Marco Barlotti

Appunti di

Logica

per l'insegnamento di "Matematica Discreta e Logica"
del corso di laurea triennale in Informatica

Vers. 2.0
Anno Accademico 2019-2020

In copertina un disegno di autore ignoto.

PERCHE’ QUESTI APPUNTI, E COME USARLI

(Prefazione alla vers. 2.0)

Questi appunti costituiscono il supporto scritto alle lezioni di logica che ho tenuto fino all’Anno Accademico 2019-2020 nell’ambito dell’insegnamento di “Matematica discreta e Logica” per il Corso di Laurea triennale in Informatica presso la Scuola di Scienze Matematiche, Fisiche e Naturali all’Università di Firenze.

È una versione ormai stabile degli argomenti che ho svolto a lezione, riveduta e corretta grazie alle segnalazioni ricevute da chi ha letto le versioni precedenti ([1]), e si può considerare definitiva.

Nell’usare questi appunti, lo studente deve tenere presente che, in quanto testo scritto, essi non sono la stessa cosa delle lezioni tenute in aula: ad esempio, la trattazione è talvolta più formale (e quindi più “pesante”) rispetto a quanto consente l’immediatezza dell’esposizione orale, dove ci si può “lasciare andare” all’uso di qualche notazione non del tutto ortodossa.

Inoltre, rispetto al programma effettivamente svolto a lezione, questi appunti comprendono alcune dimostrazioni in più ([2]) ma molti esercizi in meno (e quelli compresi non sono, in generale, risolti). Le dimostrazioni in più vogliono consentire allo studente interessato qualche approfondimento e servire come riferimento per eventuali consultazioni future. Esercizi in quantità certamente sufficiente per una buona preparazione all’esame vengono invece messi a disposizione (con la loro dettagliata risoluzione) sulla pagina *e*-learning dell’insegnamento via via che i relativi argomenti sono trattati a lezione; tali esercizi, e le loro soluzioni, non costituiscono però parte di questi appunti e quindi non possono essere consultati durante le prove di esame.

Firenze, 30.10.2020

Marco Barlotti

---

[1] Ringrazio in particolare Lorenzo Iosue per la sua attenta lettura delle versioni 1.7 e 1.8 e per i commenti che mi ha inviato.

[2] Per la preparazione all’esame lo studente è invitato a fare riferimento al programmma disponibile sulla pagina *e*-learning (“Moodle”) dell’insegnamento.

BIBLIOGRAFIA

[1] D. Mundici
Logica: metodo breve
Springer – Verlag Italia, Milano (2011)

[2] R. M. Smullyan
A beginner’s guide to Mathematical Logic
Dover Publications, Inc., Mineola, New York (2014)

[3] M. Barlotti
Appunti di Algebra

[4] M. Barlotti
Appunti di Teoria dei Grafi

SOMMARIO

3. - Logica dei predicati

1.- INTRODUZIONE

1.1 - La logica matematica.

La matematica si sviluppa mediante assiomi, definizioni e teoremi: questi ultimi esprimono proprietà degli oggetti definiti (o dei concetti primitivi) che vengono “dimostrate” attraverso “ragionamenti logici”.

La logica matematica studia le “regole di ragionamento” che vengono utilizzate nelle dimostrazioni: precisa il significato di “conseguenza logica” e consente di verificare la correttezza delle deduzioni riconducendole a procedimenti automatici, un po’ come il “metodo delle coordinate” riduce i problemi geometrici a problemi algebrici.

Esempio 1.1.1

Un teorema ben noto della geometria euclidea si può enunciare come segue:

“Sia $\mathcal{T}$ un triangolo.
Se i lati di $\mathcal{T}$ sono tutti fra loro congruenti, allora anche gli angoli (interni) di $\mathcal{T}$ sono tutti fra loro congruenti”.

In questo enunciato ([1]), alcuni termini (“triangolo”, “lato”, “angolo”, “congruente”) sono specifici della geometria, ma altri (“se”, “allora”, “tutti”) appartengono all’ambito più generale del ragionamento logico. Per dimostrare il teorema si devono utilizzare, oltre agli strumenti del ragionamento logico, anche fatti puramente geometrici (teoremi precedenti, oppure assiomi) che esprimono proprietà di $\mathcal{T}$ legate al fatto che è un triangolo (del resto se $\mathcal{T}$ è un poligono con più di tre lati sappiamo bene che per $\mathcal{T}$ l’enunciato sopra riportato è falso).

[1] In questa sezione, che ha soltanto lo scopo di presentare l’argomento, usiamo il termine “enunciato” in senso intuitivo; nella prossima cercheremo di darne una spiegazione meno vaga.

Adesso consideriamo questi altri due teoremi, anch’essi ben noti:

“Sia $\mathcal{T}$ un triangolo.
Se gli angoli (interni) di $\mathcal{T}$ sono tutti fra loro congruenti, allora anche i lati di $\mathcal{T}$ sono tutti fra loro congruenti”.

“Sia $\mathcal{T}$ un triangolo.
Se non è vero che gli angoli (interni) di $\mathcal{T}$ sono tutti fra loro congruenti, allora non è vero nemmeno che i lati di $\mathcal{T}$ sono tutti fra loro congruenti”.

In tutti e tre gli enunciati si usano sostanzialmente le stesse parole (con l’aggiunta, nel terzo, dell’espressione di valore logico “non è vero che”, usata per negare il concetto che la segue), ma i tre enunciati esprimono fatti diversi (si osservi a questo proposito che le parole sono disposte in modo diverso). Cerchiamo ora di evidenziare l’aspetto logico-formale dei tre enunciati rispetto al loro contenuto geometrico; un modo grossolano ma abbastanza efficace per farlo è quello di stabilire in primo luogo che stiamo considerando un triangolo $\mathcal{T}$ e poi indicare con simboli quelle parti dell’enunciato che

(i) si ripetono uguali nei tre enunciati

e

(ii) contengono i termini strettamente geometrici.

Porremo ad esempio

$\alpha :=$ i lati di $\mathcal{T}$ sono tutti fra loro congruenti;

$\beta :=$ gli angoli (interni) di $\mathcal{T}$ sono tutti fra loro congruenti.

In questo modo i tre enunciati si scrivono rispettivamente

“Se α allora β”, “Se β allora α”, “Se non è vero che β allora non è vero nemmeno che α”.

Supponiamo di avere dimostrato, e quindi di sapere per certo, che il primo enunciato è vero (abbiamo già osservato che per tale dimostrazione è necessario entrare nella specificità geometrica dell’enunciato). Anche per dimostrare il secondo enunciato sarà necessario utilizzare, oltre al ragionamento puramente logico, fatti geometrici legati alla natura di triangolo di $\mathcal{T}$; ma per il terzo enunciato la situazione è molto diversa: non ci serve sapere che cosa esprimano le “parti di enunciato” che abbiamo indicato con α e β, ma possiamo ragionare velocemente così: se α fosse vero, per il primo enunciato anche β sarebbe vero, e invece noi sappiamo che β non è vero; dunque α non può essere vero, come si voleva dimostrare. Abbiamo dedotto il terzo enunciato dal primo per via puramente logica.

Esempio 1.1.2

Una delle forme più antiche di “deduzione formalizzata” è quella del *sillogismo*. Vediamo alcuni casi.

(i) Se piove, Andrea prende l’ombrello. Piove. Dunque Andrea prende l’ombrello.

(ii) Ogni uomo è mortale. Socrate è un uomo. Dunque, Socrate è mortale.

(iii) *Caro salsa facit bibere*. *Atqui, bibere extinguit sitim*. *Ergo, caro salsa extinguit sitim*.

Una traduzione un po’ libera del (iii): Se mangio cose salate, bevo. Se bevo, mi disseto. Dunque, se mangio cose salate mi disseto.

Vedremo che tutti e tre i sillogismi sopra esposti sono corretti (nel senso che l’affermazione dopo il “dunque” è conseguenza logica delle due affermazioni che precedono il “dunque”). Il primo e il terzo si potranno dimostrare con mezzi relativamente poveri (quello che si chiama usualmente “calcolo proposizionale”, del quale ci occuperemo come prima cosa); per il secondo serve una struttura più complessa (la “logica dei predicati” o “logica del primo ordine”), che studieremo in seguito.

1.2 - Enunciati. Il principio aristotelico del “terzo escluso”.

Oggetto del nostro studio saranno gli *enunciati*. Fino a quando saremo in grado di darne una definizione formale rigorosa (ma ciò avverrà soltanto nella sez. 3.4) chiameremo *enunciato* qualunque frase che, opportunamente interpretata ([2]) in un insieme, assuma un “*valore di verità*” (“*vero*” oppure “*falso*”) che rispetta le condizioni fissate circa $2\,400$ anni fa dal filosofo greco Aristotele, dette “*principio del terzo escluso*”:

Un enunciato è certamente vero oppure falso, e non può essere sia vero che falso.

Esempio 1.2.1

“*Zero è il numero più piccolo*” è un enunciato vero, se interpretato nell’insieme $\mathbb{N}$ dei numeri naturali; è un enunciato falso se interpretato nell’insieme $\mathbb{Z}$ dei numeri interi relativi.

[2] Ma che cosa vuol dire “interpretare” una frase? Anche di questo si occupa la logica…

Esempio 1.2.2

“*Questo enunciato è falso*” non può essere interpretato come enunciato: infatti se fosse vero risulterebbe anche falso, quindi non può essere vero; ma se fosse falso risulterebbe anche vero, quindi non può essere nemmeno falso. Poiché non è dato un terzo possibile “valore di verità” oltre a “*vero*” e “*falso*” (i filosofi scolastici dicevano in latino: *tertium non datur*), questa frase non rispetta le condizioni fissate da Aristotele e quindi non è un enunciato.

Esempio 1.2.3

Abbiamo tutti (o crediamo di avere?) un’idea abbastanza precisa di che cos’è “un mucchio di sabbia”. Certamente, dieci granelli di sabbia non costituiscono “un mucchio di sabbia”. Ma proviamo ad effettuare il seguente esperimento: prendiamo un mucchio di sabbia (cioè tanta sabbia quanta basta per poterla chiamare “un mucchio di sabbia”) e togliamo un granello per volta dal mucchio. Prima o poi, arriveremo a dieci granelli, cioè a qualcosa che non chiameremmo più “un mucchio di sabbia”. Di fatto, nemmeno undici granelli, nemmeno dodici… costituiscono “un mucchio di sabbia”. Dunque, ***quando esattamente*** di fronte a una certa quantità di sabbia possiamo dire che l’affermazione “questo è un mucchio di sabbia” è vera, e quando invece possiamo dire che è falsa?

Dati alcuni enunciati, possiamo costruirne altri più complessi utilizzando i cosiddetti *connettivi logici* (ad esempio: “*non*”, “*o*”, “*e*”, “*se* … *allora* … ”) e assegnare a questi nuovi enunciati valori di verità che dipendono opportunamente sia dal valore di verità degli enunciati dati in partenza sia dai connettivi usati. La logica matematica studia gli enunciati più complessi così costruiti, definendo i concetti di *equivalenza logica*, *conseguenza logica*, *tautologia* e *contraddizione* e fornendo gli strumenti per riconoscere, anche con procedimenti sostanzialmente automatici, quando si realizzano.

2.- LOGICA PROPOSIZIONALE

2.1 - Alfabeto e formule.

In questo capitolo supporremo fissato un *alfabeto*, cioè un insieme A i cui elementi sono:

(i) infiniti simboli $p_0, p_1, p_2, \ldots, p_n, \ldots$
che “etichettiamo” con i numeri naturali ([3]) e chiamiamo *variabili proposizionali*;

(ii) quattro simboli $\neg,\ \wedge,\ \vee,\ \rightarrow$ detti rispettivamente “*non*”, “*e*”, “*o*”, “*implica*”
e detti complessivamente “*connettivi logici*” ;

(iii) due simboli (e) detti rispettivamente “*parentesi aperta*” e “*parentesi chiusa*”
e detti complessivamente “*simboli extralogici*” ;

Si dice *parola* sull’alfabeto A ogni sequenza finita di elementi di A: gli elementi di una parola vengono usualmente scritti di seguito senza altri segni grafici.

Esempio 2.1.1

Sono esempi di parole sull’alfabeto A sopra descritto (separati fra loro dal simbolo “;”):

$$)p_1 p_2 \wedge \neg \vee p_1 \wedge\ ; \qquad p_1 p_1 p_5 \vee \rightarrow ())\neg\ ; \qquad (\neg(p_1 \wedge p_2)) \vee (p_1 \rightarrow p_2)\ ; \qquad \wedge \neg \vee)(p_0\ .$$

A noi in generale non interesseranno tutte le parole sull’alfabeto A ma soltanto quelle “costruite” secondo opportune regole.

---

[3] Rigorosamente, tali simboli si dicono *in corrispondenza biunivoca* con l’insieme $\mathbb{N}$ dei numeri naturali. Questo fatto si esprime anche dicendo che le variabili proposizionali sono *un’infinità numerabile*.

Una parola w sull’alfabeto A sopra descritto si dice *formula* (o anche *formula ben formata*, abbreviato in *fbf*) se esiste una sequenza finita $w_0, w_1, w_2, \ldots, w_k$ di parole su A tale che

FBF-PROP.1 $\quad w = w_k$

e

FBF-PROP.2 $\quad$ per ogni $h \leq k$ la parola w_h è

– una variabile proposizionale

oppure

– della forma $(\neg w_i)$ (e si dice “negazione della formula w_i”)

oppure

– della forma $(w_i \wedge w_j)$ (e si dice “congiunzione delle formule w_i e w_j”)

oppure

– della forma $(w_i \vee w_j)$ (e si dice “disgiunzione delle formule w_i e w_j”)

oppure

– della forma $(w_i \rightarrow w_j)$ (e si dice “implicazione tra le formule w_i e w_j”)

con $i, j < h$.

La sequenza finita $w_0, w_1, w_2, \ldots, w_k$ si dice in questo caso una *costruzione* della formula w.

Il minimo numero naturale k per cui esiste una costruzione $w_0, w_1, w_2, \ldots, w_k$ della formula w si dice *lunghezza* di w.

L’insieme delle formule costituisce il *linguaggio della logica proposizionale*.

Osservazione 2.1.2

Sia w una formula e sia $w_0, w_1, w_2, \ldots, w_k$ una costruzione di w .

Ogni w_i per $i := 0, \ldots, k$ è una formula.

Esempio 2.1.3

Sono esempi di formule sull’alfabeto A sopra descritto (ciascuna su un rigo diverso):

$((p_1 \wedge (\neg p_2)) \vee p_1)$

$((((p_1 \rightarrow p_1) \rightarrow p_5) \vee (\neg p_3)) \wedge p_0)$

$((\neg(p_1 \wedge p_2)) \vee (p_1 \rightarrow p_2))$.

Osservazione 2.1.4

Per definizione, ogni variabile proposizionale è una formula (di lunghezza 1); d'altro lato, se una formula ha lunghezza 1 essa non può essere altro che una variabile proposizionale.

Le formule di lunghezza 1 (cioè le variabili proposizionali) si dicono *formule atomiche*.

Osservazione 2.1.5 ("Principio di unica lettura delle formule")

Sia w una formula che non è una variabile proposizionale. Per la condizione FBF-PROP.2 c'è un unico modo di leggere w come negazione di altra formula oppure (ma soltanto se ciò non è possibile) come congiunzione di altre due formule oppure (ma soltanto se le due precedenti letture non sono possibili) come disgiunzione di altre due formule oppure (ma soltanto se nessuna delle precedenti letture è possibile) come implicazione fra due formule.

Osservazione 2.1.6

Si intuisce dalla definizione, e si può già verificare nell'esempio 2.1.3, che in una formula generalmente abbondano le parentesi. È dunque opportuno stabilire alcune convenzioni che consentano di limitare l'uso delle parentesi nella scrittura di una formula.

Per adesso, stabiliremo che:

(i) il connettivo "$\neg$" si riferisce esclusivamente alla formula immediatamente successiva; ciò consente di omettere la parentesi aperta che precede "$\neg$" e la parentesi chiusa che segue la formula immediatamente successiva;

(ii) la parentesi aperta e la parentesi chiusa più esterne possono essere omesse.

Pertanto, ad esempio, scriveremo $\neg p_1 \wedge \neg p_2$ anziché $((\neg p_1) \wedge (\neg p_2))$ e scriveremo $p_1 \to \neg p_2$ anziché $(p_1 \to (\neg p_2))$.

2.2 - Valutazioni di verità.

Si dice *valutazione di verità* ogni funzione v dall'insieme delle variabili proposizionali nell'insieme $\{0, 1\}$.

Teorema 2.2.1

Per ogni valutazione di verità v esiste una e una sola funzione $\overline{v}$ dall'insieme delle formule nell'insieme $\{0,1\}$ tale che

(i) $\overline{v}$ estende v, ossia: $\overline{v}(p) = v(p)$ per ogni variabile proposizionale p;

(ii) comunque prese le formule φ_1, φ_2 si ha

$\overline{v}(\neg\varphi_1) = 1 - \overline{v}(\varphi_1)$;

$\overline{v}(\varphi_1 \wedge \varphi_2) = \mathit{min}\{\overline{v}(\varphi_1), \overline{v}(\varphi_2)\}$, ossia:

$\overline{v}(\varphi_1 \wedge \varphi_2) = 1$ se $\overline{v}(\varphi_1) = 1$ e $\overline{v}(\varphi_2) = 1$, $\overline{v}(\varphi_1 \wedge \varphi_2) = 0$ altrimenti;

$\overline{v}(\varphi_1 \vee \varphi_2) = \mathit{max}\{\overline{v}(\varphi_1), \overline{v}(\varphi_2)\}$, ossia:

$\overline{v}(\varphi_1 \vee \varphi_2) = 0$ se $\overline{v}(\varphi_1) = 0$ e $\overline{v}(\varphi_2) = 0$, $\overline{v}(\varphi_1 \vee \varphi_2) = 1$ altrimenti;

$\overline{v}(\varphi_1 \to \varphi_2) = 0$ se $\overline{v}(\varphi_1) = 1$ e $\overline{v}(\varphi_2) = 0$, $\overline{v}(\varphi_1 \to \varphi_2) = 1$ altrimenti.

Tale funzione $\overline{v}$ si indica ancora con v e si dice "*valutazione di verità sull'insieme delle formule associata a* v" oppure anche "interpretazione associata a v".

Dimostrazione — Per come si è definita una formula, è sufficiente applicare le relazioni espresse dalla condizione (ii) ad ogni termine della costruzione. La dimostrazione si formalizza per induzione (cfr. [3], cap. 2) sulla lunghezza delle formule coinvolte.

Osservazione 2.2.2

I connettivi logici $\neg$ ("non") e $\wedge$ ("e") corrispondono abbastanza bene all'avverbio "non" ([4]) e alla congiunzione "e" della lingua italiana. L'uso nella lingua corrente della particella "o" risulta però di fatto ambiguo.

Quando alla biglietteria di una mostra leggiamo "Hanno diritto al biglietto con tariffa ridotta i visitatori che sono in possesso di un biglietto dell'autobus o hanno più di 65 anni di età" sappiamo che le due condizioni ("essere in possesso di un biglietto dell'autobus" e "avere più di 65 anni di età") non si escludono a vicenda: il visitatore, per avere diritto al biglietto con tariffa ridotta, deve soddisfare una o l'altra ma può anche soddisfare entrambe le condizioni. Si dice in questo caso che la particella "o" esprime una *disgiunzione inclusiva*.

[4] Ma si faccia attenzione al fatto che talvolta nella lingua italiana la doppia negazione ha un valore rafforzativo ("***Non*** vedo ***niente***", "***Non*** odio ***nessuno***"): vedremo che in logica non è mai così.

Quando invece nelle istruzioni del modello “Redditi PF” per la dichiarazione dei redditi leggiamo “Il contribuente che risulta in credito nelle imposte sui redditi può chiedere il rimborso della somma per cui è in credito o utilizzarla a compensazione di altre imposte dovute” sappiamo che le due possibilità sono mutuamente esclusive: se si chiede il rimborso della somma per cui si è in credito non la si può utilizzare a compensazione di altre imposte dovute, e viceversa. Si dice che in questo caso la particella “o” esprime una *disgiunzione esclusiva*.

Il fatto che nel teorema 2.2.1 si sia posto $\overline{v}(\varphi_1 \vee \varphi_2) = 0$ se e soltanto se $\overline{v}(\varphi_1) = 0$ e $\overline{v}(\varphi_2) = 0$ significa che il connettivo logico “ $\vee$ ” corrisponde alla disgiunzione “o” usata in senso inclusivo (cioè quando in latino si tradurrebbe con “*vel*” anziché con “*aut*”).

Esercizio 2.2.3

Per ognuna delle seguenti frasi della lingua italiana si indichi se la particella “o” indica una “disgiunzione esclusiva” o una “disgiunzione inclusiva”:

– prenderemo il volo per gli Stati Uniti sabato prossimo o domenica prossima;

– per entrare in Svizzera bisogna presentare il passaporto o la carta di identità;

– per essere promossi bisogna studiare o essere molto fortunati;

– in italiano la particella “o” indica una “disgiunzione esclusiva” o una “disgiunzione inclusiva”.

Siano φ una formula e v una valutazione di verità sull’insieme delle formule. Se $v(\varphi) = 1$, si dice che “φ *è vera sotto* v”, oppure che “φ *è vera nell’interpretazione* v”, oppure anche che “v *soddisfa* φ”; scriveremo talvolta

$$v \models \varphi \,.$$

Sia φ una formula. Se ogni valutazione di verità soddisfa φ, si dice che φ è una *tautologia*; se esiste almeno una valutazione di verità che soddisfa φ, si dice che φ è *soddisfacibile*; se nessuna valutazione di verità soddisfa φ (cioè se φ non è soddisfacibile), si dice che φ è *insoddisfacibile* (oppure che φ è una *contraddizione*).

Esempio 2.2.4

Per ogni formula φ, la formula $\varphi \vee \neg\varphi$ è una tautologia, mentre la formula $\varphi \wedge \neg\varphi$ è insoddisfacibile; per ogni variabile proposizionale p, la formula $p \to \neg p$ è soddisfacibile (ma non è una tautologia).

Sia infatti v una qualsiasi valutazione di verità; allora certamente $v(\varphi) = 1$ oppure $v(\varphi) = 0$. In entrambi i casi, $v(\varphi \vee \neg\varphi) = 1$ e $v(\varphi \wedge \neg\varphi) = 0$.

Siano poi p una variabile proposizionale, v una valutazione di verità per la quale $v(p) = 0$ e w una valutazione di verità per la quale $w(p) = 1$. Si ha $v(p \to \neg p) = 1$ (cosicché $p \to \neg p$ è soddisfacibile) mentre $w(p \to \neg p) = 0$ (cosicché $p \to \neg p$ non è una tautologia).

Teorema 2.2.5

La formula φ è una tautologia se e soltanto se la formula $\neg\varphi$ è insoddisfacibile.

Dimostrazione — Supponiamo in primo luogo che φ sia una tautologia. Per ogni valutazione v, deve allora essere $v(\varphi) = 1$ e quindi $v(\neg\varphi) = 0$, dunque $\neg\varphi$ è insoddisfacibile.

Viceversa, supponiamo che $\neg\varphi$ sia insoddisfacibile. Per ogni valutazione v, deve allora essere $v(\neg\varphi) = 0$ e quindi $v(\varphi) = 1$, dunque φ è una tautologia.

Teorema 2.2.6

Siano α, β formule. Sono fatti equivalenti:

(i) La formula $\alpha \wedge \beta$ è una tautologia;

(ii) Le formule α e β sono entrambe tautologie.

Dimostrazione — Supponiamo in primo luogo che $\alpha \wedge \beta$ sia una tautologia. Per ogni valutazione v, deve allora essere $v(\alpha \wedge \beta) = 1$ e quindi $v(\alpha) = v(\beta) = 1$, dunque α e β sono entrambe tautologie.

Viceversa, supponiamo che le formule α e β siano entrambe tautologie. Per ogni valutazione v deve allora essere $v(\alpha) = v(\beta) = 1$ e quindi $v(\alpha \wedge \beta) = 1$, dunque $\alpha \wedge \beta$ è una tautologia.

Siano φ, ψ formule. Si dice che ψ *è conseguenza logica di* φ e si scrive

$$\varphi \models \psi$$

se ogni valutazione che soddisfa φ soddisfa anche ψ.

$\boxed{\text{Esercizio 2.2.7}}$

Siano α, β formule. Si dimostri che

- $\alpha \vee \beta$ è conseguenza logica di α;
- α è conseguenza logica di $\alpha \wedge \beta$;
- α è conseguenza logica di $(\alpha \vee \beta) \wedge (\neg\beta)$;

Siano φ, ψ formule. Si dice che φ e ψ sono *logicamente equivalenti* e si scrive

$$\varphi \equiv \psi$$

se ciascuna di esse è conseguenza logica dell’altra, cioè se ogni valutazione che soddisfa φ soddisfa anche ψ e viceversa ogni valutazione che soddisfa ψ soddisfa anche φ.

$\boxed{\text{Teorema 2.2.8}}$

Le formule φ e ψ sono logicamente equivalenti se e soltanto se per ogni valutazione di verità v si ha $v(\varphi) = v(\psi)$.

Dimostrazione – Supponiamo in primo luogo che φ e ψ siano logicamente equivalenti, cioè che ciascuna di esse sia conseguenza logica dell’altra. Sia v una valutazione di verità: se $v(\varphi) = 1$, deve essere $v(\psi) = 1$ (perché ψ è conseguenza logica di φ); se $v(\varphi) = 0$, non può essere $v(\psi) = 1$ (perché in tal caso dovrebbe essere $v(\varphi) = 1$, essendo φ conseguenza logica di ψ), dunque $v(\psi) = 0$. Per l’arbitrarietà della valutazione di verità v, si è così provato che ogni valutazione di verità assume lo stesso valore su φ e ψ.

Supponiamo adesso che sia $v(\varphi) = v(\psi)$ per ogni valutazione di verità v. Allora ogni valutazione di verità che soddisfa φ soddisfa anche ψ (cioè ψ è conseguenza logica di φ) e viceversa ogni valutazione di verità che soddisfa ψ soddisfa anche φ (cioè φ è conseguenza logica di ψ): dunque φ e ψ sono logicamente equivalenti.

Teorema 2.2.9

Siano φ, ψ formule. Sono fatti equivalenti:

(i) $\varphi \vDash \psi$;

(ii) $\varphi \to \psi$ è una tautologia.

Dimostrazione — Proviamo in primo luogo che dalla (i) segue la (ii).

Supponiamo che valga la (i), e sia v una qualsiasi valutazione di verità: dobbiamo provare che $v(\varphi \to \psi) = 1$. Se $v(\varphi) = 0$, ciò è ovvio; se $v(\varphi) = 1$, per la (i) deve essere anche $v(\psi) = 1$ e dunque $v(\varphi \to \psi) = 1$, come si voleva dimostrare.

Ora proviamo che dalla (ii) segue la (i).

Supponiamo che valga la (ii), e sia v una valutazione di verità tale che $v(\varphi) = 1$; poiché deve essere $v(\varphi \to \psi) = 1$, non può essere $v(\psi) = 0$, e dunque $v(\psi) = 1$. Per l'arbitrarietà della valutazione v, si è così provato che ψ è conseguenza logica di φ, come si voleva.

Teorema 2.2.10

Siano φ, ψ formule. Sono fatti equivalenti:

(i) φ e ψ sono logicamente equivalenti;

(ii) $(\varphi \to \psi) \wedge (\psi \to \varphi)$ è una tautologia.

Dimostrazione — Segue immediatamente dai teoremi 2.2.9 e 2.2.6.

Sia $\Sigma := \{\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n\}$ un insieme di formule.

Si dice che una valutazione di verità v *soddisfa* Σ se v soddisfa ciascuna delle formule $\varphi_1, \varphi_2, \varphi_3, \dots$ e φ_n.

Se esiste almeno una valutazione che soddisfa Σ, si dice che Σ è *soddisfacibile*; se nessuna valutazione soddisfa Σ, si dice che Σ è *insoddisfacibile*.

Siano $\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n, \varphi$ formule. Si dice che “φ *è conseguenza logica di* $\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n$” e si scrive

$$\{\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n\} \vDash \varphi$$

se ogni valutazione di verità che soddisfa $\{\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n\}$ soddisfa anche φ .

Teorema 2.2.11

Siano α, β, φ tre formule. Sono fatti equivalenti:

(i) $\{\alpha, \beta\} \vDash \varphi$;

(ii) $\alpha \wedge \beta \vDash \varphi$.

Dimostrazione — Proviamo in primo luogo che dalla (i) segue la (ii). Sia v una valutazione di verità che soddisfa $\alpha \wedge \beta$; allora v soddisfa α e soddisfa β, quindi per la (i) soddisfa φ . Per l'arbitrarietà di v, abbiamo provato che φ è conseguenza logica di $\alpha \wedge \beta$.

Ora proviamo che dalla (ii) segue la (i). Sia v una valutazione di verità che soddisfa α e soddisfa β ; allora v soddisfa $\alpha \wedge \beta$, quindi per la (ii) soddisfa φ . Per l'arbitrarietà di v, abbiamo provato che φ è conseguenza logica di α e β .

2.3 - Le tabelle dei valori di verità.

Se le formule φ e ψ sono costruite mediante i connettivi logici a partire da certe altre formule φ_1, φ_2, $\ldots$, φ_k, per ogni valutazione di verità v i valori $v(\varphi)$ e $v(\psi)$ dipendono soltanto da $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$.

Questa banale osservazione si applica di fatto a tutte le formule, perché ciascuna di esse è costruita mediante i connettivi logici a partire da un numero finito di formule atomiche (le variabili proposizionali).

Le informazioni essenziali sulle formule, cioè la verifica delle possibili proprietà che abbiamo definito nella sez. 2.2, si possono quindi ottenere tabulando i valori di $v(\varphi)$ e $v(\psi)$ in funzione dei 2^k possibili valori di $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$. Dunque:

– la formula φ è una tautologia se e soltanto se per ogni possibile valore di $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$ si ha $v(\varphi) = 1$;

– la formula φ è soddisfacibile se e soltanto se esiste una scelta di valori per $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$ per la quale si abbia $v(\varphi) = 1$;

– la formula φ è insoddisfacibile se e soltanto se per ogni possibile valore di $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$ si ha $v(\varphi) = 0$;

– la formula ψ è conseguenza logica della formula φ se e soltanto se per ogni valore di $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$ tale che $v(\varphi) = 1$ si ha anche $v(\psi) = 1$;

– le formula φ e ψ sono logicamente equivalenti se e soltanto se per ogni possibile valore di $v(\varphi_1)$, $v(\varphi_2)$, $\ldots$, $v(\varphi_k)$ si ha $v(\varphi) = v(\psi)$ (cfr. teor. 2.2.8) ;

Le tabelle che esprimono i valori di verità per una o più formule in funzione dei valori di verità delle formule (eventualmente delle variabili proposizionali) a partire dalle quali esse sono costruite si dicono *tabelle dei valori di verità* (o talvolta, più pomposamente, *tavole di verità*) per quelle formule.

$\boxed{\text{Teorema 2.3.1}}$

Siano α, β, γ formule.

(i) Le formule α e $\neg(\neg\alpha)$ sono logicamente equivalenti;

(ii) Le formule α, $\alpha \wedge \alpha$ e $\alpha \vee \alpha$ sono logicamente equivalenti;

(iii) Le formule $\alpha \wedge \beta$ e $\beta \wedge \alpha$ sono logicamente equivalenti;

(iv) le formule $\alpha \vee \beta$ e $\beta \vee \alpha$ sono logicamente equivalenti;

(v) le formule $(\alpha \wedge \beta) \wedge \gamma$ e $\alpha \wedge (\beta \wedge \gamma)$ sono logicamente equivalenti;

(vi) le formule $(\alpha \vee \beta) \vee \gamma$ e $\alpha \vee (\beta \vee \gamma)$ sono logicamente equivalenti.

Dimostrazione — Basta tabulare i valori di verità delle formule considerate in funzione di tutti i possibili valori di verità per α, β e γ:

α	$\neg\alpha$	$\neg(\neg\alpha)$	$\alpha \wedge \alpha$	$\alpha \vee \alpha$
0	1	0	0	0
1	0	1	1	1

α	β	$\alpha \vee \beta$	$\beta \vee \alpha$	$\alpha \wedge \beta$	$\beta \wedge \alpha$
0	0	0	0	0	0
0	1	1	1	0	0
1	0	1	1	0	0
1	1	1	1	1	1

α	β	γ	$\alpha \wedge \beta$	$\beta \wedge \gamma$	$(\alpha \wedge \beta) \wedge \gamma$	$\alpha \wedge (\beta \wedge \gamma)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

α	β	γ	$\alpha \vee \beta$	$\beta \vee \gamma$	$(\alpha \vee \beta) \vee \gamma$	$\alpha \vee (\beta \vee \gamma)$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Osservazione 2.3.2

Siano α, β formule. Le equivalenze logiche espresse dai punti (v) e (vi) del terorema 2.3.1 suggeriscono altre situazioni (oltre a quelle già esplicitate con l'osservazione 2.1.6) nelle quali certe parentesi possono essere omesse. Senza ambiguità circa il valore di verità della formula risultante, infatti, potremo scrivere $\alpha \wedge \beta \wedge \gamma$ in luogo di $(\alpha \wedge \beta) \wedge \gamma$ e di $\alpha \wedge (\beta \wedge \gamma)$, e potremo scrivere $\alpha \vee \beta \vee \gamma$ in luogo di $(\alpha \vee \beta) \vee \gamma$ e di $\alpha \vee (\beta \vee \gamma)$.

Teorema 2.3.3

Siano $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ formule.

(i) La scrittura $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \ldots \wedge \alpha_n$ non presenta ambiguità, perché comunque si inseriscano le parentesi (e) al suo interno il valore di verità della formula risultante in qualsiasi interpretazione v è 1 se e soltanto se

$$v(\alpha_1) = v(\alpha_2) = v(\alpha_3) = \ldots = v(\alpha_n) = 1\,;$$

(ii) La scrittura $\alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \ldots \vee \alpha_n$ non presenta ambiguità, perché comunque si inseriscano le parentesi (e) al suo interno il valore di verità della formula risultante in qualsiasi interpretazione v è 0 se e soltanto se

$$v(\alpha_1) = v(\alpha_2) = v(\alpha_3) = \ldots = v(\alpha_n) = 0\,.$$

Dimostrazione — Questo teorema generalizza le (v) e (vi) del teor. 2.3.1. Pur essendo un risultato abbastanza intuitivo, per formalizzarne la dimostrazione bisogna procedere per induzione su n (cfr. [3], cap. 2).

Teorema 2.3.4

Siano $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n, \varphi$ formule. Sono fatti equivalenti:

(i) $\{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\} \models \varphi$;

(ii) $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \ldots \wedge \alpha_n \models \varphi$.

Dimostrazione — Si tratta di una generalizzazione del teorema 2.2.11 che si dimostra facilmente utilizzando il teorema 2.3.3. I dettagli della dimostrazione si lasciano al lettore quale utile esercizio.

Osservazione 2.3.5

Siano $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n, \varphi$ formule. Per i teoremi 2.3.4 e 2.2.9, il problema di decidere se φ sia conseguenza logica di $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ si riconduce al problema di decidere se la formula $(\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \ldots \wedge \alpha_n) \rightarrow \varphi$ sia una tautologia, cioè (per il teorema 2.2.5) al problema di decidere se la negazione di tale formula sia soddisfacibile.

2.4 - Digressione sul connettivo “ $\rightarrow$ ”.

Siano α, β formule. Alla formula $\alpha \rightarrow \beta$ corrispondono molte espressioni del linguaggio corrente, formalmente diverse ma sostanzialmente equivalenti. Essa infatti si può leggere

– “se α allora β”

oppure

– “α implica β”

oppure

– “β se α”

oppure

– “α solo se β”.

Le formule costruite mediante il connettivo “ $\rightarrow$ ” rivestono particolare importanza in matematica, perché compaiono nell’enunciato di molti teoremi.

Siano α, β formule. Se la formula $\alpha \rightarrow \beta$ è una tautologia, si dice che “α è condizione sufficiente per β” oppure, equivalentemente, che “β è condizione necessaria per α”.

Teorema 2.4.1

Siano α, β formule. Le formule $\alpha \to \beta$ e $\neg\alpha \vee \beta$ sono logicamente equivalenti.

Dimostrazione — Basta tabulare i valori di verità di $\alpha \to \beta$ e di $\neg\alpha \vee \beta$ in funzione di tutti i possibili valori di verità per α e β:

α	β	$\alpha \to \beta$	$\neg\alpha$	$\neg\alpha \vee \beta$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	0
1	1	1	0	1

Siano α, β formule.

Si dice *contronominale* dell'implicazione $\alpha \to \beta$ l'implicazione $\neg\beta \to \neg\alpha$.

Teorema 2.4.2

Siano α, β formule. L'implicazione $\alpha \to \beta$ e la sua contronominale $\neg\beta \to \neg\alpha$ sono logicamente equivalenti.

Dimostrazione — Si potrebbero tabulare i valori di verità di $\alpha \to \beta$ e di $\neg\beta \to \neg\alpha$ in funzione di tutti i possibili valori di verità per α e β (si lascia questa strada al lettore come utile esercizio). Ma si può anche procedere a una verifica diretta utilizzando il teorema 2.4.1 e le (i) e (iv) del teorema 2.3.1:

$$\neg\beta \to \neg\alpha \equiv \neg(\neg\beta) \vee \neg\alpha \equiv \beta \vee \neg\alpha \equiv \neg\alpha \vee \beta \equiv \alpha \to \beta\,.$$

Esercizio 2.4.3

Si stabilisca se in generale per tre formule α, β e γ si ha che

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

oppure no.

Esempio 2.4.4

Verifichiamo, con l’aiuto delle tabelle dei valori di verità, che, qualunque siano le formule α e β, la formula β è conseguenza logica di $\alpha \wedge (\alpha \to \beta)$.

Dimostrazione — Basta tabulare i valori di verità di $\alpha \wedge (\alpha \to \beta)$ in funzione di tutti i possibili valori di verità per α e β e osservare che in tutte le righe per le quali nella colonna di $\alpha \wedge (\alpha \to \beta)$ compare il valore 1 anche nella colonna di β compare il valore 1:

α	β	$\alpha \to \beta$	$\alpha \wedge (\alpha \to \beta)$
0	0	1	0
0	1	1	0
1	0	0	0
1	1	1	1

Si noti che le due formule $\alpha \wedge (\alpha \to \beta)$ e β non sono però logicamente equivalenti: infatti nella seconda riga della tabella compaiono in corrispondenza delle loro colonne due valori diversi.

Osservazione 2.4.5

Si riconsideri il sillogismo (i) dell’esempio 1.1.1 .

Posto $\alpha :=$ “Piove” e $\beta :=$ “Andrea prende l’ombrello”, tale sillogismo esprime il fatto che β è conseguenza logica di α e $\alpha \to \beta$, ossia (per il teorema 2.2.11) di $\alpha \wedge (\alpha \to \beta)$. Dunque l’esempio 2.4.4 fornisce la formalizzazione del sillogismo (i) dell’esempio 1.1.1 .

Esercizio 2.4.6

Si riconsideri il sillogismo (iii) dell’esempio 1.1.1 . Lo si formalizzi mediante la scelta di opportune variabili logiche e se ne dimostri la validità.

Esercizi

Per ciascuna delle seguenti formule si stabilisca se è soddisfacibile, precisando in tal caso se è una tautologia oppure no:

2.4.7 $(p \to q) \vee (q \to p)$;

2.4.8 $(\neg p \to q) \wedge \neg(p \vee q)$;

2.4.9 $p \to (q \to p)$;

2.4.10 $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$;

2.4.11 $((p \vee q) \wedge (\neg p \vee r)) \to (p \vee r)$;

2.4.12 $((p \to q) \wedge \neg q) \wedge ((q \to p) \wedge \neg p)$;

2.4.13 $((p \to q) \to r) \wedge \neg(p \to (q \to r))$;

2.5 - Quanti connettivi logici?

Teorema 2.5.1 (“leggi di De Morgan”)

Siano α, β formule. Allora

(i) $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$;

(ii) $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$.

Dimostrazione — Basta tabulare i valori di verità delle formule considerate in funzione di tutti i possibili valori di verità per α e β:

α	β	$\alpha \vee \beta$	$\neg(\alpha \vee \beta)$	$\neg\alpha$	$\neg\beta$	$\neg\alpha \wedge \neg\beta$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

α	β	$\alpha \wedge \beta$	$\neg(\alpha \wedge \beta)$	$\neg\alpha$	$\neg\beta$	$\neg\alpha \vee \neg\beta$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Corollario 2.5.2

Siano α, β formule. Le formule $\neg(\alpha \to \beta)$ e $\alpha \wedge \neg\beta$ sono logicamente equivalenti.

Dimostrazione — Infatti

$$\neg(\alpha \to \beta) \overset{\text{(teor. 2.4.1)}}{\equiv} \neg(\neg\alpha \vee \beta) \overset{\text{(teor. 2.5.1 }(i))}{\equiv} \neg(\neg\alpha) \wedge \neg\beta \overset{\text{(teor. 2.3.1 }(i))}{\equiv} \alpha \wedge \neg\beta\,.$$

Osservazione 2.5.3

Comunque scelte le formule α e β, si ha che

$$\alpha \vee \beta \equiv \neg(\neg\alpha \wedge \neg\beta) \qquad \text{e} \qquad \alpha \wedge \beta \equiv \neg(\neg\alpha \vee \neg\beta)\,.$$

In effetti, per le leggi di De Morgan,

$$\neg(\neg\alpha \wedge \neg\beta) \equiv \neg(\neg\alpha) \vee \neg(\neg\beta) \equiv \alpha \vee \beta \quad \text{e} \quad \neg(\neg\alpha \vee \neg\beta) \equiv \neg(\neg\alpha) \wedge \neg(\neg\beta) \equiv \alpha \wedge \beta$$

Dunque ciascuno dei due connettivi $\vee$ e $\wedge$ può essere espresso mediante un’opportuna combinazione dell’altro e della negazione.

Osservazione 2.5.4

Per i teoremi 2.4.1 e 2.5.1, ogni formula è logicamente equivalente a una formula costruita utilizzando soltanto i connettivi logici “$\neg$” e “$\wedge$”. Se definiamo un nuovo connettivo logico $\downarrow$ (detto “*negazione congiunta*”) estendendo la generica valutazione di verità v con la regola

$$\overline{v}(\varphi \downarrow \psi) = 1 \text{ se } \overline{v}(\varphi) = \overline{v}(\psi) = 0, \quad \overline{v}(\varphi \downarrow \psi) = 0 \text{ altrimenti}$$

si verifica facilmente (mediante le tabelle dei valori di verità) che per ogni coppia di formule α, β si ha

$\alpha \downarrow \alpha \equiv \neg\alpha\,;$

$(\alpha \downarrow \alpha) \downarrow (\beta \downarrow \beta) \equiv \alpha \wedge \beta\,;$

$(\alpha \downarrow \beta) \downarrow (\alpha \downarrow \beta) \equiv \alpha \vee \beta\,;$

$(\alpha \downarrow (\beta \downarrow \beta)) \downarrow (\alpha \downarrow (\beta \downarrow \beta)) \equiv \alpha \to \beta\,.$

Dunque ogni formula è logicamente equivalente a una formula costruita utilizzando soltanto il connettivo logico $\downarrow$.

Esercizio 2.5.5

Sia $\downarrow$ il connettivo logico introdotto nell’osservazione 2.5.4. Si stabilisca se in generale per due formule α e β si ha che

$$\alpha \downarrow \beta \equiv \beta \downarrow \alpha$$

oppure no.

Esercizio 2.5.6

Sia $\downarrow$ il connettivo logico introdotto nell’osservazione 2.5.4. Si stabilisca se in generale per tre formule α, β e γ si ha che

$$(\alpha \downarrow \beta) \downarrow \gamma \equiv \alpha \downarrow (\beta \downarrow \gamma)$$

oppure no.

Esercizi

Definiamo cinque nuovi connettivi logici $\leftarrow$ (“se”), $\sqcup$ (“disgiunzione esclusiva”), ↲ (“la prima formula”), ↳ (“la seconda formula”) e $\triangle$ (“sempre vero”) (attenzione! questi simboli e questi nomi ***non*** sono standard!) estendendo la generica valutazione di verità v con le regole

$$\overline{v}\,(\varphi \leftarrow \psi) = 0 \text{ se } \overline{v}\,(\varphi) = 0 \text{ e } \overline{v}\,(\psi) = 1, \quad \overline{v}\,(\varphi \leftarrow \psi) = 1 \text{ altrimenti;}$$

$$\overline{v}\,(\varphi \sqcup \psi) = 1 \text{ se } \overline{v}\,(\varphi) \neq \overline{v}\,(\psi), \quad \overline{v}\,(\varphi \sqcup \psi) = 0 \text{ altrimenti;}$$

$$\overline{v}\,(\varphi \mathbin{\text{↲}} \psi) = \overline{v}\,(\varphi), \quad \text{qualunque sia il valore di } \overline{v}\,(\psi);$$

$$\overline{v}\,(\varphi \mathbin{\text{↳}} \psi) = \overline{v}\,(\psi), \quad \text{qualunque sia il valore di } \overline{v}\,(\varphi);$$

$$\overline{v}\,(\varphi \mathbin{\triangle} \psi) = 1, \quad \text{qualunque sia il valore di } \overline{v}\,(\varphi) \text{ e } \overline{v}\,(\psi).$$

2.5.7 Si esprima ciascuno dei cinque connettivi logici sopra considerati mediante i quattro connettivi “standard” $\neg$, $\wedge$, $\vee$, $\rightarrow$.

2.5.8 Si esprima ciascuno dei cinque connettivi logici sopra considerati mediante il connettivo $\downarrow$ considerato nell’osservazione 2.5.4.

$\boxed{2.5.9}$ Si dimostri che ogni possibile connettivo binario è uno dei tre connettivi binari “standard” $\wedge$, $\vee$, $\rightarrow$ oppure uno dei cinque connettivi $\leftarrow$, $\sqcup$, $\dashv$, $\hookrightarrow$, $\triangle$ oppure la negazione di uno di questi otto connettivi binari.

$\boxed{2.5.10}$ Si esprima il connettivo logico $\wedge$ mediante i connettivi logici $\rightarrow$ e $\neg$.

$\boxed{2.5.11}$ Si esprima il connettivo logico $\rightarrow$ mediante i connettivi logici $\wedge$ e $\neg$.

$\boxed{2.5.12}$ Si esprima il connettivo logico $\vee$ mediante il connettivo logico $\rightarrow$.

2.6 - La “forma normale congiuntiva”.

La scelta del connettivo logico unario “$\neg$” e dei tre connettivi logici binari “ $\wedge$ ”, “ $\vee$ ” e “ $\rightarrow$ ” per la costruzione delle formule rappresenta un compromesso tra economia (numero dei connettivi logici usati) e chiarezza (senso della formula costruita nel linguaggio comune). Abbiamo visto con l’osservazione 2.5.4 che ogni formula non solo è logicamente equivalente a una formula costruita utilizzando soltanto i connettivi logici “$\neg$” e “ $\wedge$ ” ma addirittura a una formula costruita utilizzando soltanto il connettivo binario $\odot$. In questa sezione introduciamo un’altra possibile forma standard (anzi, altre due) a cui ogni formula può essere ridotta; questa risulterà molto importante nella prossima sezione, dove la utilizzeremo per introdurre un algoritmo di soddisfacibilità.

Si dice *letterale* una formula che consiste in una formula atomica (cioè è una variabile proposizionale) oppure nella negazione di una formula atomica (cioè è la negazione di una variabile proposizionale).

Una formula si dice in *forma normale congiuntiva* (“FNC”) se è della forma

$$\varphi_1 \quad \text{oppure} \quad \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_h$$

con ciascun φ_i della forma

$$\varphi_{i,1} \quad \text{oppure} \quad \varphi_{i,1} \vee \varphi_{i,2} \vee \ldots \vee \varphi_{i,k_i}$$

dove ciascun $\varphi_{i,j}$ è un letterale.

Una formula si dice in *forma normale disgiuntiva* (“FND”) se è della forma

$$\varphi_1 \quad \text{oppure} \quad \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_h$$

con ciascun φ_i della forma

$$\varphi_{i,1} \quad \text{oppure} \quad \varphi_{i,1} \wedge \varphi_{i,2} \wedge \ldots \wedge \varphi_{i,k_i}$$

dove ciascun $\varphi_{i,j}$ è un letterale.

Osservazione 2.6.1

Ogni letterale, ogni congiunzione di letterali e ogni disgiunzione di letterali, in base alla definizione è sia una formula in FNC che una formula in FND.

Vogliamo dimostrare che ogni formula è logicamente equivalente a una formula in forma normale congiuntiva. A tale scopo ci serve ancora qualche risultato preliminare:

Teorema 2.6.2

Siano α, β, γ formule.

(i) Le formule $(\alpha \wedge \beta) \vee \gamma$ e $(\alpha \vee \gamma) \wedge (\beta \vee \gamma)$ sono logicamente equivalenti;

(ii) Le formule $\alpha \vee (\beta \wedge \gamma)$ e $(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ sono logicamente equivalenti;

(iii) Le formule $(\alpha \vee \beta) \wedge \gamma$ e $(\alpha \wedge \gamma) \vee (\beta \wedge \gamma)$ sono logicamente equivalenti;

(iv) Le formule $\alpha \wedge (\beta \vee \gamma)$ e $(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ sono logicamente equivalenti;

Dimostrazione — Come al solito, basta tabulare i valori di verità delle formule considerate in funzione di tutti i possibili valori di verità per α, β e γ. Ad esempio,

α	β	γ	$\alpha \wedge \beta$	$\alpha \vee \gamma$	$\beta \vee \gamma$	$(\alpha \wedge \beta) \vee \gamma$	$(\alpha \vee \gamma) \wedge (\beta \vee \gamma)$
0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1
0	1	0	0	0	1	0	0
0	1	1	0	1	1	1	1
1	0	0	0	1	0	0	0
1	0	1	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

α	β	γ	$\alpha \vee \beta$	$\alpha \vee \gamma$	$\beta \wedge \gamma$	$\alpha \vee (\beta \wedge \gamma)$	$(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

che dimostrano la (i) e la (ii). Si lasciano al lettore, quale utile esercizio, le tabulazioni dei valori di verità che provano la (iii) e la (iv) .

Lemma 2.6.3

Se φ è una formula in FND, allora $\neg\varphi$ è logicamente equivalente a una formula in FNC; se φ è una formula in FNC, allora $\neg\varphi$ è logicamente equivalente a una formula in FND.

Dimostrazione — Se φ è in FND; allora φ è della forma

$$\varphi = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_h$$

dove ciascun φ_i è della forma

$$\varphi_i = \varphi_{i,1} \wedge \varphi_{i,2} \wedge \ldots \wedge \varphi_{i_k}$$

dove ciascun $\varphi_{i,j}$ è un letterale.

Per le leggi di De Morgan (teor. 2.5.1) ne segue che $\neg\varphi$ è della forma

$$\neg\varphi = \neg\varphi_1 \wedge \neg\varphi_2 \wedge \ldots \wedge \neg\varphi_h$$

dove ciascun $\neg\varphi_i$ è della forma

$$\neg\varphi_i = \neg\varphi_{i,1} \vee \neg\varphi_{i,2} \vee \ldots \vee \neg\varphi_{i,k_i}$$

dove ciascun $\neg\varphi_{i,j}$ è la negazione di un letterale, dunque è logicamente equivalente a un letterale. Pertanto si è visto che: se φ è in FND, allora $\neg\varphi$ è logicamente equivalente a una formula in FNC.

Supponiamo poi che φ sia in FNC; allora φ è della forma

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_h$$

dove ciascun φ_i è della forma

$$\varphi_i = \varphi_{i,1} \vee \varphi_{i,2} \vee \ldots \vee \varphi_{i,k_i}$$

dove ciascun $\varphi_{i,j}$ è un letterale.

Per le leggi di De Morgan (teor. 2.5.1) ne segue che $\neg\varphi$ è della forma

$$\neg\varphi = \neg\varphi_1 \vee \neg\varphi_2 \vee \ldots \vee \neg\varphi_h$$

dove ciascun $\neg\varphi_i$ è della forma

$$\neg\varphi_i = \neg\varphi_{i,1} \wedge \neg\varphi_{i,2} \wedge \ldots \wedge \neg\varphi_{i,k_i}$$

dove ciascun $\neg\varphi_{i,j}$ è la negazione di un letterale, dunque è logicamente equivalente a un letterale.

Pertanto si è visto che: se φ è in FNC, allora $\neg\varphi$ è logicamente equivalente a una formula in FND.

Il lemma è così completamente dimostrato.

$\boxed{\text{Lemma 2.6.4}}$

Se φ e ψ sono formule in FNC, allora $\varphi \vee \psi$ è logicamente equivalente a una formula in FNC.

Dimostrazione — Per ipotesi, φ è della forma

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_h$$

dove ciascun φ_i è della forma

$$\varphi_i = \varphi_{i,1} \vee \varphi_{i,2} \vee \ldots \vee \varphi_{i_k}$$

dove ciascun $\varphi_{i,j}$ è un letterale; e ψ è della forma

$$\psi = \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_{\overline{h}}$$

dove ciascun ψ_i è della forma

$$\psi_i = \psi_{i,1} \vee \psi_{i,2} \vee \ldots \vee \psi_{i,\overline{k}_i}$$

dove ciascun $\psi_{i,j}$ è un letterale.

Ripetute applicazioni della (i) e della (ii) del teorema 2.6.2 ci permettono di concludere che $\varphi \vee \psi$ è logicamente equivalente alla formula

$$(\varphi_1 \vee \psi_1) \wedge (\varphi_1 \vee \psi_2) \wedge \ldots \wedge (\varphi_1 \vee \psi_{\overline{h}}) \wedge (\varphi_2 \vee \psi_1) \wedge (\varphi_2 \vee \psi_2) \wedge \ldots \wedge (\varphi_2 \vee \psi_{\overline{h}}) \wedge \ldots$$
$$\ldots \wedge (\varphi_h \vee \psi_1) \wedge (\varphi_h \vee \psi_2) \wedge \ldots \wedge (\varphi_h \vee \psi_{\overline{h}})$$

che (per la forma delle formule φ_i e ψ_i) è in FNC.

Lemma 2.6.5

Se φ e ψ sono formule in FND, allora $\varphi \wedge \psi$ è logicamente equivalente a una formula in FND.

Dimostrazione — Per ipotesi, φ è della forma

$$\varphi = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_h$$

dove ciascun φ_i è della forma

$$\varphi_i = \varphi_{i,1} \wedge \varphi_{i,2} \wedge \ldots \wedge \varphi_{i,k_i}$$

dove ciascun $\varphi_{i,j}$ è un letterale; e ψ è della forma

$$\psi = \psi_1 \vee \psi_2 \vee \ldots \vee \psi_{\overline{h}}$$

dove ciascun ψ_i è della forma

$$\psi_i = \psi_{i,1} \wedge \psi_{i,2} \wedge \ldots \wedge \psi_{i,\overline{k}_i}$$

dove ciascun $\psi_{i,j}$ è un letterale.

Ripetute applicazioni della (iii) e della (iv) del teorema 2.6.2 ci permettono di concludere che $\varphi \wedge \psi$ è logicamente equivalente alla formula

$$(\varphi_1 \wedge \psi_1) \vee (\varphi_1 \wedge \psi_2) \vee \ldots \vee (\varphi_1 \wedge \psi_{\overline{h}}) \vee (\varphi_2 \wedge \psi_1) \vee (\varphi_2 \wedge \psi_2) \vee \ldots \vee (\varphi_2 \wedge \psi_{\overline{h}}) \vee \ldots$$
$$\ldots \vee (\varphi_h \wedge \psi_1) \vee (\varphi_h \wedge \psi_2) \vee \ldots \vee (\varphi_h \wedge \psi_{\overline{h}})$$

che (per la forma delle formule φ_i e ψ_i) è in FND.

Teorema 2.6.6 (di “forma normale congiuntiva”)

Ogni formula è logicamente equivalente a una formula in FNC (e a una formula in FND).

Dimostrazione — Basta procedere per induzione sulla lunghezza della formula data: infatti l’asserto è banalmente vero per i letterali (ciascuno dei quali è sia in FNC che in FND).

Supposto vero l’asserto per le formule di lunghezza minore di n, con $n > 1$, per il teorema 2.4.1 resta da distinguere fra tre casi:

(i) la formula data è della forma $\neg\alpha$, con α di lunghezza minore di n;

(ii) la formula data è della forma $\alpha \wedge \beta$, con α e β di lunghezza minore di n;

(iii) la formula data è della forma $\alpha \vee \beta$, con α e β di lunghezza minore di n.

Nel caso (i), per concludere che $\neg\alpha$ è logicamente equivalente a una formula in FNC basta supporre (per l’ipotesi di induzione) che α sia logicamente equivalente a una formula in FND e applicare a questa il lemma 2.6.3; per concludere invece che $\neg\alpha$ è logicamente equivalente a una formula in FND basta supporre (per l’ipotesi di induzione) che α sia logicamente equivalente a una formula in FNC e applicare a questa il lemma 2.6.3.

Nel caso (ii), la conclusione che $\alpha \wedge \beta$ è logicamente equivalente a una formula in FNC segue immediatamente dall'ipotesi di induzione applicata ad α e a β; mentre per concludere che $\alpha \wedge \beta$ è logicamente equivalente a una formula in FND bisogna supporre (per l'ipotesi di induzione) che α e β siano logicamente equivalenti a formule α' e β' in FND ed applicare il lemma 2.6.5 alla formula $\alpha' \wedge \beta'$.

Nel caso (iii), infine, la conclusione che $\alpha \vee \beta$ è logicamente equivalente a una formula in FND segue immediatamente dall'ipotesi di induzione applicata ad α e a β; mentre per concludere che $\alpha \vee \beta$ è logicamente equivalente a una formula in FNC bisogna supporre (per l'ipotesi di induzione) che α e β siano logicamente equivalenti a formule α' e β' in FNC ed applicare il lemma 2.6.4 alla formula $\alpha' \vee \beta'$.

Esempio 2.6.7

Siano a, b, c variabili proposizionali. Troviamo una formula in FNC logicamente equivalente alla

$$\neg((a \to (\beta \wedge \gamma)) \wedge (\beta \to \neg a)).$$

In primo luogo eliminiamo il connettivo “ $\to$ ”; poi utilizzando le leggi di De Morgan portiamo il connettivo “$\neg$” a contatto diretto con le variabili proposizionali; infine applichiamo le (i) e (ii) del teorema 2.6.2 .

$$\neg((a \to (b \wedge c)) \wedge (b \to \neg a)) \equiv$$

$$\equiv \neg((\neg a \vee (b \wedge c)) \wedge (\neg b \vee \neg a)) \equiv$$

$$\equiv \neg(\neg a \vee (b \wedge c)) \vee \neg(\neg b \vee \neg a) \equiv$$

$$\equiv (\neg(\neg a) \wedge \neg(b \wedge c)) \vee (\neg(\neg b) \wedge \neg(\neg a)) \equiv$$

$$\equiv (a \wedge (\neg b \vee \neg c)) \vee (b \wedge a) \equiv$$

$$\equiv (a \vee b) \wedge (a \vee a) \wedge (\neg b \vee \neg c \vee b) \wedge (\neg b \vee \neg \gamma \vee a)$$

La formula così ottenuta, che è in FNC come si voleva, può naturalmente essere semplificata osservando che $a \vee a \equiv a$ e che $\neg b \vee b$ è una tautologia, cosicché anche $\neg b \vee \neg c \vee b$ è una tautologia e può essere soppressa dalla congiunzione; si giunge così alla più semplice formula in FNC

$$(a \vee b) \wedge a \wedge (\neg b \vee \neg c \vee a).$$

Vedremo più avanti (teorema 2.9.6) che questa formula può ancora essere notevolmente semplificata.

Esercizio 2.6.8

Siano a, b, c variabili proposizionali. Si trovi una formula in FNC logicamente equivalente alla

$$((a \to b) \to (\neg b \to c)) \to (a \to c)\,.$$

Esercizio 2.6.9

Trovare una formula in FNC logicamente equivalente alla

$$(a \to b) \to ((b \wedge c) \to a)\,.$$

2.7 - La notazione “in clausole”.

Introduciamo adesso una notazione alternativa per le formule in FNC. Questa notazione sarà molto utile nella sezione 2.9 per descrivere un efficace algoritmo che consente di decidere se una formula è soddisfacibile.

Si dice *clausola* un insieme di letterali.

Alla formula

$$\varphi_{i,1} \vee \varphi_{i,2} \vee \ldots \vee \varphi_{i,k_i} \quad \text{(dove ciascun } \varphi_{i,j} \text{ è un letterale)}$$

si associa la clausola $\{\varphi_{i,1}, \varphi_{i,2}, \ldots, \varphi_{i,k_i}\}$. Si noti che a formule diverse (che possono ad esempio differire per l'ordine dei $\varphi_{i,j}$ o perché qualche $\varphi_{i,j}$ è ripetuto) può venire associata la stessa clausola: tali diverse formule, però, hanno tutte lo stesso valore di verità sotto ogni interpretazione (per le (ii) e (iii) del teorema 2.3.1, e per la (ii) del teorema 2.3.3).

Alla formula in FNC

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_h$$

dove ciascun φ_i è della forma

$$\varphi_i = \varphi_{i,1} \vee \varphi_{i,2} \vee \ldots \vee \varphi_{i,k_i}$$

dove ciascun $\varphi_{i,j}$ è un letterale si associa l'insieme di clausole

$$\{\{\varphi_{1,1}, \varphi_{1,2}, \ldots, \varphi_{1,k_1}\}, \{\varphi_{2,1}, \varphi_{2,2}, \ldots, \varphi_{2,k_2}\}, \ldots, \{\varphi_{i,1}, \varphi_{i,2}, \ldots, \varphi_{i,k_i}\}, \ldots, \{\varphi_{h,1}, \varphi_{h,2}, \ldots, \varphi_{h,k_h}\}\}\,.$$

Ancora una volta, la notazione insiemistica ingloba alcune regole di calcolo dei valori di verità, come le (ii) e (iii) del teorema 2.3.1 e la (i) del teorema 2.3.3.

Siano C $:= \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ una clausola (dove ciascun λ_i è un letterale) e v una valutazione di verità sull’insieme delle formule. Se esiste almeno un $\lambda_{\bar{i}} \in$ C tale che $v(\lambda_{\bar{i}}) = 1$, si dice che “C *è vera sotto* v”, oppure che “C *è vera nell’interpretazione* v”, oppure anche che “v *soddisfa* C”; scriveremo talvolta

$$v \models \mathrm{C}\,.$$

Questa definizione è scelta in modo che una clausola risulti vera sotto una valutazione di verità v se e soltanto se qualsiasi formula a cui essa resta associata è vera sotto v. Analogamente a quanto stabilito nella sez. 2.2, se ogni valutazione di verità soddisfa la clausola C si dice che C è una *tautologia*; se esiste almeno una valutazione di verità che soddisfa C, si dice che C è *soddisfacibile*; se nessuna valutazione di verità soddisfa C, si dice che C è *insoddisfacibile*.

Siano ora S $:= \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_h\}$ un insieme di clausole e v una valutazione di verità sull’insieme delle formule. Se per ogni $\mathrm{C}_i \in$ S è $v \models \mathrm{C}_i$, si dice che “S *è vero sotto* v”, oppure che “S *è vero nell’interpretazione* v”, oppure anche che “v *soddisfa* S”; scriveremo talvolta

$$v \models \mathrm{S}\,.$$

Questa definizione è scelta in modo che un insieme di clausole risulti vero sotto una valutazione di verità v se e soltanto se qualsiasi formula a cui esso resta associato è vera sotto v. Analogamente a quanto stabilito nella sez. 2.2, se ogni valutazione di verità soddisfa l’insieme di clausole S si dice che S è una *tautologia*; se esiste almeno una valutazione di verità che soddisfa S, si dice che S è *soddisfacibile*; se nessuna valutazione di verità soddisfa S, si dice che S è *insoddisfacibile*.

Siano S e S’ due insiemi di clausole. Si dice che S’ *è conseguenza logica di* S e si scrive

$$\mathrm{S} \models \mathrm{S'}$$

se ogni valutazione di verità che soddisfa S soddisfa anche S’. Ovviamente, S’ è conseguenza logica di S se e soltanto se qualsiasi formula a cui resta associato S’ è conseguenza logica di qualsiasi formula a cui resta associato S.

Due insiemi di clausole si dicono *logicamente equivalenti* se ciascuno di essi è conseguenza logica dell’altro; ciò avviene se e soltanto se qualsiasi formula a cui resta associato il primo insieme di clausole è logicamente equivalente a qualsiasi formula a cui resta associato l’altro insieme di clausole.

Coerentemente con le definizioni poste sopra, la clausola vuota (che di solito si indica con la notazione “[]” anziché con la notazione “{}”) si considera insoddisfacibile ([5]), cosicché un insieme di clausole al quale appartenga la clausola vuota è insoddisfacibile; mentre l’insieme vuoto di clausole {} si considera soddisfacibile ([6]).

2.8 - Confutazioni.

Si dice *confutazione* di un insieme di clausole S una sequenza finita

$$\mathrm{S}_0,\, \mathrm{S}_1,\, \mathrm{S}_2,\, \ldots,\, \mathrm{S}_n$$

di insiemi di clausole tale che:

(i) $\mathrm{S}_0 = \mathrm{S}$;

(ii) S_i è conseguenza logica di S_{i-1} per $i := 1, 2, \ldots, n$;

(iii) la clausola vuota appartiene a S_n .

Teorema 2.8.1

Sia S un insieme di clausole. Se esiste una confutazione di S, allora S è insoddisfacibile.

Dimostrazione — Sia

$$\mathrm{S}_0,\, \mathrm{S}_1,\, \mathrm{S}_2,\, \ldots,\, \mathrm{S}_n$$

una confutazione di S . Se S fosse soddisfacibile, esisterebbe una valutazione di verità v che soddisfa S ; sarebbe allora immediato (la dimostrazione, per induzione su i, si lascia al lettore quale utile esercizio) che v soddisfa ogni S_i per $i := 1, 2, \ldots, n$ e in particolare soddisfa S_n: assurdo per la condizione (iii) della definizione di “confutazione”.

Nella prossima sezione vedremo che il teorema 2.8.1 è invertibile: proveremo infatti che se S è un insieme insoddisfacibile di clausole esiste una confutazione di S . Non solo: descriveremo un algoritmo che, in un numero di passi mai superiore al numero delle variabili proposizionali presenti in S, genera una confutazione di S (se S è insoddisfacibile) o permette di trovare una interpretazione che soddisfa S.

---

[5] Se la clausola vuota fosse soddisfacibile, esisterebbero una valutazione di verità v e un letterale λ appartenente alla clausola vuota tali che $v(\lambda) = 1$; ma, per definizione, alla clausola vuota non appartiene alcun letterale.

[6] Se l’insieme vuoto fosse insoddisfacibile, in esso vi sarebbe una clausola insoddisfacibile; ma, per definizione, all’insieme vuoto non appartiene alcuna clausola.

In preparazione di quanto sopra, osserviamo due casi in cui possiamo affermare con certezza che un insieme S’ di clausole è conseguenza logica di un altro insieme S : il primo caso (oss. 2.8.2) è assolutamente banale, il secondo (teor. 2.8.4) è più ingegnoso. Entrambi sono alla base dell’algoritmo che descriveremo nella prossima sezione.

Osservazione 2.8.2

Sia S un insieme di clausole. Ogni sottoinsieme di S è conseguenza logica di S.

Siano C_1 e C_2 due clausole, e sia p una variabile proposizionale tale che $p \in C_1$ e $\neg p \in C_2$. Si dice *risolvente* (rispetto a p) di C_1 e C_2 la clausola

$$Ris_p(C_1, C_2) := C_1 \cup C_2 \setminus \{p, \neg p\} .$$

Osservazione 2.8.3

Sia S un insieme di clausole, siano $C_1, C_2 \in S$, e sia p una variabile proposizionale tale che $p \in C_1$ e $\neg p \in C_2$. La clausola $Ris_p(C_1, C_2)$ è la clausola vuota se e soltanto se $C_1 = \{p\}$ e $C_2 = \{\neg p\}$.

Teorema 2.8.4

Sia S un insieme di clausole privo di tautologie, siano $C_1, C_2 \in S$, e sia p una variabile proposizionale tale che $p \in C_1$ e $\neg p \in C_2$. L’insieme di clausole

$$S \cup \{Ris_p(C_1, C_2)\}$$

è conseguenza logica di S.

Dimostrazione — Sia v una valutazione di verità che soddisfa S; dobbiamo dimostrare che v soddisfa $S \cup \{Ris_p(C_1, C_2)\}$.

Per ipotesi, v soddisfa ogni clausola che appartiene a S, dunque dobbiamo soltanto dimostrare che v soddisfa $Ris_p(C_1, C_2)$, cioè che in $Ris_p(C_1, C_2)$ esiste un letterale λ_0 tale che $v(\lambda) = 1$. Poiché $C_1 \in S$, in C_1 esiste un letterale λ_1 tale che $v(\lambda_1) = 1$; se $\lambda_1 \neq p$, è anche $\lambda_1 \neq \neg p$ (poiché $\neg p \notin C_1$, dato che per ipotesi in S non vi sono tautologie) e quindi $\lambda_1 \in Ris_p(C_1, C_2)$, cosicché possiamo scegliere $\lambda_0 := \lambda_1$. Se invece $\lambda_1 = p$, è $v(p) = 1$ e quindi $v(\neg p) = 0$; poiché $C_2 \in S$, in C_2 esiste un letterale λ_2 tale che $v(\lambda_2) = 1$ e deve essere $\lambda_2 \neq \neg p$ (perché si è appena osservato che $v(\neg p) = 0$) e $\lambda_2 \neq p$ (perché $p \notin C_2$, dato che per ipotesi in S non vi sono tautologie), quindi $\lambda_2 \in Ris_p(C_1, C_2)$, cosicché possiamo scegliere $\lambda_0 := \lambda_2$.

2.9 - L’algoritmo di Martin Davis e Hilary Putnam.

Questo algoritmo, ideato nel 1960, opera su un insieme S di clausole e ne genera una confutazione (se S è insoddisfacibile) oppure genera una valutazione di verità che ne soddisfa ogni clausola (se S è soddisfacibile).

L’algoritmo costruisce una sequenza finita di insiemi di clausole

$$S_0, S_1, S_2, \dots, S_n$$

come segue. Si definisce in primo luogo S_1 come l’insieme ottenuto da S eliminando le eventuali tautologie. Il passo $j-$simo dell’algoritmo trasforma S_j in S_{j+1} mediante quattro sotto-passi:

(i) si sceglie una variabile proposizionale p che compare, direttamente o come negazione, nelle clausole di S_j: questa viene detta il *pivot* del passo;

(ii) si costruisce un nuovo insieme S_j' di clausole che ha come elementi tutte le clausole di S_j e tutti i possibili risolventi rispetto al pivot fra due clausole di S_j;

(iii) si eliminano da S_j' tutte le clausole in cui compare il pivot, e si indica con S_{j+1} l’insieme di clausole così ottenuto.

(iv) si eliminano da S_{j+1} le tautologie;

Osserviamo subito esplicitamente che, per il teorema 2.8.4 e l’osservazione 2.8.2, al $j-$simo passo l’insieme S_{j+1} è conseguenza logica di S_j.

L’algoritmo termina quando fra le clausole di S_{j+1} compare la clausola vuota oppure quando S_{j+1} risulta vuoto. Per l’osservazione 2.8.3, se fra le clausole di S_{j+1} compare la clausola vuota la sequenza finita

$$S_0, S_1, S_2, \dots, S_j, S_{j+1}$$

è una confutazione di S.

Osservazione 2.9.1

Nel passaggio dall’insieme di clausole S_j all’insieme S, S_{j+1} al passo $j-$simo dell’algoritmo di Davis e Putnam scompare certamente la variabile proposizionale assunta come pivot. Può accadere che scompaiano altre variabili proposizionali, che diremo *esodate*. In ogni caso, dopo un numero di passi pari al più al numero delle variabili proposizionali presenti nell’insieme iniziale, l’algoritmo termina (perché non vi sono più variabili nelle clausole).

Teorema 2.9.2

Sia S un insieme di clausole. Se l’algoritmo di Davis e Putnam per S termina con l’insieme vuoto, esiste una valutazione v che soddisfa S.

Dimostrazione — Basterà definire una valutazione v che soddisfa S sull’insieme delle variabili proposizionali che compaiono nelle clausole di S. Lo facciamo procedendo a ritroso sulla sequenza di insiemi di clausole

$$\mathrm{S}_0, \mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_n \ (= \{\})$$

generata dall’algoritmo.

Sia p il pivot per il quale dall’insieme di clausole S_{n-1} l’algoritmo ha prodotto l’insieme vuoto di clausole S_n ; necessariamente p compare in ogni clausola di S_{n-1}, e ci sono tre possibilità:

(i) in S_{n-1} l’ultimo pivot p compare come tale (cioè senza mai essere preceduto dalla negazione “$\neg$”) in ogni clausola;

(ii) in S_{n-1} l’ultimo pivot p compare preceduto dalla negazione “$\neg$” in ogni clausola;

(iii) in S_{n-1} l’ultimo pivot p compare in tutte le clausole, ma in alcune di esse (almeno una!) compare come tale e in altre (almeno una!) compare preceduto dalla negazione “$\neg$” ; e tutti i risolventi sono tautologie.

Nel caso (i), S_{n-1} è soddisfatto ponendo $v(p) := 1$. Nel caso (ii), S_{n-1} è soddisfatto ponendo $v(p) := 0$.

Nel caso (iii), osserviamo che le variabili proposizionali q_i diverse da p eventualmente ancora presenti in S_{n-1} risulteranno esodate; definiamo per esse $v(q_i)$ in un modo qualsiasi (per fissare le idee: $v(q_i) := 0$). La v finora definita sulle q_i può essere estesa a p in due modi diversi, che indicheremo con v^+ e v^- ; precisamente porremo $v^+(p) := 1$ e $v^-(p) := 0$. Se, per assurdo, né v^+ né v^- soddisfacessero S_{n-1}, in S_{n-1} ci sarebbero una clausola C_1 non soddisfatta da v^+ (cosicché $\neg p \in \mathrm{C}_1$) e una clausola C_2 non soddisfatta da v^- (cosicché $p \in \mathrm{C}_2$) : mostriamo che ciò conduce a un assurdo. Per ipotesi, il risolvente R di C_1 e C_2 è una tautologia, quindi esiste una variabile esodata $\overline{q}$ tale che sia $\overline{q}$ che $\neg\overline{q}$ appartengono a R; e certamente uno dei due letterali $\overline{q}$ e $\neg\overline{q}$ è soddisfatto da v (e quindi anche da v^+ e v^- perché ciascuna di esse estende v): sia esso λ. Inevitabilmente, $\lambda \in \mathrm{C}_1$ oppure $\lambda \in \mathrm{C}_2$ (per definizione del risolvente R): se $\lambda \in \mathrm{C}_1$, la clausola C_1 è soddisfatta da v (e quindi anche da v^+); se $\lambda \in \mathrm{C}_2$, la clausola C_2 è soddisfatta da v (e quindi anche da v^-). In ogni caso abbiamo raggiunto un assurdo.

Resta così provato che v può essere estesa a una valutazione che soddisfa S_{n-1} .

Adesso supponiamo di aver definito v sulle variabili che compaiono in S_{i+1} in modo che S_{i+1} risulti soddisfatto da v; estendiamo v in un modo qualsiasi alle variabili che eventualmente risultano esodate nel passaggio da S_i a S_{i+1} (definendola ad esempio uguale a 0 su ciascuna di esse), e mostriamo che v si può ulteriormente estendere al pivot p del passo $i-$ simo in modo da soddisfare S_i .

Come si è fatto sopra, indichiamo con v^+ e v^- le due possibili estensioni di v a p $(v^+(p) := 1$ e $v^-(p) := 0)$. Se, per assurdo, né v^+ né v^- soddisfacessero S_i, in S_i ci sarebbero una clausola C_1 non soddisfatta da v^+ e una clausola C_2 non soddisfatta da v^- ; in particolare, dovrebbe essere $\neg p \in C_1$ e $p \in C_2$. Per come abbiamo definito l’algoritmo di Davis e Putnam, il risolvente di C_1 e C_2 appartiene a S_{i+1} e dunque è soddisfatto da v: a tale risolvente dunque appartiene almeno un letterale λ tale che $v(\lambda) = 1$ (e quindi $v^+(\lambda) = 1$ e $v^-(\lambda) = 1$, perché v^+ e v^- estendono v); d’altro lato, λ proviene da una delle due clausole C_1 e C_2 : se $\lambda \in C_1$ si ha che v^+ soddisfa C_1, se $\lambda \in C_2$ si ha che v^- soddisfa C_2 e comunque si è raggiunto un assurdo; pertanto v^+ oppure v^- sodisfa S_i, e si è provato che v può essere estesa a una valutazione che soddisfa S_i . Il teorema è così completamente dimostrato.

Osservazione 2.9.3

Dalla dimostrazione del teorema 2.9.2 si comprende che se l’insieme S è soddisfacibile non è in generale unica la valutazione di verità che lo soddisfa

Osservazione 2.9.4

Nella descrizione che abbiamo dato dell’algoritmo di Davis e Putnam (e della costruzione di una valutazione di verità che soddisfi S qualora S sia soddisfacibile, cfr. la dimostrazione del teorema 2.9.2) restano arbitrari alcuni passaggi: la scelta a ciascun passo del pivot, ad esempio, e la scelta del valore di verità da assegnare alle variabili esodate. Ciò non si addice ad un algoritmo in quanto tale: ma il problema può essere agevolmente superato, ad esempio stabilendo di scegliere come pivot fra quelle presenti la variabile proposizionale che precede le altre in un ordine predeterminato (come potrebbe essere l’ordine alfabetico; ma si veda anche l’oss. 2.9.5) e stabilendo di assegnare valore “0” a tutte le variabili esodate.

Osservazione 2.9.5

Si potrebbe dimostrare che l’algoritmo di Davis e Putnam risulta mediamente più veloce se ogni volta si sceglie come pivot una variabile che compare in una clausola di lunghezza minima.

Teorema 2.9.6

Sia $\mathcal{K}$ un insieme di clausole, e siano $C_1, C_2 \in \mathcal{K}$ con $C_1 \subsetneq C_2$. Allora $\mathcal{K}$ è soddisfacibile se e soltanto se $\mathcal{K} \setminus \{C_2\}$ è soddisfacibile.

Dimostrazione — Se $\mathcal{K}$ è soddisfacibile, esiste una valutazione di verità v che soddisfa ogni clausola di $\mathcal{K}$; allora v soddisfa in particolare ogni clausola di $\mathcal{K} \setminus \{C_2\}$, e quindi $\mathcal{K} \setminus \{C_2\}$ è soddisfacibile.

Viceversa, supponiamo che $\mathcal{K} \setminus \{C_2\}$ sia soddisfacibile; allora esiste una valutazione di verità v che soddisfa ogni clausola di $\mathcal{K} \setminus \{C_2\}$: dimostriamo che v soddisfa anche la clausola C_2, cosicché v soddisfa $\mathcal{K}$ e quindi $\mathcal{K}$ è soddisfacibile. Per ipotesi, v soddisfa ogni clausola di $\mathcal{K} \setminus \{C_2\}$ e quindi in particolare soddisfa C_1 (che è diversa da C_2 e quindi appartiene a $\mathcal{K} \setminus \{C_2\}$); esiste cioè in C_1 almeno un letterale λ per il quale $v(\lambda) = 1$. Ma $\lambda \in C_2$ (perché $C_1 \subset C_2$) e quindi v soddisfa C_2 (come si voleva dimostrare).

Osservazione 2.9.7

Per il teorema 2.9.6, se vogliamo valutare la soddisfacibilità di un insieme $\mathcal{K}$ di clausole possiamo semplificare il nostro compito eliminando da $\mathcal{K}$ tutte le clausole che contengono propriamente altre clausole di $\mathcal{K}$.

Esempio 2.9.8

Sia

$$\mathcal{K} := \{\{a, b, \neg c\}, \{a, \neg c, d\}, \{a, \neg c, \neg e\}, \{a, b, c, d, e\}, \{\neg a, c, \neg e\}, \{a, \neg c\}, \{c\}\}.$$

Per decidere se $\mathcal{K}$ è soddisfacibile, come primo passo conviene eliminare da $\mathcal{K}$ le clausole $\{a, b, \neg c\}$, $\{a, \neg c, d\}$ e $\{a, \neg c, \neg e\}$ perché ciascuna di esse contiene $\{a, \neg c\}$; ed anche conviene eliminare da le clausole $\{a, b, c, d, e\}$ e $\{\neg a, c, \neg e\}$ perché ciascuna di esse contiene $\{c\}$. Ci si riconduce dunque a valutare la soddisfacibilità di

$$\mathcal{K}_0 := \{\{a, \neg c\}, \{c\}\}$$

che risulta soddisfatto da qualsiasi v per la quale si abbia $v(a) = v(c) = 1$.

Esempio 2.9.9

Sia
$\mathcal{K} := \{\{a,b,c,x\}, \{a,\neg b,\neg y,\neg z\}, \{\neg a,\neg b\}, \{\neg a,c,\neg x\}, \{b,c,y\}, \{\neg b,\neg c\}, \{x,y,z\}, \{b,\neg y,\neg z\}\}$.

Per decidere se $\mathcal{K}$ è soddisfacibile, applichiamo l’algoritmo di Davis e Putnam. Come primo pivot scegliamo a (la prima lettera in ordine alfabetico che compare in una clausola di lunghezza minima, in questo caso di lunghezza 2):

Pivot a:

clausole non contenenti né a né $\neg a$: $\{b, c, y\}$, $\{\neg b, \neg c\}$, $\{x, y, z\}$, $\{b, \neg y, \neg z\}$;
$\text{Ris}_a(\{a, b, c, x\}, \{\neg a, \neg b\}) = \{b, \neg b, c, x\}$ (si sopprime perché tautologia) ;
$\text{Ris}_a(\{a, b, c, x\}, \{\neg a, c, \neg x\}) = \{b, c, x, \neg x\}$ (si sopprime perché tautologia) ;
$\text{Ris}_a(\{a, \neg b, \neg y, \neg z\}, \{\neg a, \neg b\}) = \{\neg b, \neg y, \neg z\}$;
$\text{Ris}_a(\{a, \neg b, \neg y, \neg z\}, \{\neg a, c, \neg x\}) = \{\neg b, c, \neg x, \neg y, \neg z\}$
(si sopprime perché contiene la precedente) ;

$$\{\{b,c,y\}, \{\neg b,\neg c\}, \{x,y,z\}, \{b,\neg y,\neg z\}, \{\neg b,\neg y,\neg z\}\}$$

Come nuovo pivot scegliamo b (la prima lettera in ordine alfabetico che compare in una clausola di lunghezza minima, cioè ancora di lunghezza 2):

Pivot b:

clausole non contenenti né b né $\neg b$: $\{x, y, z\}$;
$\text{Ris}_b(\{b, c, y\}, \{\neg b, \neg c\}) = \{c, \neg c, y\}$ (si sopprime perché tautologia) ;
$\text{Ris}_b(\{b, c, y\}, \{\neg b, \neg y, \neg z\}) = \{c, y, \neg y, \neg z\}$ (si sopprime perché tautologia) ;
$\text{Ris}_b(\{b, \neg y, \neg z\}, \{\neg b, \neg c\}) = \{\neg c, \neg y, \neg z\}$;
$\text{Ris}_b(\{b, \neg y, \neg z\}, \{\neg b, \neg y, \neg z\}) = \{\neg y, \neg z\}$;

$$\{\{x,y,z\}, \{\neg c,\neg y,\neg z\}, \{\neg y,\neg z\}\}$$

Poiché $\{\neg y,\neg z\} \subset \{\neg c,\neg y,\neg z\}$, la clausola $\{\neg c,\neg y,\neg z\}$ può essere soppressa; siamo dunque ricondotti a considerare l’insieme

$$\{\{x,y,z\}, \{\neg y,\neg z\}\}$$

Come nuovo pivot scegliamo y (la prima lettera in ordine alfabetico che compare in una clausola di lunghezza minima, cioè ancora di lunghezza 2):

Pivot y:

$\text{Ris}_y(\{x, y, z\}, \{\neg y, \neg z\}) = \{x, z, \neg z\}$ (si sopprime perché tautologia) ;

$$\{\ \}$$

Poiché l’insieme vuoto di clausole è soddisfacibile, anche $\mathcal{K}$ è soddisfacibile. Procedendo a ritroso, vediamo che una valutazione di verità che soddisfi $\mathcal{K}$ può essere costruita ponendo $v(z) := 0,\ v(x) := 0,\ v(y) := 1,\ v(c) := 0,\ v(b) := 0,\ v(a) := 1.$

Si osservi comunque che le variabili proposizionali c, x e z risultano esodate: ad esse potremmo assegnare qualunque valore di verità (ridefinendo opportunamente di conzeguenza i valori di verità per a, b e y).

2.10 - Delitto al castello del duca Glomgold.

Il duca Flintheart Glomgold è stato trovato assassinato nel suo maniero! La polizia sospetta il maggiordomo. Vediamo se, in base ai dati raccolti durante le indagini, è possibile dimostrare in modo inoppugnabile che davvero è lui il colpevole.

In base alle indagini degli inquirenti, si è stabilito con certezza che:

(i) se il maggiordomo è innocente, allora se il dottore è colpevole anche la cameriera è colpevole;

(ii) se la cameriera è colpevole, anche il maggiordomo è colpevole;

(iii) almeno uno fra maggiordomo, cameriera e dottore è colpevole.

Per formalizzare il nostro ragionamento, abbiamo bisogno di introdurre tre variabili proposizionali, cioè:

$p :=$ “il maggiordomo è colpevole” ; $q :=$ “la cameriera è colpevole” ; $r :=$ “il dottore è colpevole” .

I risultati delle indagini si possono esprimere nel modo seguente:

(i) diventa: $\neg p \to (r \to q)$;

(ii) diventa: $q \to p$;

(iii) diventa: $p \vee q \vee r$.

Vogliamo stabilire se la colpevolezza del maggiordomo è conseguenza logica delle (i), (ii) e (iii), cioè se $\{\neg p \to (r \to q),\ \ q \to p,\ \ p \vee q \vee r\} \vDash p$
ossia se

$$(\neg p \to (r \to q)) \wedge (q \to p) \wedge (p \vee q \vee r) \vDash p$$

ossia se

$$(\neg p \to (r \to q)) \wedge (q \to p) \wedge (p \vee q \vee r) \wedge \neg p \quad \text{è} \quad \text{insoddisfacibile.}$$

Possiamo usare l’algoritmo di tabulazione dei valori di verità (ancora ragionevole, poiché ci sono soltanto tre variabili proposizionali) oppure l’algoritmo di Davis e Putnam. Vediamoli entrambi.

p	q	r	$\neg p$	$r \to q$	$\neg p \to (r \to q)$	$q \to p$	$p \vee q \vee r$
0	0	0	1	1	1	1	0
0	0	1	1	0	0	1	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	0	1
1	0	0	0	1	1	1	1
1	0	1	0	0	1	1	1
1	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1

In ogni riga nella quale le ultime tre colonne (formule corrispondenti ai risultati delle indagini) presentano il valore “1” anche la colonna di p presenta il valore “1”: dunque ogni valutazione di verità che soddisfa i risultati delle indagini soddisfa anche la colpevolezza del maggiordomo, ossia

$$\{\neg p \rightarrow (r \rightarrow q), \quad q \rightarrow p, \quad p \vee q \vee r\} \vDash p\,;$$

si è così provato che la colpevolezza del maggiodomo è conseguenza logica dei risultati delle indagini.

In alternativa, possiamo applicare l’algoritmo di Davis e Putnam. A tale scopo dobbiamo in primo luogo ridurre in FNC la formula

$$(\neg p \rightarrow (r \rightarrow q)) \wedge (q \rightarrow p) \wedge (p \vee q \vee r) \wedge \neg p\,.$$

Eliminiamo il connettivo “ $\rightarrow$ ” ricordando che $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$, cosicché

$r \rightarrow q$ diventa $\neg r \vee q$,

$q \rightarrow p$ diventa $\neg q \vee p$ e

$\neg p \rightarrow (r \rightarrow q)$ (cioè $\neg p \rightarrow (\neg r \vee q)$) diventa $\neg(\neg p) \vee (\neg r \vee q)$

(ossia $p \vee \neg r \vee q$) .

La formula considerata si scrive dunque in FNC

$$(p \vee \neg r \vee q) \wedge (\neg q \vee p) \wedge (p \vee q \vee r) \wedge \neg p$$

e come insieme di clausole $\{\{p, q, \neg r\}, \{p, \neg q\}, \{p, q, r\}, \{\neg p\}\}$.

Applichiamo l’algoritmo di Davis e Putnam.

Pivot p:

clausole non contenenti né p né $\neg p$: nessuna ;

$\mathrm{Ris}_p(\{p, q, \neg r\}, \{\neg p\}) = \{q, \neg r\}$;

$\mathrm{Ris}_p(\{p, \neg q\}, \{\neg p\}) = \{\neg q\}$;

$\mathrm{Ris}_p(\{p, q, r\}, \{\neg p\}) = \{q, r\}$;

$$\{\{q, \neg r\}, \{\neg q\}, \{q, r\}\}$$

Pivot q:

clausole non contenenti né q né $\neg q$: nessuna ;

$\text{Ris}_q(\{q, \neg r\}, \{\neg q\}) = \{\neg r\}$;

$\text{Ris}_q(\{q, r\}, \{\neg q\}) = \{r\}$;

$$\{\{\neg r\}, \{r\}\}$$

Questo insieme di clausole non è soddisfacibile, quindi nemmeno la formula di partenza è soddisfacibile e dunque p è conseguenza logica di $\{\neg p \to (r \to q), \quad q \to p, p \vee q \vee r\}$, come si voleva dimostrare.

Ovviamente si poteva condurre a termine l’algoritmo di Davis e Putnam, ricavando da $\{\{\neg r\}, \{r\}\}$ (scegliendo come pivot l’unica variabile superstite, cioè r) l’insieme $\{[\,]\}$ che contiene soltanto la clausola vuota, e confermando così che l’insieme di clausole da cui eravamo partiti non è soddisfacibile.

2.11 - Confutazioni rapide.

La definizione di “confutazione” per un insieme di clausole data nella sez. 2.8 risulta talvolta eccessivamente farraginosa e ingombrante. Segnaliamo dunque una possibile alternativa, che per distiguerla dalla precedente denomineremo “confutazione rapida”.

Si dice *confutazione rapida* di un insieme di clausole S una sequenza finita di clausole

$$C_0, C_1, C_2, \dots, C_n$$

tale che

(i) ogni clausola C_i appartiene a S oppure è il risolvente (rispetto a un’opportuna variabile logica) di due clausole che la precedono nella sequenza;

(ii) C_n è la clausola vuota.

È facile convincersi che da ogni confutazione si può ricavare una confutazione rapida e viceversa. In generale, le confutazioni rapide, consistendo soltanto di clausole, sono più agili e veloci a scriversi; d’altro lato, richiedono una certa inventiva: l’algoritmo di Davis e Putnam produce (per il generico insieme insoddisfacibile) una confutazione nel senso definito nella sez. 2.8 e non direttamente una confutazione rapida.

Esempio 2.11.1

Sia $\mathrm{S} := \{\{p, q\}, \{\neg p, r\}, \{p, \neg q, r\}, \{p, \neg q, \neg r\}, \{\neg p, \neg r\}\}$. Una confutazione rapida di S è

$\mathrm{C}_0 := \{p, \neg q, r\} \in \mathrm{S}$;

$\mathrm{C}_1 := \{p, \neg q, \neg r\} \in \mathrm{S}$;

$\mathrm{C}_2 := \{p, \neg q\} = \mathrm{Ris}_r(\mathrm{C}_0, \mathrm{C}_1)$;

$\mathrm{C}_3 := \{p, q\} \in \mathrm{S}$;

$\mathrm{C}_4 := \{p\} = \mathrm{Ris}_q(\mathrm{C}_2, \mathrm{C}_3)$;

$\mathrm{C}_5 := \{\neg p, r\} \in \mathrm{S}$;

$\mathrm{C}_6 := \{\neg p, \neg r\} \in \mathrm{S}$;

$\mathrm{C}_7 := \{\neg p\} = \mathrm{Ris}_r(\mathrm{C}_5, \mathrm{C}_6)$;

$\mathrm{C}_8 := [\,] = \mathrm{Ris}_p(\mathrm{C}_4, \mathrm{C}_7)$.

Esempio 2.11.2

Sia $\mathrm{S} := \{\{p, q, r, s\}, \{q, s\}, \{\neg q, r, \neg s\}, \{\neg q, \neg r, \neg s\}, \{p, q, \neg s\}, \{\neg p, q, r\}, \{\neg q\}, \{\neg p, q, \neg s\}\}$. Una confutazione rapida di S è

$\mathrm{C}_0 := \{p, q, \neg s\} \in \mathrm{S}$;

$\mathrm{C}_1 := \{\neg p, q, \neg s\} \in \mathrm{S}$;

$\mathrm{C}_2 := \{q, \neg s\} = \mathrm{Ris}_p(\mathrm{C}_0, \mathrm{C}_1)$;

$\mathrm{C}_3 := \{q, s\} \in \mathrm{S}$;

$\mathrm{C}_4 := \{q\} = \mathrm{Ris}_s(\mathrm{C}_2, \mathrm{C}_3)$;

$\mathrm{C}_5 := \{\neg q\} \in \mathrm{S}$;

$\mathrm{C}_6 := [\,] = \mathrm{Ris}_q(\mathrm{C}_4, \mathrm{C}_5)$.

2.12 - Il teorema di compattezza.

Come vedremo nel capitolo 3, può accadere che si debbano considerare insiemi infiniti di clausole.

Sia $\mathcal{K}$ un insieme (eventualmente infinito) di clausole. Si dice che $\mathcal{K}$ è *finitamente soddisfacibile* se ogni sottoinsieme finito di $\mathcal{K}$ è soddisfacibile, cioè se per ogni sottoinsieme finito F di $\mathcal{K}$ esiste una valutazione di verità v_F (che in generale dipenderà da F) che soddisfa ogni clausola di F. Se $\mathcal{K}$ è soddisfacibile, esiste una valutazione di verità v_0 che soddisfa ogni clausola di $\mathcal{K}$: ovviamente v_0 soddisfa in particolare ogni clausola di ogni sottoinsieme finito di $\mathcal{K}$, e quindi $\mathcal{K}$ è finitamente soddisfacibile. Scopo di questa sezione è dimostrare che (abbastanza sorprendentemente) vale anche l’implicazione inversa: se $\mathcal{K}$ è finitamente soddisfacibile, allora $\mathcal{K}$ è soddisfacibile ([7]).

Lemma 2.12.1

Sia $\mathcal{K}$ un insieme di clausole, e sia p una variabile proposizionale. Se $\mathcal{K}$ è finitamente soddisfacibile, allora almeno uno dei due insiemi

$$\mathcal{K}_+ := \mathcal{K} \cup \{\{p\}\} \qquad \text{e} \qquad \mathcal{K}_- := \mathcal{K} \cup \{\{\neg p\}\}$$

è finitamente soddisfacibile.

Dimostrazione — Per assurdo, supponiamo che nessuno dei due insiemi $\mathcal{K}_+$ e $\mathcal{K}_-$ sia finitamente soddisfacibile: allora esistono un sottoinsieme finito

$$N_+ := \{C_1, C_2, \ldots, C_s, \{p\}\}$$

di $\mathcal{K}_+$ e un sottoinsieme finito

$$N_- := \{D_1, D_2, \ldots, D_t, \{\neg p\}\}$$

di $\mathcal{K}_-$ entrambi non soddisfacibili. Sicuramente $\{p\} \in N_+$ e $\{\neg p\} \in N_-$ perché per ipotesi $\mathcal{K}$ è finitamente soddisfacibile.

Poiché $\mathcal{K}$ è finitamente soddisfacibile, esiste una valutazione di verità v_0 che soddisfa

$$\{C_1, C_2, \ldots, C_s, D_1, D_2, \ldots, D_t\} .$$

Necessariamente, $v_0(p) = 1$ (e quindi v_0 soddisfa $\{p\}$) oppure $v_0(p) = 0$ (e quindi v_0 soddisfa $\{\neg p\}$); nel primo caso, v_0 soddisfa N_+, nel secondo caso v_0 soddisfa N_-: si ottiene comunque un assurdo, e il lemma resta così dimostrato.

---

[7] Di tale implicazione inversa a noi interesserà soprattutto l’implicazione contronominale: se $\mathcal{K}$ non è soddisfacibile, allora $\mathcal{K}$ non è finitamente soddisfacibile (e quindi esiste almeno un sottoinsieme finito di $\mathcal{K}$ che non è soddisfacibile).

Teorema 2.12.2 (“di compattezza”; Gödel, 1930)

Ogni insieme di clausole che sia finitamente soddisfacibile è soddisfacibile.

Dimostrazione — Sia $\{p_1,\ p_2,\ p_3,\ \dots,\ p_n,\ \dots\}$ l’insieme delle variabili proposizionali.

Dato l’insieme di clausole $\mathcal{K}$ finitamente soddisfacibile, definiamo induttivamente

$$\mathcal{K}_0 := \mathcal{K}\,;$$

e, per ogni $i \geq 1$,

$$\mathcal{K}_i := \begin{cases} \mathcal{K}_{i-1} \cup \{\{p_i\}\} & \text{se } \mathcal{K}_{i-1} \cup \{\{p_i\}\} \text{ è soddisfacibile;} \\ \mathcal{K}_{i-1} \cup \{\{\neg p_i\}\} & \text{altrimenti (}^{8}\text{)}\,. \end{cases}$$

Posto $\overline{\mathcal{K}} := \bigcup\limits_{i\in\mathbb{N}} \mathcal{K}_i$, dimostreremo che $\overline{\mathcal{K}}$ è soddisfacibile; poiché $\mathcal{K} \subset \overline{\mathcal{K}}$, ciò prova il teorema.

Per il lemma 2.12.1, ogni $\mathcal{K}_i$ è finitamente soddisfacibile. Di conseguenza, $\overline{\mathcal{K}}$ è finitamente soddisfacibile: se $\mathrm{F} = \{\mathrm{C}_1, \mathrm{C}_2, \dots, \mathrm{C}_s\}$ è un sottoinsieme finito di $\overline{\mathcal{K}}$, ogni C_i appartiene a un $\mathcal{K}_i$; ma i $\mathcal{K}_i$ sono inclusi uno nell’altro, quindi esiste un $\mathcal{K}_n$ a cui appartengono tutte le clausole C_i: in altre parole, $\mathrm{F} \subset \mathcal{K}_n$ e quindi F è soddisfacibile perché $\mathcal{K}_n$ è finitamente soddisfacibile.

Ora osserviamo che

(*) per ogni variabile proposizionale p_i, esattamente una delle due clausole $\{p_i\}$ e $\{\neg p_i\}$ appartiene a $\overline{\mathcal{K}}$.

Una almeno delle due clausole, infatti, appartiene a $\mathcal{K}_i$ (e quindi a $\overline{\mathcal{K}}$) per definizione di $\mathcal{K}_i$. Ma se a $\overline{\mathcal{K}}$ appartenessero entrambe allora $\overline{\mathcal{K}}$ non sarebbe finitamente soddisfacibile perché conterrebbe il sottoinsieme finito non soddisfacibile $\{\{p_i\}, \{\neg p_i\}\}$.

Siamo finalmente in grado di costruire una valutazione di verità v_0 che soddisfa $\overline{\mathcal{K}}$: per ogni variabile proposizionale p_i poniamo

$$v_0(p_i) := \begin{cases} 1 & \text{se } \{p_i\} \in \overline{\mathcal{K}}\,; \\ 0 & \text{altrimenti (}^{9}\text{)}\,. \end{cases}$$

[8] cioè se $\mathcal{K}_i \cup \{\{p_i\}\}$ è insoddisfacibile e quindi, per il lemma 2.12.1, $\mathcal{K}_i \cup \{\{\neg p_i\}\}$ è soddisfacibile

[9] cioè se $\{p_i\} \notin \overline{\mathcal{K}}$ e quindi, per la (*), $\{\neg p_i\} \in \overline{\mathcal{K}}$

Sia C una clausola che appartiene a $\overline{\mathcal{K}}$ e siano $\ell_1,\ \ell_2,\ \ell_3,\ \ldots,\ \ell_s$ i letterali che costituiscono C; dimostriamo che v_0 soddisfa C, cioè che $v_0(\ell_i) = 1$ per almeno un $\ell_i \in$ C. Se così non fosse, sarebbe $v_0(\ell_i) = 0$ (e quindi $v_0(\overline{\ell_i}) = 1$) per ogni i (abbiamo indicato con $\overline{\ell_i}$ il letterale “opposto” a ℓ_i, cioè: $\overline{\ell_i} := \neg p_i$ se $\ell_i = p_i$, $\overline{\ell_i} := p_i$ se $\ell_i = \neg p_i$). Quindi, per definizione di v_0, tutte le clausole della forma $\{\overline{\ell_i}\}$ apparterrebbero a $\overline{\mathcal{K}}$; poiché $\overline{\mathcal{K}}$ è finitamente soddisfacibile, dovrebbe esistere una valutazione di verità w che soddisfa

$$\{\mathrm{C}, \{\overline{\ell_1}\}, \{\overline{\ell_2}\}, \{\overline{\ell_3}\}, \ldots, \{\overline{\ell_s}\}\}.$$

Ma questo è assurdo: infatti, se w soddisfa $\{\overline{\ell_1}\},\ \{\overline{\ell_2}\},\ \{\overline{\ell_3}\},\ \ldots,\ \{\overline{\ell_s}\}$ non soddisfa nessuno dei letterali $\ell_1, \ell_2, \ell_3, \ldots, \ell_s$ e quindi non soddisfa C ($= \{\ell_1, \ell_2, \ell_3, \ldots, \ell_s)\}$.

Dunque v_0 soddisfa C, e per l’arbitrarietà di C in $\overline{\mathcal{K}}$ si è provato che v_0 soddisfa $\overline{\mathcal{K}}$, come si voleva dimostrare.

2.13 - Altri esempi ed esercizi.

Esempio 2.13.1

Verifichiamo che la seguente formula è una tautologia:

$$\alpha := (p \to \neg q) \to ((r \to \neg q) \to ((p \vee r) \to \neg q))\,.$$

Si tratta di dimostrare che $\neg\alpha$ è insoddisfacibile, cioè che è insoddisfacibile l’insieme di clausole associato a $\neg\alpha$.

In primo luogo scriviamo $\neg\alpha$ in FNC, ricordando che $\neg(\varphi \to \psi) \equiv \varphi \wedge \neg\psi$ per il corollario 2.5.2 :

$$\neg\alpha \equiv (p \to \neg q) \wedge \neg((r \to \neg q) \to ((p \vee r) \to \neg q)) \equiv$$

$$\equiv (p \to \neg q) \wedge ((r \to \neg q) \wedge \neg((p \vee r) \to \neg q)) \equiv (p \to \neg q) \wedge ((r \to \neg q) \wedge ((p \vee r) \wedge q))$$

Ricordando l’osservazione 2.3.2, $\neg\alpha$ si può scrivere senza ambiguità come

$$(p \to \neg q) \wedge (r \to \neg q) \wedge (p \vee r) \wedge q$$

ossia, ricordando i teoremi 2.4.1 e 2.3.1, come

$$(\neg p \vee \neg q) \wedge (\neg r \vee \neg q) \wedge (p \vee r) \wedge q$$

che è in FNC come si voleva.

L’insieme di clausole associato è

$$\{\{\neg p, \neg q\}, \{\neg r, \neg q\}, \{p, r\}, \{q\}\}\,.$$

Applichiamo l’algoritmo di Davis e Putnam.

Pivot q:

clausole non contenenti né q né $\neg q$: $\{p, r\}$;

$\text{Ris}_q(\{\neg p, \neg q\}, \{q\}) = \{\neg p\}$;

$\text{Ris}_q(\{\neg r, \neg q\}, \{q\}) = \{\neg r\}$;

$$\{\{p, r\}, \{\neg p\}, \{\neg r\}\}$$

Pivot p:

clausole non contenenti né p né $\neg p$: $\{\neg r\}$;

$\text{Ris}_p(\{p, r\}, \{\neg p\}) = \{r\}$;

$$\{\{\neg r\}, \{r\}\}$$

Pivot r:

clausole non contenenti né r né $\neg r$: nessuna ;

$\text{Ris}_r(\{\neg r\}, \{r\}) = [\,]$;

$$\{[\,]\}$$

Si può dunque concludere che l’insieme di clausole considerato è insoddisfacibile, cioè che è insoddisfacibile $\neg\alpha$ e quindi α è una tautologia, come si voleva dimostrare.

Esempio 2.13.2

Verifichiamo se la seguente formula è soddisfacibile:

$$\alpha := (p \to (q \to r)) \wedge \neg((p \to q) \to (p \to r))\,.$$

In primo luogo scriviamo α in FNC:

$$\alpha \equiv (\neg p \vee (\neg q \vee r)) \wedge ((p \to q) \wedge \neg(p \to r)) \equiv$$
$$\equiv (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q) \wedge (p \wedge \neg r) \equiv$$
$$\equiv (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q) \wedge p \wedge \neg r$$

L’insieme di clausole associato è

$$\{\{\neg p, \neg q, r\}, \{\neg p, q\}, \{p\}, \{\neg r\}\}\,.$$

Applichiamo l’algoritmo di Davis e Putnam.

Pivot p:

clausole non contenenti né p né $\neg p$: $\{\neg r\}$;

$\text{Ris}_p(\{\neg p, \neg q, r\}, \{p\}) = \{\neg q, r\}$;

$\text{Ris}_p(\{\neg p, q\}, \{p\}) = \{q\}$;

$$\{\{\neg r\}, \{\neg q, r\}, \{q\}\}$$

Pivot q:

clausole non contenenti né q né $\neg q$: $\{\neg r\}$;

$\text{Ris}_q(\{\neg q, r\}, \{q\}) = \{r\}$;

$$\{\{\neg r\}, \{r\}\}$$

Pivot r:

clausole non contenenti né r né $\neg r$: nessuna ;

$\text{Ris}_r(\{\neg r\}, \{r\}) = [\,]$;

$$\{[\,]\}$$

Si può dunque concludere che l’insieme di clausole considerato è insoddisfacibile, cioè che è insoddisfacibile la formula α.

Esempio 2.13.3

Controlliamo se il seguente insieme $\mathcal{K}$ di clausole è soddisfacibile:

$$\mathcal{K} := \{\{a, b, s\}, \{a, c, k\}, \{\neg a, \neg b, s\}, \{\neg a, b, \neg s\}, \{\neg b, k, s\}, \{\neg b, s\}, \{\neg c, k\}, \{c, k\}, \{b, s\}, \{\neg h, \neg k\}, \{c, \neg h\}, \{b, \neg s\}, \{h, \neg k\}\} .$$

Osserviamo in primo luogo che è possibile eliminare alcune clausole perché contengono propriamente altre clausole (cfr. oss. 2.9.7). Eliminiamo dunque $\{a, b, s\}$ che contiene propriamente $\{b, s\}$, $\{a, c, k\}$ che contiene propriamente $\{c, k\}$, $\{\neg a, b, \neg s\}$ che contiene propriamente $\{b, \neg s\}$, $\{\neg b, k, s\}$ che contiene propriamente $\{\neg b, s\}$ e $\{\neg a, \neg b, s\}$ che contiene propriamente $\{\neg b, s\}$.

Siamo ricondotti a considerare l’insieme di clausole

$$\mathcal{K}_1 := \{\{\neg b, s\}, \{\neg c, k\}, \{c, k\}, \{b, s\}, \{\neg h, \neg k\}, \{c, \neg h\}, \{b, \neg s\}, \{h, \neg k\}\}$$

al quale applicheremo l’algoritmo di Davis e Putnam.

Pivot b:

clausole non contenenti né b né $\neg b$: $\{\neg c, k\}$, $\{c, k\}$, $\{\neg h, \neg k\}$, $\{c, \neg h\}$, $\{h, \neg k\}$;

$\mathrm{Ris}_b(\{\neg b,\, s\},\, \{b,\, s\}) = \{s\}$;

$\mathrm{Ris}_b(\{\neg b,\, s\},\, \{b,\, \neg s\}) = \{s,\, \neg s\}$ (si sopprime perché tautologia) ;

$$\{\{s\},\, \{\neg c, k\},\, \{c, k\},\, \{\neg h, \neg k\},\, \{c, \neg h\},\, \{h, \neg k\}\}$$

Pivot c:

clausole non contenenti né c né $\neg c$: $\{s\}$, $\{\neg h, \neg k\}$, $\{h, \neg k\}$;

$\mathrm{Ris}_c(\{\neg c,\, k\},\, \{c,\, k\}) = \{k\}$;

$\mathrm{Ris}_c(\{\neg c,\, k\},\, \{c,\, \neg h\}) = \{\neg h,\, k\}$ (si sopprime perché contiene propriamente $\{k\}$;

$$\{\{k\},\, \{s\},\, \{\neg h, \neg k\},\, \{h, \neg k\}\}$$

Pivot h:

clausole non contenenti né h né $\neg h$: $\{k\}$, $\{s\}$;

$\mathrm{Ris}_h(\{\neg h,\, \neg k\},\, \{h,\, \neg k\}) = \{\neg k\}$;

$$\{\{\neg k\},\, \{k\},\, \{s\}\}$$

Pivot k:

clausole non contenenti né k né $\neg k$: $\{s\}$;

$\mathrm{Ris}_k(\{\neg k\},\, \{k\}) = [\,]$;

$$\{\{s\},\, [\,]\}$$

Poiché si è ottenuta la clausola vuota, l’insieme di clausole dato non è soddisfacibile!

Esercizio 2.13.4

Verificare se la seguente formula è soddisfacibile:

$$(p \to q) \land \neg((r \land s) \to \neg(q \to p))\,.$$

Esercizio 2.13.5

Stabilire se

$$(p \to q) \land (r \to q) \vDash ((p \lor t) \to q) \lor \neg r\,.$$

Suggerimento: Si applichi il contenuto dell’osservazione 2.3.5.

Esempio 2.13.6

Forse non tutti sanno che il castello di Vincigliata, alle porte di Firenze, si dice sia infestato dai fantasmi, soprattutto di notte. Il coraggioso Andrea è forse disponibile a trascorrere la notte al castello di Vincigliata. Incontrerà i fantasmi? Ma, secondo voi, esistono veramente i fantasmi?

Forse potremmo tutti concordare sulla seguente affermazione:

(i) se non esistono i fantasmi, allora è falso che: se Andrea trascorre la notte nel castello di Vincigliata, Andrea incontra i fantasmi;

(in effetti: come può Andrea incontrare i fantasmi, se essi non esistono?)

Bene, allora siamo tutti d’accordo che la (i) è vera, e possiamo accettarla come un dato di fatto. Ma ecco una notizia dell’ultim’ora: Andrea ha dichiarato che non intende affatto trascorrere la notte nel castello di Vincigliata. Beh, anche questo è un dato di fatto e non possiamo che accettarlo come tale... e dunque:

(ii) Andrea non trascorre la notte nel castello di Vincigliata.

L’aspetto clamoroso di questo esempio è che dalle affermazioni (i) e (ii), sulle quali non possiamo non concordare ([10]), segue come conseguenza logica la

(iii) I fantasmi esistono.

È facile controllare quanto affermato sopra. Introduciamo opportune variabili proposizionali:

$f :=$ i fantasmi esistono;
$v :=$ Andrea trascorre la notte nel castello di Vincigliata;
$i :=$ Andrea incontra i fantasmi.

La premessa (i) si può esprimere così: $(\neg f) \to \neg(v \to i)$; la premessa ($ii$) si esprime ovviamente come $\neg v$; il fatto che la (iii) sia conseguenza logica della (i) e della (ii) si traduce nella circostanza che non è soddisfacibile la formula ottenuta congiungendo la (i), la (ii) e la negazione della (iii), cioè la

$$((\neg f) \to \neg(v \to i)) \wedge (\neg v) \wedge (\neg f)\,.$$

Scriviamo questa formula in FNC:

$$(f \vee (v \wedge \neg i)) \wedge (\neg v) \wedge (\neg i) \equiv (f \vee v) \wedge (f \vee \neg i) \wedge (\neg v) \wedge (\neg f)$$

e poi come insieme di clausole:

$$\{\{f, v\}, \{f, \neg i\}, \{\neg v\}, \{\neg f\}\}$$

[10] O forse, alla fine, qualche dubbio ci verrà. E siccome per Andrea garantisco io, dovremo ripensare al valore di verità della (i): fatelo!

Applichiamo l’algoritmo di Davis e Putnam.

Pivot f:

clausole non contenenti né f né $\neg f$: $\{\neg v\}$;

$\mathrm{Ris}_f(\{f, v\}, \{\neg f\}) = \{v\}$;

$\mathrm{Ris}_f(\{f, \neg i\}, \{\neg f\}) = \{\neg i\}$;

$$\{\{v\}, \{\neg i\}, \{\neg v\}\}$$

Pivot v:

clausole non contenenti né v né $\neg v$: $\{\neg i\}$;

$\mathrm{Ris}_v(\{v\}, \{\neg v\}) = [\,]$;

$$\{[\,], \{\neg i\}\}$$

Poiché si è ottenuta la clausola vuota, l’insieme di clausole dato non è soddisfacibile! Dunque la (iii) è conseguenza logica della (i) e della (ii). Partendo da premesse (apparentemente?) del tutto condivisibili, abbiamo dimostrato che i fantasmi esistono!

$\boxed{\text{Esempio 2.13.7}}$

Un altro delitto al castello Glomgold! Questa volta la vittima è il figlio maggiore del duca. Pochi fatti ma chiari sono stati appurati con certezza:

(i) se il delitto non è avvenuto nella stanza del biliardo, allora l’assassino è il maggiordomo;

(ii) o il maggiordomo è innocente, oppure l’arma del delitto è un candelabro;

(iii) l’arma del delitto non è un candelabro.

Ci chiediamo, come al solito, se l’assassino sia il maggiordomo. Per formalizzare il nostro ragionamento, abbiamo bisogno di introdurre tre variabili proposizionali, cioè:

$p :=$ “l’assassino è il maggiordomo” ;
$q :=$ “il delitto è avvenuto nella stanza del biliardo” ;
$r :=$ “l’arma del delitto è un candelabro” .

I risultati delle indagini si possono esprimere nel modo seguente:

(i) diventa: $\neg q \to p$;
(ii) diventa: $\neg p \vee r$;
(iii) diventa: $\neg r$.

Vogliamo stabilire se la colpevolezza del maggiordomo è conseguenza logica delle (i), (ii) e (iii), cioè se $\{\neg q \to p,\ \ \neg p \vee r,\ \ \neg r\} \vDash p$
ossia se

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \vDash p$$

ossia se

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \wedge \neg p \quad \text{è} \quad \text{insoddisfacibile.}$$

Utilizziamo l’algoritmo di Davis e Putnam, lasciando per esercizio al lettore la tecnica (alternativa) di tabulazione dei valori di verità.

Per ridurre in FNC la formula

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \wedge \neg p$$

basta osservare che $\neg q \to p \;\equiv\; \neg(\neg q) \vee p \;\equiv\; q \vee p$.

La formula considerata si scrive dunque in FNC

$$(q \vee p) \wedge (\neg p \vee r) \wedge \neg r \wedge \neg p$$

e come insieme di clausole $\{\{p, q\}, \{\neg p, r\}, \{\neg r\}, \{\neg p\}\}$.

Applichiamo l’algoritmo di Davis e Putnam.

Pivot p:

clausole non contenenti né p né $\neg p$: $\{\neg r\}$;
$\mathrm{Ris}_p(\{p, q\}, \{\neg p, r\}) = \{q, r\}$;
$\mathrm{Ris}_p(\{p, q\}, \{\neg p\}) = \{q\}$;

$$\{\{\neg r\}, \{q, r\}, \{q\}\}.$$

Pivot r:

clausole non contenenti né r né $\neg r$: $\{q\}$;
$\mathrm{Ris}_r(\{\neg r\}, \{q, r\}) = \{q\}$;

$$\{\{q\}\}.$$

Pivot q:

$$\{\ \}.$$

L’insieme vuoto di clausole è soddisfacibile. Procedendo a ritroso, vediamo che una valutazione di verità che soddisfi la formula

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \wedge \neg p$$

può essere costruita ponendo

$$v(q) := 1,\ \ v(r) := 0,\ \ v(p) := 0.$$

Una possibile ricostruzione dei fatti coerente con i dati acquisiti dalla polizia è dunque che il delitto sia avvenuto nella stanza del biliardo con un'arma che non è il candelabro ad opera di un assassino che non è il maggiordomo.

Ma possiamo fare meglio: possiamo addirittura dimostrare che il maggiordomo è inocente! Ossia, che

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \models \neg p$$

cioè che

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \wedge p \quad \text{è} \quad \text{insoddisfacibile.}$$

Ragionando come sopra, ci si riconduce infatti all'insieme di clausole

$$\{\{p, q\}, \{\neg p, r\}, \{\neg r\}, \{p\}\}$$

che si verifica essere insoddisfacibile con l'algoritmo di Davis e Putnam.

Possiamo anche dimostrare che il delitto è avvenuto nella stanza del biliardo, cioè che

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \models q$$

ossia che

$$(\neg q \to p) \wedge (\neg p \vee r) \wedge \neg r \wedge \neg q \quad \text{è} \quad \text{insoddisfacibile.}$$

Questa volta ci si riconduce all'insieme di clausole $\{\{p, q\}, \{\neg p, r\}, \{\neg r\}, \{\neg q\}\}$ che anche in questa occasione con l'algoritmo di Davis e Putnam risulta insoddisfacibile.

Esempio 2.13.8

Ennesimo delitto al castello Glomgold! Questa volta la vittima è l'unica figlia femmina del duca. Esaminiamo i fatti appurati dagli inquirenti:

(i) se il maggiordomo è innocente, il delitto è avvenuto in cucina;
(ii) se il delitto è avvenuto a mezzanotte, il maggiordomo è colpevole;
(iii) la domestica è innocente se e solo se l'arma del delitto è un candelabro;
(iv) l'arma del delitto è un candelabro o il delitto è avvenuto in biblioteca;
(v) o il maggiordomo o la domestica è colpevole;
(vi) il delitto è avvenuto in biblioteca.

Ci chiediamo, come al solito, se l'assassino sia il maggiordomo.

Per formalizzare il nostro ragionamento, abbiamo bisogno di introdurre sei variabili proposizionali, cioè:

$a :=$ “il maggiordomo è colpevole” ;
$b :=$ “la cameriera è colpevole” ;
$c :=$ “il delitto è avvenuto a mezzanotte” ;
$d :=$ “l’arma del delitto è un candelabro” .
$e :=$ “il delitto è avvenuto in cucina” ;
$f :=$ “il delitto è avvenuto in biblioteca” .

I risultati delle indagini si possono esprimere nel modo seguente:

(i) diventa: $\neg a \to e$;
(ii) diventa: $c \to a$;
(iii) diventa: $(\neg b \to d) \wedge (d \to \neg b)$;
(iv) diventa: $d \vee f$;
(v) diventa: $a \vee b$;
(vi) diventa: f .

A ciò bisogna aggiungere l’ovvia considerazione che $(e \to \neg f) \wedge (f \to \neg e)$ (peraltro equivalente a $e \to \neg f$ essendo $f \to \neg e$ l’implicazione contronominale di $e \to \neg f$).

Vogliamo stabilire se il maggiordomo è colpevole. Ciò equivale al fatto che sia

$$\{\neg a \to e\,,\ \ c \to a\,,\ \ (\neg b \to d) \wedge (d \to \neg b)\,,\ \ d \vee f,\ \ a \vee b\,,\ \ f,\ \ e \to \neg f\} \models a$$

ovvero

$$(\neg a \to e) \wedge (c \to a) \wedge (\neg b \to d) \wedge (d \to \neg b) \wedge (d \vee f) \wedge (a \vee b) \wedge f \wedge (e \to \neg f) \models a$$

ossia che

$$(\neg a \to e) \wedge (c \to a) \wedge (\neg b \to d) \wedge (d \to \neg b) \wedge (d \vee f) \wedge (a \vee b) \wedge f \wedge (e \to \neg f) \wedge \neg a$$

è insoddisfacibile.

Scriviamo l’ultima formula in FNC e poi come insieme di clausole:

$$(a \vee e) \wedge (\neg c \vee a) \wedge (b \vee d) \wedge (\neg d \vee \neg b) \wedge (d \vee f) \wedge (a \vee b) \wedge f \wedge (\neg e \vee \neg f) \wedge \neg a$$

$$\{\{a, e\}, \{a, \neg c\}, \{b, d\}, \{\neg d, \neg b\}, \{d, f\}, \{a, b\}, \{f\}, \{\neg e, \neg f\}, \{\neg a\}\}$$

Applichiamo l’algoritmo di Davis e Putnam.

Pivot a:

clausole non contenenti né a né $\neg a$: $\{b, d\}, \{\neg d, \neg b\}, \{d, f\}, \{f\}, \{\neg e, \neg f\}$;

$\text{Ris}_a(\{a, e\}, \{\neg a\}) = \{e\}$;

$\text{Ris}_a(\{a, \neg c\}, \{\neg a\}) = \{\neg c\}$;

$\text{Ris}_a(\{a, b\}, \{\neg a\}) = \{b\}$;

$$\{\{b, d\}, \{\neg d, \neg b\}, \{d, f\}, \{f\}, \{\neg e, \neg f\}, \{e\}, \{\neg c\}, \{b\}\}$$

Pivot b:

clausole non contenenti né b né $\neg b$: $\{d, f\}, \{f\}, \{\neg e, \neg f\}, \{e\}, \{\neg c\}$;

$\text{Ris}_b(\{b, d\}, \{\neg d, \neg b\}) = \{d, \neg d\}$ (si sopprime perché è una tautologia);

$\text{Ris}_b(\{b\}, \{\neg d, \neg b\}) = \{\neg d\}$;

$$\{\{d, f\}, \{f\}, \{\neg e, \neg f\}, \{e\}, \{\neg c\}, \{\neg d\}\}$$

Pivot c:

$$\{\{d, f\}, \{f\}, \{\neg e, \neg f\}, \{e\}, \{\neg d\}\}$$

Pivot d:

clausole non contenenti né d né $\neg d$: $\{f\}, \{\neg e, \neg f\}, \{e\}$;
$\text{Ris}_d(\{d, f\}, \{\neg d\}) = \{\neg d\}$ (si sopprime perché già presente);

$$\{\{f\}, \{\neg e, \neg f\}, \{e\}\}$$

Pivot e:

clausole non contenenti né e né $\neg e$: $\{f\}$;
$\text{Ris}_e(\{\neg e, \neg f\}, \{e\}) = \{\neg f\}$;

$$\{\{f\}, \{\neg f\}\}$$

Pivot f:

clausole non contenenti né f né $\neg f$: nessuna;
$\text{Ris}_f(\{f\}, \{\neg f\}) = [\,]$;

$$\{[\,]\}$$

L’insieme di clausole è insoddisfacibile, dunque la formula associata non è soddisfacibile e in effetti il maggiordomo è colpevole.

Esercizio 2.13.9

Nello stesso scenario dell’esempio 2.13.8, si stabilisca se è possibile dedurre l’innocenza della cameriera; in caso contrario, si descriva un’interpretazione dei fatti coerente coi risultati delle indagini nella quale la cameriera è colpevole.

$\boxed{\text{Esercizio 2.13.10}}$

Nello stesso scenario dell’esempio 2.13.8, si stabilisca se è possibile dedurre la colpevolezza della cameriera; in caso contrario, si descriva un’interpretazione dei fatti coerente coi risultati delle indagini nella quale la cameriera è innocente.

$\boxed{\text{Esercizi}}$

Si modifichi lo scenario dell’esercizio 2.13.8 sostituendo il fatto (vi) rispettivamente con

$\boxed{2.13.11}$ (vi) il delitto non è avvenuto in biblioteca.
$\boxed{2.13.12}$ (vi) il delitto è avvenuto nel salone a mezzanotte.
$\boxed{2.13.13}$ (vi) il delitto è avvenuto a mezzogiorno e l’arma è un grosso sasso.

Caso per caso, si stabilisca se si può dedurre la colpevolezza o l’innocenza del maggiordomo e/o della cameriera.

3.- LOGICA DEI PREDICATI

3.1 - Arricchire il linguaggio.

Anche se la logica proposizionale contiene già gli elementi essenziali del processo deduttivo, il suo linguaggio risulta incapace di esprimere proprietà degli elementi di un insieme e di affermare che determinate caratteristiche valgono per qualche oggetto (o per tutti gli oggetti che consideriamo). Così alcuni ragionamenti elementari non possono venire tradotti (e quindi nemmeno giustificati o confutati) nel linguaggio della logica proposizionale.

Ad esempio, non si può esprimere con i mezzi della logica proposizionale il sillogismo (ii) dell’esempio 1.1.1; e nemmeno un (banale) ragionamento del tipo: ogni numero è pari o dispari; 5 non è pari, dunque 5 è dispari.

La logica dei predicati (o “logica del primo ordine”), che studiamo in questo capitolo, considera al posto delle varibili proposizionali (che abbiamo utilizzato nel capitolo 2) degli oggetti più complessi atti ad esprimere relazioni e proprietà di elementi scelti in opportuni ambienti (tipicamente strutture matematiche, ma con qualche libertà anche scenari più fantasiosi e “umani”). Il valore di verità non è più assegnato apoditticamente ipotizzando per le variabili ogni scelta possibile, ma è legato alla “interpretazione” (questa parola diventerà un termine tecnico!) della formula a cui assegnarlo.

Inevitabilmente, alla luce di quanto sopra accennato, ci serviranno alfabeti più articolati di quello introdotto nella sez. 2.1. E il plurale non è casuale, perché la maggiore ricchezza della logica dei predicati deriva anche dal fatto che per ogni situazione si può considerare un diverso linguaggio, fondato su un diverso alfabeto.

Procediamo gradualmente, iniziando con la parte descrittiva dei linguaggi. Un insieme A si dice un *alfabeto per un linguaggio descrittivo di primo livello* (o anche, brevemente, un *alfabeto descrittivo di primo livello*) se i suoi elementi sono:

(i) infiniti simboli $x_0, x_1, x_2, \ldots, x_n, \ldots$ ancora una volta “etichettati” con i numeri naturali, che chiameremo *variabili individuali* (o, più semplicemente, *variabili*);

(ii) un numero finito (ma arbitrario) di simboli $c_0, c_1, c_2, \ldots, c_k$ detti “*simboli di costante* (*individuale*)” ;

(iii) un numero finito (ma arbitrario) di simboli $f_0, f_1, f_2, \ldots, f_s$ detti “*simboli di funzione*”, a ciascuno dei quali resta associato un numero intero positivo detto *arietà* del simbolo di funzione; talvolta si scrive $f_i^{(j_i)}$ per indicare che al simbolo di funzione f_i è associata la arietà j_i (o, come spesso si dice, che il simbolo di funzione f_i ha arietà j_i) ;

(iv) le parentesi (,) e la virgola , .

Il numero k dei simboli di costante, il numero s dei simboli di funzione e la sequenza delle rispettive arietà costituiscono il *tipo* dell’alfabeto A.

Analogamente a quanto fatto nella sez. 2.1 (e seguendo peraltro una terminologia del tutto generale), diremo *parola* sull’alfabeto descrittivo A sopra individuato ogni sequenza finita di elementi di A: gli elementi di una parola vengono usualmente scritti di seguito senza altri segni grafici. Anche in questo caso non ci interessano tutte le parole sull’alfabeto, ma soltanto alcune con particolari caratteristiche strutturali.

Una parola t su A si dice un *termine* (*su* A) se esiste una sequenza finita t_0, t_1, t_2, $\ldots$, t_k di parole su A tali che:

(i) $t = t_k$

e

(ii) per ogni $h \leq k$ la parola t_h è una variabile individuale oppure è un simbolo di costante individuale oppure è della forma

$$f_i^{(j_i)} t_{s_1} t_{s_2} \ldots t_{s_{j_i}}$$

dove j_i è l’arietà di f_i e $s_1, s_2, \ldots, s_{j_i} < h$. Generalmente, per maggiore chiarezza, si utilizzano parentesi e virgola e si scrive

$$f_i\left(t_{s_1}, t_{s_2}, \ldots, t_{s_{j_i}}\right) \qquad \text{anziché} \qquad f_i^{(j_i)} t_{s_1} t_{s_2} \ldots t_{s_{j_i}}$$

La sequenza finita t_0, t_1, t_2, $\ldots$, t_k si dice una *costruzione* del termine t.

Il minimo numero naturale k per cui esiste una costruzione $t_0, t_1, t_2, \ldots, t_k$ del termine t si dice *lunghezza* di t.

Un termine si dice *chiuso* se in esso non compaiono variabili, *aperto* altrimenti.

L’insieme dei termini costituisce il *linguaggio descrittivo di primo livello* associato all’alfabeto A. I termini servono ad individuare (in modo articolato e flessibile) gli elementi dello scenario in cui vogliamo lavorare. Esistono a priori infinite possibili situazioni nelle quali possiamo ambientare i nostri ragionamenti; avere individuato un alfabeto descrittivo di primo livello, però, restringe già il campo.

Sia A un alfabeto per un linguaggio descrittivo di primo livello. Si dice *struttura adeguata ad* A ogni terna ordinata (S, $\mathcal{C}$, $\mathcal{F}$) tale che (per opportuni $k, w \in \mathbb{N}$)

– S è un insieme non vuoto (detto il *sostegno* della struttura);

– $\mathcal{C}$ è un sottoinsieme di S formato da k elementi (che si dicono *costanti*);

– $\mathcal{F}$ è un insieme $\{\overline{f}_0, \overline{f}_1, \overline{f}_2, \ldots, \overline{f}_w\}$ di w funzioni tali che per $i := 1, \ldots, w$ $\overline{f}_i$ è una funzione $\mathrm{S}^{j_i} \to \mathrm{S}$ dove j_i è la arietà di f_i .

Sia A un alfabeto per un linguaggio descrittivo di primo livello, e sia $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F})$ una struttura adeguata ad A. Si dice *interpretazione* di A su Σ ogni funzione da A in $\mathcal{C} \cup \mathcal{F}$ che associa ad ogni simbolo di costante un elemento di $\mathcal{C}$ e ad ogni simbolo di funzione f un elemento $\overline{f}$ di $\mathcal{F}$ “rispettando le arietà”, ossia con la condizione che $\overline{f}$ sia una funzione $\mathrm{S}^j \to \mathrm{S}$ dove j è la arietà di f.

Sia A un alfabeto per un linguaggio descrittivo di primo livello, e sia $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F})$ una struttura adeguata ad A. Se è data una interpretazione di A su Σ, ad ogni termine chiuso t su A resta associato in modo univoco un elemento di S (che diremo *valore* di t nell’interpretazione considerata): quello che si ottiene sostituendo ad ogni simbolo di costante la costante corrispondente, ad ogni simbolo di funzione la funzione corrispondente e calcolando i valori delle funzioni così ottenute. Ai termini aperti, invece, con questo procedimento non si riesce in generale ad associare un elemento di S.

Esempio 3.1.1

Sia A un alfabeto per un linguaggio descrittivo di primo livello con due simboli di costante (c_0 e c_1) e due simboli di funzione (f e g, entrambi di arietà 2). Sia

$$t_1 := f(c_0, g(c_1, c_1))\,;$$
$$t_2 := f(x_0, g(x_0, c_1))\,.$$

Il termine t_1 è chiuso, mentre il termine t_2 è aperto. Vediamo alcuni esempi di strutture adeguate a A ed alcune possibili interpretazioni di A su esse.

$\boxed{\text{Caso 1}}$ $\mathrm{S} := \mathbb{Z}$, $\mathcal{C} := \{0, 1\}$, $\mathcal{F} := \{+, \cdot\}$.
Una possibile interpretazione associa 0 a c_0, 1 a c_1, $+$ a f e $\cdot$ a g. Sotto tale interpretazione, il valore di t_1 è $0 + (1 \cdot 1)$, cioè 1, mentre al termine t_2 possiamo associare la (comoda, e utile) scrittura simbolica $2x_0$ (osservando che $x_0 + (x_0 \cdot 1) = 2x_0$), ma non un preciso elemento di $\mathbb{Z}$.
Un'altra possibile interpretazione associa 1 a c_0, 0 a c_1 e poi ancora $+$ a f e $\cdot$ a g. Sotto tale interpretazione, il valore di t_1 è $1 + (0 \cdot 0)$, cioè ancora 1, mentre al termine t_2 possiamo associare la (comoda e utile) scrittura simbolica x_0 (osservando che $x_0 + (x_0 \cdot 0) = x_0$), ma non un preciso elemento di $\mathbb{Z}$.
Una terza possibile interpretazione associa 1 a c_0, 0 a c_1, $\cdot$ a f e $+$ a g. Sotto tale interpretazione, il valore di t_1 è $1 \cdot (0 + 0)$, cioè 0, mentre al termine t_2 possiamo associare la (comoda, e utile) scrittura simbolica x_0^2 (osservando che $x_0 \cdot (x_0 + 0) = x_0^2$), ma non un preciso elemento di $\mathbb{Z}$.

$\boxed{\text{Caso 2}}$ $\mathrm{S} := \mathbb{N}$, $\mathcal{C} := \{0, 1\}$, $\mathcal{F} := \{\mathrm{MCD}, \mathit{mcm}\}$.
Una possibile interpretazione associa 0 a c_0, 1 a c_1, MCD a f e *mcm* a g. Sotto tale interpretazione, il valore di t_1 è $\mathrm{MCD}(0, \mathit{mcm}(1, 1))$, cioè 1, mentre al termine t_2 possiamo associare il simbolo x_0 (osservando che $\mathrm{MCD}(x_0, \mathit{mcm}(x_0, 1)) = \mathrm{MCD}(x_0, x_0) = x_0$), ma non un preciso elemento di $\mathbb{Z}$.
Un'altra possibile interpretazione associa 1 a c_0, 0 a c_1 e poi ancora MCD a f e *mcm* a g. Sotto tale interpretazione, il valore di t_1 è $\mathrm{MCD}(1, \mathit{mcm}(0, 0))$, cioè ancora 1, mentre al termine t_2 possiamo associare il simbolo x_0 (osservando che $\mathrm{MCD}(x_0, \mathit{mcm}(x_0, 0)) = \mathrm{MCD}(x_0, 0) = x_0$), ma non un preciso elemento di $\mathbb{Z}$.
Una terza possibile interpretazione associa 1 a c_0, 0 a c_1, *mcm* a f e MCD a g. Sotto tale interpretazione, il valore di t_1 è $\mathit{mcm}(1, \mathrm{MCD}(0, 0))$, cioè 0, mentre al termine t_2 possiamo associare il simbolo x_0 (osservando che $\mathit{mcm}(x_0, \mathrm{MCD}(x_0, 0)) = \mathit{mcm}(x_0, x_0) = x_0$), ma non un preciso elemento di $\mathbb{Z}$.

$\boxed{\text{Caso 3}}$ $\mathrm{S} := \mathbb{Z}_8$, $\mathcal{C} := \{[2], [5]\}$, $\mathcal{F} := \{+, \cdot\}$.
Una possibile interpretazione associa $[2]$ a c_0, $[5]$ a c_1, $+$ a f e $\cdot$ a g. Sotto tale interpretazione, il valore di t_1 è $[2] + ([5] \cdot [5])$, cioè $[3]$, mentre al termine t_2 possiamo associare la (comoda, e utile) scrittura simbolica $x_0 + (x_0 \cdot [5])$, ma non un preciso elemento di $\mathbb{Z}_8$.
Un'altra possibile interpretazione associa $[5]$ a c_0, $[2]$ a c_1, $\cdot$ a f e $+$ a g. Sotto tale interpretazione, il valore di t_1 è $[5] \cdot ([2] + [2])$, cioè $[4]$, mentre al termine t_2 possiamo associare la (comoda, e utile) scrittura simbolica $x_0 \cdot (x_0 + [2])$, ma non un preciso elemento di $\mathbb{Z}_8$.

Sia A un alfabeto per un linguaggio descrittivo di primo livello, sia $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F})$ una struttura adeguata ad A e sia data una interpretazione di A su Σ. Sia $\mathcal{X}$ l'insieme delle variabili individuali di S. Si dice *assegnazione* (all'interno della interpretazione data) ogni funzione $\mathcal{X} \to \mathrm{S}$.

La scelta di una assegnazione ci consente di associare un elemento di S a ciascun termine su A; notiamo esplicitamente che, nel caso dei termini aperti, l'elemento associato dipenderà in generale dalla assegnazione scelta. Precisamente:

Sia A un alfabeto per un linguaggio descrittivo di primo livello, sia $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F})$ una struttura adeguata ad A, sia data una interpretazione di A su Σ e sia a_0 un'assegnazione all'interno di tale interpretazione. Ad ogni termine t su A resta associato in modo univoco un elemento di S (che diremo ancora *valore* di t nell'interpretazione data e nell'assegnazione considerata): quello che si ottiene sostituendo ad ogni simbolo di costante la costante corrispondente, ad ogni variabile individuale la sua immagine mediante a_0, ad ogni simbolo di funzione la funzione corrispondente e calcolando i valori delle funzioni così ottenute.

Esempio 3.1.2

Riprendiamo in considerazione l’alfabeto A considerato nell’esempio 3.1.1 e le varie interpretazioni lì esaminate, per mostrare come la scelta di assegnazioni consente di associare un elemento della struttura anche ai termini aperti. Ricordiamo che A comprende due simboli di costante (c_0 e c_1) e due simboli di funzione (f e g, entrambi di arietà 2), e che abbiamo posto $t_2 := f(x_0, g(x_0, c_1))$.

Caso 1 $\mathrm{S} := \mathbb{Z}$, $\mathcal{C} := \{0, 1\}$, $\mathcal{F} := \{+, \cdot\}$.

Una possibile interpretazione associa 0 a c_0, 1 a c_1, $+$ a f e $\cdot$ a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica $x_0 + (x_0 \cdot 1)$. Scegliendo un’assegnazione che porta x_0 in 0, potremo associare a t_2 l’elemento 0 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in 1, potremo associare a t_2 l’elemento 2 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in -1, potremo associare a t_2 l’elemento -2 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in 2, potremo associare a t_2 l’elemento 4 di $\mathbb{Z}$; e così via…

Un’altra possibile interpretazione associa 1 a c_0, 0 a c_1 e poi ancora $+$ a f e $\cdot$ a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica $x_0 + (x_0 \cdot 0)$. Scegliendo un’assegnazione che porta x_0 in 0, potremo associare a t_2 l’elemento 0 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in 1, potremo associare a t_2 l’elemento 1 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in -1, potremo associare a t_2 l’elemento -1 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in 2, potremo associare a t_2 l’elemento 2 di $\mathbb{Z}$; e così via…

Una terza possibile interpretazione associa 1 a c_0, 0 a c_1, $\cdot$ a f e $+$ a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica $x_0 \cdot (x_0 + 0)$. Scegliendo un’assegnazione che porta x_0 in 0, potremo associare a t_2 l’elemento 0 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in 1, potremo associare a t_2 l’elemento 1 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in -1, potremo associare a t_2 l’elemento 1 di $\mathbb{Z}$; scegliendo un’assegnazione che porta x_0 in 2, potremo associare a t_2 l’elemento 4 di $\mathbb{Z}$; e così via…

Caso 2 $\mathrm{S} := \mathbb{N}$, $\mathcal{C} := \{0, 1\}$, $\mathcal{F} := \{\mathrm{MCD}, \mathit{mcm}\}$.

Una possibile interpretazione associa 0 a c_0, 1 a c_1, MCD a f e *mcm* a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica x_0. Scegliendo un’assegnazione che porta x_0 in 0, potremo associare a t_2 l’elemento 0 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 1, potremo associare a t_2 l’elemento 1 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 2, potremo associare a t_2 l’elemento 2 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 3, potremo associare a t_2 l’elemento 3 di $\mathbb{N}$; e così via…

Un’altra possibile interpretazione associa 1 a c_0, 0 a c_1 e poi ancora MCD a f e *mcm* a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica x_0. Scegliendo un’assegnazione che porta x_0 in 0, potremo associare a t_2 l’elemento 0 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 1, potremo associare a t_2 l’elemento 1 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 2, potremo associare a t_2 l’elemento 2 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 3, potremo associare a t_2 l’elemento 3 di $\mathbb{N}$; e così via…

Una terza possibile interpretazione associa 1 a c_0, 0 a c_1, *mcm* a f e MCD a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica x_0. Scegliendo un’assegnazione che porta x_0 in 0, potremo associare a t_2 l’elemento 0 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 1, potremo associare a t_2 l’elemento 1 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 2, potremo associare a t_2 l’elemento 2 di $\mathbb{N}$; scegliendo un’assegnazione che porta x_0 in 3, potremo associare a t_2 l’elemento 3 di $\mathbb{N}$; e così via…

$\boxed{\text{Caso 3}}$ $\mathrm{S} := \mathbb{Z}_8$, $\mathcal{C} := \{[2],[5]\}$, $\mathcal{F} := \{+, \cdot\}$.

Una possibile interpretazione associa [2] a c_0, [5] a c_1, $+$ a f e $\cdot$ a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica $x_0 + (x_0 \cdot [5])$. Scegliendo un’assegnazione che porta x_0 in [0], potremo associare a t_2 l’elemento [0] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [1], potremo associare a t_2 l’elemento [6] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [2], potremo associare a t_2 l’elemento [5] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [3], potremo associare a t_2 l’elemento [4] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [4], potremo associare a t_2 l’elemento [3] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [5], potremo associare a t_2 l’elemento [2] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [6], potremo associare a t_2 l’elemento 1 di $\mathbb{Z}_8$.
Un’altra possibile interpretazione associa [5] a c_0, [2] a c_1, $\cdot$ a f e $+$ a g; sotto tale interpretazione, al termine t_2 abbiamo associato la scrittura simbolica $x_0 \cdot (x_0 + [2])$. Scegliendo un’assegnazione che porta x_0 in [0], potremo associare a t_2 l’elemento [0] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [1], potremo associare a t_2 l’elemento [3] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [2], potremo associare a t_2 l’elemento [1] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [3], potremo associare a t_2 l’elemento [1] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [4], potremo associare a t_2 l’elemento [3] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [5], potremo associare a t_2 l’elemento [0] di $\mathbb{Z}_8$; scegliendo un’assegnazione che porta x_0 in [6], potremo associare a t_2 l’elemento [6] di $\mathbb{Z}_8$.

3.2 - Formule atomiche.

Per poter ragionare in termini di verità e falsità, ci serve un modo per esprimere predicati ([11]) sugli elementi della struttura in cui interpreteremo i termini del nostro linguaggio, che quindi va arricchito di opportuni simboli.

Un insieme A si dice un *alfabeto per un linguaggio descrittivo di secondo livello* (o anche, brevemente, un *alfabeto descrittivo di secondo livello*) se è un alfabeto per un linguaggio descrittivo di primo livello e inoltre ad esso appartengono

(v) un numero finito (ma arbitrario) t di simboli $\mathrm{P}_0, \mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_t$ detti “*simboli di predicato*”, a ciascuno dei quali resta associato un numero intero positivo detto *arietà* del simbolo di predicato; talvolta si scrive $\mathrm{P}_i^{(j_i)}$ per indicare che al simbolo di predicato P_i è associata la arietà j_i (o, come spesso si dice, che il simbolo di predicato f_i ha arietà j_i) .

Il *tipo* di un alfabeto per un linguaggio descrittivo di secondo livello è costituito non solo dal numero k dei simboli di costante, dal numero s dei simboli di funzione e dalla sequenza delle rispettive arietà, ma anche dal numero t dei simboli di predicato e dalla sequenza delle rispettive arietà.

---

[11] Ricordiamo da [3] (sez. 1.6) che si dice *predicato* (o anche *proposizione aperta*) *con variabile libera x su* X una scrittura $\mathrm{P}(x)$ con la proprietà che: per ogni $x_0 \in \mathrm{X}$, la scrittura $\mathrm{P}(x_0)$ ottenuta sostituendo ovunque x_0 a x in $\mathrm{P}(x)$ risulta un enunciato (nell’insieme X).

Parole e termini (aperti o chiusi) in un alfabeto per un linguaggio descrittivo di secondo livello si definiscono come in un alfabeto descrittivo di primo livello (quindi, in particolare, nei termini non sono coinvolti i simboli di predicato). L’aggiunta dei simboli di predicato ci permette però di introdurre un nuovo concetto, essenziale per lo sviluppo della logica dei predicati.

Sia A un alfabeto per un linguaggio descrittivo di secondo livello. Si dice *formula atomica* di A ogni parola della forma

$$\mathrm{P}_i t_1 t_2 \ldots t_{k_i}$$

dove P_i è un simbolo di predicato di arietà k_i e t_1, t_2, …, t_{k_i} sono termini su A. Generalmente, per una maggiore chiarezza, si introducono parentesi e virgole come già per indicare i termini, e anziché $\mathrm{P}_i t_1 t_2 \ldots t_{k_i}$ si scrive

$$\mathrm{P}_i(t_1, t_2, \ldots, t_{k_i}).$$

Una formula atomica si dice *chiusa* se in essa non compaiono variabili, *aperta* altrimenti.

L’insieme delle formule atomiche costituisce il *linguaggio descrittivo di secondo livello* associato all’alfabeto A.

Sia A un alfabeto per un linguaggio descrittivo di secondo livello. Si dice *struttura adeguata ad* A ogni quaterna ordinata (S, $\mathcal{C}$, $\mathcal{F}$, $\mathcal{P}$) tale che

– (S, $\mathcal{C}$, $\mathcal{F}$) è una struttura adeguata ad A in quanto alfabeto descrittivo di primo livello;
– $\mathcal{P}$ è un insieme $\{\overline{\mathrm{P}}_0, \overline{\mathrm{P}}_1, \overline{\mathrm{P}}_2, \ldots, \overline{\mathrm{P}}_t\}$ di t predicati su S tali che il predicato $\overline{\mathrm{P}}_t$ richiede k_i argomenti (dove k_i è la arietà di P_i) per $i := 1, \ldots, t$.

Sia A un alfabeto per un linguaggio descrittivo di secondo livello, e sia $\Sigma =$ (S, $\mathcal{C}$, $\mathcal{F}$, $\mathcal{P}$) una struttura adeguata ad A. Si dice *interpretazione* di A su Σ ogni funzione da A in $\mathcal{C} \cup \mathcal{F} \cup \mathcal{P}$ che fa corrispondere ad ogni simbolo di costante un elemento di $\mathcal{C}$, ad ogni simbolo di funzione f un elemento $\overline{f}$ di $\mathcal{F}$ “rispettando le arietà”, ossia con la condizione che $\overline{f}$ sia una funzione $\mathrm{S}^k \to \mathrm{S}$ dove k è la arietà di f, e ad ogni simbolo di predicato P un elemento $\overline{\mathrm{P}}$ di $\mathcal{P}$ ancora “rispettando le arietà”, ossia con la condizione che il predicato $\overline{\mathrm{P}}$ richieda tanti argomenti quanta è la arietà di P.

Sia A un alfabeto per un linguaggio descrittivo di secondo livello, e sia $\Sigma =$ (S, $\mathcal{C}$, $\mathcal{F}$, $\mathcal{P}$) una struttura adeguata ad A. Se è data una interpretazione i_0 di A su Σ, resta definita una funzione v dall’insieme delle formule atomiche *chiuse* su A nell’insieme {0, 1} ponendo

$$v(\mathrm{P}_i(t_1, t_2, \ldots, t_{k_i})) := \begin{cases} 1 & \text{se il predicato } \overline{\mathrm{P}} \text{ che } i_0 \text{ fa corrispondere a P risulta vero per gli} \\ & \text{elementi } \overline{t}_1, \overline{t}_2, \ldots, \overline{t}_{k_i} \text{ che } i_0 \text{ fa corrispondere a } t_1, t_2, \ldots, t_{k_i}\,; \\ 0 & \text{altrimenti} \end{cases}$$

Tale funzione v si dice *valutazione di verità* associata all’interpretazione i_0 data sulla struttura Σ.

Esempio 3.2.1

Sia A un alfabeto per un linguaggio descrittivo di secondo livello con due simboli di costante (c_0 e c_1) e un simbolo di predicato P di arietà 2. Sia

$$t_1 \coloneqq c_0\,;\, t_2 \coloneqq c_1\,; \qquad t_3 \coloneqq x\,;.$$

I termini t_1 e t_2 sono chiusi, mentre il termine t_3 è aperto. Vediamo alcuni esempi di strutture adeguate a A, alcune possibili interpretazioni di A su esse e le valutazioni di verità ad esse associate per la formula atomica $\alpha \coloneqq \mathrm{P}(t_1, t_2)$.

Caso 1 $\mathrm{S} \coloneqq \mathbb{Z}$, $\mathcal{C} \coloneqq \{0, 1\}$, $\mathcal{P} \coloneqq \{ < \}$.

Una possibile interpretazione associa 0 a c_0, 1 a c_1 e (senza possibilità di scelta) $<$ a P. Sotto tale interpretazione, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 0 < 1\,; \\ 0 & \text{se } 0 \not< 1\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

Per la formula atomica aperta $\beta \coloneqq \mathrm{P}(t_1, t_3)$ non siamo in grado di definire una valutazione di verità.

Un’altra possibile interpretazione associa 1 a c_0, 0 a c_1 e (senza possibilità di scelta) $<$ a P. Sotto tale interpretazione, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 1 < 0\,; \\ 0 & \text{se } 1 \not< 0\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

Per la formula atomica aperta $\beta \coloneqq \mathrm{P}(t_1, t_3)$ non siamo in grado di definire una valutazione di verità.

Caso 2 $\mathrm{S} \coloneqq \mathbb{N}$, $\mathcal{C} \coloneqq \{0, 1\}$, $\mathcal{P} \coloneqq \{ \leq\,,\, = \}$.

Una possibile interpretazione associa 0 a c_0, 1 a c_1 e $\leq$ a P. Sotto tale interpretazione, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 0 \leq 1\,; \\ 0 & \text{se } 0 \not\leq 1\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

Un’altra possibile interpretazione associa 1 a c_0, 0 a c_1 e poi ancora $\leq$ a P. Sotto tale interpretazione, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 1 \leq 0\,; \\ 0 & \text{se } 1 \not\leq 0\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

Una terza possibile interpretazione associa 1 a c_0, 0 a c_1, e = a P. Sotto tale interpretazione, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 1 = 0\,; \\ 0 & \text{se } 1 \neq 0\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

Notiamo che in nessuna delle tre interpretazioni per la formula atomica aperta $\beta \coloneqq \mathrm{P}(t_1, t_3)$ siamo in grado di definire una valutazione di verità.

Sia A un alfabeto per un linguaggio descrittivo di secondo livello, sia $\Sigma = ($S, $\mathcal{C}$, $\mathcal{F}$, $\mathcal{P})$ una struttura adeguata ad A e sia data una interpretazione i_0 di A su Σ. Sia $\mathcal{X}$ l'insieme delle variabili individuali di S. Si dice ancora *assegnazione* (all'interno della interpretazione data) ogni funzione $\mathcal{X} \to$ S.

La scelta di una assegnazione a_0, che ci consente di associare un elemento di S a ciascun termine su A, individua una funzione v dall'insieme di tutte le formule atomiche su A nell'insieme $(0, 1)$ ponendo

$$v(\mathrm{P}_i(t_1, t_2, \ldots, t_{k_i})) := \begin{cases} 1 & \text{se il predicato } \overline{\mathrm{P}} \text{ che } i_0 \text{ fa corrispondere a P risulta vero per gli} \\ & \text{elementi } \overline{t}_1, \overline{t}_2, \ldots, \overline{t}_{k_i} \text{ che } i_0 \text{ e } a_0 \text{ fanno corrispondere a } t_1, t_2, \ldots, t_{k_i}\,; \\ 0 & \text{altrimenti} \end{cases}$$

Tale funzione v si dice *valutazione di verità* associata all'interpretazione i_0 data sulla struttura Σ *sotto l'assegnazione* a_0 .

$\boxed{\text{Esempio 3.2.2}}$

Sia A un alfabeto per un linguaggio descrittivo di secondo livello con tre simboli di predicato P, Q, R tutti di arietà 2. Siano x, y variabili dell'alfabeto.

Vediamo alcuni esempi di strutture adeguate a A, alcune possibili interpretazioni di A su esse, alcune assegnazioni e le valutazioni di verità ad esse associate sotto tali assegnazioni per la formula atomica $\alpha :=$ P(x, y) .

$\boxed{\text{Caso 1}}$ S $:= \mathbb{Z}$, $\mathcal{P} := \{ <\,, \ \leq\,, \ = \}$.

Una possibile interpretazione associa $<$ a P, $\leq$ a Q e $=$ a R . Con tale interpretazione,

– sotto l'assegnazione che fa corrispondere 2 a x e 3 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 < 3\,; \\ 0 & \text{se } 2 \not< 3\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

– sotto l'assegnazione che fa corrispondere 2 a x e 1 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 < 1\,; \\ 0 & \text{se } 2 \not< 1\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

– sotto l'assegnazione che fa corrispondere 2 sia a x che a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 < 2\,; \\ 0 & \text{se } 2 \not< 2\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

Un’altra possibile interpretazione associa $\leq$ a P, $<$ a Q e $=$ a R . Con tale interpretazione,

– sotto l’assegnazione che fa corrispondere 2 a x e 3 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 \leq 3\,; \\ \\ 0 & \text{se } 2 \not\leq 3\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

– sotto l’assegnazione che fa corrispondere 2 a x e 1 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 \leq 1\,; \\ \\ 0 & \text{se } 2 \not\leq 1\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

– sotto l’assegnazione che fa corrispondere 2 sia a x che a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 \leq 2\,; \\ \\ 0 & \text{se } 2 \not\leq 2\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

Una terza possibile interpretazione associa $=$ a P, $<$ a Q e $\leq$ a R . Con tale interpretazione,

– sotto l’assegnazione che fa corrispondere 2 a x e 3 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 = 3\,; \\ \\ 0 & \text{se } 2 \neq 3\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

– sotto l’assegnazione che fa corrispondere 2 a x e 1 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 = 1\,; \\ \\ 0 & \text{se } 2 \neq 1\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

– sotto l’assegnazione che fa corrispondere 2 sia a x che a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 = 2\,; \\ \\ 0 & \text{se } 2 \neq 2\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

$\boxed{\text{Caso 2}}$ S $:= \mathbb{N}$, $\mathcal{P} := \{|, \leq, =\}$.

Una possibile interpretazione associa $|$ a P, $\leq$ a Q e $=$ a R. Con tale interpretazione,

– sotto l’assegnazione che fa corrispondere 2 a x e 3 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 \mid 3\,; \\ \\ 0 & \text{se } 2 \nmid 3\,. \end{cases}$$

Dunque, $v(\alpha) = 0$.

– sotto l’assegnazione che fa corrispondere 2 a x e 6 a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 \mid 6\,; \\ 0 & \text{se } 2 \nmid 6\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

– sotto l’assegnazione che fa corrispondere 2 sia a x che a y, la valutazione di verità per la formula atomica α è

$$v(\alpha) = \begin{cases} 1 & \text{se } 2 \mid 2\,; \\ 0 & \text{se } 2 \nmid 2\,. \end{cases}$$

Dunque, $v(\alpha) = 1$.

Più avanti ci sarà utile la seguente notazione.

Sia A un alfabeto per un linguaggio descrittivo di secondo livello, sia $\Sigma =$ (S, $\mathcal{C}$, $\mathcal{F}$, $\mathcal{P}$) una struttura adeguata ad A e sia data una interpretazione di A su Σ. Sia a una *assegnazione* di valori alle variabili individuali, sia $x_0 \in \mathcal{X}$ e sia $s \in$ S . Si indica con

$$a(x_0 \leftarrow s)$$

l’assegnazione di valori alle variabili individuali per la quale

$$a(x_0 \leftarrow s)(x) := a(x) \text{ se } x \neq x_0, \qquad\qquad a(x_0 \leftarrow s)(x_0) := s\,.$$

3.3 - Formule nella logica dei predicati.

Un alfabeto per un linguaggio descrittivo di secondo livello si dice un *alfabeto per la logica dei predicati* se ad esso appartengono, oltre alle variabili individuali, ai simboli di costante, ai simboli di funzione, ai simboli di predicato, alle parentesi e alla virgola anche i simboli “strettamente logici”

(vi) $\neg$, $\wedge$, $\vee$, $\rightarrow$ detti rispettivamente “*non*”, “*e*”, “*o*”, “*implica*” e detti complessivamente “*connettivi logici*” ;

(vii) $\forall$, $\exists$ detti rispettivamente “*quantificatore universale*” e “*quantificatore esistenziale*” .

Costituiscono il *tipo* di un alfabeto per la logica dei predicati gli stessi elementi che ne costituiscono il tipo come alfabeto descrittivo del secondo ordine, e quindi tutti i seguenti: il numero dei simboli di costante, il numero dei simboli di funzione con la sequenza delle rispettive arietà, il numero t dei simboli di predicato con la sequenza delle rispettive arietà.

Sia A un alfabeto per la logica dei predicati. Una parola w su A si dice *formula* di A (o anche *formula ben formata*, abbreviato in *fbf*) se esiste una sequenza finita w_0, w_1, w_2, $\dots$, w_k di parole su A tale che

FBF-PRED.1 $\quad w = w_k$

e

FBF-PRED.2 $\quad$ per ogni $h \leq k$ la parola w_h è

– una formula atomica di A

oppure

– della forma $(\neg w_i)$ (e si dice “negazione della formula w_i”)

oppure

– della forma $(w_i \wedge w_j)$ (e si dice “congiunzione delle formule w_i e w_j”)

oppure

– della forma $(w_i \vee w_j)$ (e si dice “disgiunzione delle formule w_i e w_j”)

oppure

– della forma $(w_i \rightarrow w_j)$ (e si dice “implicazione tra le formule w_i e w_j”)

oppure

– della forma $(\forall x)(w_i)$ con x variabile individuale di A

oppure

– della forma $(\exists x)(w_i)$ con x variabile individuale di A

con $i, j < h$.

La sequenza finita w_0, w_1, w_2, $\dots$, w_k si dice in questo caso una *costruzione* della formula w.

Il minimo numero naturale k per cui esiste una costruzione w_0, w_1, w_2, $\dots$, w_k della formula w si dice *lunghezza* di w.

L’insieme delle formule costituisce il *linguaggio della logica dei predicati associato all’alfabeto dato*.

Osservazione 3.3.1

Sia w una formula e sia w_0, w_1, w_2, $\dots$, w_k una costruzione di w .

Ogni w_i per $i := 0, \dots, k$ è una formula.

Esempio 3.3.2

Sia A un alfabeto per la logica dei predicati con variabili x, y, z, simboli di costante c e d, simboli di funzione f, g e h di arietà rispettivamente 1, 2 e 2, e simboli di predicato P, Q e R di arietà rispettivamente $1, 2$ e 3. Sono tre esempi di formule sull’alfabeto A (ciascuno su una diversa riga):

$$((\mathrm{P}(x) \wedge (\forall y)(\mathrm{Q}(x, f(z))$$
$$((\mathrm{R}(x, y, x) \rightarrow \mathrm{P}(z)) \rightarrow (\exists z)(\mathrm{Q}(x, y) \vee (\neg \mathrm{P}(x))))$$
$$(\forall x)(\exists z)((\neg(\mathrm{P}(y) \wedge \mathrm{Q}(x, x)) \vee (\mathrm{Q}(x, y) \rightarrow \mathrm{R}(y, y, y))))$$

Osservazione 3.3.3

Manterremo la convenzione introdotta con l’osservazione 2.1.6 per limitare l’uso delle parentesi. Vedremo più avanti che sarà possibile adottare anche la convenzione introdotta con l’osservazione 2.3.2

Osservazione 3.3.4 (“Principio di unica lettura delle formule”)

Sia w una formula che non è una formula atomica di A. Per la condizione FBF-PRED.2 c’è un unico modo di interpretare w come negazione di altra formula oppure (ma soltanto se ciò non è possibile) come congiunzione di altre due formule oppure (ma soltanto se le due precedenti interpretazioni non sono possibili) come disgiunzione di altre due formule oppure (ma soltanto se le tre precedenti interpretazioni non sono possibili) come implicazione fra due formule oppure (ma soltanto se le quattro precedenti interpretazioni non sono possibili) come formula preceduta da un quantificatore universale riferito a una certa variabile individuale oppure (ma soltanto se nessuna delle precedenti interpretazioni è possibile) come formula preceduta da un quantificatore esistenziale riferito a una certa variabile individuale.

3.4 - Formule aperte e formule chiuse. Comparse libere e vincolate delle variabili.

Sia A un alfabeto per la logica dei predicati, sia x una variabile individuale di A e sia φ una formula su A . Si dice che x *compare* in φ se x è stata usata nella costruzione di almeno un termine in almeno una formula atomica che è stata usata nella costruzione di φ.

Si dice poi che la variabile individuale x *compare libera* oppure *compare vincolata* in φ secondo la seguente definizione:

(i) se φ è una formula atomica, x *compare libera* in φ;

(ii) se φ è ottenuta come negazione di un’altra formula ψ oppure come congiunzione, disgiunzione o implicazione fra due altre formule ψ_1 e ψ_2, le *comparse libere* della x in φ sono tutte e sole le comparse libere della x in ψ oppure in ψ_1 e ψ_2;

(iii) se φ è ottenuta premettendo a un’altra formula ψ uno dei due quantificatori $(\forall x)$ o $(\exists x)$, ogni comparsa libera della x in ψ diventa una *comparsa vincolata* (da quel quantificatore) della x in φ.

Una formula si dice *aperta* se in essa c’è almeno una variabile individuale che compare libera; si dice invece chiusa se tutte le sue variabili individuali compaiono soltanto vincolate. Si noti che in una formula atomica se compare una variabile individuale essa vi compare certamente libera: dunque questa definizione di “formula aperta” e “formula chiusa” estende quella che abbiamo dato per le formule atomiche nella sez. 3.2 .

Una formula chiusa si dice anche un *enunciato*: come vedremo, ogni interpretazione dell’alfabeto A permette di dare una valutazione di verità a qualunque enunciato, mentre in generale per dare una valutazione di verità ad una formula aperta occorre scegliere non solo un’interpretazione dell’alfabeto A ma anche una assegnazione di valore alle variabili individuali che compaiono libere nella formula.

Esempio 3.4.1

Sia A un alfabeto per la logica dei predicati con variabili individuali x e y . Nella formula

$$(\forall x)(\mathrm{P}(x) \to \mathrm{Q}(y)) \vee (\exists y)\mathrm{Q}(x)$$

la variabile individuale x che compare come argomento del predicato P è vincolata (dal quantificatore universale $(\forall x)$), mentre quella che compare come argomento del predicato Q è libera. La variabile individuale y compare soltanto libera.

Si noti che è il principio di unica lettura (oss. 3.3.4) che consente di decidere se una comparsa di una variabile individuale è libera o vincolata; si noti inoltre che nelle scritture $(\forall x)$ e $(\exists y)$ anche se compaiono esplicitamente i simboli x e y non si parla di “comparsa” delle variabili individuali x e y (tali scritture servono soltanto a “vincolare” le variabili individuali nella formula a cui vengono premesse).

3.5 - Valutazioni di verità.

Sia A un alfabeto per la logica dei predicati, e sia Σ una struttura adeguata ad A; nella sez. 3.2 abbiamo visto come per ogni interpretazione e per ogni assegnazione resta definita una valutazione di verità sulle formule atomiche (che, nel caso delle formule atomiche chiuse, dipende soltanto dall'interpretazione e non dalla particolare assegnazione). Tale valutazione di verità, grazie al “principio di unica lettura” (oss. 3.3.4) si estende facilmente all'insieme di tutte le formule con le stesse regole viste nel punto (iii) del teorema 2.2.1 e con un trattamento particolare per le formule ottenute premettendo un quantificatore universale o un quantificatore esistenziale.

Teorema 3.5.1

Sia A un alfabeto per la logica dei predicati, e sia $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F}, \mathcal{P})$ una struttura adeguata ad A; per ogni interpretazione i_0 di A su Σ e per ogni assegnazione a_0 sull'insieme $\mathcal{X}$ delle variabili individuali di A, esiste un'unica funzione v dall'insieme delle formule di A nell'insieme $\{0, 1\}$ tale che

(i) per ogni formula atomica α, v coincide con la valutazione di verità di α associata all'interpretazione i_0 sotto l'assegnazione a_0, come definita nella sez. 3.2;

(ii) se la formula φ è della forma $(\neg\psi)$ oppure $(\psi_1 \wedge \psi_2)$ oppure $(\psi_1 \vee \psi_2)$ oppure $(\psi_1 \rightarrow \psi_2)$, si ha rispettivamente

$v(\neg\varphi_1) = 1 - v(\varphi_1)$;

$v(\varphi_1 \wedge \varphi_2) = \mathit{min}\{v(\varphi_1), v(\varphi_2)\}$, ossia: $v(\varphi_1 \wedge \varphi_2) = 1$ se $v(\varphi_1) = 1$ e $v(\varphi_2) = 1$, $v(\varphi_1 \wedge \varphi_2) = 0$ altrimenti;

$v(\varphi_1 \vee \varphi_2) = \mathit{max}\{v(\varphi_1), v(\varphi_2)\}$, ossia: $v(\varphi_1 \vee \varphi_2) = 0$ se $v(\varphi_1) = 0$ e $v(\varphi_2) = 0$, $v(\varphi_1 \vee \varphi_2) = 1$ altrimenti;

$v(\varphi_1 \rightarrow \varphi_2) = 0$ se $v(\varphi_1) = 1$ e $v(\varphi_2) = 0$, $v(\varphi_1 \rightarrow \varphi_2) = 1$ altrimenti;

(iii) se la formula φ è della forma $(\forall x)\psi$ oppure $(\exists x)\psi$ (con x variabile individuale dell'alfabeto A), si ha

$$v((\forall x)\psi) := \begin{cases} 1 & \text{se per ogni } s \in \mathrm{S} \text{ si ha } \overline{v}(\psi) = 1 \\ & \text{per la valutazione di verità } \overline{v} \text{ associata all'interpretazione } i_0 \\ & \text{sotto l'assegnazione } a_0(x \leftarrow s) \text{ definita nella sez. 3.2} \\ & \\ 0 & \text{altrimenti} \end{cases}$$

e

$$v((\exists x)\psi) := \begin{cases} 1 & \text{se esiste un } s \in \mathrm{S} \text{ tale che } \overline{v}(\psi) = 1 \\ & \text{per la valutazione di verità } \overline{v} \text{ associata all'interpretazione } i_0 \\ & \text{sotto l'assegnazione } a_0(x \leftarrow s) \text{ definita nella sez. 3.2} \\ & \\ 0 & \text{altrimenti} \end{cases}$$

Tale funzione v si dice “*valutazione di verità sull’insieme delle formule di* A *associata all’interpretazione* i_0 *e all’assegnazione* a_0”.

Esempio 3.5.2

Sia A un alfabeto per la logica dei predicati con tre variabili x, y e z, un simbolo di funzione binaria p e un simbolo di predicato binario D. Sia

$$\varphi(x) := (\forall y)(\forall z)(\mathrm{D}(x, p(y, z)) \rightarrow (\mathrm{D}(x, y) \vee \mathrm{D}(x, z)) .$$

La $\varphi(x)$ è una formula di A con variabile libera x (dunque non è un enunciato!).

Se interpretiamo A nell’insieme $\mathbb{N}$ dei numeri naturali associando al simbolo p l’operazione *prodotto* ([12]) e al simbolo D la relazione *divide* (cosicché $\mathrm{D}(x, y)$ significa “x divide y”), la $\varphi(x)$ risulta vera sotto l’assegnazione che alla x associa 1, sotto quella che alla x associa 2 e sotto quella che alla x associa 3, ma non sotto quella che alla x associa 4 (o 15, o 231, o…). Ciò si esprime scrivendo che $\varphi(1)$ è vera, $\varphi(2)$ è vera, $\varphi(3)$ è vera, $\varphi(4)$ è falsa, $\varphi(15)$ è falsa, $\varphi(231)$ è falsa…

Invece le formule $(\forall x)\varphi(x)$ e $(\exists x)\varphi(x)$ sono formule chiuse, cioè enunciati, e sotto qualsiasi interpretazione possiamo dar loro un valore di verità senza dover scegliere in aggiunta una particolare assegnazione; nell’interpretazione vista sopra, e per le osservazioni fatte nel paragrafo precedente, la $(\forall x)\varphi(x)$ è falsa mentre la $(\exists x)\varphi(x)$ è vera.

Sia A un alfabeto per la logica dei predicati. Siano Σ una struttura adeguata ad A, i_0 un’interpretazione di A su Σ e a_0 un’assegnazione sull’insieme delle variabili individuali di A. Siano φ una formula di A e v la valutazione di verità associata all’interpretazione i_0 e all’assegnazione a_0.

Se $v(\varphi) = 1$, si dice che “φ *è vera sotto* v”, oppure che “φ *è vera nell’interpretazione* i_0 *sotto l’assegnazione* a_0”, oppure che “Σ *nell’interpretazione* i_0 *sotto l’assegnazione* a_0 *soddisfa* φ” oppure anche che “Σ *nell’interpretazione* i_0 *sotto l’assegnazione* a_0 *è un modello per* φ”; scriveremo talvolta

$$\Sigma \vDash_{i_0,\, a_0} \varphi .$$

[12] Ricordiamo che ogni operazione in un insieme I è una particolare funzione $\mathrm{I} \times \mathrm{I} \rightarrow \mathrm{I}$.

Se φ è una formula chiusa, $v(\varphi)$ dipende soltanto da i_0 e non dall'assegnazione a_0: in questo caso diremo semplicemente che “*φ è vera nell'interpretazione i_0*”, oppure che “*Σ nell'interpretazione i_0 soddisfa φ*” oppure anche che “*Σ nell'interpretazione i_0 è un modello per φ*”; e scriveremo

$$\Sigma \models_{i_0} \varphi\,.$$

Quando è chiaro quale sia l'interpretazione considerata sulla struttura Σ, con abuso di linguaggio e di scrittura si dice che “*φ è vera in Σ*”, oppure che “*Σ è un modello per φ*”, e si scrive

$$\Sigma \models \varphi\,.$$

Osservazione 3.5.3

Siano A un alfabeto per la logica dei predicati, $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F}, \mathcal{P})$ una struttura adeguata ad A e $\Sigma^* = (\mathrm{S}^*, \mathcal{C}, \mathcal{F}^*, \mathcal{P}^*)$ una struttura con $\mathrm{S}^* \subseteq \mathrm{S}$, intendendo opportunamente $\mathcal{F}^*$ e $\mathcal{P}^*$ come restrizioni a S^* di $\mathcal{F}$ e $\mathcal{P}$, cosicché Σ^* è una struttura adeguata ad A.

Sia inoltre φ una formula di A.

Può accadere che Σ sia un modello per φ ma Σ^* non lo sia, e può anche accadere che viceversa Σ^* sia un modello per φ ma Σ non lo sia.

Ad esempio, sia A un linguaggio senza simboli di costanti o di funzioni e con un solo simbolo di predicato P (binario) e siano $\Sigma := (\mathbb{Z}, \emptyset, \emptyset, \{\overline{\mathrm{P}}\})$ e $\Sigma^* := (\mathbb{N}, \emptyset, \emptyset, \{\overline{\mathrm{P}}\})$, dove $\overline{\mathrm{P}}$ è la relazione “essere il successivo di”. La struttura Σ è un modello per la formula φ così definita

$$\varphi := (\forall x)(\exists y)\mathrm{P}(x, y)$$

mentre la struttura Σ^* non lo è (si noti che in entrambi i casi c'è un'unica possibile interpretazione, quella che al simbolo di predicato P associa il predicato $\overline{\mathrm{P}}$).
Viceversa, la struttura Σ^* è un modello per la formula ψ definita da

$$\psi := (\exists x)(\forall y)\neg\mathrm{P}(x, y)$$

mentre la struttura Σ non lo è.

Sia φ una formula di A.

Se φ è vera in ogni struttura adeguata ad A, per ogni interpretazione e sotto ogni assegnazione di valore alle variabili individuali di A libere in φ, si dice che φ è *valida*.

Se esistono almeno una struttura Σ adeguata ad A, almeno un'interpretazione i_0 su Σ e almeno un'assegnazione a_0 sull'insieme delle variabili individuali di A libere in φ tale che φ è vera nell'interpretazione Σ sotto l'assegnazione a_0, si dice che φ è *soddisfacibile*; se ciò non avviene, cioè se nessuna struttura adeguata ad A in nessuna interpretazione e sotto nessuna assegnazione di valore alle variabili individuali è un modello per φ, si dice che φ è *insoddisfacibile*.

Naturalmente, se φ è una formula chiusa (cioè un enunciato), la sua validità, soddisfacibilità o insoddisfacibilità dipendono soltanto dalle strutture adeguate al linguaggio A e dalle possibili interpretazioni.

Teorema 3.5.4

Siano A un alfabeto per la logica dei predicati e φ una formula di A. La formula φ è valida se e soltanto se la formula $\neg\varphi$ è insoddisfacibile.

Dimostrazione – Supponiamo in primo luogo che φ sia una formula valida: allora per ogni struttura Σ adeguata ad A, per ogni interpretazione i_0 e per ogni assegnazione a_0 di valore alle variabili individuali di A libere in φ (cioè libere in $\neg\varphi$) si ha

$$v(\varphi) = 1 \quad \text{(e quindi } v(\neg\varphi) = 0\text{)}$$

per la valutazione di verità associata a Σ, i_0 e a_0. Ma ciò significa che $\neg\varphi$ è insoddisfacibile.

Viceversa, supponiamo che $\neg\varphi$ sia insoddisfacibile: allora per ogni struttura Σ adeguata ad A, per ogni interpretazione i_0 e per ogni assegnazione a_0 di valore alle variabili individuali di A libere in $\neg\varphi$ (cioè libere in φ) si ha

$$v(\neg\varphi) = 0 \quad \text{(e quindi } v(\varphi) = v(\neg\neg\varphi) = 1\text{)}$$

per la valutazione di verità associata a Σ, i_0 e a_0. Ma ciò significa che φ è una formula valida.

Teorema 3.5.5

Siano A un alfabeto per la logica dei predicati e φ, ψ formule di A. Sono fatti equivalenti:

(i) $\varphi \wedge \psi$ è una formula valida;

(ii) φ e ψ sono entrambe formule valide.

Dimostrazione — Supponiamo in primo luogo che $\varphi \wedge \psi$ sia una formula valida: allora per ogni struttura Σ adeguata ad A, per ogni interpretazione i_0 e per ogni assegnazione a_0 di valore alle variabili individuali di A si ha

$$v(\varphi \wedge \psi) = 1 \quad \text{(e quindi } v(\varphi) = v(\psi) = 1\text{)}$$

per la valutazione di verità associata a Σ, i_0 e a_0. Dunque φ e ψ sono entrambe formule valide.

Viceversa, supponiamo che φ e ψ siano entrambe formule valide: allora per ogni struttura Σ adeguata ad A, per ogni interpretazione i_0 e per ogni assegnazione a_0 di valore alle variabili individuali di A si ha

$$v(\varphi) = v(\psi) = 1 \quad \text{(e quindi } v(\varphi \wedge \psi) = 1\text{)}$$

per la valutazione di verità associata a Σ, i_0 e a_0. Dunque $\varphi \wedge \psi$ è una formula valida.

Sia A un alfabeto per la logica dei predicati, e siano φ, ψ formule di A.

Si dice che ψ *è conseguenza logica di* φ e si scrive

$$\varphi \models \psi$$

se per ogni struttura adeguata ad A ogni interpretazione (e ogni assegnazione di valori alle eventuali variabili libere di φ e ψ) che soddisfa φ soddisfa anche ψ .

Si dice che φ e ψ sono *logicamente equivalenti* e si scrive

$$\varphi \equiv \psi$$

se ciascuna di esse è conseguenza logica dell’altra, cioè se per ogni struttura adeguata ad A ogni interpretazione (e ogni assegnazione di valori alle eventuali variabili libere di φ e ψ) che soddisfa φ soddisfa anche ψ e viceversa ogni interpretazione (e ogni assegnazione di valori alle eventuali variabili libere di φ e ψ) che soddisfa ψ soddisfa anche φ.

$\boxed{\text{Osservazione 3.5.6 (importante!)}}$

La definizione di formula che abbiamo dato nella sez. 3.3 per la logica dei predicati e il teorema 3.5.1 che regola l’estensione delle valutazioni di verità dalle formule atomiche a tutte le formule ricalcano, rispettivamente, la definizione di formula che abbiamo dato nella sez. 2.1 per la logica proposizionale e il teorema 2.2.1 . Per effetto di questa corrispondenza formale, se in una formula φ della logica proposizionale a ciascuna variabile proposizionale si sostituisce una formula della logica dei predicati si ottiene una formula della logica dei predicati che si dice *esemplificazione di* φ ; e, fatto altrettanto importante, se φ e ψ sono formule logicamente equivalenti della logica proposizionale le formule della logica dei predicati ottenute da φ e ψ sostituendo a ciascuna variabile proposizionale sempre la stessa formula della logica dei predicati sono logicamente equivalenti nella logica dei predicati.

$\boxed{\text{Esempio 3.5.7}}$

Per quanto detto nell’osservazione 3.5.6, ogni esemplificazione di una tautologia è una formula valida (e spesso si dice ancora tautologia). Ma esistono formule valide che non sono esemplificazioni di tautologie (e che ***non*** si dicono tautologie!); ad esempio,

$$(\exists x)(\mathrm{P}(x) \vee \mathrm{Q}(x)) \to ((\exists x)\mathrm{P}(x)) \vee (\exists x)\mathrm{Q}(x))\,.$$

Allo stesso modo esistono formule insoddisfacibili che non sono esemplificazioni di contraddizioni; ad esempio, $(\exists x)(\mathrm{P}(x) \vee \mathrm{Q}(x)) \wedge (\forall x)\neg\mathrm{P}(x) \wedge (\forall x)\neg\mathrm{Q}(x)$.

$\boxed{\text{Teorema 3.5.8}}$

Siano A un alfabeto per la logica dei predicati, x una variabile individuale e φ una formula di A nella quale x non compare. Allora

$$\varphi \equiv (\forall x)\varphi \qquad \text{e} \qquad \varphi \equiv (\exists x)\varphi\,.$$

Dimostrazione – Ovvio.

$\boxed{\text{Teorema 3.5.9}}$

Siano A un alfabeto per la logica dei predicati e $\varphi(x)$ una formula di A nella quale ogni comparsa della variabile individuale x è libera e la variabile individuale y non compare. Se $\varphi(y)$ è la formula ottenuta da $\varphi(x)$ sostituendo alla variabile individuale x in ogni sua comparsa la variabile individuale y, allora

$$(\forall x)\varphi(x) \equiv (\forall y)\varphi(y) \qquad \text{e} \qquad (\exists x)\varphi(x) \equiv (\exists y)\varphi(y)\,.$$

Dimostrazione – Ovvio.

$\boxed{\text{Osservazione 3.5.10}}$

Per il teorema 3.5.9, se in una formula una certa variabile individuale è vincolata da un quantificatore il nome di quella variabile diviene irrilevante perché la valutazione di verità su quella formula non dipende più dalla (eventuale) particolare assegnazione con la quale si sta lavorando; di questo fatto conviene approfittare cambiando nome alle variabili vincolate ogni volta che ciò consenta di evitare confusioni e ambiguità (e qualche volta, come vedremo più avanti, anche per semplificare l'espressione dell'intera formula).

In particolare, vale la pena di osservare che teoricamente è accettabile una formula come

$$\mathrm{P}(x) \to ((\exists x)(\mathrm{Q}(x)))$$

dove la prima x è libera mentre la seconda è vincolata dal quantificatore $\exists x$, o addirittura una formula come

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

dove la prima x è vincolata dal quantificatore universale $\forall x$ mentre la seconda è vincolata dal quantificatore esistenziale $\exists x$.

Ma noi, per evitare confusioni e ambiguità, ci rifiuteremo di usare queste scritture cambiando, tutte le volte che dovesse risultare necessario, il nome a una o più fra le variabili vincolate. Questa scelta non diminuirà in alcun modo le capacità del nostro linguaggio di esprimere le situazioni che vogliamo analizzare logicamente.

Scriveremo dunque (ad esempio)

$$\mathrm{P}(x) \to ((\exists y)(\mathrm{Q}(y))) \qquad \text{anziché} \qquad \mathrm{P}(x) \to ((\exists x)(\mathrm{Q}(x)))$$

e

$$(\forall x)(\mathrm{P}(x) \to ((\exists y)(\mathrm{Q}(y)))) \qquad \text{anziché} \qquad (\forall x)(\mathrm{P}(x) \to ((\exists x)(\mathrm{Q}(x))))\,.$$

$\boxed{\text{Teorema 3.5.11}}$

Sia A un alfabeto per la logica dei predicati; siano φ, ψ formule di A. Sono fatti equivalenti:

(i) $\varphi \vDash \psi$;

(ii) $\varphi \to \psi$ è una formula valida.

Dimostrazione — Proviamo in primo luogo che dalla (i) segue la (ii).

Supponiamo che valga la (i). Siano Σ una struttura adeguata ad A, i_0 un'interpretazione di A su Σ e a_0 un'assegnazione di valori alle variabili individuali di A; dobiamo dimostrare che

$$\Sigma \vDash_{i_0,\, a_0} (\varphi \to \psi)\,.$$

Ciò significa che, detta v la valutazione di verità sull'insieme delle formule di A associata a i_0 e a_0 (cfr. teor. 3.5.1), deve essere $v(\varphi \to \psi) = 1$. Ciò è immediato se $v(\varphi) = 0$; ma se invece $v(\varphi) = 1$ allora, poiché vale la (i), è anche $v(\psi) = 1$ e dunque anche in questo caso $v(\varphi \to \psi) = 1$ come si voleva.

Adesso proviamo che dalla (ii) segue la (i).

Supponiamo dunque che valga la (ii). Siano Σ una struttura adeguata ad A, i_0 un'interpretazione di A su Σ e a_0 un'assegnazione di valori alle variabili individuali di A tale che

$$\Sigma \models_{i_0, a_0} \varphi$$

e proviamo che è anche

$$\Sigma \models_{i_0, a_0} \psi \, .$$

Ciò significa che, detta v la valutazione di verità sull'insieme delle formule di A associata a i_0 e a_0 (cfr. teor. 3.5.1), deve essere $v(\psi) = 1$. Ma noi sappiamo che $v(\varphi) = 1$ e che, poiché vale la (ii), è $v(\varphi \to \psi) = 1$; dunque deve essere $v(\psi) = 1$ come si voleva.

Teorema 3.5.12

Sia A un alfabeto per la logica dei predicati; siano φ, ψ formule di A. Sono fatti equivalenti:

(i) φ e ψ sono logicamente equivalenti;

(ii) $(\varphi \to \psi) \land (\psi \to \varphi)$ è una formula valida.

Dimostrazione — Segue immediatamente dai teoremi 3.5.11 e 3.5.5.

Come si è fatto nella sez. 2.2 per la logica proposizionale, anche nella logica dei predicati la nozione di “conseguenza logica” si può generalizzare a un insieme di formule.

Siano $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n, \varphi$ formule. Si dice che “*φ è conseguenza logica di $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n$*” e si scrive

$$\{\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n\} \models \varphi$$

se per ogni struttura adeguata ad A ogni interpretazione (e ogni assegnazione di valori alle eventuali variabili libere di $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n$ e φ) che soddisfa $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n$ soddisfa anche φ.

Vale l’analogo del teorema 2.3.4 (e dell’osservazione 2.3.5) :

Teorema 3.5.13

Siano $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n, \varphi$ formule. Sono fatti equivalenti:

(i) $\{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n\} \vDash \varphi$;

(ii) $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \dots \wedge \alpha_n \vDash \varphi$;

(iii) $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \dots \wedge \alpha_n \wedge \neg\varphi$ è insoddisfacibile .

3.6 - Forma normale prenessa.

Una formula della logica dei predicati si dice *in forma normale prenessa* ([13]) se si scrive come

$$(\mathrm{Q}_1 x_1)(\mathrm{Q}_2 x_2)\dots(\mathrm{Q}_s x_s)\varphi$$

dove ciascun Q_j è un quantificatore (cioè è $\forall$ oppure $\exists$) e φ è una formula nella cui espressione non compaiono quantificatori.

I prossimi teoremi sono propedeutici al teorema 3.6.15 (nel quale stabiliremo che ogni formula è logicamente equivalente a una formula in forma normale prenessa) ma ciascuno di essi ha anche uno specifico interesse operativo (precisa cioè uno dei passaggi da effettuarsi per trasformare una assegnata formula in altra che sia ad essa logicamente equivalente e in forma normale prenessa).

Teorema 3.6.1

Per ogni variabile individuale x e per ogni formula φ, le formule

$$\neg((\forall x)\varphi) \quad \text{e} \quad (\exists x)(\neg\varphi)$$

sono logicamente equivalenti.

Dimostrazione – Proviamo in primo luogo che $\neg((\forall x)\varphi) \vDash (\exists x)(\neg\varphi)$. Siano $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F}, \mathcal{P})$ una struttura adeguata ad A, i_0 un’interpretazione di A su Σ e a_0 un’assegnazione di valori alle variabili individuali di A diverse da x tale che

$$\Sigma \vDash_{i_0, a_0} \neg((\forall x)\varphi)$$

e proviamo che è anche $\Sigma \vDash_{i_0, a_0} (\exists x)(\neg\varphi)$.

[13] Non è un errore di stampa! L’aggettivo “pre***n***essa” (con la “n”) è una “traduzione pigra” dell’inglese *prenex*, forse dal latino “prae-nexa” (per “legata anteriormente”, cfr. il verbo “nectere”) .

Per ipotesi, è falso che per ogni $s \in$ S si abbia $\overline{v}(\varphi) = 1$ per la valutazione di verità $\overline{v}$ associata all'interpretazione i_0 sotto l'assegnazione $a_0(x \leftarrow s)$ definita nella sez. 3.2 : esiste dunque almeno un $s_* \in$ S tale che $\overline{v}(\varphi) = 0$ (e quindi $\overline{v}(\neg\varphi) = 0$) per la valutazione di verità $\overline{v}$ associata all'interpretazione i_0 sotto l'assegnazione $a_0(x \leftarrow s_*)$ definita nella sez. 3.2 ; ma ciò significa appunto che $\Sigma \vDash_{i_0,\, a_0} (\exists x)(\neg\varphi)$.

Resta da provare che è anche $(\exists x)(\neg\varphi) \vDash \neg((\forall x)\varphi)$.

Siano dunque $\Sigma = (\mathrm{S}, \mathcal{C}, \mathcal{F}, \mathcal{P})$ una struttura adeguata ad A, i_0 un'interpretazione di A su Σ e a_0 un'assegnazione di valori alle variabili individuali di A diverse da x tale che

$$\Sigma \vDash_{i_0,\, a_0} (\exists x)(\neg\varphi)$$

e proviamo che è anche

$$\Sigma \vDash_{i_0,\, a_0} \neg((\forall x)\varphi) .$$

Per ipotesi, esiste dunque almeno un $s_* \in$ S tale che $\overline{v}(\neg\varphi) = 1$ (e quindi $\overline{v}(\varphi) = 0$) per la valutazione di verità $\overline{v}$ associata all'interpretazione i_0 sotto l'assegnazione $a_0(x \leftarrow s_*)$ definita nella sez. 3.2: dunque è falso che per ogni $s \in$ S si abbia $\overline{v}(\varphi) = 1$ per la valutazione di verità $\overline{v}$ associata all'interpretazione i_0 sotto l'assegnazione $a_0(x \leftarrow s)$ definita nella sez. 3.2 ; ma ciò significa appunto che

$$\Sigma \vDash_{i_0,\, a_0} \neg((\forall x)\varphi) .$$

$\boxed{\text{Teorema 3.6.2}}$

Per ogni variabile individuale x e per ogni formula φ, le formule

$$\neg((\exists x)\varphi) \quad \text{e} \quad (\forall x)(\neg\varphi)$$

sono logicamente equivalenti.

Dimostrazione — Infatti

$$\neg((\exists x)\varphi) \overset{\text{(teor. 2.3.1 } (i))}{\equiv} \neg((\exists x)\neg(\neg\varphi)) \overset{\text{(teor. 3.6.1)}}{\equiv} \neg(\neg((\forall x)\neg\varphi)) \overset{\text{(teor. 2.3.1 } (i))}{\equiv} (\forall x)(\neg\varphi) .$$

$\boxed{\text{Teorema 3.6.3}}$

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, le formule

$$(\forall x)\varphi \wedge (\forall x)\psi \quad \text{e} \quad (\forall x)(\varphi \wedge \psi)$$

sono logicamente equivalenti.

Dimostrazione — Omettiamo la dimostrazione di questo teorema.

Teorema 3.6.4

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, se x non compare in ψ le formule

$$(\forall x)\varphi \wedge \psi \quad \text{e} \quad (\forall x)(\varphi \wedge \psi)$$

sono logicamente equivalenti.

Dimostrazione — Per il teorema 3.5.8, poiché x non compare in ψ, le formule ψ e $(\forall x)\psi$ sono logicamente equivalenti. Dal teorema 3.6.3 allora segue subito che

$$(\forall x)\varphi \wedge \psi \equiv (\forall x)\varphi \wedge (\forall x)\psi \equiv (\forall x)(\varphi \wedge \psi)\,.$$

Osservazione 3.6.5

In generale, le formule $(\forall x)\varphi \vee (\forall x)\psi$ e $(\forall x)(\varphi \vee \psi)$ ***non*** sono logicamente equivalenti.

Ad esempio, sia $\varphi := \mathrm{P}(x)$ e sia $\psi := \mathrm{D}(x)$ con P e D simboli di predicati unari; interpretiamo P e D nell’insieme $\mathbb{N}$ dei numeri naturali rispettivamente come “essere pari” e “essere dispari”. Allora $(\forall x)(\varphi \vee \psi)$ è vera (ogni numero è pari o dispari!) ma $(\forall x)\varphi \vee (\forall x)\psi$ è falsa (non è vero che ogni numero è pari, e nemmeno che ogni numero è dispari!).

Esercizio 3.6.6

Si dimostri che per ogni variabile individuale x e per ogni scelta delle formule φ e ψ si ha che

$$(\forall x)\varphi \vee (\forall x)\psi \;\vDash\; (\forall x)(\varphi \vee \psi)\,.$$

Teorema 3.6.7

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, se x non compare libera in ψ le formule

$$(\forall x)\varphi \vee \psi \quad \text{e} \quad (\forall x)(\varphi \vee \psi)$$

sono logicamente equivalenti.

Dimostrazione — Omettiamo la dimostrazione di questo teorema.

Corollario 3.6.8

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, se y non compare né in φ né in ψ le formule

$$(\forall x)\varphi(x) \vee (\forall x)\psi(x) \quad \text{e} \quad (\forall x)(\forall y)(\varphi(x) \vee \psi(y))$$

sono logicamente equivalenti.

Dimostrazione — Per il teorema 3.5.9, poiché y non compare in $\psi(x)$ le formule $(\forall x)\psi(x)$ e $(\forall y)\psi(y)$ sono logicamente equivalenti, dunque

$$(\forall x)\varphi(x) \vee (\forall x)\psi(x) \quad \text{e} \quad (\forall x)\varphi(x) \vee (\forall y)\psi(y)$$

sono logicamente equivalenti. Poiché x non compare in $\psi(y)$, per il teorema 3.6.7 le formule

$$(\forall x)\varphi(x) \vee (\forall y)\psi(y) \quad \text{e} \quad (\forall x)\big(\varphi(x) \vee (\forall y)\psi(y)\big)$$

sono logicamente equivalenti.

Ancora per il teorema 3.6.7, poiché y non compare in $\varphi(x)$ le formule

$$\varphi(x) \vee (\forall y)\psi(y) \quad \text{e} \quad (\forall y)\big(\varphi(x) \vee \psi(y)\big)$$

sono logicamente equivalenti. Dunque anche

$$(\forall x)\big(\varphi(x) \vee (\forall y)\psi(y)\big) \quad \text{e} \quad (\forall x)(\forall y)\big(\varphi(x) \vee \psi(y)\big)$$

sono logicamente equivalenti, e infine si ha l’asserto.

Teorema 3.6.9

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, le formule

$$(\exists x)\varphi \vee (\exists x)\psi \quad \text{e} \quad (\exists x)(\varphi \vee \psi)$$

sono logicamente equivalenti.

Dimostrazione — Infatti

$$(\exists x)\varphi \vee (\exists x)\psi \overset{\text{(teor. 2.3.1 }(i))}{\equiv} \neg\big(\neg\big((\exists x)\varphi \vee (\exists x)\psi\big)\big) \overset{\text{(teor. 2.5.1 }(i))}{\equiv}$$

$$\overset{\text{(teor. 2.5.1 }(i))}{\equiv} \neg\big((\neg(\exists x)\varphi \wedge \neg(\exists x)\psi)\big) \overset{\text{(teor. 3.6.2)}}{\equiv} \neg\big(((\forall x)\neg\varphi) \wedge ((\forall x)\neg\psi)\big) \overset{\text{(teor. 3.6.3)}}{\equiv}$$

$$\overset{\text{(teor. 3.6.3)}}{\equiv} \neg(\forall x)\big((\neg\varphi) \wedge (\neg\psi)\big) \overset{\text{(teor. 2.3.1 }(i))}{\equiv} \neg(\forall x)\big(\neg(\varphi \vee \psi)\big) \overset{\text{(teor. 3.6.1)}}{\equiv}$$

$$\overset{\text{(teor. 3.6.1)}}{\equiv} (\exists x)\big(\neg(\neg(\varphi \vee \psi))\big) \overset{\text{(teor. 2.3.1 }(i))}{\equiv} (\exists x)(\varphi \vee \psi)$$

Teorema 3.6.10

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, se x non compare in ψ le formule

$$(\exists x)\varphi \vee \psi \quad \text{e} \quad (\exists x)(\varphi \vee \psi)$$

sono logicamente equivalenti.

Dimostrazione — Per il teorema 3.5.8, poiché x non compare in ψ, le formule ψ e $(\exists x)\psi$ sono logicamente equivalenti. Dal teorema 3.6.9 allora segue subito che

$$(\exists x)\varphi \vee \psi \equiv (\exists x)\varphi \vee (\exists x)\psi \equiv (\exists x)(\varphi \vee \psi)\,.$$

Osservazione 3.6.11

In generale, le formule $(\exists x)\varphi \wedge (\exists x)\psi$ e $(\exists x)(\varphi \wedge \psi)$ ***non*** sono logicamente equivalenti.

Ad esempio, sia $\varphi := \mathrm{P}(x)$ e sia $\psi := \mathrm{D}(x)$ con P e D simboli di predicati unari; interpretiamo P e D nell’insieme $\mathbb{N}$ dei numeri naturali rispettivamente come “essere pari” e “essere dispari”. Allora $(\exists x)\varphi \wedge (\exists x)\psi$ è vera (esiste un numero pari ed esiste un numero dispari!) ma $(\exists x)(\varphi \wedge \psi)$ è falsa (non esiste alcun numero che sia pari e dispari).

Esercizio 3.6.12

Si dimostri che per ogni variabile individuale x e per ogni scelta delle formule φ e ψ si ha che

$$(\exists x)(\varphi \wedge \psi) \vDash (\exists x)\varphi \wedge (\exists x)\psi\,.$$

Teorema 3.6.13

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, se x non compare libera in ψ le formule

$$(\exists x)\varphi \wedge \psi \quad \text{e} \quad (\exists x)(\varphi \wedge \psi)$$

sono logicamente equivalenti.

Dimostrazione — Omettiamo la dimostrazione di questo teorema (che si potrebbe dedurre dal teorema 3.6.7 ragionando come si è fatto nella dimostrazione del teorema 3.6.9 .

Corollario 3.6.14

Per ogni variabile individuale x e per ogni scelta delle formule φ e ψ, se y non compare né in φ né in ψ le formule

$$(\exists x)\varphi(x) \wedge (\exists x)\psi(x) \quad \text{e} \quad (\exists x)(\exists y)(\varphi(x) \wedge \psi(y))$$

sono logicamente equivalenti.

Dimostrazione – Lasciamo per esercizio al lettore questa dimostrazione, del tutto analoga a quella del corollario 3.6.8 .

Teorema 3.6.15

Ogni formula è logicamente equivalente a una formula in forma normale prenessa.

Dimostrazione – Per il teorema 2.4.1, possiamo supporre che gli unici simboli logici che compaiono nella scrittura della formula oltre ai quantificatori siano i connettivi $\neg$, $\wedge$ e $\vee$. I teoremi 3.6.1, 3.6.2, 3.6.3, 3.6.7, 3.6.9 e 3.6.13 permettono allora di trasformare la formula in altra ad essa logicamente equivalente in forma normale prenessa.

Tenendo conto del teorema 2.6.6, possiamo dire qualcosa di più:

Teorema 3.6.16

Ogni formula è logicamente equivalente a una formula in forma normale prenessa nella quale la parte che segue i quantificatori è in forma normale congiuntiva.

Esempio 3.6.17

Sia x una variabile individuale e siano P, Q simboli di predicato unario. Trasformiamo la formula

$$(\forall x)\mathrm{P}(x) \to (\forall x)\mathrm{Q}(x)$$

in una formula in forma normale prenessa ad essa logicamente equivalente.

Si ha

$$(\forall x)\mathrm{P}(x) \to ((\forall x)\mathrm{Q}(x)) \equiv \neg((\forall x)\mathrm{P}(x)) \vee ((\forall x)\mathrm{Q}(x)) \equiv$$
$$\equiv ((\exists x)\neg\mathrm{P}(x)) \vee ((\forall y)\mathrm{Q}(y)) \equiv (\exists x)(\forall y)(\neg\mathrm{P}(x) \vee \mathrm{Q}(y))\,.$$

3.7 - Skolemizzazione.

Una formula della logica dei predicati si dice *in forma di Skolem* ([14]) se si scrive come

$$(\forall x_1)(\forall x_2)\dots(\forall x_s)\varphi$$

dove φ è una formula nella cui espressione non compaiono quantificatori.

Teorema 3.7.1 (Skolem)

Siano A un alfabeto per la logica dei predicati e φ una formula di A. Esistono un alfabeto A^* con $\mathrm{A} \subset \mathrm{A}^*$ e una formula φ^* in forma di Skolem su A^* tali che

φ è soddisfacibile se e soltanto se φ^* è soddisfacibile.

Dimostrazione — Per il teorema 3.6.15, possiamo supporre che φ sia in forma normale prenessa. Sia x una variabile individuale vincolata in φ da un quantificatore esistenziale; distinguiamo due casi:

se nella scrittura di φ il quantificatore $\exists$ che vincola x non è preceduto da alcun quantificatore universale, sopprimiamo la scrittura $(\exists x)$ introducendo un nuovo simbolo di costante c col quale sostituiamo la x ogni volta che appare in φ;

se nella scrittura di φ il quantificatore $\exists$ che vincola x è preceduto da k quantificatori universali che vincolano le variabili individuali x_1, x_2, $\dots$, x_k, sopprimiamo la scrittura $(\exists x)$ introducendo un nuovo simbolo di funzione f e sostituiamo la x ogni volta che appare in φ con la scrittura $f(x_1, x_2, \dots, x_k)$.

Sia A^* l'alfabeto ottenuto da A aggiungendo simboli di costante e simboli di funzione come sopra descritto, e sia φ^* la formula ottenuta da φ con le sostituzioni sopra descritte. È chiaro che φ^* è in forma di Skolem; omettiamo la dimostrazione del fatto che φ è soddisfacibile se e soltanto se φ^* è soddisfacibile.

Sia φ una formula della logica dei predicati, e sia $\overline{\varphi}$ una formula in forma normale prenessa logicamente equivalente a $\overline{\varphi}$ nella quale la parte che segue i quantificatori è in forma normale congiuntiva (cfr. teorema 3.6.16). Anche nella formula φ^* ottenuta da $\overline{\varphi}$ con l'algoritmo descritto nella dimostrazione del teorema 3.7.1 la parte che segue i quantificatori è in forma normale congiuntiva: tale formula φ^* si dice *skolemizzazione* di φ.

[14] in onore del matematico norvegese Albert Thoralf Skolem, nato a Sandsvær (l'odierna Kongsberg) il 23 maggio 1887 e morto a Oslo il 23 marzo 1963.

Esempio 3.7.2

In un opportuno linguaggio della logica dei predicati, siano x, y, z, t variabili individuali, sia P un simbolo di predicato binario e siano f, g simboli di funzioni binarie. Trasformiamo la formula

$$\varphi := (\exists x)(\exists y)(\forall z)\big(\mathrm{P}(f(x,z),z) \wedge \mathrm{P}(g(y,z),z) \wedge \big(\neg\mathrm{P}(x,z) \rightarrow (\exists t)\mathrm{P}(g(z,t),y)\big)\big)$$

in una sua skolemizzazione.

Come primo passo, troviamo una formula in forma normale prenessa logicamente equivalente a φ.

Si ha

$$\begin{aligned}\neg\mathrm{P}(x,z) \rightarrow (\exists t)\mathrm{P}(g(z,t),y) &\equiv \mathrm{P}(x,z) \vee (\exists t)\mathrm{P}(g(z,t),y) \equiv \\ &\equiv (\exists t)\big(\mathrm{P}(x,z) \vee \mathrm{P}(g(z,t),y)\big)\end{aligned}$$

e dunque

$$\varphi \equiv (\exists x)(\exists y)(\forall z)(\exists t)\big(\mathrm{P}(f(x,z),z) \wedge \mathrm{P}(g(y,z),z) \wedge \big(\mathrm{P}(x,z) \vee \mathrm{P}(g(z,t),y)\big)\big)\,.$$

Si è così ottenuta una formula in forma normale prenessa logicamente equivalente a φ nella quale la parte che segue i quantificatori è in forma normale congiuntiva.

Per procedere alla sua skolemizzazione introduciamo due simboli di costante c_1 e c_2 e un simbolo di funzione unaria h, e sostituiamo la variabile individuale x col simbolo di costante c_1, la variabile individuale y col simbolo di costante c_2 e la variabile individuale t col termine $h(z)$, ottenendo così

$$(\forall z)\big(\mathrm{P}(f(c_1,z),z) \wedge \mathrm{P}(g(c_2,z),z) \wedge \big(\mathrm{P}(c_1,z) \vee \mathrm{P}(g(z,h(z)),c_2)\big)\big)\,.$$

Sia φ un enunciato (cioè una formula chiusa) in un linguaggio di logica dei predicati. La skolemizzazione di φ è un altro enunciato φ^*, in un diverso linguaggio della logica dei predicati, nella cui scrittura compaiono soltanto quantificatori universali; la parte della scrittura di φ^* nella quale non compaiono i quantificatori è una congiunzione di disgiunzioni di formule atomiche e loro negazioni, per la quale possiamo usare la notazione in clausole introdotta nella sez. 2.7.

Esempio 3.7.3

La notazione in clausole per la parte senza quantificatori della skolemizzazione della formula considerata nell’esempio 3.7.2 è

$$\{\{\mathrm{P}(f(c_1,z),z)\},\ \{\mathrm{P}(g(c_2,z),z)\},\ \{\mathrm{P}(c_1,z),\ \mathrm{P}(g(z,h(z)),c_2)\}\}\,.$$

Esempio 3.7.4

In un opportuno linguaggio della logica dei predicati, siano x, y variabili individuali, sia P un simbolo di predicato unario, sia Q un simbolo di predicato binario e sia f un simbolo di funzione unaria. Skolemizziamo la formula

$$\big((\exists x)(\neg P(x)) \vee ((\forall x)P(x) \to (\exists y)Q(f(y), y))\big) \wedge \big((\forall x)\neg\big((\forall y)\, Q(x, y)\big)\big) .$$

In primo luogo, trasformiamola in forma normale prenessa. Essa è formata con un “ $\wedge$ ” a partire dalle due formule

$$\alpha := (\exists x)(\neg P(x)) \vee ((\forall x)P(x) \to (\exists y)Q(f(y), y))$$

e

$$\beta := (\forall x)\neg\big((\forall y)Q(x, y)\big)$$

ciascuna delle quali va trasformata in forma normale prenessa.

La formula α è formata con un “ $\vee$ ” a partire dalle due formule

$$\alpha_1 := (\exists x)(\neg P(x)) \text{ (già in FNP)}$$

$$\alpha_2 := (\forall x)P(x) \to (\exists y)Q(f(y), y) \equiv \neg(\forall x)P(x) \vee (\exists y)Q(f(y), y) \equiv$$
$$\equiv (\exists x)\neg P(x) \vee (\exists y)Q(f(y), y) \equiv (\exists x)\neg P(x) \vee (\exists x)Q(f(x), x) \equiv$$
$$\equiv (\exists x)\big(\neg P(x) \vee Q(f(x), x)\big) \text{ (ora in FNP)}$$

e dunque la forma normale prenessa di α è

$$(\exists x)\big((\neg P(x)) \vee \big(\neg P(x) \vee \mathrm{Q}(f(x), x)\big)\big)$$

che naturalmente può essere semplificata in

$$(\exists x)\big(\neg P(x) \vee \mathrm{Q}(f(x), x)\big)$$

mentre la forma normale prenessa di β è

$$(\forall x)(\exists y)\neg \mathrm{Q}(x, y) \equiv (\forall z)(\exists y)\neg \mathrm{Q}(z, y)$$

cosicché la FNP di $\alpha \wedge \beta$ è

$$(\exists x)(\forall z)(\exists y)\big(\big(\neg P(x) \vee \mathrm{Q}(f(x), x)\big) \wedge \neg \mathrm{Q}(z, y)\big)$$

Infine, per la skolemizzazione basterà introdurre una costante c in luogo di x e una funzione unaria $g(z)$ in luogo di y, ottenendo

$$(\forall z)\big(\big(\neg P(c) \vee \mathrm{Q}(f(c), c)\big) \wedge \neg \mathrm{Q}(z, g(z))\big) .$$

3.8 - Il teorema di Herbrand.

Poiché nella notazione in clausole di un enunciato sottintendiamo i quantificatori universali, in ogni clausola considerata possiamo sostituire ad ogni variabile individuale qualsiasi termine sull’alfabeto, e un insieme scritto con poche clausole ne rappresenta in realtà tantissime, spesso addirittura infinite (ad esempio, quando è infinita la struttura in cui interpretiamo il nostro alfabeto).

Inoltre, per provare che una formula φ è insoddisfacibile dovremmo verificare tutte le possibili interpretazioni di φ su tutte le possibili strutture adeguate a φ: improponibile!

Sia A un alfabeto della logica dei predicati. Si dice *universo di Herbrand* ([15]) (di A) l’insieme $\mathcal{H}$(A) i cui elementi sono

— i termini chiusi su A (cfr. sez. 3.1) se ad A appartiene almeno un simbolo di costante

oppure

— i termini chiusi su A $\cup \{c\}$ (dove c è un simbolo di costante) se ad A non appartiene alcun simbolo di costante (e quindi l’insieme dei termini chiusi su A è vuoto).

La necessità di aggiungere all’alfabeto un simbolo di costante, se non ve ne sono, allo scopo di evitare che l’universo di Herbrand risulti vuoto apparirà evidente con l’osservazione 3.8.2 ricordando che il sostegno di una struttura non può essere vuoto

Osservazione 3.8.1

Sia A un alfabeto della logica dei predicati. Se ad A appartiene almeno un simbolo di costante c e almeno un simbolo di funzione f, l’universo di Herbrand di A è infinito: ad esso infatti appartengono c, $f(c)$, $f(f(c))$, $f(f(f(c)))$, $\ldots$.

---

[15] in onore del matematico francese Jacques Herbrand, nato a Parigi il 12 febbraio 1908 e morto il 27 luglio 1931 scalando le alpi francesi.

Osservazione 3.8.2

Sia A un alfabeto della logica dei predicati, sia $\mathcal{C}_{\mathrm{A}}$ l'insieme dei simboli di costante che appartengono ad A, sia $\mathcal{F}_{\mathrm{A}}$ l'insieme dei simboli di funzione che appartengono ad A e sia $\mathcal{H}(\mathrm{A})$ l'universo di Herbrand di A.

Per ogni simbolo di funzione f di arietà k in $\mathcal{F}_{\mathrm{A}}$ si definisce in modo naturale una funzione $\overline{f} : \mathcal{H}(\mathrm{A}) \to \mathcal{H}(\mathrm{A})$ della stessa arietà k ponendo per $t_1, t_2, \ldots, t_k \in \mathcal{H}(\mathrm{A})$

$$\overline{f}(t_1, t_2, \ldots, t_k) := f(t_1, t_2, \ldots, t_k)\,.$$

Indichiamo con $\overline{\mathcal{F}_{\mathrm{A}}}$ l'insieme delle funzioni $\mathcal{H}(\mathrm{A}) \to \mathcal{H}(\mathrm{A})$ così ottenute da $\mathcal{F}_{\mathrm{A}}$.

Per ogni insieme $\mathcal{P}$ di predicati su $\mathcal{H}(\mathrm{A})$ di arietà corrispondente a quella dei simboli di predicato di A, la quaterna ordinata $(\mathcal{H}(\mathrm{A}), \mathcal{C}_{\mathrm{A}}, \overline{\mathcal{F}_{\mathrm{A}}}, \mathcal{P})$ è una struttura adeguata ad A.

Sia A un alfabeto della logica dei predicati. Si dice *interpretazione di Herbrand* per A qualsiasi interpretazione su $(\mathcal{H}(\mathrm{A}), \mathcal{C}_{\mathrm{A}}, \overline{\mathcal{F}_{\mathrm{A}}}, \mathcal{P})$ che a ogni elemento di $\mathcal{C}_{\mathrm{A}}$ associa se stesso e ad ogni elemento f di $\mathcal{F}_{\mathrm{A}}$ associa la funzione $\overline{f}$ definita nella osservazione 3.8.2 (mentre non c'è restrizione all'interpretazione dei simboli di predicato).

Teorema 3.8.3 (Herbrand)

Siano A un alfabeto per la logica dei predicati e φ una formula di A in forma di Skolem. Allora

φ è soddisfacibile se e soltanto se esiste una interpretazione di Herbrand che la soddisfa.

Dimostrazione — Omettiamo la dimostrazione di questo teorema.

3.9 - Alcuni esempi.

Esempio 3.9.1

Riprendiamo il sillogismo (ii) dell'esempio 1.1.2:

Ogni uomo è mortale.

Socrate è un uomo.

Dunque, Socrate è mortale.

Costruiamo un alfabeto per la logica dei predicati con un simbolo di costante (s) e due simboli di predicato unario U e M. Interpreteremo U(x) come “x è un uomo”, M(x) come “x è mortale” e s come Socrate. La validità del sillogismo corrisponde al fatto che

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

A sua volta, il fatto che la formula M(s) sia conseguenza logica delle due formule $(\forall x)(\mathrm{U}(x) \to \mathrm{M}(x))$ e U(s) equivale al fatto che non sia soddisfacibile la formula

$$(\forall x)(\mathrm{U}(x) \to \mathrm{M}(x)) \land \mathrm{U}(s) \land \neg\mathrm{M}(s)$$

che scriviamo come

$$(\forall x)(\neg\mathrm{U}(x) \lor \mathrm{M}(x)) \land \mathrm{U}(s) \land \neg\mathrm{M}(s)$$

e infine come

$$(\forall x)((\neg\mathrm{U}(x) \lor \mathrm{M}(x)) \land \mathrm{U}(s) \land \neg\mathrm{M}(s))$$

già in forma di skolem, con la parte che segue l’unico quantificatore in forma normale congiuntiva. Dobbiamo quindi dimostrare che è insoddisfacibile l’insieme di clausole

$$\{\{\neg\mathrm{U}(x),\, \mathrm{M}(x)\},\, \{\mathrm{U}(s)\},\, \{\neg\mathrm{M}(s)\}\}$$

in qualsiasi interpretazione di Herbrand.

Poiché l’universo di Herbrand del nostro alfabeto consiste della sola s, non abbiamo altra scelta che assegnare alla x il valore s ottenendo l’insieme di clausole

$$\{\{\neg\mathrm{U}(s),\, \mathrm{M}(s)\},\, \{\mathrm{U}(s)\},\, \{\neg\mathrm{M}(s)\}\}$$

al quale possiamo applicare l’algoritmo di Davis-Putnam:

pivot U(s):

$$\{\{\mathrm{M}(s)\},\, \{\neg\mathrm{M}(s)\}\}$$

pivot M(s):

$$\{[\,]\}$$

Avendo ottenuto la clausola vuota, si è provato che l’insieme di clausole dato non è soddisfacibile, e quindi si è dimostrata la validità del sillogismo proposto.

Esempio 3.9.2

Vediamo se dalle premesse

(i) alcuni felini sono gatti;
(ii) nessun cane è un felino;

si può dedurre logicamente che

(iii) alcuni gatti non sono cani.

Introduciamo tre simboli di predicato unario (F, G, C), che interpretiamo come segue per tradurre le premesse date:

F(x) := x è un felino;
G(x) := x è un gatto;
C(x) := x è un cane.

In questo linguaggio della logica dei predicati, le premesse diventano:

(i) $(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x))$;
(ii) $\neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x))$;

e la supposta conclusione diventa

(iii) $(\exists x)\big(\mathrm{G}(x) \wedge \neg\mathrm{C}(x)\big)$.

Dobbiamo pertanto vedere se

$$(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \models (\exists x)(\mathrm{G}(x) \wedge \neg\mathrm{C}(x)$$

ossia se la formula φ definita da

$$(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge \neg(\exists x)(\mathrm{G}(x) \wedge \neg\mathrm{C}(x)$$

è o non è soddisfacibile.

Scriviamo φ in forma normale prenessa:

$$\begin{aligned}\varphi &\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge \neg(\exists x)(\mathrm{G}(x) \wedge \neg\mathrm{C}(x) \equiv \\ &\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall x)\neg(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge (\forall x)\neg(\mathrm{G}(x) \wedge \neg\mathrm{C}(x) \equiv \\ &\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall x)(\neg\mathrm{C}(x) \vee \neg\mathrm{F}(x)) \wedge (\forall x)(\neg\mathrm{G}(x) \vee \mathrm{C}(x) \equiv \\ &\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall y)(\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\forall y)(\neg\mathrm{G}(y) \vee \mathrm{C}(y) \equiv \\ &\equiv (\exists x)(\forall y)\big((\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\neg\mathrm{G}(y) \vee \mathrm{C}(y)\big)\end{aligned}$$

La forma di Skolem di questa formula si ottiene sopprimendo il quantificatore esistenziale e sostituendo la variabile individuale x da esso vincolata con un simbolo di costante c :

$$(\forall y)\big((\mathrm{F}(c) \wedge \mathrm{G}(c)) \wedge (\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\neg\mathrm{G}(y) \vee \mathrm{C}(y)\big)$$

A questa formula resta associato il seguente schema di clausole:

$$\{\{\mathrm{F}(c)\}, \{\mathrm{G}(\mathrm{c})\}, \{\neg\mathrm{C}(y), \neg\mathrm{F}(y)\}, \{\neg\mathrm{G}(y), \mathrm{C}(y)\}\} .$$

Poiché l’universo di Herbrand consiste della sola costante c, assegniamo direttamente alla y il valore c e otteniamo

$$\{\{\mathrm{F}(c)\}, \{\mathrm{G}(\mathrm{c})\}, \{\neg\mathrm{C}(c), \neg\mathrm{F}(c)\}, \{\neg\mathrm{G}(c), \mathrm{C}(c)\}\} .$$

Possiamo applicare l’algoritmo di Davis-Putnam!

Pivot $\mathrm{F}(c)$:

clausole non contenenti né $\mathrm{F}(c)$ né $\neg\mathrm{F}(c)$: $\{\mathrm{G}(c)\}$, $\{\neg\mathrm{G}(c), \mathrm{C}(c)\}$;

$\mathrm{Ris}_{\mathrm{F}(c)}(\{\mathrm{F}(c)\}, \{\neg\mathrm{C}(c), \neg\mathrm{F}(c)\}) = \{\neg\mathrm{C}(c)\}$;

$$\{\{\neg\mathrm{C}(c)\}, \{\mathrm{G}(c)\}, \{\neg\mathrm{G}(c)\}, \{\mathrm{C}(c)\}\}\,.$$

Pivot $\mathrm{C}(c)$:

clausole non contenenti né $\mathrm{C}(c)$ né $\neg\mathrm{C}(c)$: $\{\mathrm{G}(c)\}$;

$\mathrm{Ris}_{\mathrm{C}(c)}(\{\neg\mathrm{C}(c)\}, \{\neg\mathrm{G}(c), \mathrm{C}(c)\}) = \{\neg\mathrm{G}(c)\}$;

$$\{\{\neg\mathrm{G}(c)\}, \{\mathrm{G}(c)\}\}\,.$$

Pivot $\mathrm{G}(c)$:

$\mathrm{Ris}_{\mathrm{G}(c)}(\{\neg\mathrm{G}(c)\}, \{\mathrm{G}(c)\}) = \{[\,]\}$;

$$\{[\,]\}\,.$$

Poiché abbiamo ottenuto la clausola vuota, l’insieme di clausole considerato non è soddisfacibile ed abbiamo dunque dimostrato che la (iii) è conseguenza logica delle (i) e (ii).

Esempio 3.9.3

Vediamo se dalle premesse

(i) alcuni felini sono gatti;
(ii) nessun cane è un felino;

si può dedurre logicamente che

(iii) alcuni cani non sono gatti.

Introduciamo tre simboli di predicato unario (F, G, C), che interpretiamo come segue per tradurre le premesse date:

$\mathrm{F}(x) := x$ è un felino;
$\mathrm{G}(x) := x$ è un gatto;
$\mathrm{C}(x) := x$ è un cane.

In questo linguaggio della logica dei predicati, le premesse diventano:

(i) $(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x))$;

(ii) $\neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x))$;

e la supposta conclusione diventa

(iii) $(\exists x)\big(\mathrm{C}(x) \wedge \neg\mathrm{G}(x)\big)$.

Dobbiamo pertanto vedere se

$$(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \vDash (\exists x)(\mathrm{C}(x) \wedge \neg\mathrm{G}(x)$$

ossia se la formula φ definita da

$$(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \neg\mathrm{G}(x)$$

è o non è soddisfacibile.

Scriviamo φ in forma normale prenessa:

$$\varphi \equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \neg\mathrm{G}(x) \equiv$$
$$\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall x)\neg(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge (\forall x)\neg(\mathrm{C}(x) \wedge \neg\mathrm{G}(x) \equiv$$
$$\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall x)(\neg\mathrm{C}(x) \vee \neg\mathrm{F}(x)) \wedge (\forall x)(\neg\mathrm{C}(x) \vee \mathrm{G}(x) \equiv$$
$$\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall y)(\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\forall y)(\neg\mathrm{C}(y) \vee \mathrm{G}(y) \equiv$$
$$\equiv (\exists x)(\forall y)\big((\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\neg\mathrm{C}(y) \vee \mathrm{G}(y)\big) .$$

La forma di Skolem di questa formula si ottiene sopprimendo il quantificatore esistenziale e sostituendo la variabile individuale x da esso vincolata con un simbolo di costante c :

$$(\forall y)\big((\mathrm{F}(c) \wedge \mathrm{G}(c)) \wedge (\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\neg\mathrm{C}(y) \vee \mathrm{G}(y)\big)$$

A questa formula resta associato il seguente schema di clausole:

$$\{\{\mathrm{F}(c)\}, \{\mathrm{G(c)}\}, \{\neg\mathrm{C}(y), \neg\mathrm{F}(y)\}, \{\neg\mathrm{C}(y), \mathrm{G}(y)\}\} .$$

Poiché l’universo di Herbrand consiste della sola costante c, assegniamo direttamente alla y il valore c e otteniamo

$$\{\{\mathrm{F}(c)\}, \{\mathrm{G(c)}\}, \{\neg\mathrm{C}(c), \neg\mathrm{F}(c)\}, \{\neg\mathrm{C}(c), \mathrm{G}(c)\}\} .$$

Possiamo applicare l’algoritmo di Davis-Putnam!

Pivot F(c):

clausole non contenenti né F(c) né ¬F(c): {G(c)}, {¬C(c), G(c)} ;

$\mathrm{Ris}_{\mathrm{F}(c)}(\{\mathrm{F}(c)\}, \{\neg\mathrm{C}(c), \neg\mathrm{F}(c)\}) = \{\neg\mathrm{C}(c)\}$;

$$\{\{\neg\mathrm{C}(c)\}, \{\mathrm{G}(c)\}, \{\neg\mathrm{C}(c), \mathrm{G}(c)\}\} .$$

Pivot C(c):

clausole non contenenti né C(c) né ¬C(c): {G(c)} ;
non ci sono risolventi da calcolare!

$$\{\{\mathrm{G}(c)\}\} .$$

Pivot G(c):

$$\{\} .$$

Poiché abbiamo ottenuto l'insieme vuoto, l'insieme di clausole considerato è soddisfacibile e dunque la (iii) non è conseguenza logica delle (i) e (ii).

Osservazione 3.9.4

Se non vi fidate del signor Herbrand e del suo teorema 3.8.3 (del quale in fondo non avete visto la dimostrazione!) potreste non essere convinti che nell'esempio 3.9.3 davvero la

$$(iii) \qquad (\exists x)(\mathrm{C}(x) \land \neg\mathrm{G}(x))$$

non sia conseguenza logica delle

$$(i) \qquad (\exists x)(\mathrm{F}(x) \land \mathrm{G}(x))$$

e $$(ii) \qquad \neg(\exists x)(\mathrm{C}(x) \land \mathrm{F}(x)) .$$

Dubbi analoghi certamente non possono esservi venuti guardando gli esempi 3.9.1 e 3.9.2, nei quali l'algoritmo di Davis e Putnam ha prodotto una esplicita confutazione; mentre nell'esempio 3.9.3 potreste sospettare che una diversa assegnazione di valore alla variabile individuale y possa dar luogo a una contraddizione fra le clausole così generate.

Convinciamoci allora che davvero la (iii) non è conseguenza logica delle (i) e (ii) applicando la definizione (vista nella sez. 3.5), cioè costruendo una struttura adeguata al nostro linguaggio e una interpretazione nella quale valgono le (i) e (ii) ma non vale la (iii).

Nell'insieme $\mathbb{N}$ dei numeri naturali, interpretiamo i simboli di predicato unario F, G e C come segue:

F(x) := x è un numero pari;
G(x) := x è un multiplo di 3;
C(x) := x è una potenza di 3.

Con questa interpretazione in $\mathbb{N}$ valgono la (i) (alcuni numeri pari sono multipli di 3, ad esempio 6 è pari ed è multiplo di 3) e la (ii) (nessuna potenza di 3 è un numero pari), ma non vale la (iii) (se valesse, qualche potenza di 3 non sarebbe multiplo di 3: invece sappiamo che ogni potenza di 3 è necessariamente multiplo di 3).

Esempio 3.9.5

Vediamo se dalle premesse

(i) alcuni felini sono gatti;
(ii) nessun cane è un felino;

si può dedurre logicamente che

(iii) nessun gatto è un cane.

Introduciamo tre simboli di predicato unario (F, G, C), che interpretiamo come segue per tradurre le premesse date:

F(x) := x è un felino;
G(x) := x è un gatto;
C(x) := x è un cane.

In questo linguaggio della logica dei predicati, le premesse diventano:

(i) $(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x))$;

(ii) $\neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x))$;

e la supposta conclusione diventa

(iii) $\neg(\exists x)\big(\mathrm{G}(x) \wedge \mathrm{C}(x)\big)$.

Dobbiamo pertanto vedere se

$$(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \vDash \neg(\exists x)\big(\mathrm{G}(x) \wedge \mathrm{C}(x)\big)$$

ossia se la formula φ definita da

$$(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge \neg\neg(\exists x)\big(\mathrm{G}(x) \wedge \mathrm{C}(x)\big)$$

è o non è soddisfacibile.

Scriviamo φ in forma normale prenessa:

$$\varphi \equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge \neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge (\exists x)\big(\mathrm{G}(x) \wedge \mathrm{C}(x)\big) \equiv$$

$$\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall x)\neg(\mathrm{C}(x) \wedge \mathrm{F}(x)) \wedge (\exists x)\big(\mathrm{G}(x) \wedge \mathrm{C}(x)\big) \equiv$$

$$\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall x)(\neg\mathrm{C}(x) \vee \neg\mathrm{F}(x)) \wedge (\exists x)\big(\mathrm{G}(x) \wedge \mathrm{C}(x)\big) \equiv$$

$$\equiv (\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\forall y)(\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\exists z)\big(\mathrm{G}(z) \wedge \mathrm{C}(z)\big) \equiv$$

$$\equiv (\exists x)(\exists z)(\forall y)\big((\mathrm{F}(x) \wedge \mathrm{G}(x)) \wedge (\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\mathrm{G}(z) \wedge \mathrm{C}(z)\big)$$

La forma di Skolem di questa formula si ottiene sopprimendo i due quantificatori esistenziali e sostituendo: la variabile individuale x vincolata dal primo di essi con un simbolo di costante c, la variabile individuale z vincolata dal secondo di essi con un (diverso) simbolo di costante d:

$$(\forall y)\big((\mathrm{F}(c) \wedge \mathrm{G}(c)) \wedge (\neg\mathrm{C}(y) \vee \neg\mathrm{F}(y)) \wedge (\mathrm{G}(d) \wedge \mathrm{C}(d)\big)$$

A questa formula resta associato il seguente schema di clausole:

$$\{\{\mathrm{F}(c)\}, \{\mathrm{G}(\mathrm{c})\}, \{\neg\mathrm{C}(y), \neg\mathrm{F}(y)\}, \{\mathrm{G}(d)\}, \{\mathrm{C}(d)\}\} .$$

Poiché l’universo di Herbrand consiste delle sole due costanti c e d, assegniamo alla y il valore c e poi anche il valore d, ottenendo

$$\{\{\mathrm{F}(c)\}, \{\mathrm{G}(\mathrm{c})\}, \{\neg\mathrm{C}(c), \neg\mathrm{F}(c)\}, \{\neg\mathrm{C}(d), \neg\mathrm{F}(d)\}, \{\mathrm{G}(d)\}, \{\mathrm{C}(d)\}\} .$$

Possiamo applicare l’algoritmo di Davis-Putnam!

Pivot $\mathrm{F}(c)$:

clausole non contenenti né $\mathrm{F}(c)$ né $\neg\mathrm{F}(c)$: $\{\mathrm{G}(c)\}, \{\neg\mathrm{C}(d), \neg\mathrm{F}(d)\}, \{\mathrm{G}(d)\}, \{\mathrm{C}(d)\}$;

$\mathrm{Ris}_{\mathrm{F}(c)}(\{\mathrm{F}(c)\}, \{\neg\mathrm{C}(c), \neg\mathrm{F}(c)\}) = \{\neg\mathrm{C}(c)\}$

$$\{\{\neg\mathrm{C}(c)\}, \{\mathrm{G}(c)\}, \{\neg\mathrm{C}(d), \neg\mathrm{F}(d)\}, \{\mathrm{G}(d)\}, \{\mathrm{C}(d)\}\} .$$

Pivot C(d):

clausole non contenenti né C(d) né ¬C(d): {¬C(c)}, {G(c)}, {G(d)} ;

$\mathrm{Ris}_{\mathrm{C}(d)}(\{\neg \mathrm{C}(d), \neg \mathrm{F}(d)\}, \{\mathrm{C}(d)\}) = \{\neg \mathrm{F}(d)\}$

$$\{\{\neg \mathrm{C}(c)\}, \{\mathrm{G}(c)\}, \{\mathrm{G}(d)\}, \{\neg \mathrm{F}(d)\}\}\,.$$

Pivot G(c):

$$\{\{\neg \mathrm{C}(c)\}, \{\mathrm{G}(d)\}, \{\neg \mathrm{F}(d)\}\}\,.$$

Pivot G(d):

$$\{\{\neg \mathrm{C}(c)\}, \{\neg \mathrm{F}(d)\}\}\,.$$

Pivot C(c):

$$\{\{\neg \mathrm{F}(d)\}\}\,.$$

Pivot F(d):

$$\{\}\,.$$

Poiché abbiamo ottenuto l’insieme vuoto, l’insieme di clausole considerato è soddisfacibile e dunque la (iii) non è conseguenza logica delle (i) e (ii).

Osservazione 3.9.6

Ancora non vi fidate del signor Herbrand e del suo teorema 3.8.3? Avete il dubbio che nell’esempio 3.9.5 la

(iii) $\neg(\exists x)(\mathrm{G}(x) \wedge \mathrm{C}(x))$

possa invece essere conseguenza logica delle

(i) $(\exists x)(\mathrm{F}(x) \wedge \mathrm{G}(x))$

e (ii) $\neg(\exists x)(\mathrm{C}(x) \wedge \mathrm{F}(x))$?

Convinciamoci che la (iii) non è conseguenza logica delle (i) e (ii) utilizzando la stessa struttura e la stessa interpretazione che abbiamo scelto nell’osservazione 3.9.4 (in fondo il linguaggio di logica dei predicati utilizzato nell’esempio 3.9.5 è proprio lo stesso che era stato utilizzato nell’esempio 3.9.3).

Nell’insieme $\mathbb{N}$ dei numeri naturali, interpretiamo i simboli di predicato unario F, G e C come segue:

F$(x) :=$ x è un numero pari;
G$(x) :=$ x è un multiplo di 3;
C$(x) :=$ x è una potenza di 3.

Con questa interpretazione in $\mathbb{N}$ valgono la (i) (alcuni numeri pari sono multipli di 3, ad esempio 6 è pari ed è multiplo di 3) e la (ii) (nessuna potenza di 3 è un numero pari), ma non vale la (iii) (infatti esistono numeri che sono multipli di 3 e anche potenza di 3...).