

CAPITOLO 5

[indice](#)

IL CALCOLO DEI PREDICATI.

1. Linguaggi del primo ordine: alfabeto e termini.

Il calcolo proposizionale parte da formule atomiche che indica con simboli come p_1, p_2, \dots . Invece il calcolo dei predicati effettua una ulteriore scomposizione delle formula atomiche evidenziandone la struttura "soggetto-predicato". Pertanto un linguaggio adeguato al calcolo dei predicati dovrà contenere dei nomi per le cose e dei nomi per i predicati. Nell'ambito matematico è inoltre naturale introdurre anche nomi per funzioni ed operazioni. Sembra ad esempio ragionevole avere un linguaggio in cui poter scrivere una frase come "2 è pari" che si avvale del "nome proprio" 2 e del predicato "pari". Oppure deve essere possibile scrivere parole come " $\log(2+3) \geq 0$ " che si avvale del nome di funzione \log , dei nomi 2, 3 e 0, del nome di operazione + e del simbolo di relazione binaria \geq . Ancora, in matematica sono di uso frequente le variabili ed i quantificatori (esiste, per ogni) che permettono, ad esempio, di affermare che vale una legge generale come la proprietà commutativa $\forall x \forall y (x \cdot y = y \cdot x)$ oppure di dire che non vale una legge generale, come ad esempio che non vale la proprietà commutativa $\exists x \exists y (x \cdot y \neq y \cdot x)$. Ciò suggerisce di considerare un alfabeto del tipo seguente.

Definizione 1.1. Chiamiamo *alfabeto di un linguaggio del primo ordine* un alfabeto A costituito da:

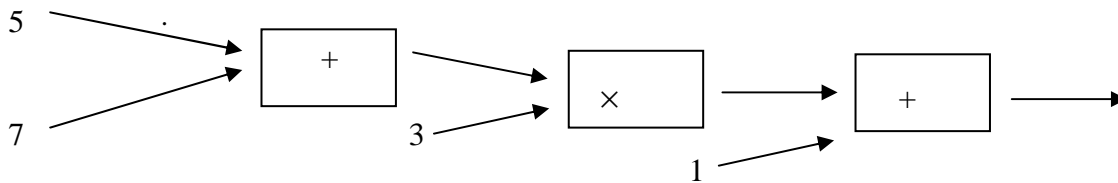
- una serie infinita di variabili, x_1, x_2, x_3, \dots ;
- i connettivi proposizionali $\{\wedge, \vee, \neg\}$;
- il quantificatore esistenziale \exists ;
- la parentesi aperta "(", la parentesi chiusa ")" e la virgola "," ;
- un insieme finito c_1, \dots, c_p di elementi detti *costanti* ;
- un insieme finito o_1, \dots, o_s di elementi chiamati *nomi di operazioni* ;
- un insieme finito r_1, \dots, r_t di elementi chiamati *nomi di relazioni*.
- una funzione $ar : \{o_1, \dots, o_s, r_1, \dots, r_t\} \rightarrow N$ chiamata *arità*.

Se $ar(x) = n$ allora si dice che x rappresenta una operazione oppure una relazione n -aria. Si suppone che nell'alfabeto ci sia almeno un simbolo di relazione in modo che, come vedremo, sia possibile costruire le asserzioni (quasi sempre si ammette l'uso del simbolo "="). Da notare che nel linguaggio comune si parla di *predicati* per indicare le relazioni 1-arie mentre si preferisce parlare di *relazioni* nel caso $n \geq 2$. In generale si considerano solo operazioni e relazioni unarie (arità uno) o binarie (arità due). Il ruolo delle costanti è di dare nomi a particolari elementi delle strutture che si vogliono descrivere. Naturalmente esistono tanti alfabeti, e quindi tanti linguaggi del primo ordine, quanti sono i modi di specificare le costanti, i nomi delle relazioni e delle operazioni. Avremo ad esempio un linguaggio (o più linguaggi) adeguato alla teoria dei gruppi, uno per la teoria degli anelli, uno per le strutture ordinate, e così via. In generale vengono presi in considerazione solo un numero finito di relazioni e di funzioni.

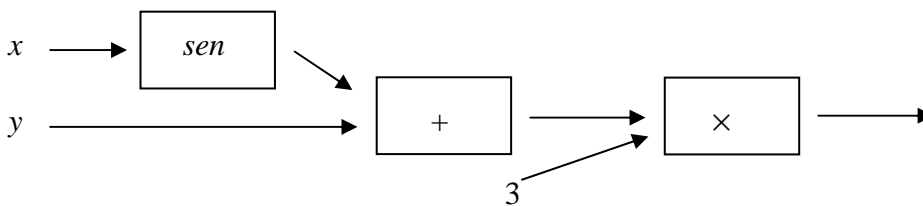
Il linguaggio dei termini: Per poter definire il linguaggio del primo ordine corrispondente ad un dato alfabeto dobbiamo prima definire il linguaggio dei termini. Intuitivamente i termini sono tutte le "descrizioni di funzioni" che si possono costruire a partire dalle variabili, dalle costanti e dai nomi di funzioni di un dato linguaggio. Un esempio di termine è la parola $\log(\sin(x)+y)$. Un altro modo di vedere un termine è come descrizione di un algoritmo per il calcolo di una funzione (se il termine contiene variabili) o di un elemento (se il termine non contiene variabili). Ad esempio al termine $3 \cdot (5+7)+1$ corrisponde l'algoritmo:

1. prendi i numeri 5 e 7
2. sommalì
3. moltiplica il risultato per 3
4. aggiungi 1

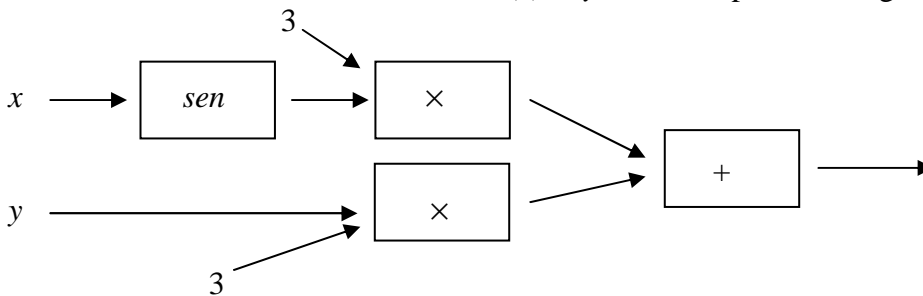
Possiamo schematizzare tale algoritmo anche al modo seguente dove in ogni rettangolo abbiamo supposto di mettere una “processore” che esegue una operazione:



Similmente al termine $3 \cdot (\text{sen}(x)+y)$ corrisponde l'algoritmo



Tale termine è diverso dal termine $3 \cdot \text{sen}(x)+3 \cdot y$ che corrisponde all'algoritmo



Naturalmente questi due algoritmi, se si interpretano al modo usuale sull'insieme dei numeri reali, determinano la stessa funzione.

Informalmente un termine è una parola che si può costruire secondo le seguenti regole:

- ogni variabile è un termine
- ogni costante è un termine
- se f è il nome di una funzione n -aria e t_1, \dots, t_n sono termini allora $f(t_1, \dots, t_n)$ è un termine.

Una definizione più rigorosa, in termini di grammatiche, è la seguente:

Definizione 1.2. Il *linguaggio dei termini* è il linguaggio la cui grammatica

- ha come alfabeto terminale l'insieme delle variabili, delle costanti, delle parentesi e dei nomi di funzioni

- ha come variabili ausiliarie $\{ter\}$

- ha le seguenti produzioni

$ter \rightarrow c$ dove c è una costante

$ter \rightarrow x_i$ dove x_i è una variabile

$ter \rightarrow (ter)o(ter)$ dove o è un nome di una operazione binaria

$ter \rightarrow o(ter, \dots, ter)$ dove o è il nome di una operazione n -aria, $n \neq 2$ e ter è scritto n volte.

Naturalmente i termini di complessità zero sono le variabili e le costanti; i termini di complessità 1 sono del tipo $o(t_1, \dots, t_n)$ dove t_1, \dots, t_n sono o variabili oppure costanti.

La notazione utilizzata per costruire i termini con $n \neq 2$ si dice *prefissa* e consiste nel mettere il simbolo che denota l'operazione avanti ai termini cui si applica. La notazione nel caso $n = 2$ si dice *infissa* e consiste nel mettere il simbolo denotante l'operazione in mezzo ai due termini. Questo significa che se \otimes è un nome di operazione binaria allora di solito si scrive $(t_1)\otimes(t_2)$ oppure $t_1\otimes t_2$ al posto di $\otimes(t_1,t_2)$. Così scriveremo x_1+x_2 e non $+(x_1,x_2)$ e $(x_1+x_2)+x_3$ al posto di $+(+(x_1,x_2),x_3)$ ¹.

A volte utilizzeremo altri simboli per denotare le variabili x_1, x_2, \dots , ad esempio i simboli x, y, z . Infine, dato un termine t , scriveremo $t(x_1, \dots, x_n)$ per indicare che sue variabili sono tra x_1, \dots, x_n .

Esempio. Supponiamo che tra i nomi di funzioni vi siano *log*, *sen*, $+$, e la costante 1, allora le produzioni sono

$$ter \rightarrow 1$$

$$ter \rightarrow x_i$$

$$ter \rightarrow \log(ter)$$

$$ter \rightarrow \text{sen}(ter)$$

$$ter \rightarrow (ter) + (ter).$$

In tale caso è facile vedere che sono esempi di termini le formule

$$(\text{sen}(x_2)+1)+\log(\text{sen}(x_1)), \log(\text{sen}(x_2+1)), \log(\log(x_4)), \dots$$

Ad esempio una derivazione di $\log(\text{sen}(x_2+1))$ è data dalla sequenza

$$ter,$$

$$\log(ter),$$

$$\log(\text{sen}(ter)),$$

$$\log(\text{sen}(ter+ter)),$$

$$\log(\text{sen}(ter+1)),$$

$$\log(\text{sen}(x_2+1))$$

Il linguaggio della teoria degli anelli. Il linguaggio della teoria degli anelli ha come operazioni " $+$ ", " $-$ " e " \cdot ", e come costanti 0 e 1. Sono esempi di termini chiusi le parole 1, $1+0$, $1+1$, $(1+1)+1 \dots (1+1)\cdot(1+0) \dots$

In tale linguaggio a volte si denota con 2 il termine $1+1$, con 3 il termine $(1+1)+1$ e così via. Pertanto un termine come $((1+1)+1)\cdot x + (1+1)\cdot x$ viene indicato con $3\cdot x+2\cdot x$.

Problema. Scrivere cinque termini nel linguaggio il cui alfabeto contiene $\{\log, x, y, \text{sen}, +, 1\}$.

2. Linguaggi del primo ordine: le asserzioni.

Passiamo ora a definire l'insieme \mathcal{L} delle "asserzioni" o "formule" di un linguaggio del primo ordine. In maniera intuitiva possiamo dire che \mathcal{L} è l'insieme delle parole su di un alfabeto del primo ordine definito dalle seguenti regole di formazione:

- sono *formule atomiche* tutte le espressioni del tipo $r(t_1, \dots, t_n)$ dove r è il nome di una relazione di arità n e t_1, \dots, t_n sono termini;
- se α e $\beta \in \mathcal{L}$ allora $(\alpha)\wedge(\beta)$, $(\alpha)\vee(\beta)$ e $\neg(\alpha)$ appartengono a \mathcal{L} ;
- se x_i è una variabile ed $\alpha \in \mathcal{L}$ allora $\exists x_i(\alpha) \in \mathcal{L}$

¹ Esistono anche notazioni "*post-fisse*" per le operazioni unarie, ad esempio la funzione fattoriale si indica con $x!$ Esistono inoltre notazioni "*esponenziali*" come la funzione inverso x^{-1} . Nel seguito non seguiremo in modo pedante la grammatica per i linguaggi del primo ordine che abbiamo proposto ed accetteremo tutti i modi di scrivere usualmente utilizzati dai matematici.

In termini di grammatica la definizione è la seguente.

Definizione 2.1. Un *linguaggio del primo ordine* è il linguaggio \mathcal{L} definito da una grammatica che ha

- come alfabeto terminale l'insieme dei: nomi di relazione, nomi di funzione, costanti, variabili, connettivi logici, le parentesi e la virgola;
- come variabili ausiliarie "*ter*", "*fbf*", "*at*"
- come start-symbol il simbolo "*fbf*"
- come regole di produzione

$ter \rightarrow c_i$ per ogni costante c_i

$ter \rightarrow x_i$ per ogni indice i

$ter \rightarrow o(ter, \dots, ter)$ con o nome di operazione e, se $ar(o) = n$, ter ripetuto n volte

$at \rightarrow r(ter, \dots, ter)$ con r nome di relazione e se $ar(r) = n$, ter ripetuto n volte

$fbf \rightarrow at$

$fbf \rightarrow (fbf) \wedge (fbf)$; $fbf \rightarrow (fbf) \vee (fbf)$; $fbf \rightarrow \neg (fbf)$; $fbf \rightarrow \exists x_i (fbf)$.

Gli elementi di \mathcal{L} vengono chiamate *formule ben formate*, o, più semplicemente, *formule*².

Nota. Come abbiamo già osservato per le operazioni, i matematici preferiscono utilizzare per le relazioni binarie la *notazione infissa*. Ad esempio, se si considerano le relazioni binarie $=$ e \leq , preferiscono scrivere $t_1 = t_2$, $t_1 \leq t_2$ al posto di $=(t_1, t_2)$ e $\leq(t_1, t_2)$. Se si accetta tale modo di scrivere le formule allora la parte della grammatica che definisce *at*, cioè le formule atomiche, dovrebbe essere riscritta al modo seguente

$at \rightarrow r(ter, \dots, ter)$ dove r è un nome di relazione e se $ar(r) = n$, ter è ripetuto n volte

$at \rightarrow ter = ter$ (se l'alfabeto contiene l'eguaglianza)

Nota. Abbiamo deciso di assumere come quantificatore solo il quantificatore esistenziale \exists . Naturalmente in matematica è usato sistematicamente anche il quantificatore universale \forall . D'altra parte poiché ogni formula del tipo $\forall x_i(\alpha)$ è equivalente alla formula $\neg \exists x_i(\neg \alpha)$, è preferibile utilizzare l'espressione $\forall x_i$ come abbreviazione di $\neg \exists x_i \neg$, cioè scrivere $\forall x_i(\alpha)$ per denotare la formula $\neg \exists x_i(\neg \alpha)$. In altri termini è preferibile assumere come primitivo il simbolo \exists e poi definire opportunamente \forall . Questa scelta consente di non complicare inutilmente le dimostrazioni per induzione sulla complessità delle formule considerando separatamente i due casi $\forall x_i(\alpha)$ e $\exists x_i(\alpha)$. Da notare che nel fare questa scelta risulta che $\forall x_i(\alpha)$ è un simbolo del mio linguaggio naturale (metalinguaggio) con cui denoto una

² Abbiamo esposto la nozione di linguaggio del primo ordine in cui, come vedremo, le variabili e le costanti sono interpretate con un solo tipo di oggetti, quelli che appartengono al dominio D . Ad esempio la nozione di spazio vettoriale che richiede due tipi di variabili (scalari e vettori) non esprimibile in tale logica. Bisogna ricorrere alle *logiche a più tipi* che permettono di trattare tipi diversi di oggetti. Comunque tutti i risultati che valgono per la logica ad un tipo che stiamo considerando valgono anche nelle logiche a più tipi.

Altre volte è necessario quantificare sui sottoinsiemi del dominio e non solo sugli elementi. Ad esempio una affermazione del tipo "ogni insieme di numeri naturali ammette minimo", essendo della forma $\forall X (X \subseteq \mathbb{N} \rightarrow \dots)$ non può essere descritta nella logica del primo ordine. Le logiche che permettono la quantificazione sui sottoinsiemi del dominio prendono il nome di *logiche del secondo ordine* mentre la logica definita in questo capitolo viene chiamata logica del primo ordine ed il calcolo proposizionale del capitolo precedente logica di ordine zero. Mentre le logiche del primo ordine a più tipi non sono molto differenti da quella del primo ordine ad un solo tipo, le logiche del secondo ordine hanno una teoria completamente diversa che noi non esamineremo.

asserzione del linguaggio oggetto. Naturalmente sarebbe stato anche possibile assumere come primitivo il quantificatore \forall e definire \exists come abbreviazione di $\neg\forall\neg$. A volte si assumono come primitivi entrambi i quantificatori.

Abbreviazioni. Una ulteriore abbreviazione consiste nel considerare una espressione del tipo $\alpha \rightarrow \beta$ come abbreviazione della formula $\neg\alpha \vee \beta$ e $\alpha \leftrightarrow \beta$ una abbreviazione di $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. Con questa convenzione i simboli \rightarrow e \leftrightarrow appartengono al mio linguaggio (metalinguaggio) e servono a denotare particolari formule del linguaggio oggetto.

I seguenti sono alcuni esempi di linguaggi del primo ordine utilizzati in matematica.

Linguaggio usato per le strutture ordinate. E' un linguaggio che contiene il solo simbolo \leq di relazione binaria ed il simbolo di eguaglianza. A volte si aggiunge anche una costante 0 (da interpretare come minimo elemento) e una costante 1 (da interpretare come massimo elemento). Poiché non ci sono nomi di operazioni, gli unici termini sono le variabili e le costanti. Sono esempi di formule

$$\forall x(x \geq 0); \forall x \exists y(x \geq y); \forall x_1(x_1 \leq x_1); \forall x_1(\forall x_2((x_1 \leq x_2) \wedge (x_2 \leq x_3) \wedge (x_1 \leq x_3))).$$

Linguaggio usato per la teoria dei gruppi. L'alfabeto di tale linguaggio contiene l'uguaglianza, il simbolo "." per rappresentare l'operazione binaria, "inv" per rappresentare l'operazione che associa ad ogni x il suo inverso e la costante 1 per rappresentare l'elemento neutro. In generale si preferisce la notazione x^{-1} al posto di $inv(x)$. I termini sono l'espressioni del tipo $(x \cdot y) \cdot x^{-1}$, $(1 \cdot x)^{-1} \cdot y$. Sono esempi di formule:

$$\forall x(x \cdot y = y \cdot x); \quad x \cdot 1 = x; \quad \forall x(x = 1 \rightarrow x \cdot x = 1).$$

Naturalmente è possibile utilizzare anche linguaggi diversi per trattare la stessa classe di strutture matematiche. Ad esempio per la teoria dei gruppi si usa spesso la notazione additiva invece di quella moltiplicativa che abbiamo indicato. In questo caso si utilizza 0 come costante, il simbolo + per l'operazione binaria ed il simbolo - per l'operazione unaria.

Alcuni tipi di formule prendono un nome particolare, ad esempio nel seguito chiameremo

- *letterale* ogni formula che sia atomica (*letterale positivo*) o negata di una formula atomica (*letterale negativo*)
- *matrice* ogni formula priva di quantificatori
- *matrice positiva* ogni matrice in cui non compare la negazione
- *formula universale* ogni formula che sia una matrice o che si ottiene da una matrice antepoendo quantificatori universali.
- *formula universale positiva* ogni formula universale in cui non è presente la negazione.

Esempi. La formula $x \leq 5$ è un letterale positivo, la formula $\neg(x \geq 5)$ è un letterale negativo. Le formule $x < 5$ e $(x \leq 5) \vee (x \geq 5)$ sono matrici positive. La formula $(x \leq 5) \vee (\neg(x \geq 5))$ è un esempio di matrice (non positiva). Sono esempi di formule universali le formule

$$\forall x((x \leq 5) \vee (\neg(x \geq 5))), \quad \forall x \forall y(\neg((x \leq 5) \wedge (x \geq 5)))$$

Sono esempi di formule universali positive le formule

$$\forall x((x \leq 5) \vee (x \geq 5)), \quad \forall x \forall y(\neg((x \leq 5) \wedge (x \geq 5))).$$

Da notare che una formula come $(x \leq 5) \rightarrow (x \leq 7)$ è una matrice ma non è da considerarsi positiva. Infatti è una abbreviazione metalinguistica della formula $\neg(x \leq 5) \vee (x \leq 7)$. Le formule con grado di complessità zero sono le formule atomiche, esempi di formule di grado di complessità 1 sono $(x+1 = 1+x) \vee (x=1)$, $\exists x_2(x_2 \cdot x_2 = 2)$.

La nozione di *sottoformula* di una data formula è evidente. La sottoformula α di $\exists x_i(\alpha)$ viene detta *campo di azione* del quantificatore \exists . Una occorrenza di x_i in una formula viene detta *vincolata* se entra nel campo di azione di un quantificatore del tipo $\exists x_i$, altrimenti viene detta *libera*. Una formula senza variabili libere viene detta *chiusa*. Ad esempio in $\exists x_2(x_2+3x_1 \geq x_2)$ l'occorrenza di x_1 è libera, quella di x_2 è vincolata.

La formula $\exists x_1(\exists x_2(x_2+3x_1 \geq x_2))$ è chiusa. Una stessa variabile può avere una occorrenza libera ed una occorrenza vincolata nella stessa formula. Ad esempio, nella formula $(\exists x_2(x_2+3x_1 \geq x_2)) \wedge (x_2+3x_1 \geq x_2)$ la variabile x_2 ha una occorrenza libera ed una vincolata.

Problema. Dire in quali formule la variabile x_1 è libera

$$\exists x_2(x_2+3x_1 \geq x_2) \rightarrow \forall x_1(x_1=x_1), \quad x_2+3x_1 \geq x_2, \quad \forall x_1(\exists x_2(x_2+3x_1 \geq x_2)), \quad \exists x_3(\exists x_2(x_2+3x_1 \geq x_2)).$$

Esercizio. Dimostrare che $((1+1) \cdot (x_2+0)) - x_4$ è un termine del linguaggio della teoria degli anelli.

Esempio. Consideriamo un linguaggio individuato dalle costanti a, b , dalla relazione binaria r e dalla funzione unaria f , allora sono esempi di termini,

$$a, b, x_2, f(a), f(b), f(f(a)), f(x_2), \dots$$

sono esempi di formule:

$$r(x_3, x_4), \quad r(a, f(x_4)), \quad \exists x_3(r(a, b)), \quad \forall x_1((r(f(a), x_1))), \quad \exists x_3((r(a, f(x_1))) \wedge (r(a, b))).$$

Supponiamo che vi sia anche un nome di una operazione binaria $+$, allora sono esempi di termini (oltre e quelli scritti sopra)

$$a+b, a+x_1, x_1+f(x_2), x_2+(f(x_2+x_3)+a), \dots$$

Noi non ci atterremo sempre alla rigida definizione di formula sopra data e spesso ci avvicineremo all'uso corrente del linguaggio matematico. Scriveremo ad esempio $x \leq y$ al posto di $\leq(x, y)$ ed $x=y$ al posto di $=(x, y)$. Ancora, potremo scrivere $\forall x_1(r(x, y))$ per denotare la formula $\neg \exists x_1(\neg(r(x_1, x_3)))$.

Nota. Negli usuali linguaggi di programmazione è possibile considerare solo una piccolissima parte della logica dei predicati. Ad esempio sono usualmente permesse istruzioni del tipo

$$\text{IF } (x-1 > 0) \text{ AND } (\text{NOT}(y=3)) \text{ THEN } \dots \text{ ELSE } \dots$$

dove però la presenza delle variabili x ed y nella condizione è solo apparente poiché in ogni istante dell'esecuzione del programma tali variabili denotano sempre un preciso numero. Naturalmente sarebbe interessante costruire un linguaggio che permetta istruzioni in cui la condizione dopo IF sia di tipo più complesso. Ad esempio un linguaggio che permetta istruzioni del tipo

$$\text{IF } \exists x(x^2=y) \text{ THEN } \dots \text{ ELSE } \dots$$

in cui la condizione esprime il fatto che y è un quadrato perfetto, oppure un poco più complicate come

$$\text{IF } \forall n \exists x \exists y (x^n + y^n = 1) \text{ THEN } \dots \text{ ELSE } \dots$$

Purtroppo un tale linguaggio (che permetterebbe di risolvere ogni problema dell'aritmetica) non esiste e, come mostreremo nel seguito parlando del primo teorema di Gödel, non può esistere.

3. Cosa è la verità: interpretare un linguaggio

Allora Pilato gli disse: «Ma dunque, sei tu re?» Gesù rispose: «Tu lo dici; sono re; io sono nato per questo, e per questo sono venuto nel mondo: per testimoniare della verità. Chiunque è dalla parte della verità ascolta la mia voce». Pilato gli disse: «Che cos'è la verità?» E detto

questo, uscì di nuovo verso i Giudei ... (Giovanni 18:33-38
NRV)

Il problema di definire cosa è la verità non è cosa semplice. Sembra addirittura che Pilato dubiti che esista la verità. Il primo tentativo di definire la verità da un punto di vista matematico fu fatto da logico matematico Tarski e noi ci atterremo alla sua proposta. Per prima cosa è evidente che una asserzione può essere vera o falsa a seconda del modo come viene interpretata. Ad esempio se dico $2+3 = 0$ scrivo una cosa falsa se interpreto tale asserzione sui numeri naturali. Invece scrivo una cosa vera se la interpreto sugli interi modulo 5. Se \mathcal{L} è un linguaggio del primo ordine, allora una sua interpretazione si ottiene specificando di quali oggetti si parla e quindi fissando un insieme D . Inoltre si deve associare

- ad ogni nome di operazione una operazione in D ,
- ad ogni costante un elemento di D ,
- ad ogni nome di relazione una relazione in D .

Definizione 3.1. Una *interpretazione* di un linguaggio del primo ordine \mathcal{L} è costituita da un insieme D , detto *dominio dell'interpretazione*, e da una funzione I , che associa:

- a) ad ogni nome di operazione n -aria o una operazione n -aria $I(o) : D^n \rightarrow D$
- b) ad ogni costante c un elemento $I(c)$ di D
- c) ad ogni nome di relazione n -aria r una relazione n -aria $I(r) \subseteq D^n$.

Esempio. Consideriamo un linguaggio \mathcal{L} in cui vi sia solo il nome di una relazione binaria "ama". Allora una interpretazione di tale linguaggio si ottiene fissando un insieme D di persone (ad esempio quelle presenti in una certa stanza) ed una relazione binaria $I(ama)$, cioè un sottoinsieme di $D \times D$. Ad esempio potremmo supporre che D sia l'insieme di persone,

$\{mario, maria, carlo, luigi\}$

e che l'interpretazione sia definita ponendo

$I(ama) = \{(maria, mario), (mario, carlo), (maria, carlo), (carlo, luigi)\}$.

Naturalmente vi possono essere più interpretazioni dello stesso linguaggio, ad esempio se si cambia il gruppo di persone cui ci si riferisce.

Esempio. Riferiamoci al linguaggio che si usa per le strutture ordinate. Poiché è costituito da un solo simbolo di relazione binaria, \leq , una interpretazione di tale linguaggio è costituita da un insieme D e da una relazione binaria $I(\leq)$. Ad esempio possiamo supporre che D sia l'insieme dei numeri interi e $I(\leq)$ l'usuale relazione di ordine tra interi, oppure che D sia l'insieme delle parti di un insieme S e $I(\leq)$ sia la relazione di inclusione. Si noti che l'unica cosa che si richiede in una interpretazione è che $I(\leq)$ sia una relazione binaria e non necessariamente una relazione d'ordine. Ad esempio otteniamo una interpretazione ponendo D uguale all'insieme dei numeri interi e $I(\leq) = \{(n,m) \mid n = m+1\}$. Naturalmente una tale interpretazione non è un insieme ordinato. Di fatto, almeno che non si impongano opportuni assiomi (cosa che faremo nel seguito) le interpretazioni di questo linguaggio possono essere accettate anche come interpretazioni del linguaggio dell'esempio precedente, e viceversa.

Esempio. Consideriamo un linguaggio \mathcal{L} con un solo simbolo per una operazione binaria \otimes . Allora una interpretazione di \mathcal{L} è costituita da un insieme D e da una operazione binaria. Ad esempio si ottiene una interpretazione ponendo $D = \{1,2,3,4\}$ e $I(\otimes)$ uguale al massimo, cioè $I(\otimes)(n,m) = \max\{n,m\}$ per ogni $(n,m) \in D^2$. Un'altra interpretazione si ottiene ponendo $D = A^+$, cioè D uguale all'insieme delle parole su di un dato alfabeto A e interpretando \otimes come la giustapposizione (mettere una parola dopo l'altra). Ad esempio, se $A = \{a,b\}$ allora

$I(\otimes)(babb,aab) = babbaab$. In entrambi i casi si suppone che $=$ sia interpretata come identità. Questo consente, ad esempio, di esprimere la proprietà associativa $\forall x \forall y \forall z ((x \otimes y) \otimes z = x \otimes (y \otimes z))$.

Naturalmente la nozione di interpretazione coincide sostanzialmente con quella di struttura del primo ordine che abbiamo data nel capitolo precedente. Infatti ad ogni interpretazione è possibile associare una struttura del primo ordine al modo seguente.

Definizione 3.2. Supponiamo che in \mathcal{L} i nomi di operazione siano o_1, \dots, o_s , che i nomi di relazioni siano r_1, \dots, r_t e che le costanti siano c_1, \dots, c_p . Chiamiamo *struttura associata ad una interpretazione* I la struttura

$$(D, I(o_1), \dots, I(o_s), I(r_1), \dots, I(r_t), I(c_1), \dots, I(c_p)).$$

Viceversa ogni struttura del primo ordine $(D, h_1, \dots, h_s, \mathcal{R}_1, \dots, \mathcal{R}_t, e_1, \dots, e_p)$ può essere vista come l'interpretazione di un opportuno linguaggio del primo ordine che contenga un nome r_i per ogni relazione \mathcal{R}_i , un nome o_i per ogni operazione h_i ed un nome (cioè una costante) c_i per ogni elemento e_i . Ovviamente:

- le interpretazioni di un linguaggio che non abbia nomi di relazioni coincidono con le strutture algebriche;
- le interpretazioni di un linguaggio che non ha nomi di operazioni coincidono con le strutture relazionali.

Poiché non esiste sostanziale differenza tra la nozione di struttura e quella di interpretazione, possiamo estendere tutte le definizioni e tutti i risultati che abbiamo elencato nel capitolo sulle strutture del primo ordine alle interpretazioni di un linguaggio del primo ordine.

Definizione 3.3. Sia \mathcal{L} un linguaggio del primo ordine e (D', I') una interpretazione di \mathcal{L} .

Allora una *sottointerpretazione* di (D', I') è un'interpretazione (D, I) di \mathcal{L} tale che:

- i) $D \subseteq D'$
- ii) per ogni nome di operazione o , D è una parte stabile rispetto $I'(o)$
- iii) per ogni costante c , D contiene $I'(c)$,
- iv) $I(o) = I'(o)/D$ per ogni nome di operazione o
- v) $I(c) = I'(c)$ per ogni costante c
- vi) $I(r) = I'(r)/D$ per ogni nome di relazione r .

E' chiaro che una sottointerpretazione di (D', I') definisce una sottostruttura di (D', I') e quindi a volte parleremo anche di *sottostruttura* di una struttura invece che di sottointerpretazione di una interpretazione. Si osservi che per ottenere una sottointerpretazione si deve considerare un sottoinsieme D di D' che sia una parte stabile rispetto alle operazioni e che contenga gli elementi che sono interpretazioni di costanti. Naturalmente vale la seguente proposizione.

Proposizione 3.4. Sia (D, I) una interpretazione e definiamo l'operatore $T : \mathcal{P}(D) \rightarrow \mathcal{P}(D)$ ponendo, per ogni $X \subseteq D$

$$T(X) = X \cup \{I(o)(d_1, \dots, d_n) \mid ar(o) = n, d_1 \in X, \dots, d_n \in X\} \cup \{I(c_1), \dots, I(c_p)\}.$$

Allora T è un operatore algebrico i cui punti fissi sono le parti stabili di D e quindi le sottointerpretazioni di (D, I) . Ne segue che:

- i) l'intersezione di una famiglia di sottointerpretazioni di (D, I) è una sottointerpretazione di (D, I)
- ii) dato un sottoinsieme X , la sottointerpretazione generata da X ha come sostegno l'insieme $\langle X \rangle$ con $\langle X \rangle = \bigcup_{n \in \mathbb{N}} T^n(X)$.

4. Interpretazione dei termini e delle formule.

Abbiamo già detto che un termine in cui compaiano delle variabili libere è la descrizione di un algoritmo per calcolare una funzione. Chiamiamo interpretazione di tale termine appunto la funzione calcolata da tale algoritmo. Ad esempio al termine $3 \cdot (x \cdot x) + 2 \cdot x + 5$ corrisponde la funzione polinomiale corrispondente. Se invece il termine non contiene variabili libere la sua interpretazione è l'elemento da esso calcolato. Naturalmente tali interpretazioni dipendono dalla struttura algebrica di cui si sta parlando, cioè dalla interpretazione fissata. Un modo più preciso di definire l'interpretazione di un termine è il seguente.

Definizione 4.1. Sia (D, I) una interpretazione, e t un termine. Allora per ogni intero n che sia maggiore o uguale agli indici delle variabili che compaiono in t è definita una funzione n -aria

$I(t) : D^n \rightarrow D$ definita per ricorsione sulla complessità di t al modo seguente:

a) se t è la costante c allora $I(t)$ è la funzione costantemente uguale a $I(c)$;

b) se t è la variabile x_i allora $I(t)$ è la proiezione i -esima³ cioè la funzione definita da

$$I(t)(d_1, \dots, d_n) = d_i$$

c) se $t = o(t_1, \dots, t_p)$ allora $I(t)$ è la funzione composta da $I(o)$ e $I(t_1), \dots, I(t_p)$, cioè la funzione tale che

$$I(t)(d_1, \dots, d_n) = I(o)(I(t_1)(d_1, \dots, d_n), \dots, I(t_p)(d_1, \dots, d_n)).$$

Ad esempio sia t il termine $\log(x_1) + \text{sen}(x_2)$ che scritto in forma prefissa sarà $+(\log(x_1), \log(x_2))$ allora

$$I(t)(d_1, d_2) = I(+)(I(\log(x_1))(d_1, d_2), I(\text{sen}(x_2))(d_1, d_2)).$$

D'altra parte

$$I(\log(x_1))(d_1, d_2) = I(\log)(I(x_1)(d_1, d_2)) = I(\log)(d_1)$$

$$I(\text{sen}(x_2))(d_1, d_2) = I(\text{sen})(I(x_2)(d_1, d_2)) = I(\text{sen})(d_2)$$

e quindi

$$I(t)(d_1, d_2) = I(+)(I(\log)(d_1), I(\text{sen})(d_2)).$$

Questo significa che il termine $\log(x_1) + \text{sen}(x_2)$ viene interpretato come la funzione che, data la coppia (d_1, d_2) di elementi di D ,

1. applica la funzione $I(\log)$ a d_1 ,
2. applica la funzione $I(\text{sen})$ a d_2 ,
3. compone i risultati ottenuti nei passi 1. e 2. tramite $I(+)$.

Nota. Da notare che se in un termine compaiono le variabili x_1 e x_3 allora la sua interpretazione è comunque una funzione di (almeno) tre variabili anche se poi i valori che assume non dipendono effettivamente dai valori di x_2 . Ad esempio il termine $2 \cdot x_1 + \text{seno}(x_3)$ è interpretato come una funzione di tre variabili in quanto viene considerato equivalente al termine $2 \cdot x_1 + 0 \cdot x_2 + \text{seno}(x_3)$. In tale caso si dice che la variabile x_2 è *muta*. Lo stesso termine può essere interpretato anche come funzione di quattro variabili in quanto può essere considerato equivalente al termine $2 \cdot x_1 + 0 \cdot x_2 + \text{seno}(x_3) + 0 \cdot x_4$. Un termine che si riduca ad una costante può essere interpretato come funzione di n variabili per ogni intero n . Ad esempio se l'interpretazione è il campo dei numeri reali allora la costante 3 può essere interpretata come funzione definita in R in R^2 in R^3 e così via. La cosa non appare tanto strana in quanto se, ad esempio, si accetta che la costante 3 sia un termine equivalente al termine $0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 + 3$ (oppure al termine $0 \cdot x_1 + 0 \cdot x_2 + 3$ oppure al termine $0 \cdot x_1 + 3$) allora tale costante deve poter essere interpretata come funzione di tre variabili (di due variabili e di una variabile, rispettivamente).

³ Prende il nome di *proiezione i-esima* in un prodotto cartesiano $X_1 \times \dots \times X_n$ con $i \leq n$ la funzione Pr_i definita dal porre $Pr_i(x_1, \dots, x_n) = x_i$. In altri termini la proiezione i -esima è la funzione che associa ad ogni vettore la sua componente di posto i . La terminologia geometrica è dovuta al fatto che, ad esempio, in un piano euclideo le due possibili proiezioni corrispondono alle proiezioni sugli assi cartesiani.

Le funzioni associate a termini permettono di caratterizzare al modo seguente le sottostrutture generate da un sottoinsieme.

Proposizione 4.2. Sia (D, I) una interpretazione ed X un sottoinsieme di D , allora

$$\langle X \rangle = \{I(t)(d_1, \dots, d_n) \mid t \text{ è un termine le cui variabili sono tra } x_1, \dots, x_n \text{ e } d_1 \in X, \dots, d_n \in X\}.$$

Dim. Posto $M = \{I(t)(d_1, \dots, d_n) \mid d_1 \in X, \dots, d_n \in X\}$, dobbiamo provare che M è la più piccola parte stabile contenente X . Il fatto che M contenga X segue dal fatto che se t è il termine x_1 allora $I(t)$ è l'applicazione identica. Per provare che M è una parte stabile supponiamo ad esempio che h sia il nome di una qualunque operazione binaria e che $a \in M, b \in M$. Allora esistono due termini t_1 e t_2 tali che $a = I(t_1)(a_1, \dots, a_t)$ e $b = I(t_2)(b_1, \dots, b_s)$ con $a_i \in X$ e $b_i \in X$. Per semplicità supponiamo che le variabili di t_1 siano diverse dalle variabili di t_2 . Allora

$$\begin{aligned} I(h)(a, b) &= I(h)(I(t_1)(a_1, \dots, a_t), I(t_2)(b_1, \dots, b_s)) \\ &= I(h(t_1, t_2))(a_1, \dots, a_t, b_1, \dots, b_s) = I(t)(a_1, \dots, a_t, b_1, \dots, b_s) \in M. \end{aligned}$$

Questo prova che M è chiuso rispetto l'operazione h . In modo analogo si procede per le operazioni che non sono binarie e quindi possiamo concludere che M è una parte stabile. Infine è evidente che ogni parte stabile contenente X contiene anche M . \square

Ad esempio se $X = \{a, b\}$ è un sottoinsieme di un gruppo allora $\langle X \rangle$ è l'insieme di tutti gli elementi del tipo

$$a, b, a \cdot a, b \cdot b, a \cdot b, b \cdot a, (a \cdot b) \cdot a^{-1}, (a^{-1} \cdot b) \cdot a^{-1}, \dots$$

cioè l'insieme di tutti gli elementi che si possono ottenere a partire da a e b tramite le operazioni del gruppo.

Si consideri ora una formula α di un dato linguaggio e proponiamoci di definire in maniera rigorosa che cosa significa l'espressione " α è vera". Ora naturalmente la verità o falsità di una formula dipende dalla interpretazione del linguaggio. Ad esempio la formula $\forall x_1 (\forall x_2 (x_1 \cdot x_2 = x_2 \cdot x_1))$ sarà vera se il dominio D dell'interpretazione è l'insieme degli interi relativi ed il simbolo "." è interpretato con l'usuale moltiplicazione di numeri. Tale formula sarà invece falsa se invece D è l'insieme delle funzioni di R in R e l'interpretazione di "." è di essere la composizione di due funzioni (siccome, ad esempio $\log(\text{sen}(x)) \neq \text{sen}(\log(x))$, tale operazione non è commutativa). Allora è più corretto definire che cosa si debba intendere per

$$"\alpha \text{ è vera rispetto ad una interpretazione } I".$$

Anche ciò crea qualche difficoltà, infatti se in α vi è una variabile libera x_1 , allora α può essere vera o falsa a seconda dell'elemento rappresentato da x_1 . Ad esempio la formula $\forall x_2 (x_2 \cdot x_1 = x_2)$ sarà vera negli interi relativi se x_1 rappresenta l'unità, sarà falsa se x_1 rappresenta il numero 2. Ciò significa che ha senso dire se α è vera o falsa non solo dopo aver fissato una interpretazione I del linguaggio ma anche dopo aver assegnato ad ogni variabile libera di α un particolare elemento nel relativo dominio. Per evitare complicazioni formali nel seguito supponiamo che anche le variabili vincolate siano interpretate da elementi del dominio. Pertanto se le variabili libere o vincolate di α sono tra x_1, \dots, x_t , dati d_1, \dots, d_t elementi di D vogliamo dare una definizione precisa del concetto:

" α è vera rispetto alla interpretazione I quando le sue eventuali variabili libere sono interpretate con d_1, \dots, d_t ".

Indicheremo in breve con $I \models \alpha [d_1, \dots, d_t]$ una tale asserzione e ne daremo la seguente definizione formale. \models

Definizione 4.3. Sia I una interpretazione, sia α una formula le cui variabili libere o vincolate siano tra x_1, \dots, x_t e siano d_1, \dots, d_m elementi del dominio D . Allora la relazione $I \models \alpha [d_1, \dots, d_m]$ è definita per induzione sulla complessità di α tramite:

- a) $I \models r(t_1, \dots, t_n) [d_1, \dots, d_m]$ se $(I(t_1)(d_1, \dots, d_m), \dots, I(t_n)(d_1, \dots, d_m)) \in I(r)$
- b) $I \models \alpha \wedge \beta [d_1, \dots, d_m]$ se $I \models \alpha [d_1, \dots, d_m]$ e $I \models \beta [d_1, \dots, d_m]$
- c) $I \models \alpha \vee \beta [d_1, \dots, d_m]$ se $I \models \alpha [d_1, \dots, d_m]$ oppure $I \models \beta [d_1, \dots, d_m]$
- d) $I \models \neg \alpha [d_1, \dots, d_m]$ se non è vero che $I \models \alpha [d_1, \dots, d_m]$
- e) $I \models \exists x_i(\alpha) [d_1, \dots, d_m]$ se esiste $d \in D$ tale che $I \models \alpha [d_1, \dots, d_{i-1}, d, d_{i+1}, \dots, d_m]$.

Nota. Risulta che

$$I \models \forall x_i(\alpha) [d_1, \dots, d_m] \Leftrightarrow I \models \alpha [d_1, \dots, d_i, \dots, d_m] \text{ per ogni } d_i \in D.$$

Infatti, poiché $\forall x_i(\alpha)$ è una abbreviazione di $\neg(\exists x_i(\neg\alpha))$, avremo che

$$I \models \forall x_i(\alpha) [d_1, \dots, d_m] \Leftrightarrow \text{non è vero che } I \models \exists x_i(\neg\alpha) [d_1, \dots, d_m] \Leftrightarrow \text{non è vero che esiste } d_i \in D \text{ tale che } I \models \neg\alpha [d_1, \dots, d_i, \dots, d_m] \Leftrightarrow \text{per ogni } d_i \in D \text{ } I \models \alpha [d_1, \dots, d_i, \dots, d_m].$$

Nota. Invece di interpretare tutte le variabili che appaiono libere o vincolate si potrebbero interpretare solo quelle che sono libere nella formula α . Infatti, se la variabile x_i è vincolata, allora l'essere $I \models \alpha [d_1, \dots, d_i]$ non dipende dall'elemento d_i considerato. Ad esempio se risulta che $I \models \forall x_1(x_1 \cdot x_2 = x_2 \cdot x_1) [d_1, d_2]$ allora risulta anche $I \models \forall x_1(x_1 \cdot x_2 = x_2 \cdot x_1) [d, d_2]$ per un qualunque $d \neq d_1$. In particolare se una formula è chiusa allora si può omettere la stringa $[d_1, \dots, d_n]$ che interpreta le variabili occorrenti in α . Ad esempio, invece di scrivere $I \models \forall x_1 \forall x_2(x_1 \cdot x_2 = x_2 \cdot x_1) [d_1, d_2]$ possiamo più semplicemente scrivere $I \models \forall x_1 \forall x_2(x_1 \cdot x_2 = x_2 \cdot x_1)$.

Definizione 4.4. Se $I \models \alpha [d_1, \dots, d_m]$ diciamo che *la formula α è vera rispetto ad I negli elementi d_1, \dots, d_m* . Diciamo che *α è vera in I* o che *I è un modello di α* e scriviamo $I \models \alpha$, se risulta $I \models \alpha [d_1, \dots, d_m]$ per ogni d_1, \dots, d_m in D .

In altre parole dire che una formula con eventuali variabili libere è vera in una interpretazione I equivale a dire che la sua chiusura universale è vera. Ad esempio diciamo che $x_1 \cdot x_2 = x_2 \cdot x_1$ è vera in una interpretazione I se $I \models x_1 \cdot x_2 = x_2 \cdot x_1 [d_1, d_2]$ comunque si scelgano d_1 e d_2 nel dominio di interpretazione, cioè se $I \models \forall x_1 \forall x_2(x_1 \cdot x_2 = x_2 \cdot x_1)$.

Definizione 4.5. Una teoria T si dice *soddisfacibile* se ammette un modello si dice *completa* se per ogni formula α risulta o che $T \models \alpha$ oppure che $T \models \neg\alpha$.

5. Formule logicamente vere ed equivalenze logiche.

Se una formula α è vera (falsa) rispetto a tutte le interpretazioni, allora viene detta *logicamente vera* (*logicamente falsa*).

Esempio. Le formule $\alpha \vee \neg\alpha$, $\alpha \rightarrow \alpha$, $(\forall x(\alpha)) \rightarrow \alpha(t)$, sono esempi di formule logicamente vere; le formule $\alpha \wedge \neg\alpha$, $(\neg\alpha(t)) \wedge \forall x(\alpha)$, sono esempi di formule logicamente false.

Proposizione 5.1. Sia α una tautologia (una contraddizione) del calcolo proposizionale e sia α^* una formula della logica del primo ordine che si ottiene sostituendo ad ogni variabile proposizionale una formula del primo ordine. Allora α^* è una formula logicamente vera (logicamente falsa).

Dim. La dimostrazione è banale e ci limitiamo ad illustrarla in un caso particolare. Sia ad esempio α la tautologia $p_1 \vee \neg p_1$ e sostituiamo p_1 con la formula $\exists x(x_2-1=0)$, allora otteniamo la formula $(\exists x(x_2-1=0)) \vee \neg(\exists x(x_2-1=0))$ che risulta vera comunque si consideri una interpretazione. \square

La formula logicamente vera α^* si dice anche che sia *un esempio* della tautologia α . Due formule α e α' si dicono *logicamente equivalenti* se ammettono gli stessi modelli. In tale caso scriveremo $\alpha \equiv \alpha'$.

Proposizione 5.2. Abbiamo che:

- i) $\alpha \equiv \alpha'$ se e solo se $\alpha \leftrightarrow \alpha'$ è logicamente vera
- ii) tutte le formule logicamente vere sono equivalenti tra loro
- iii) tutte le formule logicamente false sono equivalenti tra loro.

Proposizione 5.3. La relazione \equiv di equivalenza logica è una relazione di equivalenza.

Proposizione 5.4. Se α è una sottoformula di β , $\alpha \equiv \alpha'$ e β' si ottiene da β sostituendo al posto di α la formula α' allora $\beta \equiv \beta'$.

E' ben noto ai matematici che il "nome" con cui viene indicata una variabile quantificata non ha importanza, ad esempio la formula $\forall x_1 \forall x_2(x_1+x_2=x_2+x_1)$ ha lo stesso significato della formula $\forall x_3 \forall x_2(x_3+x_2=x_2+x_3)$ ed entrambe esprimono la proprietà commutativa. E' possibile precisare questo fatto al modo seguente.

Definizione 5.5. Chiamiamo *simili* due formule $\alpha(x_i)$ e $\alpha(x_j)$ se la prima contiene occorrenze libere di x_i esattamente in quei posti in cui la seconda possiede occorrenze libere di x_j .

Ad esempio sono simili le formule

$$\alpha(x_3) = \forall x_2(x_3+x_2=x_2+x_3) \quad \text{e} \quad \alpha(x_1) = \forall x_2(x_1+x_2=x_2+x_1)$$

perché nella prima formula x_3 compare in tutti e soli i posti in cui compare x_1 nella seconda. Equivalentemente si può dire che $\alpha(x_i)$ è simile a $\alpha(x_j)$ quando $\alpha(x_j)$ si ottiene da $\alpha(x_i)$ sostituendo al posto delle occorrenze di x_i la variabile x_j . In tale sostituzione è necessario però assicurarsi che x_j non compaia vincolata nella prima formula. Ad esempio se sostituissi x_2 al posto di x_3 nell'esempio precedente avrei la formula chiusa $\alpha(x_2) = \forall x_2(x_2+x_2=x_2+x_2)$ che non deve essere considerata simile alla formula aperta $\alpha(x_3)$.

Proposizione 5.6. (rinomina delle variabili). Siano $\alpha(x_i)$ e $\alpha(x_j)$ formule simili, allora

$$\exists x_i \alpha(x_i) \equiv \exists x_j \alpha(x_j) \quad \text{e} \quad \forall x_i \alpha(x_i) \equiv \forall x_j \alpha(x_j).$$

In definitiva tale proposizione esprime la seguente regola:

se si cambia nome ad una variabile quantificata in una formula allora si ottiene una formula equivalente.

Ad esempio sia la formula $\forall x_1 \forall x_2(x_1 \cdot x_2 = x_2 \cdot x_1)$ che la formula $\forall x_3 \forall x_4(x_3 \cdot x_4 = x_4 \cdot x_3)$ dicono la stessa cosa, cioè che vale la proprietà commutativa del prodotto. Le formule $\exists x_1(x_1^2 = 2)$ e $\exists x_3(x_3^2 = 2)$ sono equivalenti ed entrambe esprimono il fatto che esiste la radice quadrata di 2.

Proposizione 5.7. Per ogni formula α risulta

- a) $\neg(\forall x(\alpha)) \equiv \exists x(\neg\alpha)$
- b) $\neg(\exists x(\alpha)) \equiv \forall x(\neg\alpha).$

Dim. Per provare a) ricordiamo che $\forall x(\alpha)$ è una abbreviazione di $\neg(\exists x(\neg\alpha))$ e che pertanto $\neg(\forall x(\alpha)) \equiv \neg(\neg(\exists x(\neg\alpha))) \equiv \exists x(\neg\alpha)$.

Per provare b), stante il significato di \forall , dobbiamo provare l'ovvia equivalenza

$$\neg(\exists x(\alpha)) \equiv \neg\exists x(\neg(\neg\alpha)).$$

Proposizione 5.8. (leggi distributive dei quantificatori). Siano α e β due formule allora valgono le seguenti equivalenze

c) $(\forall x\alpha) \wedge (\forall x\beta) \equiv \forall x(\alpha \wedge \beta)$

d) $(\exists x\alpha) \vee (\exists x\beta) \equiv \exists x(\alpha \vee \beta)$.

Dim. Proviamo la proposizione c) supponendo, ad esempio, che x denoti la variabile x_1 . Allora

$$I \models (\forall x_1\alpha) \wedge (\forall x_1\beta) [d_1, \dots, d_n] \Leftrightarrow I \models \forall x_1\alpha [d_1, \dots, d_n] \text{ e } I \models \forall x_1\beta [d_1, \dots, d_n]$$

$$\Leftrightarrow \text{per ogni } d \in D \text{ } I \models \alpha [d, \dots, d_n] \text{ e per ogni } d \in D \text{ } I \models \beta [d, \dots, d_n]$$

$$\Leftrightarrow \text{per ogni } d \in D \text{ } I \models \alpha [d, \dots, d_n] \text{ e } I \models \beta [d, \dots, d_n] \Leftrightarrow I \models \forall x(\alpha \wedge \beta).$$

L'equivalenza d) può essere anche derivata da c) "per dualità". Infatti

1. applicando c) alle formule $\neg\alpha$ e $\neg\beta$ otteniamo l'equivalenza

$$(\forall x\neg\alpha) \wedge (\forall x\neg\beta) \equiv \forall x(\neg\alpha \wedge \neg\beta)$$

2. negando entrambi i membri di tale equivalenza otteniamo

$$\neg((\forall x\neg\alpha) \wedge (\forall x\neg\beta)) \equiv \neg(\forall x(\neg\alpha \wedge \neg\beta))$$

cioè

$$3. (\neg\forall x\neg\alpha) \vee (\neg\forall x\neg\beta) \equiv (\exists x\neg(\neg\alpha \wedge \neg\beta))$$

e quindi

$$4. (\exists x\neg\neg\alpha) \vee (\exists x\neg\neg\beta) \equiv \exists x(\alpha \vee \beta)$$

Basta allora applicare le leggi della doppia negazione per ottenere d).

E' subito visto che invece le equivalenze

$$(\forall x\alpha) \vee (\forall x\beta) \equiv \forall x(\alpha \vee \beta) \text{ e } (\exists x\alpha) \wedge (\exists x\beta) \equiv \exists x(\alpha \wedge \beta)$$

non valgono. Ad esempio, se $\alpha(x)$ significa "x è pari" e $\beta(x)$ significa "x è dispari", allora $(\forall x\alpha) \vee (\forall x\beta)$ significherebbe che o ogni numero è pari, oppure ogni numero è dispari (cosa falsa). Invece $\forall x(\alpha \vee \beta)$ significherebbe che ogni numero x o è pari oppure è dispari (cosa vera). Per quanto riguarda la seconda equivalenza, $(\exists x\alpha) \wedge (\exists x\beta)$ significa che esiste un numero pari ed esiste un numero dispari (cosa vera). La formula $\exists x(\alpha \wedge \beta)$ significa invece che esiste un numero che è sia pari che dispari (cosa falsa).

Tuttavia se si suppone che la variabile x non compaia libera in β , allora possiamo provare la seguente proposizione.

Proposizione 5.9. (leggi distributive dei quantificatori). Siano α e β due formule e supponiamo che x non sia libera in β , allora valgono le seguenti equivalenze:

a) $\exists x(\alpha \wedge \beta) \equiv (\exists x\alpha) \wedge (\exists x\beta) \equiv (\exists x(\alpha)) \wedge \beta$

b) $\exists x(\alpha \vee \beta) \equiv (\exists x\alpha) \vee (\exists x\beta) \equiv (\exists x(\alpha)) \vee \beta$

c) $\forall x(\alpha \wedge \beta) \equiv (\forall x\alpha) \wedge (\forall x\beta) \equiv (\forall x(\alpha)) \wedge \beta$

d) $\forall x(\alpha \vee \beta) \equiv (\forall x\alpha) \vee (\forall x\beta) \equiv (\forall x(\alpha)) \vee \beta$.

e) $\forall x(\alpha \rightarrow \beta) \equiv (\neg\exists x\alpha) \rightarrow \beta$

f) $\forall x(\beta \rightarrow \alpha) \equiv \beta \rightarrow (\forall x(\alpha)).$

Dim. Per provare a), supponiamo per semplicità che x sia x_1 . Sia I una interpretazione e siano d_1, \dots, d_n elementi del relativo dominio D . Per prima cosa osserviamo che, poiché x_1 non è libera in β , l'affermazione $I \models \beta [d_1, \dots, d_n]$ non dipende in maniera effettiva dal particolare d_1 scelto. Pertanto se vale $I \models \beta [d_1, \dots, d_n]$ allora vale $I \models \beta [d, d_2, \dots, d_n]$ qualunque sia $d \in D$. Allora

$$\begin{aligned}
I \models \exists x_1(\alpha) \wedge \beta [d_1, \dots, d_m] &\Leftrightarrow I \models \exists x_1(\alpha) [d_1, \dots, d_m] \text{ e } I \models \beta [d_1, \dots, d_m] \\
&\Leftrightarrow \text{esiste } d \in D \text{ tale che } I \models \alpha [d, d_2, \dots, d_m] \text{ e } I \models \beta [d, d_2, \dots, d_m] \\
&\Leftrightarrow \text{esiste } d \in D \text{ tale che } I \models \alpha \wedge \beta [d, d_2, \dots, d_m] \\
&\Leftrightarrow I \models \exists x_1(\alpha \wedge \beta) [d_1, \dots, d_m].
\end{aligned}$$

Le rimanenti equivalenze si provano in modo simile. \square

Si osservi che l'ipotesi che x non sia libera in β non può essere eliminata. Infatti ad esempio la formula $(\forall x(x \leq 0)) \vee (x \geq 0)$ che interpretiamo nel campo ordinato dei numeri reali. Tale formula, che ha la variabile x libera, è vera per $x=3$ ed è falsa per $x = -3$. Essa pertanto non è equivalente alla formula $\forall x((x \leq 0) \vee (x \geq 0))$ che è chiusa e vera. Invece la formula $(\forall x(x \leq 0)) \vee (1 \geq 0)$ è equivalente alla formula $\forall x((x \leq 0) \vee (1 \geq 0))$ e la formula $(\forall x(x \leq 0)) \vee (z \geq 0)$ è equivalente a $\forall x((x \leq 0) \vee (z \geq 0))$.

6. Un sistema di riscrittura per la forma normale prenessa

Possiamo applicare i concetti ora esposti per alcune riduzioni a forma normale per la logica del primo ordine. Cominciamo con estendere il “gioco” che permette di ridurre ogni formula in una formula in cui la negazione è applicata solo a formule atomiche e che abbiamo già visto per il calcolo proposizionale.

Proposizione 6.1. Esiste un sistema di riscrittura capace di trasformare ogni formula del calcolo dei predicati in una formula equivalente in cui la negazione compare solo davanti alle formule atomiche.

Dim. Basta procedere come nel caso del calcolo proposizionale "spingendo all'interno" la negazione tramite le leggi di De Morgan e la legge di doppia negazione. Abbiamo inoltre bisogno anche delle leggi $\neg \forall x \alpha \rightarrow \exists x \neg \alpha$ e $\neg \exists x \alpha \rightarrow \forall x \neg \alpha$. Più precisamente, applichiamo la seguente procedura:

1. ogni sottoformula del tipo $\neg(\alpha \vee \beta)$ può essere sostituita con $(\neg \alpha) \wedge (\neg \beta)$
2. ogni sottoformula del tipo $\neg(\alpha \wedge \beta)$ può essere sostituita con $(\neg \alpha) \vee (\neg \beta)$
3. ogni sottoformula del tipo $\neg(\neg(\alpha))$ può essere sostituita con α
4. ogni sottoformula del tipo $\neg \forall x \alpha$ può essere sostituita con $\exists x \neg \alpha$
5. ogni sottoformula del tipo $\neg \exists x \alpha$ può essere sostituita con $\forall x \neg \alpha$

\square

Da notare che non abbiamo scritto una regola per l'implicazione. Questo perché abbiamo considerato una implicazione del tipo $\alpha \Rightarrow \beta$ come un modo per rappresentare la formula $(\neg \alpha) \vee \beta$ e questo comporta che $\neg(\alpha \Rightarrow \beta)$ può essere sostituita con $\neg((\neg \alpha) \vee \beta)$ e quindi con $\alpha \wedge \neg \beta$.

Esempio. Consideriamo la definizione del concetto di limite di una successione $(a_n)_{n \in \mathbb{N}}$.

$$\lim_{n \rightarrow \infty} a_n = l \Leftrightarrow \forall \varepsilon \exists m \forall n (n \geq m \Rightarrow |a_n - l| < \varepsilon).$$

Proviamo a negare che tale successione converga ad l . Allora abbiamo la formula

$$\neg(\forall \varepsilon \exists m \forall n (n \geq m \Rightarrow |a_n - l| < \varepsilon)).$$

Allora è possibile “spingere” la negazione verso l'interno al modo seguente.

$$\neg(\forall \varepsilon \exists m \forall n (n \geq m \Rightarrow |a_n - l| < \varepsilon))$$

$$\exists \varepsilon \neg(\exists m \forall n (n \geq m \Rightarrow |a_n - l| < \varepsilon))$$

$$\exists \varepsilon \forall m \neg(\forall n (n \geq m \Rightarrow |a_n - l| < \varepsilon))$$

$$\exists \varepsilon \forall m \exists n \neg(n \geq m \Rightarrow |a_n - l| < \varepsilon)$$

$$\exists \varepsilon \forall m \exists n (n \geq m \wedge \neg |a_n - l| < \varepsilon).$$

Definizione 6.2. Una formula del tipo $Q_1x_1\dots Q_nx_n(\alpha)$, con Q_i quantificatore universale o esistenziale ed α matrice viene detta in *forma normale prenessa*.

Ad esempio le formule

$$x > 0, \forall x(x > 0), \exists x(x > 0 \wedge x + 1 > 0), \forall x \exists y(y > 0 \wedge y_2 = x)$$

sono in forma normale prenessa, mentre le formule

$$\exists x(x > 0) \vee \exists x(x < 0), \forall x(x > 0 \rightarrow \exists y(y = x))$$

non lo sono.

In una formula del tipo $Q_1x_1\dots Q_nx_n(\alpha)$ diciamo che i quantificatori $Q_1x_1\dots Q_nx_n$ sono *esterni*. Allora possiamo dire che una formula è in forma normale prenessa se tutti i quantificatori che vi compaiono sono esterni.

Proposizione 6.3. E' possibile trasformare ogni formula α in una formula equivalente in forma normale prenessa.

Dim. Aggiungiamo alle regole di riscrittura che abbiamo dato per spingere dentro la negazione le seguenti regole di riscrittura:

6. data la formula $(\exists x(\alpha)) \vee \beta$ la sostituiamo con $\exists y(\alpha(y/x)) \vee \beta$ con y variabile che non compare libera in β
7. data la formula $(\exists x(\alpha)) \wedge \beta$ la sostituiamo con $(\exists y(\alpha(x/y))) \wedge \beta$ con y variabile che non compare libera in β
8. data la formula $(\forall x(\alpha)) \wedge \beta$ la sostituiamo con $(\forall y(\alpha(x/y))) \wedge \beta$ con y variabile che non compare libera in β
9. data la formula $(\forall x(\alpha)) \vee \beta$ la sostituiamo con $(\forall y(\alpha(x/y))) \vee \beta$ con y variabile che non compare libera in β .

Data una formula γ indichiamo con $n(\gamma)$ il numero di occorrenze dei quantificatori in γ che siano non esterni. Allora una formula è in forma normale prenessa se e solo se $n(\gamma) = 0$ ed in questo caso nessuna delle regole può essere applicata. In definitiva partiamo da una qualunque formula γ ed operiamo in modo da ottenere una formula γ' in cui \neg sia applicato solo a formule atomiche. Appliciamo poi a γ' le regole di riscrittura ora elencate fino a quando ciò sia possibile. Ad ogni applicazione il valore di n diminuisce di una unità. Infatti, per esempio $n(\exists y(\alpha(y/x)) \vee \beta) = n((\exists x(\alpha)) \vee \beta) - 1$ e la stessa cosa avviene per le altre regole di riscrittura. Allora prima o poi si perviene ad una formula γ'' tale che $n(\gamma'') = 0$. A questo punto abbiamo ottenuto la forma normale prenessa. \square

Esempio. Consideriamo la formula $\neg(\forall x_1 \exists x_2(x_2 \cdot x_2 = x_1))$ che non è in forma normale prenessa. Allora è possibile "spingere dentro" la negazione al modo seguente

$$\neg(\forall x_1 \exists x_2(x_2 \cdot x_2 = x_1)) \equiv \exists x_1 \neg \exists x_2(x_2 \cdot x_2 = x_1) \equiv \exists x_1 \forall x_2(\neg(x_2 \cdot x_2 = x_1)).$$

Esempio. Consideriamo la formula (vera nel campo ordinato dei numeri reali) $(\exists y(y > 0)) \wedge (\exists y \forall x(x + y = x))$. Non essendo x libera in $\exists y \forall x(x + y = x)$, tale formula è equivalente a $\exists y((y > 0) \wedge \exists y \forall x(x + y = x))$. In tale modo abbiamo "spinto a sinistra" un primo quantificatore. Passiamo ora ad operare sulla formula $(y \geq 0) \wedge \exists y \forall x(x + y = x)$. Non possiamo "spingere a sinistra" il quantificatore $\exists y$ perché avremmo come risultato la formula $\exists y((y > 0) \wedge \forall x(x + y = x))$ che è una formula chiusa falsa la quale non è equivalente a $(y > 0) \wedge \exists y \forall x(x + y = x)$. Dobbiamo allora evitare tale difficoltà con un cambio di variabile sostituendo alla formula $(y > 0) \wedge \exists y \forall x(x + y = x)$ la formula equivalente $(y > 0) \wedge \exists z \forall x(x + z = x)$ (un tale passaggio è

giustificato dal fatto che la formula $\forall x(x+y=x)$ è simile alla formula $\forall x(x+z=x)$. Tale formula è equivalente alla formula $\exists z((y>0)\wedge\forall x(x+z=x))$ e quindi alla formula $\exists z\forall x((y>0)\wedge(x+z=x))$. In conclusione la nostra forma iniziale $(\exists y(y>0))\wedge(\exists y\forall x(x+y=x))$ è stata ridotta alla forma normale $\exists y\exists z\forall x((y>0)\wedge(x+z=x))$.

Esempio. L'espressione $\lim a_n = a$ si traduce nella formula $\forall p\exists m\forall n(n\geq m \rightarrow |a_n-a|\leq 1/p)$ che è scritta in forma normale prenessa. Se si nega tale formula, si ottiene la formula $\neg(\forall p\exists m\forall n(n\geq m \rightarrow |a_n-a|\leq 1/p))$ che esprime il fatto che la successione (a_n) non converge ad a . Tale formula si semplifica se si riduce a forma normale prenessa. Infatti

$$\begin{aligned} \neg(\forall p\exists m\forall n(n\geq m \rightarrow |a_n-a|\leq 1/p)) &\equiv \exists p\neg(\exists m\forall n(n\geq m \rightarrow |a_n-a|\leq 1/p)) \\ &\equiv \exists p(\forall m\neg\forall n(n\geq m \rightarrow |a_n-a|\leq 1/p)) \equiv \exists p(\forall m\exists n\neg(n\geq m \rightarrow |a_n-a|\leq 1/p)) \\ &\equiv \exists p(\forall m\exists n(n\geq m \wedge (|a_n-a|>1/p))). \end{aligned}$$

Ad esempio, se si vuole dimostrare che la successione $1/n$ non converge ad 1, dobbiamo dimostrare che $\exists p(\forall m\exists n(n\geq m\wedge(1-1/n>1/p))$ e per fare questo basta prendere $p = 1$.

Esercizio. Ridurre a forma normale prenessa le formule

$$(\forall x\forall y(x+y = y+x))\wedge(\neg(y\cdot y<7)) ; (\forall y\exists x(y+x=0))\wedge(\exists y\forall x(x+y=x)).$$

Esercizio. Ridurre a forma normale prenessa le formule

$$(\forall x\forall y(x+y = y+x))\wedge(\neg(y\cdot y<7)) ; (\forall y\exists x(y+x=0))\wedge(\exists y\forall x(x+y=x)).$$

La proposizione 6.3 pur essendo “costruttiva”, nel senso di indicare una procedura per ottenere la forma normale, non è rigorosa perché non dimostra che la procedura termina fino a giungere alla forma normale (anche se la terminazione è abbastanza intuitiva). Un modo più classico e rigoroso per dimostrare l'esistenza della forma normale è il seguente.

Proposizione 6.4. Per ogni formula α esiste una formula α' equivalente in forma normale prenessa.

Dim. Dimostriamo la proposizione per induzione sulla complessità delle formule. Se la formula α è atomica la proposizione è banalmente vera poiché le formule atomiche sono già in forma normale prenessa. Proviamo che se la proposizione è vera per α allora è vera anche per $\neg\alpha$. Per ipotesi di induzione esiste una matrice γ tale che $\alpha \equiv Q_1z_1 \dots Q_nz_n(\gamma)$. Allora $\neg\alpha \equiv \neg(Q_1z_1 \dots Q_nz_n(\gamma))$ ed, applicando più volte la a) della proposizione 6.3, $\neg\alpha \equiv Q'_1z_1 \dots Q'_nz_n(\neg\gamma)$ avendo posto $Q'_i = \forall$ se $Q_i = \exists$ e $Q'_i = \exists$ se $Q_i = \forall$. Ciò prova che $\neg\alpha$ è equivalente ad una formula in forma normale prenessa.

Assumiamo la proposizione vera per le formule α e β e proviamo che è vera per $\alpha\wedge\beta$. Per ipotesi di induzione sappiamo che α e β sono equivalenti a formule del tipo $Q_1z_1\dots Q_nz_n(\alpha')$ e $P_1y_1\dots P_my_m(\beta')$ con α' e β' opportune matrici. Cambiando eventualmente nome alle variabili z_1, \dots, z_n , possiamo supporre che nessuna delle variabili z_i sia libera in β . Pertanto per la proposizione precedente

$$\alpha\wedge\beta \equiv (Q_1z_1\dots Q_nz_n(\alpha'))\wedge\beta \equiv Q_1z_1(Q_2z_2\dots Q_n(\alpha'\wedge\beta)) \equiv \dots \equiv Q_1z_1Q_2z_2\dots Q_nz_n(\alpha'\wedge\beta)$$

Poiché β può contenere ancora quantificatori il processo di riduzione a forma normale non è ancora finito. Possiamo però ripetere il processo precedente per la formula $\alpha'\wedge\beta$ ed ottenere che tale formula è equivalente ad una formula del tipo $P_1y_1\dots P_my_m(\alpha'\wedge\beta')$. Sostituendo opportunamente possiamo concludere che $\alpha\wedge\beta$ è equivalente alla forma normale prenessa $Q_1z_1\dots Q_nz_nP_1y_1\dots P_my_m(\alpha'\wedge\beta')$.

Similmente si procede per provare che se α e β sono riducibili in forma normale prenessa anche $\alpha\vee\beta$ è riducibile in forma normale prenessa. D'altra parte è immediato che se α è

equivalente alla forma normale prenessa $Q_1z_1\dots Q_nz_n(\alpha')$ allora $\exists x(\alpha)$ è equivalente alla forma normale prenessa $\exists xQ_1z_1\dots Q_nz_n(\alpha')$.

Ciò prova che ogni formula è riducibile in forma normale prenessa. \square

Esempio. Consideriamo la formula $\neg(\forall x_1\exists x_2(x_2\cdot x_2=x_1))$ che non è in forma normale prenessa. Allora è possibile "spingere a destra" la negazione al modo seguente

$$\neg(\forall x_1\exists x_2(x_2\cdot x_2=x_1)) \equiv \exists x_1\neg\exists x_2(x_2\cdot x_2=x_1) \equiv \exists x_1\forall x_2(\neg(x_2\cdot x_2=x_1)).$$

Esempio. Consideriamo la formula (vera nel campo ordinato dei numeri reali) $(\exists y(y>0))\wedge(\exists y\forall x(x+y=x))$. Non essendo x libera in $\exists y\forall x(x+y=x)$, tale formula è equivalente a $\exists y((y>0)\wedge\exists y\forall x(x+y=x))$. In tale modo abbiamo "spinto a sinistra" un primo quantificatore. Passiamo ora ad operare sulla formula $(y\geq 0)\wedge\exists y\forall x(x+y=x)$. Non possiamo "spingere a sinistra" il quantificatore $\exists y$ perché avremmo come risultato la formula $\exists y((y>0)\wedge\forall x(x+y=x))$ che è una formula chiusa falsa la quale non è equivalente a $(y>0)\wedge\exists y\forall x(x+y=x)$. Dobbiamo allora evitare tale difficoltà con un cambio di variabile sostituendo alla formula $(y>0)\wedge\exists y\forall x(x+y=x)$ la formula equivalente $(y>0)\wedge\exists z\forall x(x+z=x)$ (un tale passaggio è giustificato dal fatto che la formula $\forall x(x+y=x)$ è simile alla formula $\forall x(x+z=x)$). Tale formula è equivalente alla formula $\exists z((y>0)\wedge\forall x(x+z=x))$ e quindi alla formula $\exists z\forall x((y>0)\wedge(x+z=x))$. In conclusione la nostra forma iniziale $(\exists y(y>0))\wedge(\exists y\forall x(x+y=x))$ è stata ridotta alla forma normale $\exists y\exists z\forall x((y>0)\wedge(x+z=x))$.

7. Skolemizzazione: come eliminare i quantificatori esistenziali

Abbiamo visto che ogni formula può essere ridotta a forma normale prenessa. In questo paragrafo vogliamo mostrare che, in un certo senso:

ogni formula risulta equivalente ad una formula universale purché si arricchisca opportunamente il linguaggio con nomi di costanti e nomi di operazioni.

Prima di procedere in generale facciamo alcuni esempi.

Insiemi ordinati con minimo. Consideriamo un linguaggio \mathcal{L} che contenga il simbolo \leq e consideriamo la teoria $T = Ord \cup \{\alpha, \beta\}$ che si ottiene aggiungendo agli assiomi soliti della teoria Ord degli insiemi ordinati le formule

$$\alpha = \exists x_1(\forall x_2(x_1 \leq x_2))$$

$$\beta = \exists x_1(\forall x_2(x_2 \leq x_1))$$

esprimenti l'esistenza di un elemento minimo e di un elemento massimo. Arricchiamo \mathcal{L} aggiungendo due costanti ad esempio m ed M , e sia \mathcal{L}^* il linguaggio ottenuto in tale modo. In tale linguaggio è possibile considerare la teoria $T^* = Ord \cup \{\alpha^*, \beta^*\}$ che si ottiene sostituendo α e β con le formule

$$\alpha^* = \forall x_2(m \leq x_2)$$

$$\beta^* = \forall x_2(x_2 \leq M).$$

E' evidente che, in un certo senso, i modelli di T coincidono con i modelli di T^* . Più precisamente, rimanendo nell'ambito degli insiemi ordinati, se I è una interpretazione di \mathcal{L} che verifica T allora esiste un elemento $d_1 \in S$ che è il minimo dell'insieme ordinato S ed un elemento d_2 che è il massimo. Allora si ottiene una interpretazione I^* di \mathcal{L}^* che verifica T^* ponendo $I^*(\leq) = I(\leq)$, $I^*(m) = d_1$ e $I^*(M) = d_2$. Viceversa se I^* è una interpretazione di \mathcal{L}^* che verifica T^* , allora, posto $I(\leq) = I^*(\leq)$, si ottiene una interpretazione di \mathcal{L} che verifica T .

Assiomi per la teoria dei gruppi. Un altro esempio può essere ricavato dalla teoria dei gruppi. Supponiamo di riferirci ad un linguaggio "minimale" \mathcal{L}_0 per tale teoria il quale contenga solo il nome di una operazione binaria. Allora la teoria dei gruppi dovrà essere espressa con l'insieme T dei due assiomi

$$A_1 \quad x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$$

$$A_2 \quad \exists z (\forall x_1 ((z \cdot x_1 = x_1) \wedge (x_1 \cdot z = x_1)) \wedge \forall x_2 \exists y ((x_2 \cdot y = z) \wedge (y \cdot x_2 = z))).$$

Il primo assioma è la solita proprietà associativa, il secondo assioma comprende sia l'affermazione di esistenza dell'elemento neutro z che dell'inverso y di ogni elemento x . Da notare che A_2 non si può spezzare nei due assiomi

$$\exists z (\forall x_1 ((z \cdot x_1 = x_1) \wedge (x_1 \cdot z = x_1)))$$

$$\forall x \exists y ((x \cdot y = z) \wedge (y \cdot x = z))$$

in quanto, essendo z una variabile (in quanto viene quantificata nella prima formula) nella seconda formula non è detto che z rappresenti lo stesso elemento di cui si è parlato nella prima formula. Per rendersene conto basta rinominare z nella prima formula e sostituire il sistema delle due formule con il sistema equivalente:

$$\exists u (\forall x_1 ((u \cdot x_1 = x_1) \wedge (x_1 \cdot u = x_1)))$$

$$\forall x \exists y ((x \cdot y = z) \wedge (y \cdot x = z)).$$

La riduzione a forma normale prenessa di tale assioma non migliora molto la sua comprensione. Infatti avremo che:

$$\begin{aligned} & \exists z (\forall x_1 ((z \cdot x_1 = x_1) \wedge (x_1 \cdot z = x_1)) \wedge \forall x_2 \exists y ((x_2 \cdot y = z) \wedge (y \cdot x_2 = z))) \\ & \equiv \exists z \forall x_1 ((z \cdot x_1 = x_1) \wedge (x_1 \cdot z = x_1) \wedge \forall x_2 \exists y ((x_2 \cdot y = z) \wedge (y \cdot x_2 = z))) \\ & \equiv \exists z \forall x_1 \forall x_2 (z \cdot x_1 = x_1) \wedge (x_1 \cdot z = x_1) \wedge \exists y ((x_2 \cdot y = z) \wedge (y \cdot x_2 = z)) \\ & \equiv \exists z \forall x_1 \forall x_2 \exists y ((z \cdot x_1 = x_1) \wedge (x_1 \cdot z = x_1) \wedge (x_2 \cdot y = z) \wedge (y \cdot x_2 = z)). \end{aligned}$$

Si ottiene però una semplificazione della formula introducendo un nuovo simbolo di costante, ad esempio $\mathbf{1}$, per denotare l'elemento neutro. Infatti nel linguaggio ampliato \mathcal{L}_1 ottenuto in questo modo potremo riscrivere A_2 in maniera più semplice come

$$(\forall x_1 ((\mathbf{1} \cdot x_1 = x_1) \wedge (x_1 \cdot \mathbf{1} = x_1)) \wedge \forall x_2 \exists y ((x_2 \cdot y = \mathbf{1}) \wedge (y \cdot x_2 = \mathbf{1}))).$$

Tale formula è equivalente alla formula A_2 nel senso che abbiamo specificato nel caso della teoria degli insiemi ordinati. D'altra parte una tale formula si spezza nei due assiomi

$$G_2 \quad \forall x_1 ((\mathbf{1} \cdot x_1 = x_1) \wedge (x_1 \cdot \mathbf{1} = x_1))$$

e

$$\forall x_2 \exists y ((x_2 \cdot y = \mathbf{1}) \wedge (y \cdot x_2 = \mathbf{1})).$$

La prima formula non si può ulteriormente semplificare, per quanto riguarda la seconda, una ulteriore semplificazione si ottiene introducendo un nuovo simbolo di operazione unaria, ad esempio " inv ". Infatti essa può essere riscritta come:

$$G_3 \quad \forall x ((x \cdot inv(x) = \mathbf{1}) \wedge (inv(x) \cdot x = \mathbf{1})).$$

In definitiva se scriviamo, come si usa solitamente, x^{-1} al posto $inv(x)$ abbiamo l'usuale sistema T^* di assiomi per la teoria dei gruppi

$$G_1 \quad x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$$

$$G_2 \quad \mathbf{1} \cdot x = x$$

$$G_2' \quad x \cdot \mathbf{1} = x$$

$$G_3 \quad x \cdot x^{-1} = \mathbf{1}$$

$$G_3' \quad x^{-1} \cdot x = \mathbf{1}.$$

L'idea che è alla base di un tale modo di procedere è che, come abbiamo già detto, è possibile semplificare la complessità delle formule arricchendo il linguaggio di opportuni nomi di costanti e di funzioni.

Per enunciare quanto detto sopra nella forma più generale proponiamo alcune definizioni.

Definizione 7.1. Siano \mathcal{L} ed \mathcal{L}^* linguaggi del primo ordine e supponiamo che l'alfabeto di \mathcal{L} sia contenuto nell'alfabeto di \mathcal{L}^* . Allora diciamo che una interpretazione (D, I) di \mathcal{L} è *il ridotto* di una interpretazione (D^*, I^*) di \mathcal{L}^* se $D = D^*$ ed I^* è una estensione di I . Diciamo che una teoria T in \mathcal{L} è *riducibile* ad una teoria T^* in \mathcal{L}^* se (D, I) è un modello di T se e solo se è il ridotto di un modello (D^*, I^*) di T^* .

In altre parole il ridotto di una interpretazione si ottiene cancellando dal linguaggio un certo numero di nomi di relazioni, nomi di funzioni o nomi di costanti e considerando solo l'interpretazione dei simboli rimanenti. Ad esempio il campo dei numeri reali è il ridotto del campo ordinato dei numeri reali. Il gruppo additivo dei reali è il ridotto del campo dei numeri reali.

Negli esempi che abbiamo fatto all'inizio del paragrafo la teoria T è riducibile alla teoria universale T^* .

Teorema 7.2. Sia \mathcal{L} un linguaggio del primo ordine e T una teoria in \mathcal{L} . Allora T è riducibile ad una teoria universale T^* in una estensione \mathcal{L}^* del linguaggio \mathcal{L} .

Dim. Supponiamo che T contenga una sola formula α , allora in base al teorema di riduzione a forma normale prenessa, non è restrittivo supporre α sia del tipo $Q_1 y_1 Q_2 y_2 \dots Q_n y_n (\alpha')$ con α' matrice. Supponiamo anche che non vi siano quantificatori "inutili", cioè che ogni variabile quantificata compaia effettivamente libera in α' . Se $Q_1 = \exists$ allora sia c una nuova costante e consideriamo la formula $Q_2 y_2 \dots Q_n y_n (\alpha'(y_1/c))$ ⁴. E' immediato che una interpretazione è un modello di $Q_1 y_1 Q_2 y_2 \dots Q_n y_n (\alpha)$ se e solo se è il ridotto di un modello di $Q_2 y_2 \dots Q_n y_n (\alpha'(y_1/c))$. Se Q_1 è universale, sia Q_{i+1} il primo quantificatore esistenziale che compare nella sequenza $Q_1 y_1 Q_2 y_2 \dots Q_n y_n$. Allora la nostra formula potrà essere scritta nella forma $\forall y_1 \forall y_2 \dots \forall y_i \exists y_{i+1} (\beta(y_{i+1}))$ dove β ha $n-1$ quantificatori esistenziali. Introduciamo ora nel nostro linguaggio un nuovo simbolo h di funzione ad i posti e consideriamo la formula

$$\forall y_1 \forall y_2 \dots \forall y_i (\beta(y_{i+1}/h(y_1, \dots, y_i))).$$

Anche in questo caso una interpretazione è un modello di $\forall y_1 \forall y_2 \dots \forall y_i \exists y_{i+1} (\beta(y_{i+1}))$ se e solo se è un modello della formula $\forall y_1 \forall y_2 \dots \forall y_i (\beta(y_{i+1}/h(y_1, \dots, y_i)))$. In definitiva abbiamo visto come sia possibile eliminare un quantificatore esistenziale. Iterando tale procedimento fino ad eliminare tutti i quantificatori esistenziali si ottiene la formula universale α^* cercata.

Nel caso in cui T contenga più di una formula si può procedere sostituendo ogni formula α in T con la formula α^* ottenuta con il modo ora esposto.

Il significato di tale teorema è anche che ogni teoria scientifica è costretta ad ampliare il linguaggio naturale se vuole descrivere i propri concetti in modo semplice. Questo spiega perché in ogni scienza nasca e si sviluppi un gergo specialistico sempre più esteso e perché spesso la divulgazione scientifica, cioè l'espressione di concetti complessi tramite il linguaggio naturale, sia una impresa molto difficile e forse impossibile.

⁴ Se α è una formula, x_i una variabile e t un termine, allora con $\alpha[x_i/t]$ denotiamo la formula che si ottiene sostituendo in α al posto delle occorrenze libere di x_i il termine t . Ad esempio se α è la formula $x_3(x_2+x_3=0)$ e t il termine $\log(x_4)$ allora $\alpha[x_2/t]$ è la formula $x_3(\log(x_4)+x_3=0)$, se invece t è uguale alla costante 3, allora $\alpha[x_2/t]$ coincide con $x_3(3+x_3=0)$.

8. Conseguenze logiche

Prende il nome di *sistema di assiomi del primo ordine* o semplicemente *sistema di assiomi* ogni insieme \mathcal{A} di formule. Le formule in \mathcal{A} vengono anche chiamate *assiomi*.

Definizione 8.1. Dato un sistema di assiomi \mathcal{A} , una interpretazione I è detta essere un modello di \mathcal{A} se ogni formula in \mathcal{A} è vera rispetto ad I . Chiameremo *soddisfacibili* i sistemi di assiomi che ammettono un modello.

Non tutti i sistemi di assiomi ammettono un modello, ad esempio se \mathcal{A} contiene una formula α e la sua negata $\neg\alpha$, allora è evidente che non può esistere un modello di \mathcal{A} .

Definizione 8.2. Diremo che un insieme Y di formule è *conseguenza logica* di un insieme X di formule e scriveremo $X \models Y$ se ogni modello di X è un modello di Y . Diremo che α è una conseguenza logica di X e scriveremo $X \models \alpha$, se ogni modello di X è un modello di α , diremo che α è conseguenza logica di β e scriveremo $\beta \models \alpha$ se ogni modello di β è un modello di α .

A volte useremo anche le espressioni “ X dipende logicamente da Y ” oppure “ β dipende logicamente da X ”. Intuitivamente α dipende logicamente da un insieme X di formule quando non dice niente di più di quando è detto da X .

Definizione 8.3. Un sistema di assiomi \mathcal{A} viene detto *indipendente* se nessuna formula α di \mathcal{A} dipende logicamente da $\mathcal{A} - \{\alpha\}$.

Forse il più famoso esempio di indipendenza è quello dell'assioma delle parallele dai rimanenti assiomi della geometria.

Esempio. Consideriamo il sistema di assiomi

$$A_1 \quad x_1 \leq x_1$$

$$A_2 \quad (x_1 \leq x_2) \wedge (x_2 \leq x_3) \rightarrow x_1 \leq x_3$$

$$A_3 \quad (x_1 \leq x_2) \wedge (x_2 \leq x_1) \rightarrow x_1 = x_2$$

$$A_4 \quad (x_1 \leq x_2) \vee (x_2 \leq x_1)$$

$$A_5 \quad \exists x_1 \exists x_2 ((x_1 \neq x_2) \wedge \forall x_3 (x_3 = x_1 \vee x_3 = x_2)) \text{ (esistono solo due elementi distinti).}$$

I primi tre dicono che \leq è una relazione d'ordine, il quarto asserisce che l'ordine è totale, il quinto che esistono solo due elementi nel dominio. Allora è immediato che la formula $\exists x_1 (\forall x_2 (x_1 \leq x_2))$ il cui significato è "esiste un minimo elemento" dipende da A_1, A_2, A_3, A_4 in quanto ogni insieme totalmente ordinato di due elementi contiene minimo.

Esercizio. La proprietà commutativa dipende dalla teoria dei gruppi ?

Proposizione 8.4. Un sistema di assiomi \mathcal{A} è indipendente se e solo se per ogni $\alpha \in \mathcal{A}$ esiste un modello di $\mathcal{A} - \{\alpha\}$ in cui α è falsa.

Dim. Ovvvia.

Esempio. Il sistema di assiomi A_1, A_2, A_3 per definire le strutture d'ordine è indipendente. Infatti tale teoria coincide con l'insieme $\mathcal{A} = \{R, T, A\}$ delle tre formule esprimenti rispettivamente la riflessività, la transitività e la proprietà antisimmetrica. Un esempio di modello di $\{R, T\}$ che non verifica A è dato dall'insieme dei polinomi in cui si è posto

$p \leq q \Leftrightarrow$ il grado di p è minore o uguale al grado di q .

Un modello di $\{T, A\}$ che non verifica R è dato dal porre in N

$xsy \Leftrightarrow x$ minore o uguale a y ed y è pari.

Un modello di $\{R, A\}$ che non verifica T è dato dal porre in N

$xsy \Leftrightarrow x = y$ oppure $y = x+1$.

Esercizio 4. La teoria dei gruppi è indipendente ?

Esercizio 5. La teoria delle equivalenze è indipendente?

Esempio. Consideriamo il linguaggio della teoria dei gruppi. Allora chiamiamo *gruppo* ogni modello del seguente sistema di assiomi

$G_1 \quad \forall x \forall y \forall z ((x \cdot y) \cdot z = x \cdot (y \cdot z))$ (proprietà associativa);

$G_2 \quad \forall x (x \cdot 1 = x) \wedge (1 \cdot x = x)$ (1 è elemento neutro)

$G_3 \quad \forall x (x \cdot x^{-1} = 1) \wedge (x^{-1} \cdot x = 1)$ (x^{-1} è l'inverso di x).

Consideriamo inoltre la formula

$G_4 \quad 1^{-1} = 1$

che esprime il fatto che l'inverso di 1 è 1. Allora poiché G_4 è vera in ogni gruppo, G_4 dipende da G_1, G_2 e G_3 .

Consideriamo ora la formula

$G_4' \quad \forall x \forall y (x \cdot y = y \cdot x)$

che esprime la proprietà commutativa. Allora tale formula non dipende dalle rimanenti in quanto esistono gruppi che non sono commutativi.

9. Logiche con eguaglianza

Fino ad ora abbiamo considerato il simbolo di relazione binaria "=" come un simbolo qualunque e non abbiamo fatto ipotesi sulla sua interpretazione. Tuttavia usualmente a tale simbolo viene sempre dato un significato particolare in quanto quasi sempre viene interpretato come identità, cioè si considera vera una asserzione del tipo $a = b$ se a e b denotano lo stesso oggetto. Un tale modo di interpretare il simbolo = viene detto *normale*.

Definizione 9.1. Una interpretazione (D, I) si dice *normale* se $I(=)$ è interpretato come identità, cioè se $I(=)$ è la diagonale $\{(d, d) \mid d \in D\}$ di D .

Le strutture algebriche vengono viste come interpretazioni normali in quanto, anche se non si dice esplicitamente, quando si parla di strutture algebriche si suppone sempre che esista nel linguaggio il simbolo = e che tale simbolo venga interpretato con l'identità. E' evidente che le interpretazioni normali verificano le seguenti formule:

$U_1 \quad x = x$ (riflessività) ;

$(x=y) \rightarrow (y=x)$ (simmetria) ;

$((x=y) \wedge (y=z)) \rightarrow (x=z)$ (transitività) ;

$U_2 \quad x = y \Rightarrow (r(z_1, \dots, x, \dots, z_n) \Leftrightarrow r(z_1, \dots, y, \dots, z_n))$ (sostituzione per relazioni);

$U_3 \quad x = y \Rightarrow f(z_1, \dots, x, \dots, z_n) = f(z_1, \dots, y, \dots, z_n)$ (sostituzione per funzioni)

dove U_2 e U_3 vanno intese estese a tutti i nomi di relazione r ed a tutti i nomi di funzione f e dove

- $r(z_1, \dots, y, \dots, z_n)$ è la formula atomica ottenuta da $r(z_1, \dots, x, \dots, z_n)$ sostituendo una occorrenza della variabile x con y

- $f(z_1, \dots, y, \dots, z_n)$ è il termine ottenuto da $f(z_1, \dots, x, \dots, z_n)$ rimpiazzando una occorrenza di x con y .

Viceversa si pone il seguente problema:

una interpretazione del simbolo = che verifica tali formule è necessariamente l'identità ?

La risposta è negativa in quanto tali formule sono verificate anche in tutti i casi in cui $I(=)$ è una congruenza. Infatti le condizioni date in U_1 esprimono il fatto che \equiv è una relazione di equivalenza mentre U_2, U_3 equivalgono ad affermare che \equiv è una congruenza.

D'altra parte in matematica spesso il simbolo di eguaglianza viene utilizzato per denotare una equivalenza. Ad esempio in geometria si usa dire che due segmenti sono uguali nel caso abbiano la stessa lunghezza. Quando si dice che

“due triangoli sono uguali se hanno due lati e l'angolo compreso uguali” il termine “uguali” si intende che due triangoli sono ritenuti uguali se hanno tutti i lati uguali (oppure se si possono sovrapporre con un movimento). Pertanto è possibile che due triangoli siano uguali in tale senso senza essere gli stessi (in quanto posizionati in due diverse zone del piano).

Definizione 9.2. Chiamiamo *logica con identità* la logica che presenta nel suo alfabeto il simbolo $=$ di relazione binaria e che interpreta tale relazione in modo che siano sempre soddisfatte le formule U_1, U_2 e U_3 (cioè interpreta $=$ con una congruenza).

Nel seguito se consideriamo una logica in cui è presente il simbolo $=$ supporremo sempre che sia una logica con identità, cioè supporremo sempre che siano verificate le formule U_1, U_2 e U_3 . Considereremo anche logiche in cui non sia presente $=$.

A volte le interpretazioni che verificano il sistema di assiomi U_1, U_2, U_3 vengono anche chiamate *pre-interpretazioni*. Questo nome si giustifica dal fatto che è possibile passare in modo canonico da una pre-interpretazione ad una interpretazione normale. Infatti, è possibile considerare il quoziente di I modulo la congruenza $I(=)$.

Definizione 9.3. Sia I una pre-interpretazione, allora il *quoziente normale* di I è l'interpretazione normale I^* che si ottiene come quoziente di I tramite la congruenza $I(=)$. Pertanto I^* è definita al modo seguente.

- $D^* = \{[x] \mid x \in D\}$ dove $[x] = \{y \in D \mid (x,y) \in I(=)\}$;
- $I^*(c) = [I(c)]$;
- $I^*(r) = \{([x_1], \dots, [x_n]) \mid (x_1, \dots, x_n) \in I(r)\}$;
- $I^*(h)([x_1], \dots, [x_n]) = [I(h)(x_1, \dots, x_n)]$.

Poiché $I^*(=)$ è uguale a $\{([x],[y]) \mid (x,y) \in I(=)\} = \{([x],[y]) \mid [x]=[y]\}$, I^* è normale. Come vedremo nel seguito una pre-interpretazione ed il suo quoziente normale verificano le stesse formule, pertanto non ha importanza se ci si riferisce ad una pre-interpretazione o al suo quoziente normale.

Esempio (Il campo dei razionali). Consideriamo il metodo con cui si costruisce il campo Q dei razionali a partire dall'anello Z degli interi relativi. Poniamo $D = Z \times (Z - \{0\})$ e definiamo in tale insieme le operazioni di moltiplicazione e ponendo

$$(n,m) \cdot (p,q) = (n \cdot p, m \cdot q) \quad ; \quad (n,m) + (p,q) = (n \cdot q + p \cdot m, m \cdot q). \quad (9.1)$$

Chiamiamo inoltre con 0 e 1 le coppie (0,1) e (1,1), rispettivamente. In tale modo si ottiene una struttura algebrica $(D, +, \cdot, 0, 1)$. Consideriamo in tale struttura algebrica la relazione \equiv definita da:

$$(n,m) \equiv (p,q) \Leftrightarrow nq = mp. \quad (9.2)$$

Non è difficile verificare che \equiv è una congruenza in $(D, +, \cdot, 0, 1)$. Chiamiamo *campo dei numeri razionali* il quoziente di $(D, +, \cdot, 0, 1)$ modulo \equiv . Se si accetta tale modo di procedere un numero razionale è una classe completa di equivalenza. In termini di logica del primo ordine $(Q, +, \cdot, 0, 1)$ è una interpretazione normale del linguaggio della teoria degli anelli.

Questo modo di procedere è corretto dal punto di vista matematico ma non rappresenta bene il modo con cui le persone manipolano ed utilizzano effettivamente i numeri razionali.

Infatti per una persona un numero razionale è una coppia non una classe di equivalenza. Il primo elemento della coppia viene chiamato *numeratore*, il secondo *denominatore*. Tutti i calcoli vengono fatti con tali coppie e non con le classi di equivalenza. Pertanto la struttura effettivamente utilizzata è $(D, +, \cdot, 0, 1)$ e non $(Q, +, \cdot, 0, 1)$. Tuttavia si utilizza continuamente anche la relazione \equiv che viene chiamata uguaglianza nonostante il fatto che sia una congruenza. Questo vuol dire che se si accetta questo modo di procedere il linguaggio è interpretato dal modello non normale $(D, +, \cdot, 0, 1, \equiv)$ invece che dal modello normale $(Q, +, \cdot, 0, 1, =)$. Naturalmente $(Q, +, \cdot, 0, 1, =)$ è il quoziente normale di $(D, +, \cdot, 0, 1, \equiv)$. I due modi di vedere le cose sono equivalenti.

Da notare che i programmi di calcolo simbolico come Mathematica o Derive assumono proprio tale punto di vista sia nella trattazione dei numeri razionali sia, più in generale, per tutta la matematica. Infatti per Mathematica gli oggetti da manipolare, ad esempio i razionali, sono coppie (per meglio dire parole del tipo m/n). I calcoli da effettuare sono quelli indicati da (9.1). La relazione di equivalenza definita in (10.2) viene vista come una *regola di riscrittura* o di *riduzione a forma normale*. Questo significa che dopo che si è effettuato un calcolo come $2/3 + 5/6$ ed avere ottenuto $(2 \cdot 6 + 3 \cdot 5)/(3 \cdot 6) = 27/16$ si cerca un elemento equivalente a $27/16$ che sia in forma canonica. Se si è scelta come forma canonica quella per cui numeratore e denominatore sono primi tra loro, allora si ottiene $3/2$. Da notare quindi che l'output non è 1.5 come sarebbe fatto da un qualunque linguaggio di programmazione che considera i razionali come particolari reali rappresentati in forma decimale.

Esercizio. Provare che la relazione \equiv dell'esempio precedente è una congruenza, che $(0,1) + (x,y)$ è congruo a (x,y) e che $(1,1) \cdot (x,y)$ è congruo a (x,y) .

Esempio: (Gli interi modulo m). Consideriamo l'anello degli interi relativi $(Z, +, \cdot, 0, 1)$ sia m un intero positivo e denotiamo con \equiv la *congruenza modulo m* , definita ponendo $x \equiv y$ se e solo se $x - y$ è un multiplo di m . Come è noto il quoziente $(Z_m, +, \cdot, [0], [1])$ di Z modulo \equiv è un anello finito chiamato anello degli interi modulo m . Un elemento di Z_m è una classe completa di equivalenza. Anche in questo caso possiamo interpretare tale struttura come un modello normale del linguaggio della teoria degli anelli. Un modo diverso di vedere le cose è riferirsi alla interpretazione non normale $(Z, +, \cdot, 0, 1, \equiv)$. In tale caso tutti i calcoli vengono fatti con i numeri ma siamo autorizzati ad utilizzare nei calcoli la relazione \equiv .

10. Estensione di una proprietà e relazioni definibili.

Sia α una formula le cui variabili sono comprese tra x_1, \dots, x_n , e sia (D, I) una interpretazione. Allora chiamiamo *estensione di α in I* la relazione n -aria $E(\alpha) \subseteq E^n$ definita da:

$$E(\alpha) = \{(d_1, \dots, d_n) \in D^n : I \models \alpha [d_1, \dots, d_n]\}.$$

Diciamo che una relazione è *definibile* se è l'estensione di una opportuna formula.

Esempio. Ad esempio se si considera la formula $3x + 2y + 1 = 0$ allora la sua estensione nel campo dei numeri reali è il sottoinsieme del piano costituito dalla retta $\{(x, y) : 3x + 2y + 1 = 0\}$. L'estensione della formula $x^2 + y^2 \geq 1$ è il complemento del cerchio chiuso di centro l'origine e raggio 1.

Esempio. Nell'anello Z degli interi relativi la nozione di numero primo è definita dalla formula $\exists n \exists m (n \neq 1 \wedge m \neq 1 \wedge n \cdot m = x)$.

Esempio. Nella teoria degli ordinamenti con il simbolo \leq come primitivo, allora la formula $(x \leq y) \wedge \neg (x = y)$ definisce l'ordine stretto.

Nel seguito, data una relazione n -aria R , chiamiamo i -striscia generata da R la relazione n -aria $S_i(R)$ definita da

$$S_i(R) = \{(d_1, \dots, d_n) : \text{esiste } d \in D \text{ tale che } (d_1, \dots, d_{i-1}, d, d_{i+1}, \dots, d_n) \in R\}.$$

Proposizione 10.1. La nozione di estensione di una formula può essere definita per induzione sulla complessità delle formule. Infatti risulta che:

1. $E(r(t_1, \dots, t_p)) = \{(d_1, \dots, d_n) \in D^n : (I(t_1)(d_1, \dots, d_n), \dots, I(t_p)(d_1, \dots, d_n)) \in I(r)\}$
2. $E(\alpha \wedge \beta) = E(\alpha) \cap E(\beta)$;
3. $E(\alpha \vee \beta) = E(\alpha) \cup E(\beta)$;
4. $E(\neg \alpha) = -E(\alpha)$.
5. $E(\exists x_i \alpha) = S_i(E(\alpha))$.

Dim. Immediato.

Proposizione 10.2. L'insieme delle relazioni n -arie definibili in una data interpretazione è la sottoalgebra dell'algebra di Boole $(P(D^n), \cup, \cap, D^n, \emptyset)$. Tale sottoalgebra è finita oppure numerabile.

Dim. Siano R_1 ed R_2 due relazioni definibili, allora esistono due formule α e β tali che $E(\alpha) = R_1$ ed $E(\beta) = R_2$. Per la 2. della proposizione ora dimostrata $R_1 \cap R_2 = E(\alpha \wedge \beta)$ e quindi $R_1 \cap R_2$ è l'estensione della formula $\alpha \wedge \beta$. In modo simile si prova che $R_1 \cup R_2$ è definibile in quanto estensione della formula $\alpha \vee \beta$ e che $-R_1$ è definibile in quanto estensione della formula $\neg \alpha$.

Per provare la seconda parte della dimostrazione osserviamo che la funzione E è una funzione suriettiva dell'insieme (numerabile) di formule nell'insieme delle relazioni definibili. Ciò prova che l'insieme delle relazioni definibili ha potenza minore o uguale al numerabile.

Corollario 10.3. In ogni interpretazione infinita esiste un numero infinito di insiemi e relazioni non definibili.

Dim. Osserviamo che se il dominio D è infinito allora l'insieme $P(D^n)$ di relazioni n -arie ha potenza maggiore del numerabile. D'altra parte in tale insieme solo una quantità finita è numerabile di relazioni è definibile. Pertanto esistono infinite relazioni n -arie non definibili.