Appunti di Logica Matematica

Francesco Bottacin

1 Logica Proposizionale

Una *proposizione* è un'affermazione che esprime un *valore di verità*, cioè una affermazione che è VERA oppure FALSA.

Ad esempio:

- "5 è un numero dispari"
- "Roma è la capitale della Francia"

sono due proposizioni (una vera e l'altra falsa).

Al contrario, l'affermazione "Mi piacerebbe passare l'esame senza studiare" non è una proposizione (esprime un desiderio e non un fatto che può essere vero o falso).

Le proposizioni possono essere combinate tra loro per costruire delle proposizioni più complesse utilizzando dei *connettivi*, quali "e", "o", "non", "se... allora...", ecc. Chiameremo *atomiche* quelle proposizioni che non sono ottenute da proposizioni più semplici mediante l'uso di connettivi.

Ad esempio:

- "4 è un numero pari" è una proposizione atomica,
- "*Se* c'è il sole *allora* vado al mare" è una proposizione composta (le cui componenti sono "C'è il sole" e "Vado al mare").

1.1 Linguaggi Formali

Il linguaggio naturale è troppo complesso e ambiguo. La Logica Matematica ha bisogno di un linguaggio molto più semplice, che chiameremo *linguaggio formale*.

Per costruire un linguaggio formale bisogna fissare un *alfabeto*, cioè un insieme di simboli che ci serviranno a costruire delle "frasi" (che, in questo contesto, chiameremo *formule*). Le "frasi" non sono altro che delle sequenze finite (*stringhe*) di simboli che appartengono all'alfabeto che abbiamo fissato.

Servirà poi una *sintassi*, cioè un insieme di regole per stabilire quali sequenze di simboli sono accettabili nel nostro linguaggio e quali no.

Attenzione: la sintassi si occupa solo della *forma* delle frasi e non del loro *contenuto*.

Esempio 1.1. Nel linguaggio dell'Aritmetica, consideriamo le seguenti "frasi" (cioè formule):

- $2 \cdot (3 + 1) = 8$, formula sintatticamente corretta
- $2 + (4(- ==)$, formula sintatticamente non corretta
- $2 + 1 = 6$, formula sintatticamente corretta

Si noti che la prima e la terza formula sono entrambe sintatticamente corrette (anche se la prima è VERA e la terza è FALSA). In particolare si noti che, nel caso di una formula non sintatticamente corretta, non ha nessun senso chiederci se è vera o falsa; essa è semplicemente una sequenza di simboli priva di senso.

Le formule sintatticamente corrette saranno chiamate *Formule Ben Formate* (FBF). Riassumendo: il compito della sintassi è quello di fornire un insieme di regole per costruire le FBF.

Solo quando una formula è sintatticamente corretta si può poi parlare del suo *significato*. Questo è il compito della *semantica*: assegnare un significato a tutte le frasi sintatticamente corrette (cioè a tutte le FBF).

1.2 Il linguaggio del Calcolo Proposizionale

Nel *Calcolo Proposizionale* si devono manipolare delle proposizioni. Queste verranno indicate, per comodità, con delle lettere maiuscole A, B, C, ..., P, Q, ... (eventualmente con delle lettere con indici, come A_1, A_2, A_3, ...).

Le proposizioni si possono poi combinare tra loro mediante l'uso dei connettivi. Pertanto il nostro alfabeto dovrà contenere:

- Simboli per indicare le proposizioni: A, B, C, ...
- Simboli per indicare i connettivi: $\neg$, $\wedge$, $\vee$, $\rightarrow$
- Simboli accessori, come le parentesi: (e).

Per comodità, introduciamo anche il simbolo $\perp$ che servirà ad indicare la *falsità*, l'assurdo.

Ora stabiliamo le regole della sintassi, cioè spieghiamo come si costruiscono le FBF. Le regole sono le seguenti:

- A, B, C, ..., e anche $\bot$ sono FBF;
- le altre FBF si ottengono combinando delle FBF già costruite mediante l'uso dei connettivi, nei modi seguenti: Se P e Q sono FBF, allora anche $(\neg P)$, $(P \land Q)$, $(P \lor Q)$ e $(P \to Q)$ sono FBF.

Esempio 1.2. Le seguenti sono FBF:

$$((A \lor B) \to (C \land D))$$
$$((A \to B) \to (\neg C))$$
$$(((\neg A) \lor \bot) \land (\neg C))$$

Al contrario, la seguente sequenza di simboli non è una FBF:

$$\land \to A \to ($$

Osservazione 1.3. Si noti che (per ora) i simboli $\neg$, $\land$, $\lor$, $\to$ non hanno nessun significato; essi sono solo dei simboli che devono essere manipolati in modo puramente formale, secondo le regole della sintassi enunciate prima.

Osservazione 1.4. Le regole che abbiamo stabilito conducono ad un uso eccessivo delle parentesi. Per semplificare l'aspetto delle formule conviene stabilire delle *priorità* tra i vari simboli:

priorità più alta	$\neg$
.	$\land$
.	$\lor$
priorità più bassa	$\to$

Pertanto la formula

$$A \land \neg B \to C$$

deve essere interpretata come segue

$$((A \land (\neg B)) \to C)$$

mentre la formula

$$\neg A \land B \lor C \to \neg D$$

significa

$$((((\neg A) \land B) \lor C) \to (\neg D)).$$

Invece, nella formula

$$(A \to B) \lor C$$

non si possono togliere le parentesi, perché la formula

$$A \to B \lor C$$

verrebbe interpretata come

$$(A \to (B \lor C)).$$

1.3 La Semantica del Calcolo Proposizionale

Bisogna ora attribuire un *significato* (VERO o FALSO) a tutte le formule sintatticamente corrette (le FBF).

Indichiamo i valori di verità **VERO** con **1** e **FALSO** con **0**. Si tratta quindi di definire una funzione

$$v : \text{FBF} \to \{0, 1\},$$

che associa ad ogni formula ben formata P il suo valore di verità $v(P)$.

Dato che ogni FBF si ottiene combinando tra loro delle proposizioni atomiche mediante l'uso dei connettivi (secondo le regole della sintassi), per definire una tale funzione v sull'insieme di tutte le FBF basta definirla per le proposizioni atomiche e poi analizzare il comportamento dei vari connettivi. Poniamo quindi

$$v(A) = \begin{cases} 1 & \text{se } A \text{ è vera,} \\ 0 & \text{se } A \text{ è falsa,} \end{cases}$$

per ogni proposizione atomica A. Poniamo inoltre

$$v(\perp) = 0,$$

dato che il simbolo $\perp$ rappresenta la falsità.

Analizziamo ora il comportamento dei vari connettivi. Per fare ciò scriviamo le *tavole di verità*:

Tavola della verità della *negazione* ($\neg$ = NOT)

P	$\neg P$
0	1
1	0

Tavola della verità della *congiunzione* ($\wedge$ = AND)

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

Tavola della verità della *disgiunzione* ($\vee$ = OR)

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

Tavola della verità della *implicazione* ($\to$)

P	Q	$P \to Q$
0	0	1
0	1	1
1	0	0
1	1	1

Esercizio 1.5. Verificare che $P \to Q$ ha la stessa tavola della verità di $\neg P \vee Q$. Diremo che queste due formule sono *equivalenti.*

Osservazione 1.6. I simboli usati per denotare i vari connettivi non sono standard. In contesti diversi si usano spesso simboli diversi da quelli che noi abbiamo introdotto.

Ad esempio, nel linguaggio di programmazione C, il simbolo per il connettivo logico AND è `&&`, mentre quello per il connettivo logico OR è `||`. (Sempre nel linguaggio C, si faccia molta attenzione a non confondere gli operatori logici `&&` e `||` con i cosiddetti "bitwise operators" (operatori bit-a-bit) `&` e `|`).

Definizione 1.7. Una funzione $v\colon \text{FBF} \to \{0,1\}$ che soddisfa tutte le proprietà precedentemente elencate è detta una *interpretazione.*

Osservazione 1.8. Da quanto detto risulta evidente che una interpretazione è univocamente determinata dai valori che essa assume sulle proposizioni atomiche, poiché ogni altra FBF si ottiene combinando delle proposizioni atomiche mediante l'uso di connettivi.

Definizione 1.9. Sia P una FBF e v una interpretazione. Se $v(P) = 1$, diremo che P è soddisfatta nell'interpretazione v, oppure che v è un *modello* per P. In tal caso si scrive $v \models P$ (P è vera nell'interpretazione v).

Definizione 1.10. Una formula ben formata P è *soddisfacibile* se ha almeno un modello, cioè se esiste almeno una interpretazione in cui P è soddisfatta. In caso contrario P è *insoddisfacibile* (si dice anche che P è una *contraddizione*).

Esempio 1.11. La seguente formula è soddisfacibile:

$$(A \wedge \neg B) \vee (B \to A).$$

Infatti, se consideriamo una interpretazione v tale che $v(A) = 1$ e $v(B) = 0$, si ha $v(\neg B) = 1$, quindi $v(A \wedge \neg B) = 1$, da cui segue che

$$v((A \wedge \neg B) \vee (B \to A)) = 1.$$

Un esempio di formula insoddisfacibile (contraddizione) è dato dalla formula seguente:

$$A \wedge \neg A.$$

Infatti, dato che per una qualunque interpretazione v, $v(A)$ può solo essere 0 o 1, la tavola di verità della formula precedente è:

A	$\neg A$	$A \wedge \neg A$
0	1	0
1	0	0

il che significa che la proposizione $A \wedge \neg A$ è sempre FALSA, indipendentemente dal fatto che A sia vera o falsa.

Definizione 1.12. Una formula ben formata P è una *tautologia* se ogni interpretazione v è un modello per P, cioè se P risulta VERA in ogni interpretazione. In tal caso si scriverà $\models P$.

Esempio 1.13. La formula

$$A \to A \vee B$$

è una tautologia.

Infatti, dato che per una qualunque interpretazione v, $v(A)$ e $v(B)$ possono solo essere 0 o 1, la tavola di verità della formula precedente è:

A	B	$A \vee B$	$A \to A \vee B$
0	0	0	1
0	1	1	1
1	0	1	1
1	1	1	1

il che significa che la proposizione $A \to A \vee B$ è sempre VERA, indipendentemente dal fatto che A o B siano vere o false.

Esempio 1.14. Ci chiediamo se la formula

$$A \to \neg A$$

sia soddisfacibile.

Scriviamo la tavola di verità:

A	$\neg A$	$A \to \neg A$
0	1	1
1	0	0

Ciò significa che la proposizione $A \to \neg A$ è soddisfacibile e l'interpretazione che la soddisfa è quella che assegna ad A il valore 0, cioè $v(A) = 0$ (in altre parole, la proposizione $A \to \neg A$ è VERA solo quando A è FALSA).

Proposizione 1.15. *Una formula ben formata P è una tautologia se e solo se $\neg P$ è insoddisfacibile.*

Dimostrazione. La dimostrazione è lasciata per esercizio. $\square$

Osservazione 1.16. La proposizione precedente afferma una cosa piuttosto ovvia e cioè che una proposizione P è "sempre vera" se e solo se la sua negazione $\neg P$ è "sempre falsa."

Definizione 1.17. Sia Γ un insieme di formule ben formate. Γ è *soddisfacibile* se esiste un'interpretazione v tale che $v(P) = 1$, per ogni $P \in \Gamma$ (cioè un'interpretazione che renda vere tutte le proposizioni di Γ).

Γ è *insoddisfacibile* se, per ogni interpretazione v, esiste almeno una proposizione $P \in \Gamma$ tale che $v(P) = 0$.

Definizione 1.18. Sia Γ un insieme di formule ben formate. Diremo che una proposizione Q è *conseguenza (semantica)* di Γ, e scriveremo $\Gamma \models Q$, se per ogni interpretazione v tale che $v(P) = 1$ per ogni $P \in \Gamma$, si ha anche $v(Q) = 1$.

In altre parole, $\Gamma \models Q$ significa che Q è vera in tutte le interpretazioni che sono dei modelli per Γ. In caso contrario, scriveremo $\Gamma \not\models Q$.

A titolo di esempio dimostriamo i seguenti risultati:

Proposizione 1.19. *Si ha $\Gamma \models Q$ se e solo se $\Gamma \cup \{\neg Q\}$ è insoddisfacibile.*

Dimostrazione. Ricordando la definizione, si ha $\Gamma \models Q$ se e solo se $v(Q) = 1$ per tutte le interpretazioni v che rendono vere tutte le formule di Γ. Ciò equivale a dire che, per ogni interpretazione v, o esiste qualche $P \in \Gamma$ tale che $v(P) = 0$, oppure si ha $v(Q) = 1$. Ma se $v(Q) = 1$, allora $v(\neg Q) = 0$, quindi per ogni interpretazione v, esiste almeno una formula $R \in \Gamma \cup \{\neg Q\}$ tale che $v(R) = 0$. Ciò significa che $\Gamma \cup \{\neg Q\}$ è insoddisfacibile. $\square$

Proposizione 1.20. *Si ha $P \models Q$ se e solo se $\models P \to Q$.*

Dimostrazione. Supponiamo che $P \models Q$. Allora, per ogni interpretazione v tale che $v(P) = 1$ si deve avere anche $v(Q) = 1$. Quindi vale anche $v(P \to Q) = 1$. Se invece l'interpretazione v è tale che $v(P) = 0$, si ha ancora $v(P \to Q) = 1$ (si veda la tavola di verità di $\to$). Pertanto, per ogni interpretazione v, si ha sempre $v(P \to Q) = 1$, il che equivale a dire che $\models P \to Q$.

Viceversa, supponiamo che valga $\models P \to Q$. Allora, per ogni interpretazione v, si ha $v(P \to Q) = 1$. In particolare, se consideriamo un'interpretazione v tale che $v(P) = 1$, dal fatto che $v(P \to Q) = 1$ si deduce che anche $v(Q) = 1$ (si veda la tavola di verità di $\to$). Questo significa che $P \models Q$. $\square$

Usando quest'ultimo risultato si può dimostrare il seguente teorema:

Teorema 1.21 (Deduzione semantica)**.** *Per ogni intero $n \geq 1$, si ha*

$$P_1, P_2, \ldots, P_n \models Q \text{ se e solo se } P_1, P_2, \ldots, P_{n-1} \models P_n \to Q.$$

Dimostrazione. La dimostrazione procede per induzione su n. $\square$

Enunciamo ora, senza dimostrarli, alcuni risultati di particolare importanza teorica.

Teorema 1.22 (Teorema di Compattezza)**.** *Un insieme Γ di formule ben formate è soddisfacibile se e solo se lo è ogni suo sottoinsieme finito.*

Una formulazione equivalente del Teorema di Compattezza è la seguente:

Teorema 1.23. *Un insieme Γ di formule ben formate è insoddisfacibile se e solo se esiste un sottoinsieme finito di Γ che è insoddisfacibile.*

Corollario 1.24. *Sia Γ un insieme di formule ben formate e P una proposizione. $\Gamma \models P$ se e solo se esiste un sottoinsieme finito Δ di Γ tale che $\Delta \models P$.*

1.4 Equivalenza Semantica

Due formule si dicono *semanticamente equivalenti* quando i loro valori di verità coincidono per ogni interpretazione, cioè quando hanno la stessa tavola di verità.

Diamo quindi la seguente definizione:

Definizione 1.25. Due formule ben formate P e Q si dicono *(semanticamente) equivalenti* se, per ogni interpretazione v, si ha $v(P) = v(Q)$. In tal caso scriveremo $P \equiv Q$.

Teorema 1.26. *Si hanno le seguenti equivalenze:*

idempotenza	$\begin{cases} P \vee P \equiv P \\ P \wedge P \equiv P \end{cases}$
commutatività	$\begin{cases} P \vee Q \equiv Q \vee P \\ P \wedge Q \equiv Q \wedge P \end{cases}$
associatività	$\begin{cases} (P \vee Q) \vee R \equiv P \vee (Q \vee R) \\ (P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R) \end{cases}$
assorbimento	$\begin{cases} P \vee (P \wedge Q) \equiv P \\ P \wedge (P \vee Q) \equiv P \end{cases}$
distributività	$\begin{cases} P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R) \\ P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R) \end{cases}$
leggi di De Morgan	$\begin{cases} \neg(P \vee Q) \equiv \neg P \wedge \neg Q \\ \neg(P \wedge Q) \equiv \neg P \vee \neg Q \end{cases}$
doppia negazione	$\neg\neg P \equiv P$

Dimostrazione. Basta scrivere le rispettive tavole di verità e controllare che siano uguali (farlo per esercizio). □

1.5 Completezza Funzionale

Definizione 1.27. Sia P una FBF contenente n proposizioni atomiche distinte $A_1, A_2, \ldots, A_n$. La funzione $f_P \colon \{0,1\}^n \to \{0,1\}$ tale che, per ogni $(a_1, a_2, \ldots, a_n) \in \{0,1\}^n$, si ha $f_P(a_1, a_2, \ldots, a_n) = v(P)$, dove v è una interpretazione tale che $v(A_i) = a_i$, per ogni $i = 1, \ldots, n$, è detta la *funzione di verità* di P.

La funzione di verità di una proposizione P è equivalente alla tavola di verità di P.

Esercizio 1.28. Quante sono le possibili tavole di verità per una proposizione P contenente esattamente n proposizioni atomiche distinte?

A_1	A_2	$\cdots$	A_{n-1}	A_n	P
0	0	$\cdots$	0	0	?
0	0	$\cdots$	0	1	?
0	0	$\cdots$	1	0	?
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	1	?

Anche i connettivi logici definiscono delle funzioni di verità, descritte dalle loro tavole di verità.

Ogni funzione $f\colon \{0,1\}^n \to \{0,1\}$ definisce un qualche connettivo n-ario. Ad esempio, se $n = 2$, vi sono $2^{(2^2)} = 16$ funzioni da $\{0,1\}^2$ in $\{0,1\}$, quindi esistono 16 connettivi binari differenti (ricordiamo che noi ne abbiamo definiti solo tre: $\wedge$, $\vee$ e $\to$), i quali corrispondono a tutte le possibili tavole di verità del tipo

A	B	$A * B$
0	0	?
0	1	?
1	0	?
1	1	?

Definizione 1.29. Dato un insieme di connettivi logici C e un connettivo $c \notin C$, c si dice *(semanticamente) derivabile* da C se esiste una formula proposizionale P costruita con i soli connettivi di C tale che $f_P = f_c$.

In altre parole, un connettivo c è derivabile dall'insieme di connettivi C se è possibile esprimerlo mediante connettivi di C.

Esempio 1.30. Dalle leggi di De Morgan (e dalla doppia negazione) si deduce che

$$P \vee Q \equiv \neg(\neg P \wedge \neg Q),$$

quindi il connettivo $\vee$ è derivabile dall'insieme di connettivi $\{\neg, \wedge\}$.

Analogamente, si ha che

$$P \wedge Q \equiv \neg(\neg P \vee \neg Q),$$

da cui segue che il connettivo $\wedge$ è derivabile dall'insieme di connettivi $\{\neg, \vee\}$.

Esempio 1.31. Definiamo il connettivo $\oplus$ mediante la seguente tavola di verità:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Questo connettivo è detto "o esclusivo" (eXclusive OR, XOR). Si noti che esso corrisponde alla somma in $\mathbb{Z}/2\mathbb{Z}$.

Analizzando la tavola di verità, si verifica facilmente che si ha

$$A \oplus B \equiv (A \wedge \neg B) \vee (\neg A \wedge B),$$

quindi il connettivo $\oplus$ è derivabile dall'insieme di connettivi $\{\neg, \vee, \wedge\}$.

Esercizio 1.32. Poiché abbiamo già osservato che il connettivo $\wedge$ è derivabile dall'insieme di connettivi $\{\neg, \vee\}$, si deduce che il connettivo $\oplus$ è, in effetti, derivabile anch'esso dall'insieme di connettivi $\{\neg, \vee\}$.

Si scriva dunque una formula per esprimere $A \oplus B$ usando solo i due connettivi $\neg$ e $\vee$.

Esempio 1.33. Definiamo il connettivo $\leftrightarrow$ mediante la seguente tavola di verità:

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Analizzando la tavola di verità, si verifica facilmente che si ha

$$A \leftrightarrow B \equiv (A \to B) \wedge (B \to A),$$

quindi il connettivo $\leftrightarrow$ è derivabile dall'insieme di connettivi $\{\wedge, \to\}$.

Esercizio 1.34. Usando il fatto che il connettivo $\to$ è derivabile dall'insieme di connettivi $\{\neg, \vee\}$ (vedi Esercizio 1.5), si scriva una formula per esprimere $A \leftrightarrow B$ usando solo i due connettivi $\neg$ e $\vee$.

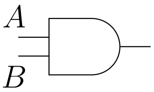
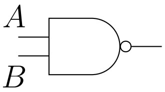
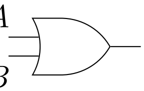
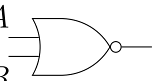
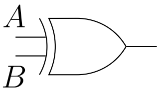
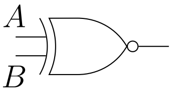
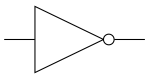
Esercizio 1.35. Si dimostrino le seguenti equivalenze:

$$A \to B \equiv \neg A \vee B$$
$$A \vee B \equiv \neg A \to B$$
$$A \vee B \equiv \neg(\neg A \wedge \neg B)$$
$$A \wedge B \equiv \neg(\neg A \vee \neg B)$$
$$A \wedge B \equiv (((A \to \bot) \to \bot) \to (B \to \bot)) \to \bot$$
$$\neg A \equiv A \to \bot$$
$$\bot \equiv A \wedge \neg A$$
$$A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$$
$$A \oplus B \equiv (A \wedge \neg B) \vee (\neg A \wedge B)$$

Definizione 1.36. Un insieme C di connettivi logici si dice *funzionalmente completo* se, per ogni $n \geq 1$ e per ogni funzione $f\colon \{0,1\}^n \to \{0,1\}$, esiste una formula ben formata P, costruita utilizzando solo i connettivi di C, tale che $f = f_P$.

In altri termini, un insieme di connettivi è funzionalmente completo se ogni altro connettivo possibile (che corrisponde ad ogni possibile tavola di verità) è derivabile da esso.

Osservazione 1.37. Esistono dei circuiti elettronici, detti *porte logiche*, che realizzano le funzioni logiche elementari (NOT, AND, OR, ecc.). I loro simboli circuitali sono i seguenti:

Porta AND	A B  C	$C = A \wedge B$
Porta NAND	A B  C	$C = \neg(A \wedge B)$
Porta OR	A B  C	$C = A \vee B$
Porta NOR	A B  C	$C = \neg(A \vee B)$
Porta XOR	A B  C	$C = A \oplus B$
Porta XNOR	A B  C	$C = \neg(A \oplus B)$
Porta NOT	A  C	$C = \neg A$

Esempio 1.38. A titolo di esempio riportiamo, nella figura 1, lo schema interno del circuito integrato SN74LS47, prodotto negli anni '70 dalla Texas Instruments. Si tratta di un "BCD-to-seven-segment decoder/driver."

Siete in grado di scrivere la corrispondente tavola di verità?

1.6 Forme Normali

Definizione 1.39. Un *letterale* è una proposizione atomica o la sua negazione.

Definizione 1.40. Una *congiunzione* di formule ben formate P_1, P_2, ..., P_n è una formula del tipo $P_1 \wedge P_2 \wedge \cdots \wedge P_n$. La *disgiunzione* delle formule P_1, P_2, ..., P_n è invece la formula $P_1 \vee P_2 \vee \cdots \vee P_n$.

Definizione 1.41. Una formula ben formata P è detta in *forma normale congiuntiva* (fnc) se P è della forma $P_1 \wedge P_2 \wedge \cdots \wedge P_n$ (per qualche $n \geq 1$), dove ciascuna P_i è una disgiunzione di letterali.

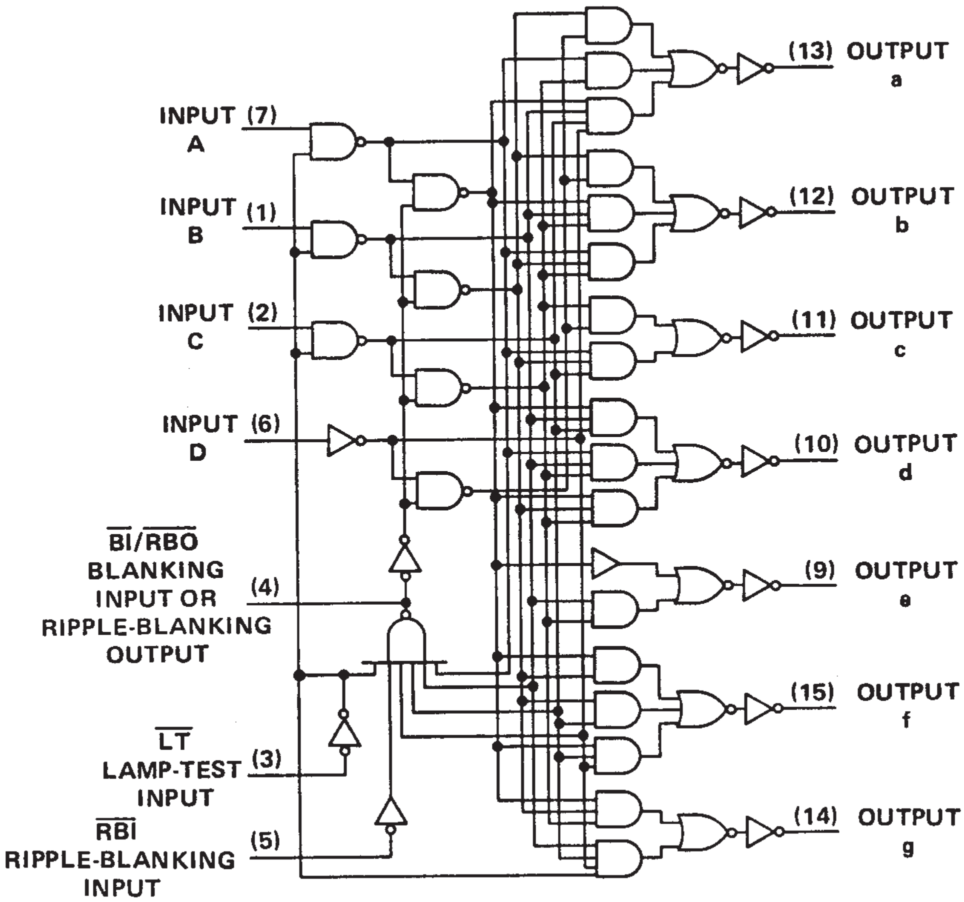


Figura 1: Schema interno del circuito integrato SN74LS47

Esempio 1.42. Le due formule seguenti sono in forma normale congiuntiva:

$$A \wedge \neg B \wedge (A \vee C)$$
$$(\neg A \vee B \vee C) \wedge (\neg C \vee A)$$

Osservazione 1.43. Un caso particolare di forma normale congiuntiva si ha quando $n = 1$. La formula $P_1 \wedge P_2 \wedge \cdots \wedge P_n$ si riduce allora alla sola P_1, la quale è una *disgiunzione* di letterali. Ciò significa che una formula come la seguente

$$\neg A \vee B \vee \neg C,$$

dove A, B e C sono proposizioni atomiche, è già nella sua forma normale *congiuntiva*!

Definizione 1.44. Una formula ben formata P è detta in *forma normale disgiuntiva* (fnd) se P è della forma $P_1 \vee P_2 \vee \cdots \vee P_n$ (per qualche $n \geq 1$), dove ciascuna P_i è una congiunzione di letterali.

Esempio 1.45. Le due formule seguenti sono in forma normale disgiuntiva:

$$A \vee (\neg B \wedge C \wedge \neg A)$$
$$(A \wedge B) \vee (C \wedge \neg A) \vee C$$

Osservazione 1.46. Un caso particolare di forma normale disgiuntiva si ha quando $n = 1$. La formula $P_1 \vee P_2 \vee \cdots \vee P_n$ si riduce allora alla sola P_1, la quale è una *congiunzione* di letterali. Ciò significa che una formula come la seguente

$$A \wedge \neg B \wedge \neg C,$$

dove A, B e C sono proposizioni atomiche, è già nella sua forma normale *disgiuntiva*!

In particolare, ciò significa che, se $A_1, A_2, \ldots, A_n$ sono dei letterali, una formula del tipo

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n$$

può essere interpretata sia come una forma normale congiuntiva, sia come una forma normale disgiuntiva. Lo stesso vale per una formula del tipo

$$A_1 \vee A_2 \vee \cdots \vee A_n.$$

L'importanza di tali forme normali è dovuta al seguente risultato:

Teorema 1.47. *Per ogni formula ben formata P esistono una formula in forma normale congiuntiva P^C e una formula in forma normale disgiuntiva P^D tali che $P \equiv P^C$ e $P \equiv P^D$.*

Dimostrazione. La dimostrazione (e la costruzione delle formule P^C e P^D a partire da P) procede nel modo seguente:

1. Si eliminano dalla formula P tutti i connettivi diversi da $\wedge$, $\vee$ e $\neg$ utilizzando le formule dimostrate nell'Esercizio 1.35 (chi non lo avesse ancora svolto lo faccia ora!).

2. Si utilizzano le leggi di De Morgan e la legge della doppia negazione per portare i simboli di negazione immediatamente davanti alle proposizioni atomiche.

3. Si utilizzano le proprietà distributive per convertire la formula così ottenuta nella forma P^C oppure P^D.

$\square$

Esempio 1.48. Consideriamo la formula $P = (A \vee \neg B) \to C$. Vogliamo determinare una sua forma normale congiuntiva P^C:

$$\begin{aligned}(A \vee \neg B) \to C &\equiv \neg(A \vee \neg B) \vee C \\ &\equiv (\neg A \wedge \neg\neg B) \vee C \\ &\equiv (\neg A \wedge B) \vee C \\ &\equiv (\neg A \vee C) \wedge (B \vee C) = P^C\end{aligned}$$

Cerchiamo ora una sua forma normale disgiuntiva P^D:

$$\begin{aligned}(A \vee \neg B) \to C &\equiv \neg(A \vee \neg B) \vee C \\ &\equiv (\neg A \wedge \neg\neg B) \vee C \\ &\equiv (\neg A \wedge B) \vee C = P^D\end{aligned}$$

Un metodo pratico per determinare una forma normale disgiuntiva P^D di una formula ben formata P contenente le proposizioni atomiche A_1, A_2, $\ldots$, A_n, è il seguente:

1. Si costruisce la tavola di verità di P.
2. Per ogni riga in cui P ha valore di verità 1 si scrive una congiunzione, i cui letterali sono determinati come segue: se nell'interpretazione v che corrisponde alla riga in questione risulta $v(A_i) = 1$ allora viene inserito A_i come letterale, altrimenti si inserisce $\neg A_i$.
3. Tutte le formule così ottenute vanno concatenate tra loro, separandole con il connettivo $\vee$.

Un esempio dovrebbe essere sufficiente a chiarire quanto sopra detto.

Esempio 1.49. Supponiamo che P sia una formula la cui tavola di verità è la seguente:

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Le righe in cui P ha valore di verità 1 sono la prima, la quinta e la sesta. P^D sarà quindi una disgiunzione di tre formule

$$P^D = (\cdots) \vee (\cdots) \vee (\cdots),$$

una per ciascuna delle tre righe menzionate. La formula corrispondente alla prima riga si ottiene nel modo seguente: dato che nella prima riga i valori di verità di A, B e C sono tutti e tre 0, dovremo scrivere una congiunzione di $\neg A$, $\neg B$ e $\neg C$. Si ha quindi

$$P^D = (\neg A \wedge \neg B \wedge \neg C) \vee (\cdots) \vee (\cdots).$$

Passiamo ora alla quinta riga: il valore di verità di A è 1, mentre quelli di B e C sono 0. Dovremo quindi scrivere una congiunzione di A, $\neg B$ e $\neg C$. Si ha pertanto

$$P^D = (\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (\cdots).$$

Infine, consideriamo la sesta riga. Il valore di verità di A è 1, quello di B è 0 e quello di C è 1. Dovremo quindi scrivere una congiunzione di A, $\neg B$ e C. Arriviamo così alla formula seguente:

$$P^D = (\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge C).$$

Esiste un algoritmo analogo per determinare una forma normale congiuntiva P^C di una formula ben formata P. Tale algoritmo si ottiene semplicemente scambiando, nel procedimento descritto in precedenza, i ruoli di 0 e 1 e quelli di $\wedge$ e $\vee$. Più precisamente, si procede come segue:

1. Si costruisce la tavola di verità di P.

2. Per ogni riga in cui P ha valore di verità 0 si scrive una disgiunzione, i cui letterali sono determinati come segue: se nell'interpretazione v che corrisponde alla riga in questione risulta $v(A_i) = 0$ allora viene inserito A_i come letterale, altrimenti si inserisce $\neg A_i$.

3. Tutte le formule così ottenute vanno concatenate tra loro, separandole con il connettivo $\wedge$.

Applichiamo questo algoritmo alla formula dell'esempio precedente:

Esempio 1.50. Supponiamo che P sia una formula la cui tavola di verità è la seguente:

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Le righe in cui P ha valore di verità 0 sono la seconda, la terza, la quarta, la settima e l'ottava. P^C sarà quindi una congiunzione di cinque formule

$$P^C = (\cdots) \wedge (\cdots) \wedge (\cdots) \wedge (\cdots) \wedge (\cdots),$$

una per ciascuna delle cinque righe menzionate. La formula corrispondente alla seconda riga si ottiene nel modo seguente: dato che nella seconda riga i valori di verità di A e B sono 0 mentre quello di C è 1, dovremo scrivere una disgiunzione di A, B e $\neg C$. Si ha quindi

$$P^C = (A \vee B \vee \neg C) \wedge (\cdots) \wedge (\cdots) \wedge (\cdots) \wedge (\cdots).$$

Passiamo ora alla terza riga: il valore di verità di A è 0, quello di B è 1 e quello di C è 0. Dovremo quindi scrivere una disgiunzione di A, $\neg B$ e C. Si ha pertanto

$$P^C = (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (\cdots) \wedge (\cdots) \wedge (\cdots).$$

Consideriamo la quarta riga. Il valore di verità di A è 0, quello di B è 1 e quello di C è 1. Dovremo quindi scrivere una disgiunzione di A, $\neg B$ e $\neg C$. Si ha pertanto

$$P^C = (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\cdots) \wedge (\cdots).$$

Consideriamo la settima riga. Il valore di verità di A è 1, quello di B è 1 e quello di C è 0. Dovremo quindi scrivere una disgiunzione di $\neg A$, $\neg B$ e C. Si ha pertanto

$$P^C = (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge (\cdots).$$

Infine, consideriamo l'ottava riga. Il valore di verità di A è 1, quello di B è 1 e quello di C è 1. Dovremo quindi scrivere una disgiunzione di $\neg A$, $\neg B$ e $\neg C$. Si arriva così alla seguente formula finale:

$$\begin{aligned} P^C = &(A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \\ &\wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C). \end{aligned}$$

Osservazione 1.51. Le formule P^C e P^D ottenute con i due metodi appena descritti non sono necessariamente le più corte possibili. A volte è possibile semplificarle, come nel seguente esempio.

Vogliamo determinare una forma normale disgiuntiva P^D della formula P la cui tavola di verità è

A	B	C	P
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Le righe in cui P ha valore di verità 1 sono la terza e la settima. Applichiamo il metodo descritto in precedenza. P^D sarà dunque una disgiunzione di due formule

$$P^D = (\cdots) \vee (\cdots).$$

La formula corrispondente alla terza riga si ottiene nel modo seguente: dato che nella terza riga il valore di verità di A è 0, quello di B è 1 e quello di C è 0, dovremo scrivere una congiunzione di $\neg A$, B e $\neg C$. Si ha quindi

$$P^D = (\neg A \wedge B \wedge \neg C) \vee (\cdots).$$

Passiamo ora alla settima riga: il valore di verità di A è 1, quello di B è 1 e quello di C è 0. Dovremo allora scrivere una congiunzione di A, B e $\neg C$. Si ha quindi

$$P^D = (\neg A \wedge B \wedge \neg C) \vee (A \wedge B \wedge \neg C).$$

Utilizzando le proprietà dei connettivi (in special modo la proprietà distributiva), questa formula può essere semplificata come segue:

$$\begin{aligned} P^D &= (\neg A \wedge B \wedge \neg C) \vee (A \wedge B \wedge \neg C) \\ &\equiv (\neg A \wedge (B \wedge \neg C)) \vee (A \wedge (B \wedge \neg C)) \\ &\equiv (\neg A \vee A) \wedge (B \wedge \neg C) \\ &\equiv B \wedge \neg C \end{aligned}$$

dove abbiamo usato il fatto che $\neg A \vee A$ è una tautologia.

Da quanto visto finora discende immediatamente il seguente risultato:

Teorema 1.52. *L'insieme di connettivi* $\{\neg, \wedge, \vee\}$ *è funzionalmente completo.*

Corollario 1.53. *Gli insiemi di connettivi* $\{\neg, \vee\}$ *e* $\{\neg, \wedge\}$ *sono funzionalmente completi.*

Dimostrazione. Basta osservare che $\wedge$ è derivabile da $\{\neg, \vee\}$, dato che si ha

$$A \wedge B \equiv \neg(\neg A \vee \neg B)$$

e, analogamente, $\vee$ è derivabile da $\{\neg, \wedge\}$, dato che si ha

$$A \vee B \equiv \neg(\neg A \wedge \neg B)$$

□

Esercizio 1.54. È possibile definire un connettivo binario (chiamiamolo $\star$) tale che l'insieme costituito da questo unico connettivo sia funzionalmente completo?

2 Logica dei Predicati

Il calcolo proposizionale che abbiamo presentato è poco espressivo. Manca la possibilità di esprimere che una certa proprietà P vale *per tutti* gli elementi di un certo insieme, oppure che *esiste* almeno un elemento che gode della proprietà P.

Se x è un elemento di un dato insieme, noi scriveremo $P(x)$ per indicare che la proprietà P vale per l'elemento x. Ad esempio, $P(x)$ potrebbe essere l'asserzione seguente: "x è un numero pari," dove x denota un elemento dell'insieme $\mathbb{N}$ dei numeri naturali.

Avremo quindi bisogno di due nuovi simboli, che indicheremo con $\forall$ e $\exists$, a cui attribuiremo i seguenti significati:

- $\forall x\, P(x)$ servirà ad indicare che la proprietà P vale *per ogni* elemento x (di un qualche insieme prefissato),
- $\exists x\, P(x)$ servirà ad indicare che la proprietà P vale *per qualche* x, cioè che esiste almeno un elemento x per cui $P(x)$ è vera.

I due simboli $\forall$ e $\exists$ si chiamano, rispettivamente, *quantificatore universale* e *quantificatore esistenziale*. Si tratterà quindi di estendere il linguaggio del calcolo proposizionale aggiungendo questi due nuovi simboli, assieme a tutta una serie di altri "oggetti" di cui avremo bisogno.

2.1 Il linguaggio del Calcolo dei Predicati

Definiamo ora il linguaggio che ci servirà a manipolare espressioni logiche contenenti i due quantificatori $\forall$ e $\exists$. Cominciamo dall'alfabeto. Esso dovrà contenere:

- Simboli di *costante*: a, b, c, ... (oppure a_1, a_2, a_3, ...)
- Simboli di *variabile*: x, y, z, ... (oppure x_1, x_2, x_3, ...)
- Simboli di *funzione*: f, g, h, ... (oppure f_1, f_2, f_3, ...)
- Simboli di *predicato*: A, B, C, ... (oppure A_1, A_2, A_3, ...)
- Connettivi: $\neg$, $\wedge$, $\vee$, $\rightarrow$
- Il simbolo $\perp$
- Quantificatori: $\forall$, $\exists$
- Simboli ausiliari: parentesi e virgole.

Di solito si richiede che i simboli di variabile siano in numero infinito (numerabile). Per quanto riguarda quelli di funzione e di predicato, solitamente si assume che siano in numero finito. L'insieme dei simboli di costante può avere cardinalità arbitraria.

Se f è un simbolo di funzione, scriveremo spesso $f^{(n)}$ per indicare in modo esplicito che f rappresenta una funzione di n variabili. Analogamente, se A è un simbolo di predicato, useremo il simbolo $A^{(n)}$ per indicare il fatto che A rappresenta un predicato n-ario.

Definiamo ora l'insieme TER dei *termini* stabilendo che:

1. ogni costante è un termine;
2. ogni variabile è un termine;
3. se t_1, t_2, ..., t_n sono dei termini e $f^{(n)}$ è un simbolo di funzione, allora $f^{(n)}(t_1, t_2, \ldots, t_n)$ è un termine.

I termini sono dunque tutte e sole quelle espressioni che si possono costruire mediante applicazioni ripetute delle tre regole date.

Possiamo ora definire le *formule atomiche* mediante le seguenti regole:

1. $\perp$ è una formula atomica;
2. ogni termine è una formula atomica;

3. se $t_1, t_2, \ldots, t_n$ sono dei termini e $A^{(n)}$ è un simbolo di predicato, allora $A^{(n)}(t_1, t_2, \ldots, t_n)$ è una formula atomica.

Anche in questo caso le formule atomiche sono tutte e sole quelle formule che si possono costruire mediante l'applicazione delle regole precedenti.

Siamo finalmente in grado di definire le formule ben formate (FBF) stabilendo che:

1. ogni formula atomica è una FBF;
2. se P e Q sono FBF, allora anche $(\neg P)$, $(P \wedge Q)$, $(P \vee Q)$ e $(P \rightarrow Q)$ sono FBF;
3. se P è una FBF, allora anche $(\forall x\, P)$ e $(\exists x\, P)$ sono FBF.

Esempio 2.1. Le seguenti sono FBF:

$$(\exists x\, ((P(x) \wedge Q) \rightarrow (\forall y\, R(y))))$$
$$(\forall x\, (\exists y\, (P(x, f(c, y)) \vee Q(y, c))))$$

Al contrario, la seguente sequenza di simboli non è una FBF:

$$\forall x\, \exists \rightarrow (\wedge P\, \forall$$

Osservazione 2.2. Le regole che abbiamo stabilito conducono ad un uso eccessivo delle parentesi. Come nel calcolo proposizionale, conviene dunque stabilire delle *priorità* tra i vari simboli:

priorità più alta	$\forall$, $\exists$, $\neg$
.	$\wedge$
.	$\vee$
priorità più bassa	$\rightarrow$

(si noti che $\forall$, $\exists$ e $\neg$ hanno la stessa priorità).

La formula

$$(\forall x\, (P(x) \rightarrow ((\exists y\, Q(x, y)) \vee (\neg P(x)))))$$

si potrà dunque scrivere come segue:

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

Si noti che l'ultima coppia di parentesi rimasta non si può togliere. Infatti, la formula

$$\forall x\, P(x) \rightarrow \exists y\, Q(x, y) \vee \neg P(x)$$

verrebbe interpretata come

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

2.2 Variabili libere e variabili legate

In una formula del tipo $\forall x\, P$ (oppure $\exists x\, P$) la variabile x si dice *legata*. Ogni quantificatore introduce un legame con le variabili presenti nel suo *campo d'azione*, ove per campo d'azione di un quantificatore si intende la sottoformula a cui "si applica" il quantificatore stesso (cioè la formula ben formata immediatamente alla sua destra). Ad esempio, nella formula

$$\exists x\, (P \to Q) \to R \wedge Q$$

il campo d'azione di $\exists x$ è la sottoformula $P \to Q$. In tale formula la variabile x è dunque legata.

Una variabile si dice *libera* se non è legata. Ad esempio, nella formula

$$\forall x\, A(x, y) \vee \neg B(y, z),$$

la variabile x è legata, mentre le variabili y e z sono libere.

Indicheremo con $\mathrm{FV}(P)$ l'insieme delle variabili libere (*free variables*) della formula ben formata P. Esso può essere definito in modo rigoroso come segue:

Definizione 2.3. Se t è un termine, l'insieme $\mathrm{FV}(t)$ è definito induttivamente mediante le seguenti regole:

1. $\mathrm{FV}(c) = \varnothing$, per ogni costante c;
2. $\mathrm{FV}(x) = \{x\}$, per ogni variabile x;
3. $\mathrm{FV}(f^{(n)}(t_1, \ldots, t_n)) = \mathrm{FV}(t_1) \cup \cdots \cup \mathrm{FV}(t_n)$, per ogni funzione n-aria $f^{(n)}$ (e, naturalmente, per ogni n).

Definizione 2.4. Se P è una FBF, l'insieme $\mathrm{FV}(P)$ è definito induttivamente mediante le seguenti regole:

1. $\mathrm{FV}(\bot) = \varnothing$;
2. $\mathrm{FV}(A^{(n)}(t_1, \ldots, t_n)) = \mathrm{FV}(t_1) \cup \cdots \cup \mathrm{FV}(t_n)$, per ogni predicato n-ario $A^{(n)}$ (e, naturalmente, per ogni n);
3. $\mathrm{FV}(\neg P) = \mathrm{FV}(P)$, per ogni formula ben formata P;
4. $\mathrm{FV}(P \wedge Q) = \mathrm{FV}(P) \cup \mathrm{FV}(Q)$, per ogni P e Q;
5. $\mathrm{FV}(P \vee Q) = \mathrm{FV}(P) \cup \mathrm{FV}(Q)$, per ogni P e Q;
6. $\mathrm{FV}(P \to Q) = \mathrm{FV}(P) \cup \mathrm{FV}(Q)$, per ogni P e Q;

7. $\mathrm{FV}(\forall x\, P) = \mathrm{FV}(P) \setminus \{x\}$, per ogni P;
8. $\mathrm{FV}(\exists x\, P) = \mathrm{FV}(P) \setminus \{x\}$, per ogni P.

Definizione 2.5. Una formula ben formata P è detta *chiusa* se essa non contiene variabili libere, cioè se $\mathrm{FV}(P) = \varnothing$. In caso contrario essa è detta *aperta*.

Indicheremo con $\mathrm{BV}(P)$ l'insieme delle variabili legate (*bound variables*) della formula P.

Facciamo subito notare che, per una formula ben formata P, è possibile avere

$$\mathrm{FV}(P) \cap \mathrm{BV}(P) \neq \varnothing,$$

cioè una variabile può comparire all'interno di una formula P sia come variabile libera che come variabile legata. Ad esempio, se indichiamo con P la seguente formula

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

si ha $\mathrm{FV}(P) = \{y, x\}$ e $\mathrm{BV}(P) = \{x, y, z\}$.

Tuttavia, ogni *occorrenza* di una data variabile all'interno di una formula ben formata è libera oppure legata.

Osservazione 2.6. Può accadere che una variabile compaia nel campo d'azione di più quantificatori, come, ad esempio, nella formula

$$\forall x\, (A(x) \to \forall x\, B(x)),$$

dove la seconda occorrenza di x (cioè quella in $B(x)$) si trova nel campo d'azione di entrambi i quantificatori. In questo caso si assume che il legame corretto sia con il quantificatore più interno.

2.2.1 Sostituzione

Anche se non abbiamo ancora parlato dell'interpretazione di una FBF contenente dei quantificatori, ricordiamo che il simbolo $\forall$ è stato introdotto allo scopo di poter esprimere il fatto che una determinata proprietà valga per tutti gli elementi di un dato insieme: $\forall x\, P(x)$ dovrà dunque significare che la proprietà P vale *per tutti* gli elementi di un prefissato insieme. Da ciò segue che il fatto di aver indicato la variabile quantificata con la lettera x è del tutto irrilevante; avremmo anche potuto scrivere $\forall y\, P(y)$, oppure $\forall z\, P(z)$, ecc. Un discorso del tutto analogo vale anche per il quantificatore esistenziale $\exists$.

È dunque naturale assumere che nel nostro linguaggio valga la seguente regola: *ogni variabile che compare nel campo d'azione di un quantificatore* *può essere sostituita con un'altra, a patto che quest'ultima non compaia come variabile libera all'interno della stessa formula.*

Cerchiamo di chiarire con un esempio quest'ultima richiesta. Se le variabili x e y rappresentano dei numeri naturali, la formula

$$\exists x\,(x > y)$$

esprime il fatto che, dato un numero intero y, esiste un qualche numero intero x che sia maggiore di y (il che, in effetti, è vero).

In base a quanto detto, noi possiamo sostituire ogni occorrenza della variabile quantificata x con un'altra variabile, ad esempio z, ottenendo la formula

$$\exists z\,(z > y)$$

che esprime esattamente la stessa affermazione.

Non possiamo invece sostituire la variabile quantificata x con la variabile y, dato che y compare come variabile libera nella formula $x > y$. Se lo facessimo otterremmo infatti la formula

$$\exists y\,(y > y)$$

la quale affermerebbe che esiste un qualche numero intero che è maggiore di sé stesso (il che è falso).

Introduciamo ora una notazione adeguata per indicare la sostituzione all'interno di una formula.

Definizione 2.7. Siano R e P due formule ben formate e sia A una formula atomica. Indicheremo con $R[{}^{P}\!/_{A}]$ la formula ben formata ottenuta rimpiazzando tutte le occorrenze di A in R con la formula P.

Esempio 2.8. Se $R = \neg A \wedge B \rightarrow B \vee A$ e $P = \neg B \vee C$, allora si ha

$$R[{}^{P}\!/_{A}] = \neg(\neg B \vee C) \wedge B \rightarrow B \vee (\neg B \vee C).$$

Definiamo ora, in modo formale, la sostituzione all'interno dei termini.

Definizione 2.9. Siano s e t due termini e x una variabile. Il termine $s[{}^{t}\!/_{x}]$ è definito mediante le seguenti regole:

1. se s è una costante c, si pone $c[{}^{t}\!/_{x}] = c$;

2. se s è una variabile y, allora si pone

$$y[{}^{t}\!/_{x}] = \begin{cases} y & \text{se } y \neq x, \\ t & \text{se } y = x; \end{cases}$$

3. se t_1, t_2, ..., t_n sono termini e $f^{(n)}$ è una funzione, allora si pone

$$f^{(n)}(t_1, \ldots, t_n)[{}^t/_x] = f^{(n)}(t_1[{}^t/_x], \ldots, t_n[{}^t/_x]).$$

Passiamo ora a definire la sostituzione all'interno delle formule ben formate. Come già accennato in precedenza, in questo caso si presentano dei problemi. Infatti, se P è una formula ben formata in cui compaiono una variabile libera x e una variabile legata y, mentre t è un termine in cui la y compare come variabile libera, operando la sostituzione $P[{}^t/_x]$, la variabile y che compariva in t come variabile libera diventa ora legata nella formula $P[{}^t/_x]$.

Possiamo illustrare questo fatto con il seguente esempio. Consideriamo la formula $P = \exists y\,(x < y)$, in cui x compare come variabile libera e y come variabile legata. Se consideriamo il termine $t = z$, in cui non compare la variabile y, non incontriamo nessun problema: la formula $P[{}^z/_x]$ è $\exists y\,(z < y)$, che esprime la stessa asserzione della formula P. Al contrario, se consideriamo il termine $t = y$, in cui compare proprio la y come variabile libera, si ottiene la formula $P[{}^y/_x] = \exists y\,(y < y)$, la quale ha un "significato" profondamente diverso. Questo problema può tuttavia essere risolto ridenominando opportunamente la variabile legata y all'interno della formula P *prima* di operare la sostituzione del termine t al posto della variabile x. In questo modo si ha, infatti,

$$P[{}^z/_y][{}^y/_x] = (\exists y\,(x < y))[{}^z/_y][{}^y/_x] = (\exists z\,(x < z))[{}^y/_x] = (\exists z\,(y < z)),$$

la quale ha, intuitivamente, lo stesso significato della formula P.

Chiarito questo aspetto, diamo ora la definizione formale:

Definizione 2.10. Siano P una formula ben formata, t un termine e x una variabile. La formula $P[{}^t/_x]$ è definita mediante le seguenti regole:

1. $\bot[{}^t/_x] = \bot$;

2. se t_1, t_2, ..., t_n sono dei termini e $A^{(n)}$ è un simbolo di predicato, allora $A^{(n)}(t_1, \ldots, t_n)[{}^t/_x] = A^{(n)}(t_1[{}^t/_x], \ldots, t_n[{}^t/_x])$;

3. se P e Q sono FBF, allora $(\neg P)[{}^t/_x] = \neg(P[{}^t/_x])$, $(P \wedge Q)[{}^t/_x] = (P[{}^t/_x] \wedge Q[{}^t/_x])$, $(P \vee Q)[{}^t/_x] = (P[{}^t/_x] \vee Q[{}^t/_x])$ e $(P \to Q)[{}^t/_x] = (P[{}^t/_x] \to Q[{}^t/_x])$;

4. se P è una FBF, allora

$$(\forall y\,P)[{}^t/_x] = \begin{cases} \forall y\,(P[{}^t/_x]) & \text{se } x \neq y \text{ e } y \notin \mathrm{FV}(t), \\ \forall z\,(P[{}^z/_y][{}^t/_x]) & \text{se } x \neq y,\ y \in \mathrm{FV}(t) \text{ e } z \text{ non occorre in } P \text{ o } t, \\ \forall y\,P & \text{se } x = y; \end{cases}$$

5. se P è una FBF, allora

$$(\exists y\, P)[{}^{t}/_{x}] = \begin{cases} \exists y\, (P[{}^{t}/_{x}]) & \text{se } x \neq y \text{ e } y \notin \mathrm{FV}(t), \\ \exists z\, (P[{}^{z}/_{y}][{}^{t}/_{x}]) & \text{se } x \neq y,\ y \in \mathrm{FV}(t) \text{ e } z \text{ non occorre in } P \text{ o } t, \\ \exists y\, P & \text{se } x = y. \end{cases}$$

Osservazione 2.11. Se t_1, t_2, ..., t_n sono dei termini e x_1, x_2, ..., x_n sono delle variabili, indicheremo con il simbolo $P[{}^{t_1,\ldots,t_n}/_{x_1,\ldots,x_n}]$ la *sostituzione simultanea* all'interno della formula P. In generale il suo effetto è diverso dalle corrispondenti sostituzioni iterate. Si ha infatti

$$P(x,y)[{}^{y,x}/_{x,y}] = P(y,x),$$

mentre

$$P(x,y)[{}^{y}/_{x}][{}^{x}/_{y}] = P(y,y)[{}^{x}/_{y}] = P(x,x).$$

2.3 La Semantica del Calcolo dei Predicati

Ci poniamo ora il problema di attribuire un significato a tutte le formule ben formate del nostro linguaggio. Se nel caso del Calcolo Proposizionale ciò era piuttosto facile (ricordiamo che si trattava solo di attribuire un valore di verità alle FBF), nel caso del Calcolo dei Predicati l'attribuzione di un valore di verità ad una formula ben formata risulta complicata a causa della presenza di costanti, variabili, funzioni, ecc.

Innanzitutto dovremo fissare un *dominio*, cioè un insieme D, sul quale assumeranno i loro valori i simboli di costante ed i simboli di variabile. Poi dovremo far corrispondere ad ogni simbolo di funzione $f^{(n)}$ una funzione vera e propria da D^n in D (cioè una funzione di n variabili a valori nel dominio D). Infine, ad ogni simbolo di predicato $A^{(n)}$ dovremo far corrispondere una *relazione* n-aria sull'insieme D, cioè una funzione definita su D^n a valori nell'insieme $\{0,1\}$ (VERO o FALSO).

Cominciamo quindi col dare la seguente definizione.

Definizione 2.12. Una *struttura* $\mathscr{A}$ è il dato di un insieme non vuoto $D = D_{\mathscr{A}}$, detto *dominio*, e di un *assegnamento* che associa:

1. ad ogni simbolo di costante c un elemento $c^{\mathscr{A}} \in D$;

2. ad ogni simbolo di funzione $f^{(n)}$ una funzione

$$f^{\mathscr{A}} : D^n \to D;$$

3. ad ogni simbolo di predicato $B^{(n)}$ una relazione n-aria

$$B^{\mathscr{A}} : D^n \to \{0, 1\}.$$

Notiamo che, in generale, non è possibile assegnare un valore di verità ad una formula che contenga delle variabili libere; sarà prima necessario assegnare a tali variabili degli elementi del dominio. Di conseguenza il valore di verità di una formula dipenderà, in generale, dallo specifico assegnamento scelto.

Diamo dunque la seguente definizione:

Definizione 2.13. Data una struttura $\mathscr{A}$ con dominio D, chiameremo *ambiente* per $\mathscr{A}$ (in inglese, *environment*) una funzione $\xi = \xi^{\mathscr{A}}$, definita sull'insieme VAR delle variabili, a valori in D (una tale funzione assegna dunque ad ogni variabile un elemento del dominio). Indicheremo con

$$\mathrm{ENV}_D = \{\xi : \mathrm{VAR} \to D\}$$

l'insieme di tutti gli ambienti per la struttura $\mathscr{A}$.

Se ξ è un ambiente per una struttura $\mathscr{A}$ e $a \in D$, indicheremo con $\xi[{}^a\!/_x]$ l'ambiente ξ modificato in modo da associare alla variabile x l'elemento a. Più precisamente, si ha:

$$\xi[{}^a\!/_x](y) = \begin{cases} \xi(y) & \text{se } y \neq x, \\ a & \text{se } y = x. \end{cases}$$

Definizione 2.14. Una *interpretazione* $\mathscr{I} = (\mathscr{A}, \xi)$ è il dato di una struttura $\mathscr{A}$ e di un ambiente ξ per tale struttura. Se a è un elemento di D, scriveremo $\mathscr{I}[{}^a\!/_x]$ per indicare l'interpretazione data dalla struttura $\mathscr{A}$ e dall'ambiente $\xi[{}^a\!/_x]$.

Definiamo ora il valore di un termine in una data interpretazione:

Definizione 2.15. Si consideri un'interpretazione $\mathscr{I} = (\mathscr{A}, \xi)$ e un termine t. Denotiamo il suo valore nell'interpretazione $\mathscr{I}$ con $[\![t]\!]^{\mathscr{A}}_{\xi}$. La funzione $[\![\ ]\!]^{\mathscr{A}}_{\xi} : \mathrm{TER} \to D$ è definita in modo induttivo come segue:

1. se t è una costante c, si ha $[\![t]\!]^{\mathscr{A}}_{\xi} = c^{\mathscr{A}}$;

2. se t è una variabile x, si ha $[\![t]\!]^{\mathscr{A}}_{\xi} = \xi(x)$;

3. se t è una funzione $f^{(n)}(t_1, \ldots, t_n)$, ove t_1, …, t_n sono termini, si ha $[\![t]\!]^{\mathscr{A}}_{\xi} = f^{\mathscr{A}}([\![t_1]\!]^{\mathscr{A}}_{\xi}, \ldots, [\![t_n]\!]^{\mathscr{A}}_{\xi})$.

Possiamo ora definire formalmente il valore di verità di una formula ben formata P nell'interpretazione $\mathscr{I} = (\mathscr{A}, \xi)$, il quale verrà indicato con $v^{\mathscr{I}}(P) = v^{(\mathscr{A},\xi)}(P)$.

Definizione 2.16. La funzione di valutazione $v^{\mathscr{I}} : \mathrm{FBF} \to \{0,1\}$ è definita induttivamente come segue:

1. $v^{\mathscr{I}}(\bot) = 0$;
2. $v^{\mathscr{I}}(P(t_1, \ldots, t_n)) = P^{\mathscr{A}}([\![t_1]\!]^{\mathscr{A}}_{\xi}, \ldots, [\![t_n]\!]^{\mathscr{A}}_{\xi})$;
3. $v^{\mathscr{I}}(\neg P) = 1 - v^{\mathscr{I}}(P)$;
4. $v^{\mathscr{I}}(P \wedge Q) = \min\{v^{\mathscr{I}}(P), v^{\mathscr{I}}(Q)\}$;
5. $v^{\mathscr{I}}(P \vee Q) = \max\{v^{\mathscr{I}}(P), v^{\mathscr{I}}(Q)\}$;
6. $v^{\mathscr{I}}(P \to Q) = \max\{1 - v^{\mathscr{I}}(P), v^{\mathscr{I}}(Q)\}$;
7. $v^{\mathscr{I}}(\forall x\, P) = \min\{v^{\mathscr{I}[a/x]}(P) \mid a \in D\}$;
8. $v^{\mathscr{I}}(\exists x\, P) = \max\{v^{\mathscr{I}[a/x]}(P) \mid a \in D\}$.

Osservazione 2.17. Si noti che le definizioni date ai punti 3, 4, 5 e 6 dei valori di verità delle formule costruite mediante l'uso dei connettivi $\neg$, $\wedge$, $\vee$ e $\to$, sono equivalenti a quelle date nel Calcolo Proposizionale mediante l'uso delle tavole di verità.

Osservazione 2.18. Si noti che il quantificatore universale $\forall$ può essere pensato come una "congiunzione iterata"

$$\forall x\, P(x) \text{ ha lo stesso significato di } \bigwedge_{a \in D} P(a).$$

Analogamente, il quantificatore esistenziale $\exists$ può essere pensato come una "disgiunzione iterata"

$$\exists x\, P(x) \text{ ha lo stesso significato di } \bigvee_{a \in D} P(a).$$

Osservazione 2.19. Al contrario di quanto avviene nel Calcolo Proposizionale, le definizioni che abbiamo dato non consentono, in generale, di determinare in modo effettivo il valore di verità di una formula ben formata P in una data interpretazione $\mathscr{I}$. Infatti, se il dominio D è un insieme infinito e se la formula P contiene dei quantificatori, per determinare il valore di verità $v^{\mathscr{I}}(P)$ sarebbe necessario calcolare il valore di verità delle *infinite* formule che si ottengono da P sostituendo alle variabili quantificate gli infiniti elementi di D.

Possiamo, infine, notare che il valore di verità di una formula ben formata P in una data interpretazione $\mathscr{I} = (\mathscr{A}, \xi)$ dipende solo dalla restrizione di ξ all'insieme delle variabili libere di P. Si ha infatti:

Teorema 2.20. *Siano P una formula ben formata e $\mathscr{A}$ una struttura. Sia $\mathrm{FV}(P) = \{y_1, y_2, \ldots, y_n\}$ l'insieme delle variabili libere di P. Allora, per tutti gli ambienti ξ_1 e ξ_2 per $\mathscr{A}$ tali che $\xi_1(y_i) = \xi_2(y_i)$, per ogni $i = 1, \ldots, n$, si ha $v^{(\mathscr{A},\xi_1)}(P) = v^{(\mathscr{A},\xi_2)}(P)$.*

Dimostrazione. La dimostrazione procede per induzione sulla struttura della formula ben formata P. □

2.4 Soddisfacibilità, Validità e Modelli

Definizione 2.21. Sia P una formula ben formata. Diremo che:

- P è *soddisfatta* in una struttura $\mathscr{A}$ rispetto all'ambiente ξ se $v^{(\mathscr{A},\xi)}(P) = 1$. In tal caso scriveremo $(\mathscr{A}, \xi) \models P$.
- P è *soddisfacibile* in una struttura $\mathscr{A}$ se esiste un ambiente ξ tale che $v^{(\mathscr{A},\xi)}(P) = 1$.
- P è *vera* in una struttura $\mathscr{A}$ se per ogni ambiente ξ si ha $v^{(\mathscr{A},\xi)}(P) = 1$. In tal caso diremo che $\mathscr{A}$ è un *modello* per P, e scriveremo $\mathscr{A} \models P$.
- P è *soddisfacibile* se esistono una struttura $\mathscr{A}$ ed un ambiente ξ tale che $v^{(\mathscr{A},\xi)}(P) = 1$.
- P è *valida* se essa è vera in ogni struttura. In tal caso scriveremo $\models P$.

Osservazione 2.22. La nozione di *formula valida* rappresenta la controparte, nel Calcolo dei Predicati, della nozione di *tautologia*.

Definizione 2.23. Sia Γ un insieme di formule ben formate. Diremo che:

- Γ è *soddisfacibile* se esistono una struttura $\mathscr{A}$ ed un ambiente ξ tali che, per ogni formula $P \in \Gamma$, si abbia $v^{(\mathscr{A},\xi)}(P) = 1$.
- Una struttura $\mathscr{A}$ è un *modello* per Γ se, per ogni $P \in \Gamma$, si ha $\mathscr{A} \models P$. In tal caso scriveremo $\mathscr{A} \models \Gamma$.
- Γ è *valido* se ogni struttura è un modello per Γ. In tal caso scriveremo $\models \Gamma$.

Definizione 2.24. Sia P una formula ben formata. Diremo che:

- P è *falsa* in una struttura $\mathscr{A}$ se essa non è soddisfacibile in $\mathscr{A}$, cioè se non esiste alcun ambiente ξ tale che $v^{(\mathscr{A},\xi)}(P) = 1$. In tal caso scriveremo $\mathscr{A} \not\models P$.

- P è *insoddisfacibile* (o *contraddittoria*) se non è soddisfacibile, cioè se essa è falsa in ogni struttura.

Diamo ora la generalizzazione della nozione di *conseguenza semantica*.

Definizione 2.25. Dato un insieme di formule ben formate Γ ed una formula Q, diremo che Q è *conseguenza semantica* di Γ, e scriveremo $\Gamma \models Q$, se per ogni struttura $\mathscr{A}$ ed ogni ambiente ξ per i quali si abbia $v^{(\mathscr{A},\xi)}(P) = 1$ per ogni $P \in \Gamma$, risulta anche $v^{(\mathscr{A},\xi)}(Q) = 1$.

Dalle definizioni precedenti segue immediatamente il seguente risultato:

Teorema 2.26. *Siano Γ un insieme di FBF e P una formula ben formata. Allora:*

- *P è valida se e solo se $\neg P$ è insoddisfacibile.*

- *P è soddisfacibile se e solo se $\neg P$ non è valida.*

- *$\Gamma \models P$ se e solo se $\Gamma \cup \{\neg P\}$ è insoddisfacibile.*

Osservazione 2.27. Notiamo che affermare che una formula P non è valida non implica che essa sia contraddittoria (cioè che $\neg P$ sia valida), ma solo che $\neg P$ è soddisfacibile.

Studiamo ora alcune proprietà della relazione di soddisfacibilità.

Definizione 2.28. Sia P una formula ben formata e $\mathrm{FV}(P) = \{x_1, \ldots, x_n\}$ l'insieme delle variabili libere di P. La *chiusura universale* di P è la formula

$$\mathrm{Cl}(P) = \forall x_1, \ldots, x_n\, P,$$

mentre la *chiusura esistenziale* di P è la formula

$$\mathrm{Ex}(P) = \exists x_1, \ldots, x_n\, P.$$

Osservazione 2.29. Un'espressione del tipo $\forall x_1, \ldots, x_n\, P$ è solo una forma abbreviata per indicare la formula

$$\forall x_1\, (\ldots (\forall x_{n-1}\, (\forall x_n\, P)) \cdots).$$

Una considerazione del tutto analoga vale anche per il quantificatore $\exists$.

Valgono i seguenti risultati, di cui omettiamo le dimostrazioni:

Teorema 2.30. *Sia P una formula ben formata. Allora P è valida se e solo se* $\mathrm{Cl}(P)$ *lo è.*

Teorema 2.31. *Sia P una formula ben formata. Allora P è soddisfacibile se e solo se* $\mathrm{Ex}(P)$ *lo è.*

Osservazione 2.32. In generale, verificare la validità di una formula in modo semantico è complicato. Si tratta infatti di provare che tale formula vale per ogni possibile interpretazione. Al contrario, la semantica è molto utile per dimostrare la non validità di una formula P; basta infatti esibire una interpretazione che non la soddisfa.

2.5 Equivalenza Semantica

Definizione 2.33. Due formule ben formate P e Q sono *(semanticamente) equivalenti* se, per tutte le interpretazioni $\mathscr{I} = (\mathscr{A}, \xi)$ si ha $v^{\mathscr{I}}(P) = v^{\mathscr{I}}(Q)$. Per indicare che P e Q sono semanticamente equivalenti scriveremo $P \equiv Q$.

Osservazione 2.34. Possiamo notare che due formule P e Q sono semanticamente equivalenti se e solo se $\models P \leftrightarrow Q$.

È possibile dimostrare che ridenominando una variabile quantificata all'interno del campo d'azione di un quantificatore si ottiene una nuova formula semanticamente equivalente a quella data (naturalmente a condizione che il nuovo simbolo di variabile non compaia già all'interno della formula in questione). Più precisamente, si ha:

Teorema 2.35. *Siano P una formula ben formata e z un simbolo di variabile che non compare in P. Allora si hanno le seguenti equivalenze:*

- $\exists x\, P \equiv \exists z\, (P[^z/_x])$,
- $\forall x\, P \equiv \forall z\, (P[^z/_x])$.

Elenchiamo adesso, senza dimostrazione, una serie di utili risultati:

Teorema 2.36. *Sia P una formula ben formata. Si hanno le seguenti equivalenze:*

- $\neg\forall x\, P \equiv \exists x\, \neg P$;
- $\neg\exists x\, P \equiv \forall x\, \neg P$;
- $\forall x\, P \equiv \neg\exists x\, \neg P$;

- $\exists x\, P \equiv \neg\forall x\, \neg P$.

Teorema 2.37. *Sia P una formula ben formata. Si hanno le seguenti equivalenze:*

- $\forall x\, \forall y\, P \equiv \forall y\, \forall x\, P$;
- $\exists x\, \exists y\, P \equiv \exists y\, \exists x\, P$;
- $\forall x\, P \equiv P$, *se* $x \notin \mathrm{FV}(P)$;
- $\exists x\, P \equiv P$, *se* $x \notin \mathrm{FV}(P)$.

Osservazione 2.38. Si noti però che, in generale, $\forall x\, \exists y\, P \not\equiv \exists y\, \forall x\, P$.

Teorema 2.39. *Siano P_1 e P_2 due formule ben formate. Si hanno le seguenti equivalenze:*

- $\forall x\, (P_1 \wedge P_2) \equiv \forall x\, P_1 \wedge \forall x\, P_2$;
- $\exists x\, (P_1 \vee P_2) \equiv \exists x\, P_1 \vee \exists x\, P_2$;
- $\forall x\, (P_1 \vee P_2) \equiv \forall x\, P_1 \vee P_2$, *se* $x \notin \mathrm{FV}(P_2)$;
- $\exists x\, (P_1 \wedge P_2) \equiv \exists x\, P_1 \wedge P_2$, *se* $x \notin \mathrm{FV}(P_2)$.

Osservazione 2.40. Si faccia tuttavia molta attenzione perché, in generale, si ha:

- $\forall x\, (P_1 \vee P_2) \not\equiv \forall x\, P_1 \vee \forall x\, P_2$;
- $\exists x\, (P_1 \wedge P_2) \not\equiv \exists x\, P_1 \wedge \exists x\, P_2$.

In generale, se vogliamo spostare i quantificatori attraverso i connettivi $\wedge$ e $\vee$, sarà necessario ridenominare in modo opportuno le variabili:

Teorema 2.41. *Siano P_1 e P_2 due formule ben formate. Indichiamo con Q_1 e Q_2 due quantificatori. Allora si ha:*

- $Q_1 x\, P_1 \vee Q_2 x\, P_2 \equiv Q_1 x\, Q_2 z\, (P_1 \vee P_2[{}^{z}/_{x}])$,
- $Q_1 x\, P_1 \wedge Q_2 x\, P_2 \equiv Q_1 x\, Q_2 z\, (P_1 \wedge P_2[{}^{z}/_{x}])$,

ove $Q_1, Q_2 \in \{\forall, \exists\}$ *e* $z \notin \mathrm{FV}(P_1) \cup \mathrm{FV}(P_2)$.

Infine, vale anche il seguente risultato:

Teorema 2.42. *Siano P e Q due formule ben formate. Se $x \notin \mathrm{FV}(Q)$, si ha:*

- $\forall x\, P \to Q \equiv \exists x\, (P \to Q)$;
- $\exists x\, P \to Q \equiv \forall x\, (P \to Q)$;
- $Q \to \exists x\, P \equiv \exists x\, (Q \to P)$;
- $Q \to \forall x\, P \equiv \forall x\, (Q \to P)$.

Dimostrazione. Dimostriamo la prima equivalenza:

$$\begin{aligned} \forall x\, P \to Q &\equiv (\neg\forall x\, P) \vee Q \\ &\equiv (\exists x\, \neg P) \vee Q \\ &\equiv \exists x\, (\neg P \vee Q) \\ &\equiv \exists x\, (P \to Q). \end{aligned}$$

La seconda si dimostra in modo analogo:

$$\begin{aligned} \exists x\, P \to Q &\equiv (\neg\exists x\, P) \vee Q \\ &\equiv (\forall x\, \neg P) \vee Q \\ &\equiv \forall x\, (\neg P \vee Q) \\ &\equiv \forall x\, (P \to Q). \end{aligned}$$

Dimostriamo ora la terza equivalenza:

$$\begin{aligned} Q \to \exists x\, P &\equiv \neg Q \vee (\exists x\, P) \\ &\equiv \exists x\, (\neg Q \vee P) \\ &\equiv \exists x\, (Q \to P). \end{aligned}$$

Per quanto riguarda la quarta equivalenza, si ha:

$$\begin{aligned} Q \to \forall x\, P &\equiv \neg Q \vee (\forall x\, P) \\ &\equiv \forall x\, (\neg Q \vee P) \\ &\equiv \forall x\, (Q \to P). \end{aligned}$$

□

Osservazione 2.43. Si noti che, nel teorema precedente, si può sempre fare in modo che l'ipotesi $x \notin \mathrm{FV}(Q)$ sia verificata. A tal fine basta infatti ridenominare opportunamente le variabili legate.

2.6 Forme Normali

In questa sezione mostreremo come sia possibile trasformare una qualunque formula ben formata in una particolare forma normale, detta *forma normale prenessa*.

Definizione 2.44. Una formula ben formata P è in *forma normale prenessa* se ha la seguente forma:

$$P = Q_1 x_1 \, Q_2 x_2 \dots Q_n x_n \, P_1,$$

per qualche $n \geq 0$, dove $Q_1, \dots, Q_n \in \{\exists, \forall\}$ e dove la sottoformula P_1 non contiene alcun quantificatore.

L'espressione $Q_1 x_1 \, Q_2 x_2 \dots Q_n x_n$ viene detta *prefisso*, mentre P_1 è detta la *matrice* della formula P.

Dimostriamo ora il seguente risultato:

Teorema 2.45. *Per ogni formula ben formata P esiste una formula in forma normale prenessa ad essa equivalente.*

Dimostrazione. La dimostrazione procede per induzione sulla struttura di P.

1. Se P è una formula atomica, essa è già in forma normale prenessa.

2. Supponiamo che sia $P = \neg P_1$, per qualche formula ben formata P_1. Per ipotesi induttiva esiste una formula P_1' in forma normale prenessa equivalente a P_1. Allora, per determinare la forma normale prenessa di P è sufficiente scambiare di posto il simbolo di negazione $\neg$ con i quantificatori che compaiono nella formula P_1'.

3. Supponiamo che sia $P = P_1 \wedge P_2$ (risp. $P = P_1 \vee P_2$, $P = P_1 \to P_2$), ove P_1 e P_2 sono formule ben formate. Per ipotesi induttiva esistono due formule in forma normale prenessa P_1' e P_2' rispettivamente equivalenti a P_1 e P_2. Allora, per determinare la forma normale prenessa di P è sufficiente scambiare di posto il connettivo $\wedge$ (risp. $\vee$, $\to$) con i quantificatori che compaiono nelle formule P_1' e P_2'.

4. Infine, supponiamo che sia $P = Qx \, P_1$, per qualche formula ben formata P_1, ove $Q \in \{\exists, \forall\}$. Per ipotesi induttiva esiste una formula P_1' in forma normale prenessa equivalente a P_1. Allora, la formula $P' = Qx \, P_1'$ è in forma normale prenessa ed è equivalente a P.

$\square$

Illustriamo il contenuto del precedente teorema con alcuni esempi.

Esempio 2.46. Consideriamo la formula

$$P = \forall x\, A(x) \to \forall y\, B(y)$$

e cerchiamo una sua forma normale prenessa. Utilizzando le equivalenze dimostrate nella sezione precedente, si ha:

$$\begin{aligned} \forall x\, A(x) \to \forall y\, B(y) &\equiv \exists x\, (A(x) \to \forall y\, B(y)) \\ &\equiv \exists x\, (\forall y\, (A(x) \to B(y))) \\ &\equiv \exists x\, \forall y\, (A(x) \to B(y)). \end{aligned}$$

Tuttavia si ha anche:

$$\begin{aligned} \forall x\, A(x) \to \forall y\, B(y) &\equiv \forall y\, (\forall x\, A(x) \to B(y)) \\ &\equiv \forall y\, (\exists x\, (A(x) \to B(y))) \\ &\equiv \forall y\, \exists x\, (A(x) \to B(y)). \end{aligned}$$

Da ciò si deduce che le due formule in forma normale prenessa $\exists x\, \forall y\, (A(x) \to B(y))$ e $\forall y\, \exists x\, (A(x) \to B(y))$ sono tra loro equivalenti (essendo entrambe equivalenti alla formula P). Si confronti questo risultato con quanto affermato nell'Osservazione 2.38.

Esempio 2.47. Vogliamo ora determinare la forma normale prenessa della formula

$$P = \forall x\, A(x) \to \neg\forall y\, B(y).$$

Utilizzando le equivalenze dimostrate nella sezione precedente, si ha:

$$\begin{aligned} \forall x\, A(x) \to \neg\forall y\, B(y) &\equiv \forall x\, A(x) \to \exists y\, \neg B(y) \\ &\equiv \exists y\, (\forall x\, A(x) \to \neg B(y)) \\ &\equiv \exists y\, \exists x\, (A(x) \to \neg B(y)). \end{aligned}$$

Quest'ultima formula è in forma normale prenessa.

Esempio 2.48. Consideriamo ora una formula un po' più complessa:

$$P = \neg\forall x\, A(x) \wedge \exists y\, B(y) \to \exists x\, C(x, y).$$

Osserviamo che la variabile legata x compare sia nella sottoformula $\forall x\, A(x)$ che nella sottoformula $\exists x\, C(x, y)$, inoltre la variabile y compare sia come variabile libera nella sottoformula $\exists x\, C(x, y)$ che come variabile legata nella sottoformula $\exists y\, B(y)$. Come già precedentemente osservato, problemi di questo tipo possono essere facilmente risolti ridenominando opportunamente le variabili:

$$\forall x\, A(x) \equiv \forall x_1\, A(x_1), \quad \exists y\, B(y) \equiv \exists x_2\, B(x_2), \quad \exists x\, C(x,y) \equiv \exists x_3\, C(x_3,y).$$

Si ha dunque:

$$\begin{aligned}
P &\equiv \neg\forall x_1\, A(x_1) \wedge \exists x_2\, B(x_2) \to \exists x_3\, C(x_3,y) \\
&\equiv \exists x_1\, \neg A(x_1) \wedge \exists x_2\, B(x_2) \to \exists x_3\, C(x_3,y) \\
&\equiv \exists x_1\, (\neg A(x_1) \wedge \exists x_2\, B(x_2)) \to \exists x_3\, C(x_3,y) \\
&\equiv \exists x_1\, \exists x_2\, (\neg A(x_1) \wedge B(x_2)) \to \exists x_3\, C(x_3,y) \\
&\equiv \forall x_1\, (\exists x_2\, (\neg A(x_1) \wedge B(x_2)) \to \exists x_3\, C(x_3,y)) \\
&\equiv \forall x_1\, \forall x_2\, (\neg A(x_1) \wedge B(x_2) \to \exists x_3\, C(x_3,y)) \\
&\equiv \forall x_1\, \forall x_2\, \exists x_3\, (\neg A(x_1) \wedge B(x_2) \to C(x_3,y)).
\end{aligned}$$