, t_n$ sono termini, $(R(t_1, \ldots, t_n))$ è una formula;*[5]

- *se A_1, A_2 sono formule, tali sono anche $(A_1 \wedge A_2)$, $(A_1 \vee A_2)$, $(A_1 \to A_2)$, $(\neg A_1)$;*

- *se A è una formula e $x \in \mathcal{V}$, allora $(\forall x A)$ e $(\exists x A)$ sono formule.*

La scelta di un opportuno linguaggio dipende da ciò di cui vogliamo parlare e da ciò che intendiamo realmente esprimere. Ad esempio, se vogliamo parlare di numeri (naturali, interi, razionali o reali) un opportuno linguaggio $\mathcal{L}_1$ potrebbe contenere: a) due costanti $0, 1$; b) due simboli di operazioni binarie, cioè la somma $+$ e il prodotto $\cdot$; c) due relazioni binarie, cioè l'identità $=$ e la relazione di minore $<$. Se invece vogliamo parlare di relazioni di parentela, una scelta ragionevole potrebbe essere il linguaggio $\mathcal{L}_2$ comprendente: a) due

[4]Si osservi che, nel caso dei simboli di funzione binaria, questa definizione obbliga ad adottare una notazione prefissa (cioè obbliga a scrivere ad esempio $+(x, y)$ anzichè $x + y$). Tuttavia, noi ci sentiremo liberi di usare notazioni infisse nei casi in cui lo riterremo opportuno.

[5]Formule di questo tipo, cioè formule ottenute applicando un simbolo di predicato a termini, si dicono formule *atomiche*. Le formule atomiche non contengono quindi altre formule come parti proprie.

simboli di funzione unarie, cioè $p =$'il padre di' e $m =$'la madre di'; b) un predicato binario, l'identità.

Come sappiamo, i termini servono a costruire designazioni di individui complesse, partendo dalle designazioni di base specificate da costanti e variabili. Nel caso di $\mathcal{L}_2$, i termini

$$p(p(x)), \qquad p(m(x))$$

servono a specificare il nonno paterno e il nonno materno dell'individuo x. Analogamente, le formule costruiscono proprietà e relazioni complesse. Nel caso di $\mathcal{L}_2$, la formula

$$p(x) = p(y) \wedge m(x) = m(y)$$

dice che x e y sono fratelli/sorelle. La formula

$$(p(p(x))=p(p(y))) \vee (p(p(x))=p(m(y))) \vee (p(m(x))=p(p(y))) \vee (p(m(x))=p(m(y)))$$

è un modo per esprimere che x e y sono cugini (tramite la comunanza di uno dei nonni maschi).

Convenzioni notazionali:

- Valgono le solite convenzioni per eliminare le parentesi, ossia le parentesi più esterne vengono di regola omesse. In più stipuliamo anche che $\neg, \forall, \exists$ legano più strettamente di $\wedge, \vee$, che a loro volta legano più strettamente di $\rightarrow$.[6]

- Un termine è *chiuso* (o '**ground**') se è costruito senza usare variabili.

- Una occorrenza di una variabile x in una formula A è detta **vincolata** qualora si trovi all'interno di una sottoformula[7] del tipo $\forall x B$ o $\exists x B$, altrimenti è detta **libera**. Ad esempio, nelle formula

  $$\forall x(R(x, y)) \vee P(x)$$

  la x ha due occorrenze, la prima delle quali è vincolata (perchè è all'interno della sottoformula $\forall x(R(x, y)))$, mentre la seconda è libera; la y ha una sola occorrenza, che è libera.

[6] Così, ad esempio, $\forall x P(x) \rightarrow \exists z P(z) \vee \exists y Q(y)$ sta per $(\forall x(P(x))) \rightarrow ((\exists z(P(z))) \vee (\exists y(Q(y))))$. Si faccia attenzione a queste convenzioni: sbagliare nell'identificare l'operatore principale (connettivo o quantificatore che sia) di una formula costituisce errore invalidante (e purtroppo molto frequente) negli esercizi scritti.

[7] La nozione di sottoformula è quella ovvia (volendo essere precisi, si potrebbe definirla induttivamente).

- Una variabile x occorre libera in A se e solo se qualche sua occorrenza in A è libera.
- Un **enunciato** ('sentence', in inglese) o formula chiusa è una formula in cui nessuna variabile occorre libera.
- Con notazioni del tipo $A(t/x)$ (o, più semplicemente $A(t)$) indicheremo il risultato dell'operazione di sostituzione del termine t al posto di tutte le occorrenze libere della variabile x in A.

3 La Semantica di Tarski

Diamo ora la semantica per i linguaggi elementari. Il problema della semantica è presto detto: simboli di funzione, simboli di predicato, termini e formule di per sè non sono altro che caratteri e stringhe di caratteri e in quanto tali non hanno nessun significato intrinseco. È ben vero che nel definire termini e formule abbiamo tenuto presente che i termini devono rappresentare individui e che le formule devono rappresentare relazioni complesse o affermazioni, tuttavia queste motivazioni intuitive non sono sufficienti ad attribuire loro un significato pieno. In altri termini, il fatto che il dizionario riporti la parola 'cane' come parola dotata di significato, non significa ancora che tale significato sia noto o compreso, non prima almeno di aver letto cosa la parola 'cane' significhi o di averlo imparato per altra via da piccoli.

La semantica dei linguaggi elementari è stata rigorosamente fissata dal logico polacco Tarski negli anni 30; tuttavia la definizione tarskiana non fa che riprendere la millenaria tradizione filosofica della definizione di verità come corrispondenza con lo stato di fatto. All'interno di un corso di logica, la definizione tarskiana di verità rappresenta un passaggio imprescindibile, perchè è solo su di essa che si possono fondare tutti i procedimenti algoritmici che vengono poi introdotti. Ciononostante, l'impatto con la semantica formalizzata può risultare difficile di primo acchito per un semplice motivo: i significati delle parole e delle locuzioni che usiamo nella vita quotidiana (e quindi anche la nozione di verità che su di essi si basa) sono fissati dalle convenzioni linguistiche e sociali, per cui non si sente il bisogno di fare un passo indietro e di riesaminarli da un punto di vista astratto. Tuttavia tale riesame è per noi indispensabile per proseguire; invitiamo lo studente perciò a leggere senza troppe pretese il presente paragrafo in prima battuta e a ritornare (magari più volte) su di esso quando la sua comprensione degli argomenti si sarà affinata.

3.1 La Nozione di Struttura

La prima cosa da tenere presente è che il significato dei simboli dipende da una situazione concreta. Nel caso della logica proposizionale, una situazione concreta è modellata da un assegnamento: l'assegnamento fa sì ad esempio che l'enuciato 'Piove' venga a denotare un significato (il vero od il falso) a seconda appunto della situazione che l'assegnamento simula in modo schematico, cioè a seconda che realmente piova o meno. Nella logica predicativa, occorerà la nozione più complessa di *struttura* invece della nozione di assegnamento: una struttura deve assegnare un insieme di oggetti ad ogni simbolo di predicato unario, una relazione (vista come insieme di coppie di oggetti) ad ogni simbolo di predicato binario, un oggetto ad ogni costante, ecc.[8] In altre parole, la nozione di struttura fotografa e schematizza (mediante strumenti insiemistici) quanto è noto sul significato degli enti linguistici in una data situazione.

Per introdurre formalmente una struttura, si fissa un insieme non vuoto $\mathbf{A}$, detto dominio della struttura, sul quale assumeranno i valori le variabili individuali; poi si fissa una funzione che associ ad ogni simbolo di funzione n-ario una funzione da $\mathbf{A}^n$ a $\mathbf{A}$ (cioè una operazione a n posti) e ad ogni simbolo di predicato n-ario una relazione n-aria (cioè un sottoinsieme di $\mathbf{A}^n$). Tutto questo è scritto nella seguente definizione:

Definizione 4. *Data una segnatura $\mathcal{L}$, una $\mathcal{L}$-***struttura** *$\mathcal{A}$ è una coppia $\langle \mathbf{A}, \mathcal{I} \rangle$ dove $\mathbf{A}$ è un insieme (non vuoto),*[9] *detto* **dominio***, e $\mathcal{I}$ è una funzione, detta* **interpretazione***, che opera come specificato qui di seguito. $\mathcal{I}$ associa*

- *ad ogni $P \in \mathcal{P}$ tale che $\alpha(P) = n$, un sottoinsieme $\mathcal{I}(P)$ di $\mathbf{A}^n$.*
- *ad ogni $f \in \mathcal{F}$ tale che $\beta(f) = n$, una funzione $\mathcal{I}(f) : \mathbf{A}^n \longrightarrow \mathbf{A}$.* $\dashv$

In particolare, se $c \in \mathcal{F}_0$, $\mathcal{I}(c)$ è un elemento di $\mathbf{A}$ e se $P \in \mathcal{P}_0$, $\mathcal{I}(P)$ è un valore di verità (si ricordi quanto convenuto nel paragrafo 1). Se $\mathcal{L}$ contiene l'identità, stipuliamo che $\mathcal{I}(=)$ sia sempre l'insieme delle coppie identiche, cioè $\{(a, a) \mid a \in \mathbf{A}\}$.

[8] Per farsi un'idea precisa, può essere utile pensare ai nomi propri di persona, come 'Pietro', 'Paolo', ecc.: questi ultimi non hanno un significato universale fissato apriori (come possiamo essere erroneamente tentati di credere che succeda per altre entità linguistiche come nomi comuni e verbi), il loro significato è di volta in volta adattato dalla nostra mente alla situazione in cui ci troviamo.

[9] Si faccia attenzione al fatto che usiamo la lettera calligrafica $\mathcal{A}$ per indicare una struttura nel suo complesso e la lettera in grassetto $\mathbf{A}$ per indicarne il dominio (invece la lettera stampata maiuscola A, come le lettere $B, C, \ldots$, continueranno ad essere usate come metavariabili per formule). Questi aspetti notazionali non vanno trascurati, pena cadere in confusione completa!

3.2 La Nozione di Verità

Fissata un'interpretazione (cioè una $\mathcal{L}$-struttura $\mathcal{A}$) per i simboli di un linguaggio $\mathcal{L}$, è possibile dire quali enunciati di $\mathcal{L}$ sono veri (in $\mathcal{A}$) e quali no. Ciò rispecchia una pratica intuitiva: possiamo dire se 'Paolo è simpatico' è vero o no, una volta che ci siamo intesi sul significato delle parole, cioè una volta che abbiamo fissato una $\mathcal{L}$-struttura. Avendo fissato una $\mathcal{L}$-struttura, sappiamo chi è 'Paolo', sappiamo quali sono i nostri criteri di simpatia perchè abbiamo fissato l'insieme delle persone simpatiche, per cui per stabilire il valore di verità della frase 'Paolo è simpatico' si tratta solo di vedere se Paolo appartiene o meno a tale insieme delle persone simpatiche. Più in generale, avremo che una formula del tipo $P(c_1, \ldots, c_n)$ (dove $c_1, \ldots, c_n$ sono costanti) sarà vera nella $\mathcal{L}$-struttura $\mathcal{A}$ sse la n-pla $(\mathcal{I}(c_1), \ldots, \mathcal{I}(c_n))$ (che fissa gli individui denotati da $c_1, \ldots, c_n$ in $\mathcal{A}$) appartiene a $\mathcal{I}(P)$ (ossia all'insieme delle n-ple che fissa il significato della relazione n-aria P in $\mathcal{A}$).

Per definire il valore di verità di formule non atomiche avremo delle ovvie clausole ricorsive. Tuttavia, nel dare tale definizione, si incontrano alcuni problemi che necessitano l'introduzione di qualche accorgimento tecnico, dovuto al fatto che non tutti gli elementi di $\mathbf{A}$ sono nominabili con termini chiusi di $\mathcal{L}$. Il problema può essere evidenziato nel modo seguente. La relazione $\mathcal{A} \models A$ ('l'enunciato A è vero nella $\mathcal{L}$-struttura $\mathcal{A}$') verrà, come si è detto, definita per induzione sul numero di connettivi e quantificatori di A; avremo ad esempio clausole ricorsive del tipo[10]

$$\mathcal{A} \models A_1 \wedge A_2 \qquad \text{sse} \qquad (\mathcal{A} \models A_1 \quad \text{e} \quad \mathcal{A} \models A_2)$$

che dicono che una congiunzione è vera sse lo sono entrambi i congiunti. Tuttavia, nel caso dei quantificatori, tali clausole non possono essere formulate ingenuamente con

$$\mathcal{A} \models \exists x A \qquad \text{sse} \qquad \mathcal{A} \models A(a/x), \text{ per qualche } \ a \in \mathbf{A}$$

perchè $A(a/x)$ non è una $\mathcal{L}$-formula, in quanto a è un elemento del dominio di interpretazione e non un simbolo del linguaggio (per cui la scrittura $A(a/x)$ semplicemente non ha senso). Se tentiamo di risolvere questo problema, ci accorgiamo subito inoltre che non c'è nessuna motivo affinchè il linguaggio $\mathcal{L}$ abbia a disposizione un nome per ogni elemento $a \in \mathbf{A}$.[11] Se avessimo a disposizione un nome $\bar{a}$ per ogni $a \in \mathbf{A}$, la clausola di verità per il

[10] Qui e nel seguito, abbreviamo 'se e solo se' con 'sse'.

[11] Si pensi soltanto al fatto seguente: il linguaggio usualmente è numerabile (cioè le formule e i termini possono di solito essere messi in corrispondenza biunivoca con i numeri naturali), mentre $\mathbf{A}$ può essere un insieme qualunque molto più grande, come quello dei numeri reali.

quantificatore esistenziale potrebbe essere agevolmente corretta con

$$\mathcal{A} \models \exists x A \qquad \text{sse} \qquad \mathcal{A} \models A(\bar{a}/x), \text{ per qualche } \ a \in \mathbf{A}.$$

Per questo motivo, decidiamo di ampliare preventivamente $\mathcal{L}$ stesso. Se $\mathcal{A}$ è una $\mathcal{L}$-struttura, $\mathcal{L}_{\mathbf{A}}$ indica il linguaggio ottenuto aggiungendo a $\mathcal{L}$ una costante $\bar{a}$ per ogni $a \in \mathbf{A}$ ($\bar{a}$ è detta essere il nome di a). Cosí $\mathcal{L}_{\mathbf{A}}$ contiene un nome per ogni elemento del dominio di $\mathcal{A}$. Allarghiamo $\mathcal{I}$ a queste nuove costanti ponendo $\mathcal{I}(\bar{a}) = a$ (in futuro, se non c'è pericolo di confusione, ometteremo spesso di distinguere fra elementi di $\mathbf{A}$ e i loro nomi, cioè scriveremo direttamente a invece di $\bar{a}$).

C'è ancora un punto da chiarire. L'interpretazione $\mathcal{I}$ di una $\mathcal{L}$-struttura $\mathcal{A} = \langle \mathbf{A}, \mathcal{I} \rangle$ fissa il significato delle costanti (e dei simboli di funzione), ma non fissa direttamente il significato dei termini composti. Ad esempio, la $\mathcal{L}$-struttura $\mathcal{A}$ fissa il significato della costante $c =$'Paolo', fissa il significato delle funzioni unarie $p =$'padre di' e $m =$'madre di'; possiamo da tutto ciò risalire al significato dell'espressione 'nonno materno di Paolo'? Certamente, tale espressione 'nonno materno di Paolo' altri non è che $p(m(c))$ e il suo significato sarà $\mathcal{I}(p)(\mathcal{I}(m)(\mathcal{I}(c)))$ (ossia il valore della funzione che interpreta 'padre di' calcolato sul valore della funzione che interpreta 'madre di' calcolato sull'elemento denotato da 'Paolo').

Formalmente, si procede così: per induzione, *definiamo $\mathcal{I}(t)$ per ogni $\mathcal{L}_{\mathbf{A}}$-termine ground t.* Se t è una costante (vecchia o nuova) $\mathcal{I}(t)$ è già stato definito. Se t è $f(t_1, \dots, t_n)$, $\mathcal{I}(t)$ sarà $\mathcal{I}(f)(\mathcal{I}(t_1), \dots, \mathcal{I}(t_n))$ (dove $\mathcal{I}(t_1), \dots, \mathcal{I}(t_n)$ sono dati per induzione e $\mathcal{I}(f)$ è l'interpretazione del simbolo f specificata dalla $\mathcal{L}$-struttura $\mathcal{A} = (\mathbf{A}, \mathcal{I})$).

Siamo ora in grado di dare le definizione di verità di una $\mathcal{L}$-formula in una $\mathcal{L}$-struttura:

Definizione 5. *Data una $\mathcal{L}$-struttura $\mathcal{A}$ e dato un $\mathcal{L}_{\mathbf{A}}$-enunciato A, la relazione $\mathcal{A} \models A$ (che si legge con 'A è* **vero** *in $\mathcal{A}$), è definita induttivamente sul numero di connettivi e* *quantificatori di A come segue:*

$$\begin{array}{rll}
\mathcal{A} \models P(t_1, \ldots, t_n) & \text{sse} & (\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n)) \in \mathcal{I}(P) \\
\mathcal{A} \models A_1 \wedge A_2 & \text{sse} & (\mathcal{A} \models A_1 \quad \text{e} \quad \mathcal{A} \models A_2) \\
\mathcal{A} \models A_1 \vee A_2 & \text{sse} & (\mathcal{A} \models A_1 \quad \text{oppure} \quad \mathcal{A} \models A_2) \\
\mathcal{A} \models \neg A_1 & \text{sse} & \mathcal{A} \not\models A_1 \\
\mathcal{A} \models A_1 \to A_2 & \text{sse} & (\mathcal{A} \not\models A_1 \quad \text{oppure} \quad \mathcal{A} \models A_2) \\
\mathcal{A} \models \forall x A_1 & \text{sse} & \mathcal{A} \models A_1(\bar{a}/x) \quad \text{per ogni } a \in \mathbf{A} \\
\mathcal{A} \models \exists x A_1 & \text{sse} & \mathcal{A} \models A_1(\bar{a}/x) \quad \text{per qualche } a \in \mathbf{A}.
\end{array}$$

Se A è una formula qualunque (non necessariamente un enunciato) $\mathcal{A} \models A$ sta per $\mathcal{A} \models \forall x_1 \cdots \forall x_n A$ (dove abbiamo assunto che $x_1, \ldots, x_n$ siano tutte e sole le variabili che occorrono libere in A).[12]

3.3 Teorie, Modelli e Conseguenza Logica

La seguente importante definizione completa il quadro della semantica della logica elementare. Chiamiamo **$\mathcal{L}$-teoria $\mathcal{T}$** un qualsiasi insieme di enunciati di $\mathcal{L}$ (le formule appartenenti a $\mathcal{T}$ saranno dette *assiomi* di $\mathcal{T}$).

[12] Si noti che nel caso in cui $\mathcal{L}$ contenga l'identità abbiamo sempre

$$\mathcal{A} \models t_1 = t_2 \quad \text{sse} \quad \mathcal{I}(t_1) = \mathcal{I}(t_2)$$

per ogni coppia t_1, t_2 di $\mathcal{L}_{\mathbf{A}}$-termini chiusi.

Inseriamo qui anche un'osservazione tecnica (al momento assolutamente marginale). Data una $\mathcal{L}$-struttura $\mathcal{A}$ e data una $\mathcal{L}_{\mathbf{A}}$-formula $A(x)$ (in cui la sola variabile x può occorrere libera), si prova facilmente per induzione (ma si deve fare un'induzione preventiva per stabilire un'analoga proprietà dei termini) che se t e u sono $\mathcal{L}_{\mathbf{A}}$-termini chiusi tali che $\mathcal{I}(t) = \mathcal{I}(u)$ allora

$$(+) \qquad \mathcal{A} \models A(t/x) \quad \text{sse} \quad \mathcal{A} \models A(u/x).$$

Ne segue che la clausola induttiva nella definizione di verità, ad esempio per il quantificatore esistenziale, si può equivalentemente dare nella seguente forma (detta sostituzionale)

$$\mathcal{A} \models \exists x A \quad \text{sse} \quad \mathcal{A} \models A(t/x) \quad \text{per qualche } t \text{ termine chiuso di } \mathcal{L}_{\mathbf{A}}$$

(infatti, se $\mathcal{I}(t) = a$, $\mathcal{A} \models A(t/x)$ equivale a $\mathcal{A} \models A(\bar{a}/x)$ per la (+)).

Definizione 6. *Se $\mathcal{T}$ è una $\mathcal{L}$-teoria, si dice che $\mathcal{A}$ è* **modello** *di $\mathcal{T}$ (in simboli $\mathcal{A} \models \mathcal{T}$) qualora $\mathcal{A} \models B$ valga per ogni enunciato B appartenente a $\mathcal{T}$. Si dice che $\mathcal{T}$ è* **soddisfacibile**, *qualora abbia almeno un modello. $\mathcal{T} \models A$ (letto con 'A è* **conseguenza logica** *di $\mathcal{T}$') significa che A è vera in ogni modello di $\mathcal{T}$. Una $\mathcal{L}$-formula A è un* **logicamente valida** *qualora $\mathcal{A} \models A$ valga per ogni $\mathcal{A}$.*

Tutti i teoremi che si trovano nei libri di matematica sono conseguenze logiche di un opportuno sistema assiomatico, cioè di una opportuna teoria. Il problema di decidere la nozione di conseguenza logica modulo una teoria è centrale in molte delle applicazioni della logica all'informatica. Anche il problema del riconoscimento della correttezza delle regole di inferenza si può ridurre al problema della conseguenza logica modulo una teoria. Ad esempio, per stabilire che l'inferenza (vista nel paragrafo 2.1)

$$\frac{\text{Tutti gli uomini sono mortali.} \quad \text{Socrate è un uomo.}}{\text{Quindi Socrate è mortale.}}$$

è corretta basta stabilire la relazione di conseguenza logica

$$\{\forall x(U(x) \to M(x)),\ U(a)\} \ \models\ M(a).$$

Per affrontare il problema di decidere la relazione $\mathcal{T} \models A$ spesso è utile ricorrere a metodi che siano specifici per la teoria $\mathcal{T}$ scelta; tuttavia, se $\mathcal{T}$ ha un numero finito di assiomi si può ridurre il problema di decidere la conseguenza logica modulo $\mathcal{T}$ al problema di decidere se una data formula è o meno logicamente valida. Infatti se $\mathcal{T} = \{B_1, \dots, B_n\}$, si ha che A è conseguenza logica di $\mathcal{T}$ sse la formula

$$B_1 \wedge \dots \wedge B_n \to A$$

è logicamente valida.

La nozione di verità logica (cioè di formula logicamente valida) corrisponde alla nozione di tautologia usata nella logica proposizionale. C'è però una cruciale differenza: per la logica predicativa non è possibile adottare un procedimento di esame esaustivo (come l'ispezione di tutta la tavola di verità) per stabilire se una data formula è o meno una verità logica (bisognerebbe esaminare tutte le possibili $\mathcal{L}$-strutture, cosa chiaramente infattibile). Per questo motivo, a differenza che nel caso proposizionale, la definizione semantica di verità non è solo inefficiente, ma proprio inutilizzabile dal punto di vista computazionale.

Per un risultato classico (*il teorema di Church*), la nozione di verità logica è indecidibile, cioè non è possibile progettare un algoritmo che, presa in ingresso una formula A, termini sempre dicendo in uscita se A è una verità logica o meno. Il calcolo che proporremo costituirà perciò solo una procedura di semidecisione: data A in ingresso, se A è una verità logica, sarà sempre possibile appurarlo in modo meccanico (pur di avere risorse di calcolo illimitate in tempo ed in spazio); se non lo è, solo nei casi fortunati saremo in grado di appurarlo, nei casi sfortunati la procedura di ricerca sarà destinata a non avere mai fine. Quindi il calcolo che presenteremo, una volta implementato su un calcolatore, ci potrà porre nella situazione imbarazzante per cui, se dopo un tempo notevole di attesa non abbiamo ancora avuto la risposta, non potremo mai sapere se tale attesa è fatalmente destinata a durare per sempre o se, dando ancora un po' di tempo alla macchina, si avrebbe invece la risposta desiderata. Nonostante questi limiti di principio, va comunque osservato che il settore del ragionamento automatico ha fatto segnare in tempi recenti notevoli successi, tali da coprire un certo numero di casi di interesse sia teorico che pratico.

4 Forme Normali Negative

Una formula è in **forma normale negativa** (**fnn**) se e solo se non contiene implicazioni e contiene negazioni solo di fronte a sottoformule atomiche.

Due formule A, B sono **logicamente equivalenti** se e solo se $A \leftrightarrow B$ è una verità logica. Per trasformare una formula A in una formula A' in fnn *logicamente equivalente* ad A, è sufficiente eseguire (**in ordine qualunque, ma in modo esaustivo**) le seguenti trasformazioni:

$$C \to D \quad \leadsto \quad \neg C \vee D \tag{2}$$

$$\neg\neg C \quad \leadsto \quad C \tag{3}$$

$$\neg(C \vee D) \quad \leadsto \quad \neg C \wedge \neg D \tag{4}$$

$$\neg(C \wedge D) \quad \leadsto \quad \neg C \vee \neg D \tag{5}$$

$$\neg\forall x C \quad \leadsto \quad \exists x \neg C \tag{6}$$

$$\neg\exists x C \quad \leadsto \quad \forall x \neg C \tag{7}$$

Le trasformazioni vanno viste come regole di riscrittura: ossia ogniqualvolta la formula corrente A contenga una sottoformula del tipo indicato a sinistra, la si rimpiazzi con la corrispondente formula del tipo indicato a destra. Il seguente teorema è conseguenza di un lemma generale di rimpiazzamento (su cui non ci soffermiamo) e del fatto che le formule a destra e a sinistra di $\rightsquigarrow$ nelle (2)-(7) sono logicamente equivalenti fra loro:

Teorema 1. *Ogni formula è logicamente equivalente ad una formula in fnn.*

Esempio 1. Trasformiamo in fnn la formula $\neg(\exists x \forall y R(x,y) \vee \exists x R(x,x))$:

$$\begin{array}{lll} & \neg(\exists x \forall y R(x,y) \vee \exists x R(x,x)) & \rightsquigarrow \\ \rightsquigarrow & \neg\exists x \forall y R(x,y) \wedge \neg\exists x R(x,x)) & \rightsquigarrow \\ \rightsquigarrow & \forall x \neg\forall y R(x,y) \wedge \forall x \neg R(x,x)) & \rightsquigarrow \\ \rightsquigarrow & \forall x \exists y \neg R(x,y) \wedge \forall x \neg R(x,x)) & \end{array}$$

Un *enunciato* A è **soddisfacibile** sse esiste una struttura $\mathcal{A}$ tale che $\mathcal{A} \models A$. Il calcolo che vedremo nel prossimo paragrafo affronta il problema della *soddisfacibilità di enunciati in fnn*. Quindi se lo vogliamo utilizzare per testare se un dato enunciato B è o meno una verità logica, **occorre preventivamente trasformare** $\neg B$ **in fnn** (attenzione a ricordarsi la negazione per non sbagliare!).

5 Il Calcolo dei Tableaux

Fissiamo da qui fino alla fine delle presenti note *una segnatura* $\mathcal{L}$ *ed un* $\mathcal{L}$*-enunciato* A *in fnn di cui vogliamo sapere se è soddisfacibile o meno.* Sia $\mathcal{L}^+$ ottenuto da $\mathcal{L}$ con l'aggiunta di un'infinità numerabile di nuove costanti (che chiameremo **parametri**) $c_0, c_1, c_2, \ldots$.

Usiamo le lettere greche $\Gamma, \Delta, \ldots$ per indicare insiemi finiti di enunciati[13] di $\mathcal{L}^+$; notazioni come Γ, C e Γ, C, D verranno usate rispettivamente al posto di $\Gamma \cup \{C\}$ e $\Gamma \cup \{C, D\}$.

[13]Se si vuole operare con formule che contengono variabili libere (cosa che non faremo per semplicità didattica), occorre tenere presente alcuni problemi relativi alla sostituzione. Li riassumiamo brevemente perchè essi sono menzionati su tutti i testi di logica. Il nostro formalismo deve consentirci di dimostrare formule del tipo $\forall x A \to A(t/x)$, tuttavia non tutte le formule tale tipo sono verità logiche. Ad esempio $\forall x \exists y P(x,y) \to \exists y P(y,y)$ non vale in tutte le $\mathcal{L}$-strutture (questa formula direbbe ad esempio che 'se ogni uomo ha un padre, allora c'è qualcuno che è padre di se stesso'). Questa anomalia è dovuta al fatto che la sostituzione di y ad x in $\exists y P(x,y)$ non è da ritenersi corretta: in casi come questi il termine sostituendo contiene una variabile che, a sostituzione avvenuta, risulta vincolata. Diciamo, in generale, che il termine t è sostituibile ad x in A qualora per nessuna variabile y che occorre in t ci sia una sottoformula di A del tipo $\exists y B$ (oppure del tipo $\forall y B$) contenente un'occorrenza di x che è libera in A. Per manipolare formule con

La nostra procedura costruisce un albero i cui nodi sono etichettati mediante insiemi finiti di enunciati di $\mathcal{L}^+$ (la radice dell'albero viene sempre etichettata con $\{A\}$).

La procedura analizza l'informazione che A è soddisfacibile, per ricavarne eventuali contraddizioni: se vengono ricavate contraddizioni, si conclude l'insoddisfacibilità di A, altrimenti se ne certifica la soddisfacibilità (ma in questo caso la procedura può divergere, per cui si può avere una certificazione per così dire ideale, dopo infiniti passi di calcolo).

Si procede nel seguente modo: si seleziona un nodo-foglia dell'albero corrente che non sia **chiuso**, ossia che non contenga *sia una formula atomica che la sua negazione* (se tutti i nodi-foglia sono chiusi la formula è dichiarata **insoddisfacibile**). Se a tale nodo-foglia nessuna delle regole di espansione della Figura 1 si applica, il nodo-foglia è detto **terminale**: la formula è dichiarata **soddisfacibile**. Altrimenti, se una delle regole di espansione della Figura 1 si applica, si aggiungono sotto tale nodo uno/due nodi-figli etichettandoli come previsto dalla regola stessa.

$$\frac{\Gamma, B \wedge C}{\Gamma, B, C} \qquad\qquad \frac{\Gamma, B \vee C}{\Gamma, B \parallel \Gamma, C}$$

$$\frac{\Gamma, \exists x B}{\Gamma, B(c/x)} \qquad\qquad \frac{\Gamma, \forall x B}{\Gamma, B(t/x), \forall x B}$$

Figura 1: **Regole di espansione.** (i) *c è un parametro nuovo;* (ii) *t è un termine ground di $\mathcal{L}^+$.*

Come si può osservare dalla figura, c'è una regola di espansione per ogni connettivo o quantificatore (salvo che per la negazione e l'implicazione che sono state pretrattate con la riduzione in fnn). Facciamo ora alcuni utili commenti su ciascuna delle regole (nei prossimi paragrafi esamineremo svariati esempi in dettaglio).

- Il significato della regola di espansione per $\wedge$ è questo: se abbiamo trovato una

variabili libere, occorre specificare che nella regola di espansione per il quantificatore universale della figura 1, la sostituzione coinvolta deve essere corretta (o, meglio, che in caso di sostituzioni scorrette, si deve passare ad una variante alfabetica, cioè ad una rinomina delle variabili vincolate). Si noti che, operando soltanto con termini ground, il problema non si pone: se t è ground, t è sempre sostituibile ad ogni x in ogni A perchè t non contiene variabili.

struttura in cui tutte le Γ e la $B \wedge C$ sono vere, abbiamo anche una struttura in cui tutte le Γ e la B e la C sono vere.

- La regola di espansione per $\vee$ dice che se abbiamo trovato un a struttura in cui tutte le Γ e la $B \vee C$ sono vere, abbiamo anche *o* una struttura in cui tutte le Γ e la B sono vere *o* una struttura in cui tutte le Γ e la C sono vere. Questa regola di espansione è responsabile delle biforcazioni che si producono nell'albero che costruiamo.

- La regola di espansione per $\exists$ dice che se abbiamo trovato una struttura in cui tutte le Γ e la $\exists x B$ sono vere, abbiamo anche una struttura in cui tutte le Γ e la $B(c/x)$ sono vere. Perchè la regola sia applicata in modo corretto, occorre che il parametro c sia **nuovo**, cioè non ancora utilizzato (attenzione: l'uso di un parametro vecchio costituisce errore invalidante negli esercizi). Il motivo di questa restrizione sta nel fatto che l'informazione 'c'è un x tale che B' può sì essere utilizzata dando un nome a tale x, ma il nome non deve collidere con altri nomi già noti, altrimenti possiamo rischiare di fare assunzioni ingiustificate su x.

- La regola di espansione per $\forall$ è responsabile della possibile non terminazione della procedura. Innanzitutto, si osservi che tale regola prevede la **ricopiatura** della formula che viene analizzata, a differenza delle regole precedenti. Il motivo è il seguente: l'informazione 'tutti gli x sono B' può essere utilizzata dicendo 't è B', 'u è B, ecc. Qui $t, u, \dots$ sono termini ground qualunque e sappiamo che i termini ground sono infiniti (semplicemente perchè i parametri sono infiniti). Quindi la regola di espansione per $\forall$ è passibile di infiniti utilizzi per ciascuna formula universalmente quantificata cui si applica. Sono però possibili, per fortuna, delle ottimizzazioni, in particolare noi useremo sempre le seguenti due restrizioni ottimizzanti (che si dimostrano non distruggere la completezza del calcolo): (a) il termine t che si utlizza nel conseguente della regola di espansione per il quantificatore universale deve essere **vecchio**, ossia già presente nel ramo che porta al nodo cui si applica la regola;[14] (b) lungo tale ramo, la regola non si applica mai due volte con identiche modalità (cioè allo stesso t per la stessa formula $\forall x B$).

In linea di principio, le regole di espansione si applicano in un ordine qualunque,

[14]Con una unica **eccezione**: se in tale ramo *non compare alcun termine ground*, si prende come t un parametro qualunque, ad esempio c_0.

ma ritorneremo sull'argomento per delle precisazioni importanti da un punto di vista computazionale.

5.1 Ricerca di Refutazioni

Il calcolo che abbiamo esposto può essere utilizzato negli esercizi sia per stabilire che un data formula è logicamente valida, sia per trovarne un contromodello (questo doppio utilizzo del calcolo è giustificato dai teoremi di validità e completezza che enunceremo più oltre). In questo paragrafo, vediamo la prima modalità di utilizzo.

Esempio 2. Per stabilire che l'inferenza

Tutti gli uomini sono mortali.
Socrate è un uomo.
Quindi Socrate è mortale.

è corretta, sappiamo che è sufficiente stabilire che l'enunciato

$$(\forall x(U(x) \to M(x))) \wedge U(s) \to M(s) \tag{8}$$

è logicamente valido. Negando e trasformando in fnn, otteniamo l'enunciato

$$\forall x(\neg U(x) \vee M(x))) \wedge U(s) \wedge \neg M(s)$$

cui applichiamo la nostra procedura. Ciò può essere fatto nel modo seguente[15]

$$\dfrac{\dfrac{\dfrac{\forall x(\neg U(x) \vee M(x)), U(s), \neg U(s), \neg M(s) \qquad \forall x(\neg U(x) \vee M(x)), U(s), M(s), \neg M(s)}{\forall x(\neg U(x) \vee M(x)), \neg U(s) \vee M(s), U(s), \neg M(s)}}{\forall x(\neg U(x) \vee M(x)), U(s), \neg M(s)}}{\forall x(\neg U(x) \vee M(x)) \wedge U(s) \wedge \neg M(s)}$$

dove abbiamo utilizzato, nell'ordine, le regole di espansione per $\wedge, \forall$ e $\vee$: siccome tutti i nodi-foglia sono chiusi, l'enunciato in radice è insoddisfacibile, pertanto l'enunciato (8) è logicamente valido e l'inferenza è corretta.

Esempio 3. Vogliamo provare che l'enunciato

$$\forall x(P(x) \wedge Q(x)) \to \forall x P(x) \wedge \forall x Q(x) \tag{9}$$

[15]Per mantenere uniformità con la trattazione della vecchia dispensa, sviluppiamo gli alberi dal basso verso l'alto (lo studente può naturalmente svilupparli dall'alto verso il basso, se preferisce).

è logicamente valido. Negando, portando in fnn e applicando le regole, si ottiene il seguente albero:[16]

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall x(P(x)\wedge Q(x)), P(a), Q(a), \neg P(a)}{\forall x(P(x)\wedge Q(x)), P(a)\wedge Q(a), \neg P(a)}}{\forall x(P(x)\wedge Q(x)), \neg P(a)}}{\forall x(P(x)\wedge Q(x)), \exists x \neg P(x)} \qquad \dfrac{\dfrac{\dfrac{\forall x(P(x)\wedge Q(x)), P(a), Q(a), \neg Q(a)}{\forall x(P(x)\wedge Q(x)), P(a)\wedge Q(a), \neg Q(a)}}{\forall x(P(x)\wedge Q(x)), \neg Q(a)}}{\forall x(P(x)\wedge Q(x)), \exists x \neg Q(x)}}{\forall x(P(x)\wedge Q(x)), \exists x \neg P(x) \vee \exists x \neg Q(x)}$$

Dunque l'enunciato (9) è una verità logica.

Esempio 4. Il seguente esempio dimostra che la ricopiatura delle formule universali nella regola di espansione per $\forall$ è indispensabile: Vogliamo provare che l'enunciato

$$\exists x(\exists y P(y) \to P(x)) \tag{10}$$

è logicamente valido. Procedendo nell'analisi otteniamo il seguente albero:

$$\begin{array}{l}
P(a), \exists y P(y), \neg P(c), \neg P(a), \forall x(\exists y P(y) \wedge \neg P(x)) \\
\hline
P(a), \neg P(c), \exists y P(y) \wedge \neg P(a), \forall x(\exists y P(y) \wedge \neg P(x)) \\
\hline
P(a), \neg P(c), \forall x(\exists y P(y) \wedge \neg P(x)) \\
\hline
\exists y P(y), \neg P(c), \forall x(\exists y P(y) \wedge \neg P(x)) \\
\hline
\exists y P(y) \wedge \neg P(c), \forall x(\exists y P(y) \wedge \neg P(x)) \\
\hline
\forall x(\exists y P(y) \wedge \neg P(x))
\end{array}$$

Nell'ordine (dal basso verso l'alto), abbiamo utilizzato: la regola di espansione per $\forall$ relativamente a c,[17] la regola di espansione per $\wedge$, la regola di espansione per $\exists$ con l'introduzione del parametro nuovo a, la regola di espansione per $\forall$ (per la seconda volta!) relativamente al termine vecchio a e infine di nuovo la regola di espansione per $\wedge$.

Negli esempi precedenti, la procedura si conclude sempre con una **refutazione** (una refutazione è *un albero i cui nodi-foglia sono tutti chiusi*).

La possibilità che la ricerca di una refutazione non abbia mai fine è frequente: se per esempio fra le formule che etichettano un nodo-foglia corrente compare una formula

[16] Spesso useremo le lettere $a, b, c, ...$ (invece che $c_0, c_1, \ldots$ come inizialmente stabilito) per indicare i parametri.

[17] Si osservi che viene qui usato il parametro (nuovo) c per istanziare il quantificatore universale: ciò è dovuto al fatto che nessun termine ground 'vecchio' era disponibile (si veda l'eccezione segnalata in nota nel paragrafo precedente). Si tratta di un evento eccezionale (negli altri casi, l'uso di parametri nuovi per la regola di espansione del quantificatore universale produce solo inutili giri viziosi).

del tipo $\forall x \exists y R(x, y)$, allora l'applicazione combinata delle due regole di espansione dei quantificatori produce un regresso all'infinito. Iniziando dalla regola per il quantificatore universale e istanziando con un termine chiuso c_0 qualunque, si ha

$$\cdots \forall x \exists y R(x, y), \exists y R(c_0, y), \cdots$$

e poi applicando la regola per il quantificatore esistenziale (qui c_1 è forzatamente un parametro nuovo) si ottiene

$$\cdots \forall x \exists y R(x, y), R(c_0, c_1) \cdots$$

A questo punto, si può applicare la regola per il quantificatore universale istanziando x su c_1

$$\cdots \forall x \exists y R(x, y), R(c_0, c_1), \exists y R(c_1, y) \cdots$$

e poi di nuovo la regola per il quantificatore esistenziale usando un parametro nuovo c_2

$$\cdots \forall x \exists y R(x, y), R(c_0, c_1), R(c_1, c_2) \cdots$$

e così via all'infinito. Certamente qualche particolare tipo di inconveniente si potrebbe eliminare con una versione meno ingenua del calcolo, ma *non esiste nessuna versione del calcolo che assicuri sempre la terminazione*, a causa del risultato di indecidibilità cui si accennava.

La possibilità di regressi all'infinito pone un problema relativamente alla strategia di ricerca di una refutazione. Consideriamo il caso in cui compaia un nodo-foglia etichettato con

$$\forall x \exists y R(x, y), \neg P(a), P(a) \wedge Q(a)$$

Se non operiamo mai sull'ultima formula a destra, otteniamo ovviamente il regresso all'infinito che abbiamo appena visto; se invece operiamo su tale formula (con la regola di espansione per $\wedge$), chiudiamo subito. Quindi non è vero che possiamo applicare le regole in un ordine qualunque, coll'idea che 'tanto poi il risultato è sempre lo stesso'. Si possono applicare le regole in un ordine qualunque, ma bisogna rispettare il seguente criterio di equità ('fairness'): **nel generare un ramo, nessuna regola applicabile deve essere dilazionata all'infinito**. Il requisito di equità è sempre presente in una forma o nell'altra in tutti i tipi di calcoli che si utilizzano nell'ambito della dimostrazione automatica.

Possiamo formulare in modo preciso il nostro requisito di equità come segue. Una qualunque esecuzione della nostra procedura produce una successione (finita o infinita) di alberi

$$T_0 \subseteq T_1 \subseteq T_2 \subseteq \cdots \tag{11}$$

etichettati con insiemi finiti di enunciati di $\mathcal{L}^+$ (se A è la formula che si intende testare per la soddisfacibilità, T_0 consite della sola radice etichettata con $\{A\}$, mentre per ogni i si ha che T_{i+1} è ottenuto da T_i applicando ad un nodo-foglia non chiuso una delle nostre quattro regole di espansione). Consideriamo ora l'albero $T_\infty := \bigcup_i T_i$;[18] diciamo che l'esecuzione (11) è **equa** qualora ogni ramo $(r_0, r_1, r_2, \ldots)$ di T_∞ soddisfi la seguente condizione:

- per ogni i, se r_i non è chiuso e l'insieme che etichetta r_i contiene $B_1 \wedge B_2$, allora c'è un j tale che l'insieme che etichetta r_j contiene B_1, B_2;
- per ogni i, se r_i non è chiuso e l'insieme che etichetta r_i contiene $B_1 \vee B_2$, allora c'è un j tale che l'insieme che etichetta r_j contiene B_1 oppure B_2;
- per ogni i, se r_i non è chiuso e l'insieme che etichetta r_i contiene $\exists x B$, allora c'è un j tale che l'insieme che etichetta r_j contiene $B(c/x)$ per qualche c;
- per ogni i, se r_i non è chiuso, se l'insieme che etichetta r_i contiene $\forall x B$ e se il termine ground t compare nell'insieme che etichetta qualche $r_{i'}$, allora c'è un j tale che l'insieme che etichetta r_j contiene $B(t/x)$.

Ottenere esecuzioni eque è un puro fatto di *strategia implementativa*. Il modo più semplice (non necessariamente il migliore) per farlo è di mantenere le formule dell'insieme-etichetta di un nodo in una coda: si applica la relativa regola di espansione sempre alla prima formula della coda che non sia atomica e, se si deve operare una ricopiatura, la formula ricopiata va in fondo alla coda.[19]

Mediante la nozione di esecuzione equa, possiamo dare una formulazione dei teoremi classici di validità e completezza che tenga conto di qualche aspetto procedurale:

[18] Definendo in modo preciso gli alberi come insiemi non vuoti di liste di numeri naturali chiusi per prefisso, si verifica subito che l'unione di una catena di alberi è un albero. Tutte le nozioni che usiamo (ramo, foglia, ecc.) sono passibili di analoghe definizioni precise.

[19] Per la regola di espansione del quantificatore universale, occorre mantenere anche code di termini da sostituire.

Teorema 2 (Validità). *Se l'esecuzione (11) termina in una refutazione, l'enunciato A è insoddisfacibile.*

Teorema 3 (Completezza). *Se l'esecuzione (11) è equa e non termina in una refutazione, allora l'enunciato A è soddisfacibile.*

Diamo qui di seguito una copiosa serie di esercizi (tratti da temi d'esame effettivamente assegnati a studenti di informatica). Lo studente potrà svolgere il quantitativo di esercizi che egli reputa sufficiente ad ottenere un buon grado di dimestichezza con il calcolo.[20]

Negli esercizi di questo e del prossimo paragrafo (e in molti esercizi che si trovano nelle raccolte di temi d'esame) abbiamo usato l'abbreviazione $A_1, \dots, A_n \Rightarrow B_1, \dots, B_m$ per indicare la singola formula $(A_1 \wedge \cdots \wedge A_n) \to (B_1 \vee \cdots \vee B_m)$. Se $n = 0$, l'abbreviazione $\Rightarrow B_1, \dots, B_m$ sta per $B_1 \vee \cdots \vee B_m$ e, se $m = 0$, l'abbreviazione $A_1, \dots, A_n \Rightarrow$ sta per $\neg(A_1 \wedge \cdots \wedge A_n)$.

<u>*ESERCIZI*</u> Provare che le seguenti formule sono tutte logicamente valide:

1 $\forall x(P(x) \to Q(x) \wedge S(x)) \Rightarrow \forall x(P(x) \to Q(x)) \wedge \forall x(P(x) \to S(x))$.

2 $\forall x \forall y(R(x, y) \vee R(y, x)) \Rightarrow \forall x \exists y R(x, y)$.

3 $\exists y R(a, y), \forall x \forall y(P(x) \wedge R(x, y) \to S(y)) \Rightarrow \exists x(P(x) \to S(x))$.

4 $\exists x(P(x) \vee Q(x)) \to P(a) \Rightarrow (\exists x P(x) \to P(a)) \wedge (\exists x Q(x) \to P(a))$.

5 $\forall y(R(a, y) \to Q(y)), \forall x \exists y R(x, y), \forall x \forall y(R(x, y) \wedge Q(x) \to \neg Q(y)) \Rightarrow \exists x \neg Q(x)$.

6 $\forall x(P(x) \to Q(a)), \forall x(R(x) \to Q(a)) \Rightarrow \exists x(P(x) \vee R(x)) \to Q(a)$.

7 $Q(a), \forall x(Q(x) \wedge \neg P(f(x)) \to P(x)) \Rightarrow \exists x P(x)$.

[20]Il lettore, dopo aver svolto un congruo numero di esercizi, si accorgerà che molto spesso, nel generare l'albero refutativo, risulta inutile ricopiare sempre lungo un ramo una formula che si vede subito che non verrà più utilizzata. Ciò è ovviamente lecito (a patto di avere intuito correttamente che la formula in questione non servirà) nell'ambito del tipo di esercizi esaminato in questo paragrafo (non è così per gli esercizi che vedremo nel prossimo paragrafo dove nessuna formula può essere 'persa per strada', perchè tutte le formule ancora attive contribuiscono alla costruzione di un modello). In effetti, la versione del calcolo che abbiamo presentato è un po' ridondante: sarebbe forse meglio non ricopiare mai niente e ripescare 'dal basso' del ramo le formule da analizzare ('asteriscando' quelle già analizzate e non da ricopiare); tuttavia ci sembra che il calcolo che abbiamo introdotto si presti meglio all'utilizzo didattico in un corso di primo livello. Inoltre, l'uso degli asterischi non è del tutto comptibile con un'esplorazione in pura profondità degli alberi di ricerca di refutazioni.

8 $\neg\exists x(P(x) \to Q(x)) \Rightarrow \forall x P(x) \wedge \neg\exists x Q(x)$.

9 $\neg P(a) \to \exists y R(a, y), \forall x \forall y(R(x, y) \to P(x) \vee P(y) \vee \exists z P(z)) \Rightarrow \exists x P(x)$.

10 $\forall x(P(x) \to Q(f(x)) \Rightarrow \exists x P(x) \to \exists x Q(x)$.

11 $\exists x(\neg Q(a) \to R(x)), \forall x(R(x) \wedge \neg Q(a) \to Q(x)) \Rightarrow \exists x(P(x) \to Q(x))$.

12 $\forall x(P(x) \vee Q(a)) \Rightarrow \neg Q(a) \to P(b)$.

13 $\forall x(Q(x) \to \exists z P(z)) \Rightarrow \exists y(Q(y) \to P(y))$.

14 $\exists x(P(x) \wedge \forall y Q(y)) \Rightarrow \forall y \exists x(P(x) \wedge Q(y))$;

15 $\neg Q(a) \to (P(f(a)) \vee Q(g(a))) \Rightarrow \exists y(\forall x \neg P(x)$

16 $\forall x(P(x) \vee Q(x)) \to \exists y S(y) \Rightarrow \exists y \exists x((P(x) \to S(y)) \wedge (Q(x) \to S(y)))$.

17 $\Rightarrow \exists x(P(x) \to Q(x) \vee \forall z \neg Q(z))$.

18 $\exists x P(x) \to \forall y \exists x R(x, y) \Rightarrow \forall y \forall x \exists z(P(x) \to R(z, y))$.

19 $\forall y(\neg P(y) \to P(f(y))) \Rightarrow \exists x P(f(x))$.

20 $\exists x P(x) \to \forall y Q(y) \Rightarrow \forall y \forall x(\neg Q(y) \to \neg P(x))$.

21 $\forall x \exists y R(x, y), \forall x \forall y(R(x, y) \to (P(x) \leftrightarrow \neg P(y))) \Rightarrow \exists x P(x) \wedge \exists x \neg P(x)$.

22 $\forall x(P(x) \vee Q(x) \to \forall z P(z)) \Rightarrow \forall z((\exists x P(x) \to P(z)) \wedge (\exists x Q(x) \to P(z)))$.

23 $\forall x(P(x) \vee \exists y Q(y)), \forall z(Q(z) \to P(z) \vee P(f(z))) \Rightarrow \exists x P(x)$.

24 $\exists y P(y) \to \forall x(Q(x) \to S(x)) \Rightarrow \neg P(a) \vee \neg Q(b) \vee S(b)$.

25 $\exists x P(x), \forall x(P(x) \to \neg P(f(x))), \forall x(Q(x) \to Q(f(x))) \Rightarrow$
$\Rightarrow \exists x \neg(P(x) \leftrightarrow Q(x))$.

26 $\forall x(P(x) \to \neg P(f(x))) \Rightarrow \exists x(P(x) \to Q(x))$.

27 $R(a, b, c), \forall x \forall y \forall z(R(x, y, z) \to R(y, y, z)), \forall x \forall y \forall z(R(x, y, z) \to R(x, y, y)) \Rightarrow \exists x R(x, x, x)$.

28 $\forall x(P(x) \vee \neg Q(x)), \exists y(P(y) \vee Q(y)) \Rightarrow \exists x P(x)$.

29 $\forall x \forall y (R(x,y) \to Q(x)), \forall x \forall y (R(x,y) \to R(y,x)) \Rightarrow$
$\Rightarrow \forall x \forall y (R(x,y) \to Q(x) \wedge Q(y))$.

30 $\forall x (P(x) \to \exists y Q(y) \wedge \forall y R(y)) \Rightarrow \forall x (P(x) \to \exists y Q(y)) \wedge \forall x (P(x) \to R(x))$.

31 $\forall x R(x, f(a)) \Rightarrow \exists y R(g(y), y)$.

32 $\forall x \exists y \exists z (D(x,y,z) \wedge P(y)), \forall x \forall y \forall z (D(x,y,z) \wedge P(x) \wedge P(y) \to R(x,y)) \Rightarrow \exists x \exists y R(x,y)$.

33 $\forall x \exists y (R(x,y) \wedge R(y,y)), \forall x \forall y (R(x,x) \wedge R(x,y) \to \neg R(y,y)) \Rightarrow .$

34 $\forall x \forall y \forall z (D(x,y,z) \to D(x,x,y)), \exists y \exists z D(a,y,z) \Rightarrow D(a,a,a)$.

35 $\forall x R(x, f(x)), \forall x \forall y (R(f(x), f(y)) \to R(f(x), y)) \Rightarrow \exists x R(x,x)$

36 $\exists y \forall z (R(y,z) \to P(z)), \forall x \forall y \exists z (R(x,z) \wedge R(y,z)) \Rightarrow \exists y (R(a,y) \wedge P(y))$

37 $\forall x \forall y (R(x,y) \to P(x) \vee P(y)) \Rightarrow \exists y (\exists x R(x,y) \to P(y))$

38 $\forall x P(x), \exists x (Q(x) \wedge P(x) \to S(x)) \Rightarrow \exists x (P(x) \wedge S(x)) \vee \exists x \neg Q(x)$

39 $\forall y (Q(y) \wedge \neg P(y) \to P(f(y))) \Rightarrow \exists x (\exists y Q(y) \to P(x))$.

5.2 Ricerca di Modelli

Se l'enunciato A è soddisfacibile, esso ha un modello: la dimostrazione del teorema 3 dice come fare per estrarlo dal fallimento del processo di ricerca di una refutazione. Anche se non abbiamo dato tale dimostrazione, ne riassumiamo qui le informazioni salienti utili negli esercizi.

Per costruire un modello, è sufficiente esplorare **un solo ramo** dell'albero costruito dalla nostra procedura equa (purchè sia un ramo che non chiuda, s'intende). Tale ramo può essere finito od infinito, distinguiamo a tal proposito due casi:

(a) caso in cui il ramo è finito (terminazione);

(b) caso in cui il ramo è infinito (non terminazione).

È qualche volta possibile che, a seconda del ramo che si sceglie, si possa avere o meno terminazione.

Caso(a) (terminazione) Siamo autorizzati a dichiarare lo stato di terminazione **solo quando**[21]

(1) abbiamo sempre ricopiato lungo il ramo tutte le formule universali che comparivano;

(2) ci troviamo di fronte nel nodo terminale solo a formule atomiche oppure a formule il cui operatore principale è il quantificatore universale;

(3) tutte le possibili istanziazioni $B(t/x)$ (con $\forall x B$ che compare nel nodo terminale) sono già state fatte, per ogni termine ground t presente nel ramo.

Avendo dichiarato lo stato di terminazione, si costruisce il contromodello nel modo seguente. La struttura $\mathcal{A} = (\mathbf{A}, \mathcal{I})$ è cosí individuata:

- $\mathbf{A}$ è l'insieme dei termini chiusi t presenti nel ramo.
- per ogni simbolo di funzione f di arietà n, $\mathcal{I}(f)$ è la funzione che, calcolata sugli elementi $t_1, \ldots, t_n \in \mathbf{A}$, dà come risultato il termine $f(t_1, \ldots, t_n)$ se tale termine appartiene ad $\mathbf{A}$ (cioè se c'è nel ramo) e dà un valore arbitrario in caso contrario.
- $\mathcal{I}(P)$ (per P simbolo di predicato di arietà n) contiene i $\langle t_1, \ldots, t_n \rangle \in \mathbf{A}^n$ tali che $P(t_1, \ldots, t_n)$ appartiene all'insieme di enunciati che etichettano il nodo terminale.[22]

Caso(b) (non terminazione) Questo caso è più difficile, in quanto occorre prevedere dall'esterno l'andamento del ramo infinito (questo caso non è eliminabile, in quanto il calcolo non è decidibile). Si noti che non ogni ramo infinito è buono per estrarre un modello: per essere buono deve soddisfare il requisito di equità illustrato nel paragrafo precedente. Qualora si riesca a prevedere l'andamento di un ramo infinito buono, si estrae il contromodello $\mathcal{A} = \langle \mathbf{A}, \mathcal{I} \rangle$ con le solite modalità:

- $\mathbf{A}$ è l'insieme dei termini chiusi t presenti nel ramo.

[21] Non accertarsi che le tre condizioni seguenti valgano davvero costituisce errore frequente negli esercizi.

[22] Questo fatto si può esprimere in modo più sintetico dicendo che, nella struttura $\mathcal{A}$ che stiamo costruendo, si avrà $\mathcal{A} \models P(t_1, \ldots, t_n)$ precisamente qualora l'enunciato atomico $P(t_1, \ldots, t_n)$ compare nell'etichetta del nodo terminale.

- $\mathcal{I}(f)$ è definito come segue:

$$\mathcal{I}(f)(t_1,\dots,t_n) = \begin{cases} f(t_1,\dots,t_n) & \text{se } f(t_1,\dots,t_n) \in \mathbf{A} \\ \text{arbitrario,} & \text{altrimenti} \end{cases}$$

- $\mathcal{I}(P) = \{\langle t_1,\dots,t_n\rangle \mid P(t_1,\dots,t_n)$ compare nelle etichette del ramo$\}$.

Vediamo alcuni esempi.

Esempio 5. Vogliamo trovare un contromodello all'enunciato

$$\forall x(P(x) \to Q(x)) \to (\exists x P(x) \to \forall x Q(x)) \tag{12}$$

(che risulterà quindi non essere logicamente valido). A tale scopo, lo neghiamo, lo portiamo in fnn e attiviamo la nostra procedura:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{P(a), Q(a), \forall x(\neg P(x) \lor Q(x)), \neg P(b), \neg Q(b) \dots}{P(a), Q(a), \neg P(b) \lor Q(b), \forall x(\neg P(x) \lor Q(x)), \neg Q(b) \dots}}{P(a), \neg P(a) \lor Q(a), \neg P(b) \lor Q(b), \forall x(\neg P(x) \lor Q(x)), \neg Q(b)}}{P(a), \neg P(a) \lor Q(a), \forall x(\neg P(x) \lor Q(x)), \neg Q(b)}}{P(a), \forall x(\neg P(x) \lor Q(x)), \neg Q(b)}}{P(a), \forall x(\neg P(x) \lor Q(x)), \exists x \neg Q(x)}}{\exists x P(x), \forall x(\neg P(x) \lor Q(x)), \exists x \neg Q(x)}}{\forall x(\neg P(x) \lor Q(x)) \land \exists x P(x) \land \exists x \neg Q(x)}$$

Abbiamo applicato nell'ordine le regole di espansione per $\land, \exists$ (due volte), $\forall$ (due volte) e $\lor$ (due volte). I puntini indicano rami in sospeso, dovuti alle altre diramazioni della regola per $\lor$ che non è necessario analizzare (poichè stiamo cercando un contromodello, ci basta aver trovato un ramo che non chiude, indipendentemente da quanto succede sugli altri rami). Nel nodo terminale, ci sono solo formule atomiche e la formula quantificata universalmente $\forall x(\neg P(x) \lor Q(x))$: quest'ultima è stata istanziata su tutti i termini ground presenti nel ramo (che sono a e b), pertanto possiamo dichiarare lo stato di terminazione. Otteniamo il seguente contromodello $\mathcal{A}$: $\mathbf{A} = \{a, b\}$ e $\mathcal{A} \models P(a)$, $\mathcal{A} \models Q(a)$. È facile verificare che si ha proprio $\mathcal{A} \models \forall x(P(x) \to Q(x))$, $\mathcal{A} \models \exists x P(x)$ e $\mathcal{A} \not\models \forall x Q(x)$: quindi l'enunciato (12) è falsificato da $\mathcal{A}$ e pertanto non è logicamente valido.

Esempio 6. La seguente inferenza è chiramente errata:

Tutte le ciminiere fumano.
Mia nonna fuma.
Quindi mia nonna è una ciminiera.

Per provarlo, basta trovare un contromodello all'enunciato

$$\forall x(C(x) \to F(x)) \wedge F(n) \to C(n). \tag{13}$$

Ciò può essere fatto nel modo seguente:

$$\cfrac{\cfrac{\cfrac{\forall x(\neg C(x) \vee F(x)), F(n), \neg C(n) \ldots}{\forall x(\neg C(x) \vee F(x)), F(n), \neg C(n) \vee F(n), \neg C(n)}}{\forall x(\neg C(x) \vee F(x)), F(n), \neg C(n)}}{\forall x(\neg C(x) \vee F(x)) \wedge F(n) \wedge \neg C(n)}$$

dove abbiamo utilizzato, nell'ordine, le regole di espansione per $\wedge, \forall$ e $\vee$, raggiungendo lo stato di terminazione (si osservi che nell'ultimo passaggio la formula $\neg C(n)$ non è stata aggiunta perchè era già presente) .

Esempio 7. Vogliamo trovare un contromodello all'enunciato

$$\neg(\exists x P(x) \wedge (\forall y(P(y) \to P(f(y))))) \tag{14}$$

che quindi non risulterà logicamente valido. Procedendo nel solito modo si ottiene

$$\begin{array}{c}
\vdots \\
P(a), P(f(a)), \ldots, P(f^i(a)), \forall y(\neg P(y) \vee P(f(y))) \\
\hline
\vdots \\
P(a), P(f(a)), P(f(f(a))), \forall y(\neg P(y) \vee P(f(y))) \ldots \\
\hline
P(a), P(f(a)), \neg P(f(a)) \vee P(f(f(a))), \forall y(\neg P(y) \vee P(f(y))) \\
\hline
P(a), P(f(a)), \forall y(\neg P(y) \vee P(f(y))) \ldots \\
\hline
P(a), \neg P(a) \vee P(f(a)), \forall y(\neg P(y) \vee P(f(y))) \\
\hline
P(a), \forall y(\neg P(y) \vee P(f(y))) \\
\hline
\exists x P(x), \forall y(\neg P(y) \vee P(f(y))) \\
\hline
\exists x P(x) \wedge \forall y(\neg P(y) \vee P(f(y)))
\end{array}$$

Qui abbiamo chiaramente un ramo che non termina, ma di cui siamo riusciti a farci un'idea precisa (si noti che l'espressione 'farsi un'idea precisa' indica un'operazione non meccanizzabile). Otteniamo quindi il seguente contromodello $\mathcal{A}$: $\mathbf{A} = \{a, f(a), f^2(a), f^3(a), \ldots\}$

e $\mathcal{A} \models P(a), \mathcal{A} \models P(f(a)), \ldots, \mathcal{A} \models P(f^i(a)), \ldots$ Il contromodello che ci ha fornito la nostra procedura semi-automatica non è certamente il più economico possibile: potevamo ad esempio prendere una struttura con un solo elemento a, porre $f(a) = a$ e $\mathcal{I}(P) = \{a\}$. Tuttavia, la procedura semiautomatica ha il merito di fornirci (in un senso che andrebbe precisato), contromodelli al massimo grado di generalità.

ESERCIZI Provare che le seguenti formule non sono logicamente valide ed esibirne un contromodello:

1 $\exists x P(x) \wedge \exists x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$.

2 $\forall x(P(x) \rightarrow Q(x) \vee S(x)) \Rightarrow \forall x(P(x) \rightarrow Q(x)) \vee \forall x(P(x) \rightarrow S(x))$.

3 $\forall x(R(x,x) \vee \exists y R(y,x)) \Rightarrow \forall x \exists y R(x,y)$.

4 $\forall x(P(x) \vee Q(x)) \Rightarrow \forall x P(x) \vee \forall x Q(x)$.

5 $\exists x(P(x) \wedge Q(x)) \rightarrow P(a) \Rightarrow \forall x(P(x) \rightarrow P(a)) \vee \forall x(Q(x) \rightarrow P(a))$.

6 $\exists x(P(x) \rightarrow Q(x)) \Rightarrow \exists x P(x) \rightarrow \exists x Q(x)$.

7 $\exists x P(x), \forall y(P(y) \rightarrow P(f(y))) \Rightarrow$.

8 $\forall x(P(x) \rightarrow Q(x) \vee R(x)) \Rightarrow \exists x P(x) \rightarrow \forall x(Q(x) \vee R(x))$.

9 $P(f(a)), \forall x(P(f(x)) \rightarrow P(x)) \Rightarrow P(f(f(a)))$.

10 $\forall x(P(x) \vee Q(x)) \Rightarrow \forall x P(x) \vee \forall x Q(x)$.

11 $\forall x R(x, f(x)) \Rightarrow \forall x R(x,x)$.

12 $\forall x P(x) \rightarrow \exists x Q(x) \Rightarrow \exists x P(x) \rightarrow \forall x Q(x)$.

13 $\forall x \exists y R(x,y) \Rightarrow \exists z R(z,z), P(a_0)$.

14 $\forall x(P(x) \rightarrow Q(x) \vee R(x)) \Rightarrow \forall x P(x) \rightarrow \forall x Q(x), \forall x P(x) \rightarrow \forall x R(x)$;

15 $\forall x R(f(x), x) \Rightarrow \exists x R(a, f(x))$.

16 $\forall x P(x) \rightarrow \forall x Q(x) \Rightarrow \forall x(P(x) \rightarrow Q(x))$.

17 $\exists x P(x) \Rightarrow \exists y(P(y) \to \forall z Q(z))$.

18 $\neg\forall x \exists z R(x,z) \Rightarrow \exists x R(a,x)$.

19 $\forall x(\neg P(x) \to \exists y(R(x,y) \wedge \neg P(y))) \Rightarrow P(a_0)$.

20 $\neg\forall x R(a,x) \Rightarrow \forall x \exists z \neg R(x,z)$.

21 $\forall x(P(x) \to \exists y(R(x,y) \wedge P(y))) \Rightarrow \neg P(a_0)$.

22 $\exists x(R(a,x) \to S(a,x)) \Rightarrow \exists y R(a,y)$.

23 $\forall x \exists y R(x,y) \Rightarrow \forall x \forall y(R(x,y) \to (P(x) \leftrightarrow \neg P(y)))$.

24 $\forall y R(a,y), \exists x(R(a,x) \to S(a,x)) \Rightarrow$.

25 $\forall x \exists y R(x,y) \Rightarrow \forall x \forall y(R(x,y) \vee (P(x) \leftrightarrow \neg P(y)))$.

26 $\forall x(P(x) \vee Q(x) \to S(x)) \Rightarrow \forall z(\forall x P(x) \vee \forall x Q(x) \to P(z))$.

27 $\exists x R(x,x) \Rightarrow \exists x R(x,f(x))$.

28 $\forall x(P(x) \vee Q(x) \to S(x)) \Rightarrow \forall z \exists x \exists y(P(x) \vee Q(y) \to P(z))$.

29 $\exists x R(f(x),x) \Rightarrow \exists x R(x,f(x))$.

30 $\exists x Q(x), \exists x \forall y R(x,y) \Rightarrow \exists x(\neg Q(x) \wedge R(x,x))$.

31 $\forall x \exists y R(x,y) \Rightarrow \forall x \forall y(R(x,y))$.

32 $\forall x(P(x) \to Q(x) \vee R(x)) \Rightarrow \forall x(P(x) \to Q(x)) \vee \forall x R(x)$.

33 $\forall x(P(x) \to P(g(x))) \Rightarrow \forall x(P(x) \to \exists y Q(y))$.

34 $\forall x(P(x) \to \forall y R(x,y)) \Rightarrow \exists x P(x) \to \forall x \forall y R(x,y)$.

35 $\forall x \forall y(R(x,y) \to Q(x)), \forall x \forall y(R(x,y) \to R(y,x)) \Rightarrow$
$\Rightarrow \forall x \forall y(Q(x) \wedge Q(y) \to R(y,x))$.

36 $\forall x(R(x,x) \to Q(x)) \Rightarrow \forall x \forall y(R(x,y) \to Q(x))$.

37 $\exists x R(x,x) \Rightarrow \exists x R(x,f(x))$.

38 $\exists x \forall y R(x,y) \Rightarrow \forall x R(x,x)$.

39 $\forall x \forall y (R(x,y) \vee R(y,x)) \Rightarrow \exists x \forall y R(x,y)$.

40 $\forall x \forall y (R(x,y) \vee R(y,x)) \Rightarrow \forall x \forall y \forall z (R(x,y) \wedge R(y,z) \rightarrow R(x,z))$.

41 $\forall x R(f(x),x) \Rightarrow \forall x \exists y R(x,y)$.

42 $\forall x R(x) \Rightarrow \forall x (\exists y P(y) \rightarrow Q(x))$.

43 $R(a_0,b_0), \forall x (R(x,a_0) \vee R(x,b_0)) \Rightarrow \exists x \forall y R(y,x)$.

44 $\forall x \forall y (R(x,y) \vee P(x)) \Rightarrow \forall x R(x,f(x))$.

45 $R(a,b), \forall x \forall y (R(x,y) \rightarrow R(f(x),f(y))) \Rightarrow$

46 $\forall x \forall y (R(x,y) \rightarrow R(y,x)) \Rightarrow \forall x \forall y \forall z (R(x,y) \wedge R(y,z) \rightarrow R(x,z))$

47 $\forall x R(x,f(x)), \forall x \forall y (R(f(x),f(y)) \rightarrow R(x,y)) \Rightarrow$

48 $\forall x (P(x) \vee Q(x)), \exists x P(x), \exists x Q(x) \Rightarrow \exists x (P(x) \wedge Q(x))$

49 $\forall x (P(f(x)) \vee P(g(x))) \Rightarrow \forall x (P(f(x)) \vee \neg P(g(x)))$

50 $\forall x R(x,x), \forall x \forall y (R(x,y) \rightarrow R(y,x)) \Rightarrow \exists x (R(a,x) \wedge R(x,b))$

51 $\exists x (R(f(x),x), \forall x \forall y (R(x,y) \rightarrow R(f(x),f(y))) \Rightarrow$

52 $\Rightarrow \forall z (\forall x D(x,x,z) \rightarrow \forall y D(z,y,y))$

53 $\exists y \forall x R(x,y) \Rightarrow \forall x \forall y (R(x,y) \vee R(y,x))$.

5.3 Ulteriori Esempi

In questo paragrafo mostriamo come utilizzare quanto abbiamo appreso in alcuni ambiti specifici. Tenendo conto della esiguità del materiale di un corso propedeutico, gli esempi che potremo presentare saranno piuttosto gracili, speriamo tuttavia che essi aiutino a comprendere l'utilità dell'approccio logico a problemi di varia natura (linguistico, matematico, informatico).

I limiti a cui dobbiamo sottostare sono di una duplice natura: a) di natura espressiva, dovuti al fatto che sappiamo trattare solo problemi formalizzabili nella logica del primo ordine senza identità; b) di natura realizzativa, dovuti al fatto che il calcolo che abbiamo presentato (mancando ad esempio del meccanismo fondamentale dell'unificazione, presente invece nella versione con variabili libere dei tableaux) non è molto efficiente per problemi significativi.

Esempio 8. Supponiamo di voler formalizzare il seguente ragionamento: '*Paolo e Carlo sono fratelli; Paolo ha un cugino, quindi anche Carlo ha un cugino*'. Questo ragionamento non è puramente logico, esso soggiace a 'postulati di significato' (nel senso del filosofo Carnap) relativi all'uso delle parole 'fratello, cugino'; tali postulati devono essere specificati interamente per procedere ad un'opportuna formalizzazione. Siccome non abbiamo analizzato le regole logiche per l'identità, invece del linguaggio $\mathcal{L}_1$ del paragrafo 2, scegliamo un linguaggio più semplice contenente una sola relazione binaria $G(x, y)$ (che leggiamo con 'x è genitore di y'). In tale linguaggio possiamo definire $F(x, y)$ ('x è fratello di y') con $\forall z(G(z, x) \leftrightarrow G(z, y))$ ed anche $C(x, y)$ ('x è cugino di y') con $\exists z \exists w(F(z, w) \wedge G(z, x) \wedge G(w, y))$. Usiamo la costante a per Andrea e la costante c per Carlo. Si tratta ora semplicemente di provare che l'enunciato

$$F(a, c) \wedge \exists y C(a, y) \to \exists y C(c, y)$$

è logicamente valido (si ricordi che C e F sono in realtà abbreviazioni per le formule che abbiamo introdotto sopra). Ritroviamo quindi un esercizio analogo a quelli del paragrafo 5.1, che può essere facilmente risolto con le tecniche di ricerca di una refutazione là illustrate.

Esempio 9. In alcuni casi si possono ritrovare inferenze che possono sembrare un po' paradossali, ma il paradosso risiede tutto nell'ambiguità del linguaggio ordinario. Dalle due premesse (che possiamo ritenere plausibilmente vere) *Alcuni pazienti amano tutti i dottori* e *Nessun paziente ama un incapace*, formalizzate rispettivamente con[23]

$$\exists x(P(x) \wedge \forall y(D(y) \to A(x, y)))$$

e con

$$\forall y \forall x(P(x) \wedge A(x, y) \to \neg I(y))$$

[23]Abbiamo usato tre predicati unari $P(x), D(x), I(x)$ (rispettivamente per 'x è un paziente, un dottore, un incapace') e un predicato binario $A(x, y)$ (per 'x ama y'). Le traduzioni dal linguaggio ordinario al linguaggio formalizzato che facciamo sono piuttosto empiriche; lo studio di metodi sistematici per tali traduzioni rientra nel settore della semantica dei linguaggi naturali.

segue il fatto (plausibilmente falso) che *Non esistono dottori incapaci.* Questo perchè possiamo appurare con le nostre tecniche che l'enunciato

$$\exists x(P(x) \land \forall y(D(y) \to A(x,y))) \land \forall y \forall x(P(x) \land A(x,y) \to \neg I(y)) \to \neg \exists x(D(x) \land I(x))$$

è logicamente valido. La spiegazione dell'apparente paradosso sta nel fatto che *incapace* è usato in senso soggettivo nella seconda premessa e in senso oggettivo nella conclusione dell'inferenza.

Esempio 10. Anche il metodo di ricerca di contromodelli del paragrafo 5.2 può essere utile in esempi linguistici concreti. Se manteniamo le due premesse dell'esempio precedente, possiamo chiederci se da esse segue che *Esistono dottori amati da tutti i pazienti*; questo equivale a chiederci se l'enunciato

$$\exists x(P(x) \land \forall y(D(y) \to A(x,y))) \land \forall y \forall x(P(x) \land A(x,y) \to \neg I(y)) \to$$

$$\to \exists x(D(x) \land \forall y(P(y) \to A(y,x))$$

sia o meno logicamente valido. Di fatto non lo è e la ricerca di un contromodello produce varie possibilità in proposito (a seconda del ramo non chiuso che si sceglie). Ad esempio, si ottiene facilmente che un contromodello è fornito da una situazione limite in cui c'è una sola persona in gioco, che è un paziente, non è un dottore e non è un incapace.

Esempio 11. Tutti i ragionamenti che si trovano nei testi matematici possono essere formalizzati in un opportuno ambito logico. Questo fatto può essere usato per costruire dimostratori automatici di teoremi. Il vero problema che si ritrova in questo settore sta non solo nell'indecidibilità dei calcoli logici, ma anche nelle svariatissime possibilità che ci sono nella ricerca delle dimostrazioni; tali possibilità provocano facilmente esplosioni combinatorie tali da non essere sopportabili nemmeno da calcolatori molto potenti. Solo con tecniche molto raffinate si può ottenere qualche risultato concreto. Vediamo però come riprodurre formalmente un ragionamento matematico semplicissimo con i pochi strumenti che abbiamo. Consideriamo un grafo[24] transitivo in cui da ogni vertice si diparta almeno un arco; vogliamo provare che tale grafo è riflessivo. Il ragionamento informalmente è il seguente: prendiamo un vertice a, tracciamo un arco aRb (che esiste per ipotesi); poi osserviamo che, per simmetria, c'è un arco bRa e, infine, per transitività da $aRbRa$ concludiamo che a è un vertice riflessivo (cioè che vale aRa). Essendo a arbitrario, abbiamo

[24]Per grafo si intende un insieme dotato di una relazione simmetrica.

la conclusione. Formalmente, si tratta di provare che l'enunciato

$$\forall x \forall y(R(x,y) \to R(y,x)) \wedge \forall x \forall y \forall z(R(x,y) \wedge R(y,z) \to R(x,z)) \wedge \forall x \exists y R(x,y) \to \forall x R(x,x)$$

è logicamente valido. I passi della dimostrazione formale fatta con le tecniche del paragrafo 5.1 corrispondono esattamente al ragionamento matematico intuitivo che abbiamo appena illustrato. 'Prendere un vertice a' corrisponde a utilizzare una prima regola di espansione per $\exists$; 'tracciare un arco aRb' corrisponde a usare le regole per $\forall$ e poi per $\exists$ relativamente alla premessa $\forall x \exists y R(x,y)$. Osservare la presenza dell'arco in senso opposto bRa corrisponde ad utilizzare la regola per $\forall$ relativamente alla premessa che esprime la simmetria di R. Infine l'utilizzo della transitività (istanziata su a, b, a) corrisponde all'utilizzo della regola per $\forall$ sulla premessa che esprime la transitività (la chiusura della ricerca della refutazione si ottiene infine mediante le regole per i connettivi proposizionali $\wedge, \vee$).

Esempio 12. Il metodo del parafrafo 5.2 può essere utilizzato per ottenere controesempi a congetture matematiche, sempre con i soliti limiti (di indecidibilità e di complessità della ricerca), che rendono la strada difficilemente praticabile su esempi significativi (quantomeno con gli strumenti che abbiamo). Ad esempio, un grafo non necessariamente è transitivo; se inizializziamo il nostro procedimento di ricerca di un modello con la negazione dell'enunciato

$$\forall x \forall y(R(x,y) \to R(y,x)) \to \forall x \forall y \forall z(R(x,y) \wedge R(y,z) \to R(x,z))$$

otteniamo (dopo aver eliminato i quantificatori esistenziali con i parametri a, b, c e dopo aver applicato la regola per $\wedge$)

$$\forall x \forall y(\neg R(x,y) \vee R(y,x)), R(a,b), R(b,c), \neg R(a,c).$$

A questo punto, dobbiamo istanziare i quantificatori universali di $\forall x \forall y(\neg R(x,y) \vee R(y,x))$ sui parametri a, b, c in tutti i modi possibili (che sono ben 9!) e applicare esaustivamente la regola per $\vee$. Alcuni rami chiudono, tra i rami che non chiudono abbiamo ad esempio

$$\forall x \forall y(\neg R(x,y) \vee R(y,x)), R(a,b), R(b,c), R(b,a), R(c,b), \neg R(a,c).$$

Possiamo ora dichiarare lo stato di terminazione ed estrarre il contromodello che sarà un grafo con tre vertici (cioè a, b, c) e con due archi (tra a e b e tra b e c, rispettivamente). Gli altri rami che non chiudono danno contromodelli leggermente differenti, in cui alcuni dei vertici a, b, c possono essere riflessivi.