Logica

Luca Telloli

12 febbraio 2003

Indice

Capitolo 1

Note introduttive

1.1 Introduzione

In questo corso verranno studiate due classi di linguaggi:

- *linguaggi proposizionali*
- *linguaggi del primo ordine (o predicativi)*

Ogni linguaggio é caratterizzato da elementi ricorrenti, che verranno richiamati di volta in volta, e che sono:

- **Vocabolario (o lessico)**: insieme fissato e finito di elementi del linguaggio
- **Sintassi**: regole di corretta formazione delle stringhe (o *frasi*) del linguaggio. Le regole sono finite e ricorsive
- **Semantica**: attribuzione di significato alle stringhe ben formate

I linguaggi proposizionali non nascono in maniera artificiale, ma dalla riflessione e dall'indagine sui linguaggi naturali, che quotidianamente parliamo.

1.2 Alberi.

Definizione: si definisce **albero** (e intenderemo un *albero* finito) un insieme finito e ordinato con una relazione d'ordine parziale; questo significa che non tutti gli elementi dell'albero sono *confrontabili* tra loro.

Un albero é detto *banale* se composto da un unico elemento, detto **radice dell'albero**. Al contrario, é detto *non banale* quando contiene più elementi; esso é composto da una radice (l'elemento più piccolo secondo la relazione d'ordine assegnata) e da altri nodi, detti **successori**.

Si dicono **nodi terminali** (o *foglie*) i nodi che non hanno successori. Gli altri sono detti **nodi non terminali**. Il numero di successori di un nodo é detto **molteplicità** del nodo. Si dice infine **ramo** l'insieme (totalmente ordinato!) di nodi il cui primo elemento é la radice, e l'ultimo é una foglia. É intuitivo che un albero *ha tanti rami quanti sono i suoi nodi terminali.* Si definiscono infine *lunghezza del ramo, o cardinalità* il numero di nodi che esso percorre, diminuito di 1; e *profondità dell'albero* il valore massimo nell'insieme delle cardinalità dell'albero.

Relazione di dominanza tra nodi

Dato un albero non banale, e due nodi X, Y appartenenti ad esso, si dice che X **domina** Y quando i due nodi sono sullo stesso ramo, e la profondit di X é minore di quella di Y [1].

Relazione di C - comando

Dato un albero non banale, e due nodi X, Y appartenenti ad esso, dicendo che X **C–comanda** Y (o *comanda per costituenti*) si afferma che siano verificate tutte le seguenti condizioni:

1. $X \neq Y$
2. X non domina Y, né Y domina X. Per cui X, Y non sono sullo stesso ramo, cioé esiste ad un livello superiore un nodo che biforca
3. Il nodo biforcante più profondo che domina X domina anche Y

[1] *Osservazione*: Ciò significa che X precede Y nella relazione d'ordine dell'albero

Capitolo 2

I linguaggi proposizionali

Evidenziamo gli elementi di un linguaggio proposizionale:

2.1 Vocabolario

1. **Frasi atomiche**: É dato un insieme $\mathcal{A}$, finito od infinito, non vuoto, i cui elementi sono detti *frasi atomiche* del linguaggio, e vengono indicati con lettere alfabetiche minuscole $\{a, b, c, \ldots\}$. L'insieme delle frasi atomiche é insieme che caratterizza un particolare linguaggio proposizionale.

2. **Connettivi**:

Simbolo	Denominazione	Significato
$\neg$	*negazione*	non
$\wedge$	*congiunzione*	e
$\vee$	*disgiunzione*	o, non esclusivo
$\rightarrow$	*implicazione*	se … allora …

3. **Simboli ausiliari**:

Simbolo	Denominazione
$\perp$	*assurdo*
(	*parentesi aperta sinistra*
)	*parentesi chiusa destra*

2.2 Sintassi del linguaggio.

La sintassi del linguaggio fornisce le regole di formazione delle stringhe. Tali regole sono di tipo ricorsivo.

Definizione 2.1 (linguaggio) *Si definisce **linguaggio**, e lo si indica con $\mathcal{L}_{\mathcal{A}}$ l'insieme delle **stringhe ben formate** (o* frasi*) ottenute dall'insieme di frasi atomiche $\mathcal{A}$ mediante l'utilizzo delle regole della sintassi.*

Indicheremo in seguito le singole frasi di un determinato linguaggio con l'utilizzo di lettere greche $\{\alpha, \beta, \gamma, ...\}$. É immediato osservare che insiemi differenti di frasi atomiche generano differenti linguaggi.

$\mathcal{L}_{\mathcal{A}}$ é il più piccolo insieme tale che:

1. L'assurdo appartiene a $\mathcal{L}_{\mathcal{A}}$:

$$\perp \in \mathcal{L}_{\mathcal{A}}$$

2. Le frasi atomiche appartengono a $\mathcal{L}_{\mathcal{A}}$

$$\mathcal{A} \subset \mathcal{L}_{\mathcal{A}}$$

3. Insieme delle regole ricorsive:
   - se $\alpha \in \mathcal{L}_{\mathcal{A}}$ allora $(\neg\alpha) \in \mathcal{L}_{\mathcal{A}}$
   - se $\alpha, \beta \in \mathcal{L}_{\mathcal{A}}$ allora $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta) \in \mathcal{L}_{\mathcal{A}}$

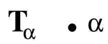
Osservazioni. [FARE] ◀

▷ `Esercizio.`
Dimostrare che se una stringa é ben formata, ci sono tante parentesi aperte a sinistra quante parentesi chiuse a destra. ◁

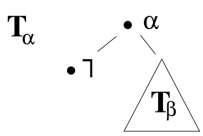
Ad ogni frase α del linguaggio é possibile associare un albero, detto *albero sintagmatico di* α, indicato con $\mathcal{T}_\alpha$ che permette di risalire alla struttura atomica della frase.

▽ *Dimostrazione.*
La dimostrazione utilizza il procedimento induttivo:

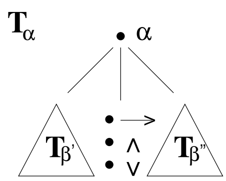
1. nel caso di una frase atomica α:



2. nel caso di una frase del tipo $\alpha = \neg\beta$:



3. nel caso di una frase del tipo $\alpha = \beta' \rightarrow \beta''$, oppure $\alpha = \beta' \wedge \beta''$, oppure $\alpha = \beta' \vee \beta''$



△

Osservazioni. Ad esempio, l'albero sintagmatico per l'espressione:

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

é la seguente:
[TODO] ◀

Si osservi che nei nodi terminali compaiono soltanto connettivi, o frasi atomiche.

Osservazioni.

- Se $\mathcal{A}', \mathcal{A}''$ sono insiemi di frasi atomiche, e $\mathcal{A}' \subset \mathcal{A}''$, allora vale che $\mathcal{L}_{\mathcal{A}'} \subset \mathcal{L}_{\mathcal{A}''}$.
- Dato $\alpha \in \mathcal{L}_{\mathcal{A}}$, se ne costruisca l'albero $\mathcal{T}_\alpha$. Si indichi con $\mathcal{A}_\alpha$ l'insieme dei nodi di $\mathcal{T}_\alpha$ che sono in $\mathcal{A}$. Valgono le seguenti affermazioni:
  1. $\mathcal{A}_\alpha$ é un insieme finito. Un linguaggio ha necessariamente infinite frasi, poiché é definito in maniera ricorsiva, ma ogni singola frase del linguaggio ha un numero finito di nodi terminali, e di conseguenza dipende da un numero finito di frasi atomiche.
  2. $\alpha \in \mathcal{L}_{\mathcal{A}\alpha}$, cioé la frase α é contenuta nel linguaggio generato dall'insieme di frasi atomiche che la compongono.

◀

2.3 Semantica del linguaggio.

2.3.1 Interpretazione del linguaggio.

Definizione 2.2 (interpretazione del linguaggio) *Dato un insieme di frasi atomiche $\mathcal{A}$ e il linguaggio $\mathcal{L}_{\mathcal{A}}$ ad esso associato, si definisce **interpretazione del linguaggio** il criterio utilizzato per attribuire un valore di verità alle sue frasi.*

Formalmente si tratta di una funzione $v : \mathcal{L}_{\mathcal{A}} \to \{0, 1\}$ ove si attribuisce a 0 il significato di 'falso' e ad 1 il significato di 'vero'. Tale funzione é costituita da:

- Una funzione arbitraria $v : \mathcal{A} \to \{0, 1\}$, che attribuisce un valore di verità alle *frasi atomiche* del linguaggio
- Un insieme di *regole semantiche ricorsive* che permettono di attribuire un valore di verità a tutte le frasi del linguaggio $\mathcal{L}_{\mathcal{A}}$. In seguito assumeremo valide le seguenti regole semantiche:
  1. $v(\perp) = 0$. L'assurdo é falso.

2. $v(\neg a) = 1 - v(a)$. Noto anche come *principio di non contraddizione.*
3. $v(\alpha \wedge \beta) = v(\alpha) \cdot v(\beta) = min\{v(\alpha), v(\beta)\}$
4. $v(\alpha \vee \beta) = max\{v(\alpha), v(\beta)\}$
5. $v(\alpha \rightarrow \beta) = max\{1 - v(\alpha), v(\beta)\}$

Osservazioni.

- La disgiunzione é definita in maniera *non esclusiva*, corrispondentemente al latino `vel`, anziché dell'esclusivo `aut`.
- Si osservi la definizione di implicazione: l'affermazione che α vera implica β falsa é inaccettabile, e genera valore 'falso'. L'affermazione α falsa implica β vera é poco intuitiva, ma viene inclusa nella definizione: essa garantisce *stabilità nell'aggiunta di nuove ipotesi*, cioé l'inclusione di una frase falsa in un insieme di frasi che implica una frase vera, non pregiudica la verità dell'enunciato a secondo membro. Si consideri ad esempio l'affermazione:

  oggi piove.

  Se la frase é vera si consideri l'implicazione:

  se ieri era giovedì oggi piove

  la condizione che ieri fosse o meno giovedì non pregiudica il valore di verità della conseguenza.

◀

L'interpretazione di una frase qualunque del linguaggio dipende in maniera diretta dall'interpretazione delle frasi atomiche. A questo proposito si enunciano i seguenti teoremi:

Teorema 2.3.1 *Sia dato un linguaggio $\mathcal{L}_{\mathcal{A}}$ e due interpretazioni v', v'' per esso, che soddisfano le regole semantiche sopra esposte. Se:*

$$v'(a) = v''(a), \quad \forall a \in \mathcal{A}$$

allora $v' = v''$, cioé le due interpretazioni sono uguali.

Un'interpretazione di un linguaggio é dunque individuata in maniera non ambigua quando é nota una interpretazione delle frasi atomiche in esso.

$\triangledown$ *Dimostrazione.*
Sia dato un insieme $\mathcal{F}$ di frasi del linguaggio $\mathcal{L}_{\mathcal{A}}$, e due interpretazioni v', v'', per le quali vale che: $v'(a) = v''(a), \forall a \in \mathcal{A}$. L'insieme $\mathcal{F}$ sia così definito: $\mathcal{F} = \{\alpha \in \mathcal{L}_{\mathcal{A}} : v'(\alpha) = v''(\alpha)\}$ cioé l'insieme delle frasi del linguaggio che hanno lo stesso valore per entrambe le interpretazioni. É sufficiente dimostrare che: $\mathcal{F} = \mathcal{L}_{\mathcal{A}}$, e tale dimostrazione si effettua mediante ricorsione. Si può innanzitutto affermate che l'insieme $\mathcal{F}$ non é vuoto: infatti esso contiene l'assurdo, poiché esso é falso per ogni interpretazione, e per ipotesi esso contiene anche le frasi atomiche dell'alfabeto del linguaggio, cioé $\mathcal{A} \subset \mathcal{F}$. Ora:

- se $\alpha \in \mathcal{F}$, segue: $v'(\neg\alpha) = 1 - v'(\alpha) = 1 - v''(\alpha) = v''(\neg\alpha)$
- se $\alpha, \beta \in \mathcal{F}$, segue che: $v'(\alpha \wedge \beta) = max\{v'(\alpha), v'(\beta)\} = max\{v''(\alpha), v''(\beta)\} = v''(\alpha \wedge \beta)$.

I restanti casi sono lasciati per esercizio. $\triangle$

Teorema 2.3.2 *Sia data una funzione a valori arbitrari $\tilde{v} : \mathcal{A} \to \{0, 1\}$. Allora esiste certamente una intepretazione v dell'intero linguaggio per la quale:*

$$v(a) = \tilde{v}(a) \quad \forall a \in \mathcal{A}$$

C'e' dunque almeno una interpretazione che interpreta il linguaggio come la funzione arbitraria assegnata. Per il teorema 2.3.1, l'interpretazione dell'intero linguaggio coincide. Tale interpretazione del resto e' unica, come conseguenza del teorema 2.3.2

Osservazioni. Per quel che riguarda il secondo teorema, si consideri un qualunque generico insieme *finito* di frasi atomiche: $\mathcal{A} = \{a_1, a_2, \ldots a_N\}$, con $N \geq 1$. Le possibili interpretazioni $\tilde{v}$ delle frasi atomiche di questo insieme sono 2^N; ognuna di queste interpretazioni genera una differente interpretazione dell'intero linguaggio.

Nel caso di insieme infinito di frasi atomiche, anche l'insieme delle possibili interpretazioni é infinito. ◀

Definizione 2.3 (tautologia) *Si definisce **tautologia** una frase vera in ogni interpretazione:*

$$\alpha : v(\alpha) = 1 \quad \forall \text{ interpretazione di } v$$

Definizione 2.4 (contraddizione) *Si definisce **contraddizione** una frase falsa in ogni interpretazione:*

$$\alpha : v(\alpha) = 0 \quad \forall \text{ interpretazione di } v$$

Osservazioni.

- la frase $(a \vee \neg a)$ una *tautologia*
- la frase $(a \wedge \neg a)$ una *contraddizione*

◀

2.3.2 Soddisfacibilità e consequenzialità semantica.

Definizione 2.5 (soddisfacibilità/insoddisfacibilità) *Sia dato $\Gamma \subset \mathcal{L}_{\mathcal{A}}$, insieme di frasi non vuoto. L'insieme Γ si dice **soddisfacibile** se esiste almeno una interpretazione v del linguaggio tale che*

$$v(\alpha) = 1 \qquad \forall \alpha \in \Gamma$$

*In caso contrario l'insieme é detto **insoddisfacibile** e non esiste nessuna interpretazione che renda vero l'insieme Γ*

Osservazioni. Un tipico esempio di insieme insoddisfacibile é *l'intero linguaggio* che, contenendo la negazione, contiene necessariamente almeno una frase la cui interpretazione é falsa. ◀

Osservazioni. Siano Γ', Γ'' due insiemi di frasi dello stesso linguaggio. Valgono allora le seguenti affermazioni, facilmente dimostrabili:

- Se Γ' é insoddisfacibile, e $\Gamma' \subset \Gamma''$, allora anche Γ'' é insoddisfacibile.
- Se Γ' é soddisfacibile, e $\Gamma'' \subset \Gamma'$, allora anche Γ'' é soddisfacibile.

◀

Osservazioni. Siano dati i due insiemi:

$$\Gamma' = \{(a \vee \neg a)\}, \Gamma'' = \{(a \vee \neg a), ((a \vee \neg a) \wedge b)\}$$

Il primo insieme é certamente soddisfacibile, poiché esso contiene una sola frase, e questa é una tautologia. Esiste poi almeno una interpretazione (tutte quelle per le quali $v(b) = 1$ che rende soddisfacibile anche il secondo insieme. Il primo insieme é contenuto nel secondo: se ne poteva dimostrare la soddisfacibilità anche in base a questa osservazione. ◀

Definizione 2.6 (conseguenza semantica) *Dato un insieme $\mathcal{C}$ di frasi di un qualche linguaggio, ed una frase α, non necessariamente in $\mathcal{C}$, si dice che α é* ***conseguenza semantica*** *di $\mathcal{C}$, e lo si indica con la simbologia*

$$\mathcal{C} \models \alpha$$

quando per ogni interpretazione v che rende vere le frasi di $\mathcal{C}$ si verifica che $v(\alpha) = 1$, cioe' anche α, con la stessa interpretazione, é vera.

Si può ottenere una *definizione equivalente* di conseguenza semantica, osservando che l'insieme $\{\mathcal{C} \cup (\neg\alpha)\}$ é sempre insoddisfacibile, quando α é conseguenza semantica di $\mathcal{C}$. Dunque anche la nozione appena presentata coinvolge quella più ampia di soddisfacibilità di un insieme di frasi; in seguito osserveremo che la soddisfabilità costituisce una nozione chiave. Si é visto in precedenza che le possibili interpretazioni di un linguaggio con un insieme finito di N frasi atomiche sono in numero di 2^N, cioé crescono con stima esponenziale. Allo stesso tempo, si può osservare che l'insoddisfacibilità di un insieme equivale alla ricerca all'interno di esso di almeno una frase falsa per ogni interpretazione: l'ordine di grandezza di un possibile algoritmo di decidibilità, nel caso peggiore (cioé quando esiste una sola interpretazione che soddisfa l'insieme, e tale interpretazione é l'ultima testata dall'algoritmo) resta di ordine esponenziale. Se effettivamente questo algoritmo sia di ordine esponenziale o di un ordine minore (presumibilmente polinomiale) non é ancora stato dimostrato; tuttavia vedremo in seguito le differenti strade intraprese per risolvere il problema della decidibilità.

Osservazioni. [TODO] ◀

2.3.3 Equivalenza semantica e completezza funzionale.

Definizione 2.7 (equivalenza semantica) *Due frasi α, β si dicono* ***semanticamente equivalenti****, e lo si indica con la scrittura $\alpha \equiv \beta$, quando si verifica che:*

$$v(\alpha) = v(\beta) \quad \forall v$$

cioé se le due frasi hanno lo stesso valore di verità per ogni interpretazione del linguaggio.

Enunciamo di seguito alcune equivalenze semantiche notevoli: si osservi che le frasi sono spesso sintatticamente molto differenti tra loro:

1. $\neg(\neg\alpha) \equiv \alpha$

2. *(Proprietà commutative)*:

$$\alpha \wedge \beta \equiv \beta \wedge \alpha$$

$$\alpha \vee \beta \equiv \vee \wedge \alpha$$

3. *(Leggi di Boole)*:

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

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

4. *(Proprietà associative)*:

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

$$\alpha \vee (\beta \vee \gamma) \equiv (\alpha \vee \beta) \wedge \gamma$$

5. *(Proprietà distributive)*:

$$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

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

$\triangledown$ *Dimostrazione.*
Dimostriamone alcune (le altre per esercizio):

- $v(\neg(\neg\alpha)) = 1 - v(\neg\alpha) = 1 - (1 - v(\alpha)) = v(\alpha)$
- (Prima legge di Boole):

$$v((\neg\alpha) \wedge \beta) = 1 - v(\alpha \wedge \beta) = 1 - min\{v(\alpha), v(\beta)\}$$

  che equivale a :

$$v(\neg\alpha \vee \neg\beta) = max\{v(\neg\alpha), v(\neg\beta)\} = max\{1 - v(\alpha), 1 - v(\beta)\}$$

  infatti:

$v(\alpha)$	$v(\beta)$	$1 - min\{v(\alpha), v(\beta)\}$	$max\{1 - v(\alpha), 1 - v(\beta)\}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

$\triangle$

Nella nostra definizione di sintassi, abbiamo ammesso l'esistenza di quattro connettivi logici: essi costituiscono il miglior compromesso tra chiarezza espositiva e necessità sintattiche. Esistono linguaggi che utilizzano un numero minore di connettivi logici ed é possibile dimostrare l'equivalenza tra insiemi differenti di connettivi.

Definizione 2.8 (completezza funzionale) *Un insieme di connettivi si dice funzionalmente completo quando mediante essi é possibile rappresentare ogni elemento dell'insieme iniziale di connettivi.*

Osservazioni. E' possibile definire un linguaggio che faccia uso dei soli connettivi $(\neg, \wedge)$, per il quale varranno le seguenti equivalenze semantiche:

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

$$\alpha \rightarrow \beta \equiv (\neg\alpha) \vee \beta = \neg(\neg(\neg\alpha) \wedge (\neg\beta)) = \neg(\alpha \wedge (\neg\beta))$$

◀

Altri esempi sono costituiti dalle coppie di connettivi:

$$\{\neg, \rightarrow\} \quad \{\neg, \vee\}$$

Si osservi che la negazione é sempre presente in tutte le coppie, ed é possibile dimostrare che non é possibile creare una sintassi equivalente a quella finora proposta senza utilizzare questo connettivo.

2.3.4 Dal linguaggio naturale al linguaggio proposizionale.

Un frammento di lingua italiana può essere un buon esempio di come sia possibile formalizzare un linguaggio naturale: *formalizzare* non é lo stesso di *creare un linguaggio proposizionale*, come sarà più chiaro tra breve. Costruiamo dunque un semplice linguaggio formale che rispecchii alcune strutture della lingua italiana (*grammatica non contestuale, o* `context-free`). Esso sarà costituito da un insieme di **regole sintattiche**:

1. *Regole ricorsive*: da enunciati E', E" si può ottenere:
   - neg E', neg E
   - E' cong E", E' disg E
   - Se E' allora E"
2. *Regole non ricorsive*: gli enunciati sono nella forma NSv, dove N é un *nome* e Sv é un *sintagma verbale* che può essere del tipo:

- V_i, Verbo intransitivo
- $V_t N$, Verbo transitivo, Nome

e da un **vocabolario**, dal quale é possibile estrarre gli elementi da combinare mediante le regole appena esposte; ad esempio:

- N:{Antonio, Lucia, il gatto Tom, il topo Jerry}
- V_i={corre, dorme, gioisce, brontola}
- V_t={ama, odia, saluta, mangia}

Ci si può chiedere, a questo punto, se un certo enunciato appartiene al linguaggio appena introdotto; affermiamo che l'enunciato appartiene al linguaggio se é possibile derivarlo a partire dal vocabolario e dalle regole sintattiche date.

Ad esempio, l'enunciato *Antonio ama Lucia o il gatto Tom dorme* si può ottenere come segue:
E=E' disg E" (regola ricorsiva)
E'=Antonio ama Lucia = $N^I\ V_t^I\ N^{II}$ (regola non ricorsiva)
E"=il gatto Tom dorme = $N^{III}\ V_i^I$ (regola non ricorsiva)

da cui gli insiemi:

$$\{N^I, N^{II}, N^{III}\} \subset N$$

$$\{V_t^I\} \subset V_t$$

$$\{V_i^I\} \subset V_i$$

Quest'ultimo esempio e' privo di ambiguità, ma si veda ora l'esempio seguente: *Antonio ama Lucia o non si dà il caso che il gatto Tom dorme e il topo Jerry corre.*

Questo enunciato può essere derivato in due modi differenti:

I modo:
$E = E^I$ disg $E^I I$
E^I = Antonio ama Lucia
$E^I = E^{III}$ cong E^{IV}
E^{III} = neg E^V
E^V = il gatto Tom dorme
E^{IV} = il topo Jerry corre

II modo:
$E = E^I$ cong E^{II}
$E^I = E^{III}$ disg E^{IV}
E^{III} = Antonio ama Lucia
E^{IV} = neg E^V
E^V = il gatto Tom dorme
E^{II} = il topo Jerry corre

La mancanza di parentesi in un linguaggio naturale genera ambiguità che pongono alcuni problemi nella traduzione in un linguaggio proposizionale. Tentiamo di farlo con il linguaggio appena introdotto: la relazione tra i connettivi proposizionali e le congiunzioni naturali é immediata; le frasi atomiche del linguaggio saranno tutti gli enunciati del linguaggio naturale nella forma NSv.

La frase *Antonio ama Lucia o non si dà il caso che il gatto Tom dorme e il topo Jerry corre*, quando si sia ammessa la corrispondenza:

$$\begin{array}{c} \text{a} \longleftrightarrow \text{Antonio ama Lucia} \\ \text{b} \longleftrightarrow \text{il gatto Tom dorme} \\ \text{c} \longleftrightarrow \text{il topo Jerry corre} \end{array}$$

genera le seguenti traduzioni in linguaggio proposizionale:

$$\begin{array}{c} a \vee \neg(b \wedge c) \\ a \vee ((\neg b) \wedge c) \\ (a \vee (\neg b)) \wedge c \end{array}$$

La corrispondenza tra frasi del linguaggio naturale e frasi del linguaggio proposizionale non é perciò di 1 a 1 bensí, nel caso più generale, di uno a molti. Possiamo affermare che una frase del linguaggio naturale ed una del linguaggio proposizionale sono in relazione tra di esse se uno dei possibili alberi di derivazione di E é traducibile in una frase α del linguaggio proposizionale.

Osservazioni. Verificare che:

$$\{a \to b, \neg c \to \neg b\} \models a \to c$$

Questo significa dimostrare che l'insieme $\{\alpha, \beta, \gamma\}$ con:

$$\alpha = a \to b \quad \beta = \neg c \to \neg b \quad \gamma = \neg(a \to c)$$

é insoddisfacibile.

E' sufficiente verificare che per ogni interpretazione possibile, almeno una delle tre frasi contenute in questo insieme é falsa:

a	b	c	α	β	γ
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	1	1	0

◀

▷ `Esercizio.`

Verificare che:

$$\{a \to b, a, (b \wedge c) \to d\} \models d$$

◁

2.3.5 Forme normali.

Le equivalenze semantiche introdotte in precedenza dimostrano che e' sempre possibile trasformare enunciati del linguaggio in altri sintatticamente differenti, ma semanticamente equivalenti. Esistono particolari trasformazioni che si possono applicare alle frasi di un linguaggio, il cui scopo e' quello di ottenere un enunciato semanticamente equivalente al primo, che rispetti una *forma* precisa; tale forma puo' essere successivamente elaborata in maniera automatica, mediante regole semplici e spesso meccaniche.

Definizione 2.9 (letterale) *Si definisce* ***letterale*** *una frase atomica o la sua negazione. Ad esempio, tutti i seguenti termini sono letterali:* $\perp, a, b, ..., \neg \perp, \neg a, \neg b$.

Definizione 2.10 (forma normale congiuntiva) *Una frase* γ *di un linguaggio proposizionale si dice in* ***forma normale congiuntiva*** *quando é scritta come congiunzione di un numero finito di frasi* q_i*, ciascuna delle quali é disgiunzione di letterali.*

Ad esempio, la frase seguente e' in forma normale congiuntiva:

$$\gamma = \underbrace{(a \vee \neg b)}_{q_1} \wedge \underbrace{(c \vee \neg d \vee e)}_{q_2} \wedge \underbrace{(\neg a \vee b)}_{q_3}$$

Definizione 2.11 (forma normale disgiuntiva) *Analogamente, una frase* δ *di un linguaggio proposizionale si dice in* ***forma normale disgiuntiva*** *quando é scritta come disgiunzione di un numero finito di frasi* q_j*, ciascuna delle quali é congiunzione di letterali.*

Ad esempio, la frase seguente e' in forma normale disgiuntiva:

$$\delta = \underbrace{(\neg a \wedge \neg b)}_{q_1} \vee \underbrace{(\neg c \wedge d)}_{q_2} \vee \underbrace{(\neg e \wedge \neg a \wedge b)}_{q_3}$$

Teorema 2.3.3 *Data una qualunque frase* $\alpha \in \mathcal{L}_{\mathcal{A}}$*:*

1. *Esiste almeno una frase* γ' *in forma normale congiuntiva tale che* $\alpha \equiv \gamma'$
2. *Esiste almeno una frase* γ'' *in forma normale disgiuntiva tale che* $\alpha \equiv \gamma''$

Osservazioni. Data la frase $\alpha = \neg(a \rightarrow (b \rightarrow \neg c))$, la si riscrive in forma normale congiuntiva e disgiuntiva.Ricordando alcune equivalenze semantiche introdotte in precedente, si possono effettuare successivamente i seguenti passaggi:

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

$$\equiv a \wedge b \wedge c$$

La frase cosí ottenuta é già in entrambe le forme:

- $a \wedge b \wedge c$, congiunzione di tre frasi atomiche
- $(a \wedge b \wedge c)$, disgiunzione di un'unica congiunzione di letterali

◀

Osservazioni. $(a \wedge \neg b) \rightarrow c \equiv \neg(a \wedge \neg b) \vee c \equiv \neg a \vee b \vee c$ (f.n.c, f.n.d.)
$(a \vee \neg b) \rightarrow c \equiv \neg(a \vee \neg b) \vee c \equiv (\neg a \vee c) \wedge (b \vee c)(f.n.c.)$ ◀

Una procedura automatica di normalizzazione.

Esiste una procedura *automatica* per convertire una frase α nella propria forma normale congiuntiva/disgiuntiva: a partire dalla tabella della verità della frase stessa:

- per ottenere una *forma normale disgiuntiva* equivalente, si selezionano tutte le interpretazioni *vere* della frase, e per ciascuna di esse si crea un congiunto delle frasi atomiche, in forma vera se hanno interpretazione vera, in forma negata se hanno interpretazione falsa; le frasi cosí formate saranno poi congiunte.
- per ottenere una *forma normale congiuntiva* equivalente, si selezionano tutte le interpretazioni *false* della frase, e per ciascuna di esse si crea un disgiunto delle frasi atomiche, in forma vera se hanno interpretazione falsa, in forma negata se hanno interpretazione vera; le frasi cosí formate saranno poi disgiunte.

Si prenda ad esempio la seguente tabella di verità:

a	b	c	α
0	0	0	**1**
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	**1**
1	0	1	**1**
1	1	0	0
1	1	1	0

Per la forma normale forma normale disgiuntiva, si considerino i valori per cui $\alpha = 1$: seguendo lo schema indicato si ottiene la seguente espressione:

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

Per la forma normale congiuntiva, si considerino i valori per cui $\alpha = 0$: seguendo lo schema indicato si ottiene la seguente espressione:

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

2.4 Metodi sintattici di decisione per la soddisfacibilità.

Il problema fondamentale finora incontrato é quello di decidere dell'insoddisfacibilità di un insieme di frasi $\Gamma \subset \mathcal{L}_{\mathcal{A}}$, che é strettamente legato allo svolgimento della tabella di verità dell'insieme Γ. Esistono procedure che utilizzano metodologie sintattiche, che cioé si limitano a manipolare i simboli, e che permettono di ottenere lo stesso risultato.

2.4.1 Risoluzione per clausole.

Un metodo che non utilizza tabelle di verità, é ad esempio il **metodo di risoluzione per clausole**.

Enunciamo il procedimento:

- si trasforma ogni frase α di Γ nella relativa forma normale congiuntiva.
- per ogni frase: da ogni congiunto, si ottiene un insieme di letterali detto **clausola**. Si indica col simbolo $\square$ la clausola vuota. Si ottiene perciò un insieme $\tilde{\Gamma}$ formato da tutte le clausole di tutte le frasi di Γ.
- si applica la seguente *regola di risoluzione*: date due clausole $\mathcal{C}', \mathcal{C}''$ si dice che la clausola $\mathcal{C}$ é **derivabile per risoluzione** da $\mathcal{C}', \mathcal{C}''$ se esiste un letterale $l \in \mathcal{C}'$ e il suo negato $\bar{l} \in \mathcal{C}''$ [1] tale che: $\mathcal{C} = \{\mathcal{C}' \setminus \{l\}) \cup (\mathcal{C}'' \setminus \{\bar{l}\}\}$

Osservazioni. Sia data la forma normale congiuntiva: $(\neg a \vee b \vee c) \wedge (a \vee \neg b \vee c)$. É una singola frase, e genera le clausole:

$$\mathcal{C}' = \{\neg a, b, c\}, \quad \mathcal{C}'' = \{a, \neg b, c\}$$

Da queste é possibile dedurre: $\mathcal{C} = \{b, \neg b, c\}$ oppure $\mathcal{C} = \{a, \neg a, c\}$ ◄

Osservazioni. Dalle clausole: $\mathcal{C}' = \{\neg a, b\}$, $\mathcal{C}'' = \{a, c\}$ é possibile dedurre: $\mathcal{C} = \{b, c\}$ ◄

Il risultato fondamentale a cui si perviene attraverso l'utilizzo di questo metodo e' il seguente:

Definizione 2.12 (risoluzione di un insieme di clausole) *Sia $\tilde{\Gamma}$ un insieme di clausole ottenute dall'insieme $\Gamma \subset \mathcal{L}_{\mathcal{A}}$. Si dice che la clausola $\mathcal{C}$ é deducibile dall'insieme $\tilde{\Gamma}$ per risoluzione e si indica $\tilde{\Gamma} \stackrel{\mathcal{R}}{\longmapsto} \mathcal{C}$ se e solo se esiste un albero finito con radice $\mathcal{C}$, con foglie in $\tilde{\Gamma}$ e dove ogni nodo terminale é dedotto per risoluzione dai due successori.*

Vale il seguente **teorema** che enuncia un criterio di decidibilità per un insieme Γ di frasi.

Teorema 2.4.1 *Sia $\Gamma \subset \mathcal{L}_{\mathcal{A}}$, e $\tilde{\Gamma}$ il corrispondente insieme di clausole. Allora Γ é insoddisfacibile se e solo se $\tilde{\Gamma} \stackrel{\mathcal{R}}{\longmapsto} \square$*

[1] $\bar{l} = \neg a$ se $l = a \in A$; $\bar{l} = a$ se $l = \neg a, a \in A$

Osservazioni. Consideriamo un insieme di frasi proposto in precedenza:

$$\Gamma = \{(a \wedge \neg b) \to c, c \to (b \vee d), \neg d \to a, \neg b, \neg d\}$$

e verifichiamo per risoluzione che esso é insoddisfacibile.

Si trasformano le singole frasi $\alpha \in \Gamma$ nella loro forma normale congiuntiva:

$$\Gamma' = \{(\neg a \vee b \vee c), (\neg c \vee b \vee d), d \vee a, \neg b, \neg d\}$$

E si costruisce l'insieme delle clausole:

$$\tilde{\Gamma} = \{\{\neg a, b, c\}, \{\neg c, b, d\}, \{d, a\}, \{\neg b\}, \{\neg d\}\}$$

Bisogna individuare un albero di derivazione che abbia radice $\Box$.

Un possibile albero é il seguente:

$\{a, d\}^*, \{\neg d\}^* \xmapsto{\mathcal{R}} \{a\}$

$\{\neg a, b, c\}^*, \{\neg b\}^* \xmapsto{\mathcal{R}} \{\neg a, c\}$

$\{a\}, \{\neg a, c\} \xmapsto{\mathcal{R}} \{c\}$

$\{b, \neg c, d\}^*, \{\neg d\}^* \xmapsto{\mathcal{R}} \{b, \neg c\}$

$\{b, \neg c\}, \{\neg b\}^* \xmapsto{\mathcal{R}} \{\neg c\}$

$\{\neg c\}, \{c\} \xmapsto{\mathcal{R}} \{\Box\}$

Le clausole contenute in $\tilde{\Gamma}$ sono seguite da un asterisco; si può verificare che esse sono nodi terminali della struttura ad albero. ◄

Una procedura automatica per la risoluzione per clausole?

Il teorema enunciato in questo paragrafo afferma che l'esistenza di un albero di risoluzione con radice $\Box$ implica l'insoddisfacibilità dell'insieme che ha per clausole i nodi terminali dell'albero stesso. Questo teorema non dice però come individuare tale albero, e neppure se tale albero esiste.

Per implementare una procedura automatica é dunque necessario considerare l'esistenza di tutti i possibili alberi che si generano da un certo insieme di clausole, e verificare se tra essi esiste almeno un albero con la clausola vuota a radice.

Definizione 2.13 *insieme risoluzione di un insieme di clausole Sia Λ un insieme finito di clausole, e siano $\mathcal{C}_i, \mathcal{C}_j$ due qualunque clausole in esso. Definiamo un nuovo insieme, denominato **Risoluzione di** Λ, tale che*

$$Ris(\Lambda) = \{\Lambda\} \cup \{\forall i, \forall j : \quad \mathcal{C}_i, \mathcal{C}_j \xmapsto{\mathcal{R}} \mathcal{C}_k\}$$

dove il secondo é l'insieme delle clausole $\mathcal{C}_k$ che si ottengono da tutte le possibili coppie in Λ.

Osservazioni. Poiché il numero di coppie non ripetitive in un insieme di n elementi é in numero $\frac{n(n-1)}{2}$, vale che $k \leq \frac{n(n-1)}{2}$. Per cui tra l'insieme Λ e la sua risoluzione vale la relazione $\Lambda \subseteq Ris\{\Lambda\}$. ◄

La definizione si può estendere ricorsivamente, per cui si pone

$$Ris^k(\Lambda) = Ris(Ris^{k-1}(\Lambda))$$

Si dimostra che esiste un $p \in \mathbf{N}$ tale che $Ris^p(\Lambda) = Ris^{p+i}(\Lambda) \quad \forall i \in \mathbf{N}$, cioé esiste un limite superiore finito alla cardinalità dell'insieme delle risoluzioni [2]. In questo caso, é sufficiente verificare se $\square \in Ris^p(\Lambda)$.

Benché questa procedura sia automatica e facilmente implementabile, si può dimostrare che $p \leq 2^{2card(\Lambda)}$, cioé che l'ordine di complessità dell'algoritmo é ancora di natura esponenziale. Sia infatti $\Gamma = \{a_1, a_2, ..., a_p\}$ con $p \geq 1$ un insieme di frasi atomiche distinte del linguaggio, se segue ogni clausola appartiene all'insieme Γ' di letterali ottenuto a partire da Γ; la cardinalità di tale insieme sarà di $2p$. L'insieme delle parti di Γ' ha di conseguenza cardinalità 2^{2p}. Poiché le clausole non si ripetono, e poiché ogni clausola é un elemento dell'insieme delle parti, la risoluzione n-esima può comunque contenere al massimo lo stesso numero di elementi di questo insieme.

L'algoritmo esposto in questa forma, però, si rivela poco efficiente. Sono stati individuati alcuni espedienti che permettono di migliorarne l'efficienza eliminando inutili confronti:

- **Eliminazione delle tautologie**: tutte le clausole contenenti un letterale e il suo negato (del tipo $\{a, ..., \neg a, ...\}$) sono tautologie, poiché ogni clausola é una disgiunzione di letterali.

- **Eliminazione di clausole che contengono altre clausole**: se in Λ ci sono due clausole $\mathcal{C}', \mathcal{C}''$ tali che $\mathcal{C}' \subset \mathcal{C}''$, é possibile eliminare la clausola $\mathcal{C}''$: ogni clausola é infatti una disgiunzione di letterali, ed ogni insieme di frasi e' una congiunzione di clausole; se pertanto e' vera la clausola *con meno letterali*, sara' di conseguenza altrettanto vera la clausola che la contiene.

Chiameremo in seguito Λ_c l'insieme delle clausole di Λ 'ripulito' secondo le regole esposte sopra.

Osservazioni. Sia dato l'insieme $\Lambda = \{\{a, \neg b, c\}\{\neg a, b, c\}\}$; verificare se esso é soddisfacibile. Si può immediamente calcolare la risoluzione di Λ:

$$Ris(\Lambda) = \Lambda \cup \{\{a, \neg a, c\}\{b, \neg b, c\}\}$$

Le due clausole ricavate per risoluzione sono entrambe tautologie, per cui vale immediatamente che: $\Lambda = Ris(\Lambda)$, e perciò tale insieme non é soddisfacibile. ◄

Osservazioni. Verificare se:

$$\{a \to b \vee c, b \vee c \to \neg d\} \models a \to \neg d$$

[2]nelle ipotesi considerate, di Λ insieme finito di clausole

Questo equivale ad indagare la soddisfacibilità dell'insieme:

$$\{a \to b \vee c, b \vee c \to \neg d, \neg(a \to \neg d)$$

Mediante trasformazioni successive, possiamo scrivere:

$$\begin{gathered}\{\neg a \vee (b \vee c), \neg(b \vee c) \vee \neg d, \neg(\neg a \vee \neg d)\} \\ \{\neg a \vee b \vee c, (\neg b \wedge \neg c) \vee \neg d, a \wedge d\} \\ \{\neg a \vee b \vee c, (\neg b \vee \neg d) \wedge (\neg c \vee \neg d) \vee \neg d, a \wedge d\}\end{gathered}$$

da cui l'insieme di clausole:

$$\Lambda = \{\{\neg a, b, c\}, \{\neg b, \neg d\}, \{\neg c, \neg d\}, \{a\}, \{d\}\}$$

e le successive risoluzioni:

$$Ris(\Lambda) = \Lambda \cup \{\{b, c\}, \{\neg c\}, \{\neg b\}\}$$

e senza procedere in ulteriori risoluzioni, é immediato ricavare dai nuovi elementi in due passaggi la clausola vuota:

$$\begin{gathered}\{b, c\}, \{\neg c\} \stackrel{\mathcal{R}}{\longmapsto} \{b\} \\ \{b\}, \{\neg b\} \stackrel{\mathcal{R}}{\longmapsto} \Box\end{gathered}$$

◀

2.4.2 Deduzione naturale.

Esiste un altro metodo sintattico, che va sotto il nome di **deduzione naturale** e permette di decidere della soddisfacibilità di un insieme di frasi. A differenza del metodo precedente esso agisce direttamente sulle frasi senza la necessità di trasformarle in una forma normale, ma ha lo svantaggio di utilizzare un insieme di regole di manipolazione più esteso rispetto all'altro. Tali regole sono usualmente divise in due classi:

- *regole elementari*
- *regole condizionali*

Regole elementari.

Sono regole di eliminazione/introduzione dei connettivi; le elenchiamo per ogni connettivo:

- Connettivo di *congiunzione*:
  1. *eliminazione* della congiunzione: data una congiunzione di due frasi, ne posso dedurre ciascuna delle frasi che la compongono.

$$(\wedge, e): \qquad \alpha \wedge \beta \stackrel{\mathcal{DN}}{\longmapsto} \begin{cases} \alpha \\ \beta \end{cases}$$

2. *inserimento* della congiunzione: date due frasi α, β, ne posso dedurre ciascuna delle loro congiunzioni.

$$(\wedge, i): \qquad \{\alpha, \beta\} \stackrel{\mathcal{DN}}{\longmapsto} \begin{cases} \alpha \wedge \beta \\ \beta \wedge \alpha \end{cases}$$

- Connettivo di *disgiunzione*:

1. *eliminazione* della disgiunzione: data una disgiunzione, e la negazione di uno dei disgiunti, ne posso dedurre l'altro.

$$(\vee, e): \qquad \begin{array}{l} \{\alpha \vee \beta, \neg\alpha\} \stackrel{\mathcal{DN}}{\longmapsto} \beta \\ \{\alpha \vee \beta, \neg\beta\} \stackrel{\mathcal{DN}}{\longmapsto} \alpha \end{array}$$

2. *inserimento* della disgiunzione: data una frase α, é sempre possibile congiungerla ad una qualunque frase β:

$$(\vee, i): \qquad \{\alpha\} \stackrel{\mathcal{DN}}{\longmapsto} \begin{cases} \alpha \vee \beta \\ \beta \vee \alpha \end{cases}$$

- Connettivo di *implicazione*:

1. *eliminazione* dell'implicazione: data una implicazione $\alpha \rightarrow \beta$, e l'implicando α, é sempre possibile dedurne l'implicato; tale regola va sotto il nome di **modus ponens**:

$$(\rightarrow, i.1): \qquad \{\alpha \rightarrow \beta\}, \{\alpha\} \stackrel{\mathcal{DN}}{\longmapsto} \beta$$

Ancora, data una implicazione, e la negazione dell'implicato, é sempre possibile dedurne la negazione dell'implicando; tale regola va sotto il nome di **modus tollens**:

$$(\rightarrow, i.2): \qquad \{\alpha \rightarrow \beta\}, \{\neg\beta\} \stackrel{\mathcal{DN}}{\longmapsto} \neg\alpha$$

2. *inserimento* dell'implicazione: é una regola condizionale, e verrà esposta nel paragrafo successivo.

- Connettivo di *negazione*:

1. *eliminazione* della negazione: data una frase in forma vera e la stessa in forma negata, é possibile dedurne l'assurdo:

$$(\neg, e): \qquad \{\alpha\}, \{\neg\alpha\} \stackrel{\mathcal{DN}}{\longmapsto} \bot$$

2. *inserimento* della negazione: dato l'assurdo, se ne può dedurre qualunque frase (e la propria negazione):

$$(\neg, i): \qquad \{\bot\} \stackrel{\mathcal{DN}}{\longmapsto} \neg\alpha$$

Regole condizionali.

Sono solamente due, e sono le seguenti:

1. Regola di *introduzione dell'implicazione*. Dato un insieme Γ di frasi del linguaggio:
$$\Gamma \overset{\mathcal{DN}}{\longmapsto} (\alpha \to \beta) \Longleftrightarrow \Gamma \cup \{\alpha\} \overset{\mathcal{DN}}{\longmapsto} \beta$$
2. Regola di *riduzione ad assurdo*: da un insieme di frasi posso derivare una certa frase α se aggiungendo la negazione di questa ne viene l'assurdo, cioé:
$$\Gamma \overset{\mathcal{DN}}{\longmapsto} \alpha \Longleftrightarrow \Gamma \cup \{\neg\alpha\} \overset{\mathcal{DN}}{\longmapsto} \perp$$

Osservazioni. Le regole formali introdotte sono per la maggior parte di natura estremamente intuitiva. Ad esempio, per quel che riguarda l'eliminazione della congiunzione, é piuttosto *naturale*, data una congiunzione di frasi vere, dichiarare vero ciascuno dei congiunti. Se é vero che

Il 30 ottobre 1977 d.C. era domenica **e** *gli uomini sono organismi pluricellulari*

ciascuno di noi intuitivamente se la sente di affermare singolarmente che:

Il 30 ottobre 1977 d.C. era domenica

oppure che:

Gli uomini sono organismi pluricellulari

Allo stesso modo, data una frase vera, la si puo' unire ad una qualunque altra frase, vera o falsa che sia, attraverso una disgiunzione. Se é infatti vero che:

Il 30 ottobre 1977 d.C. era domenica

é *intuitivamente* vero che:

Il 30 ottobre 1977 d.C. era domenica **o** *gli asini volano*

anche se la seconda frase é palesemente falsa (almeno da quanto ne so...).

Ma si consideri il seguente esempio:

Oggi é domenica **e** *questa frase contiene nove parole*

È forse possibile ricavarne la seguente frase?

Questa frase contiene nove parole

◀

Definizione 2.14 (deducibilità di una frase da un insieme) *Sia ora* $\Gamma \subset \mathcal{A}, \alpha \in \mathcal{L}_{\mathcal{A}}$*; l'affermazione* α *é deducibile da* Γ*, indicata dall'espressione:* $\Gamma \overset{\mathcal{DN}}{\longmapsto} \alpha$*, significa che esiste un albero che ha come radice* α*, i cui nodi terminali sono in* Γ *e per il quale ogni nodo non terminale é dedotto dai suoi successori mediante l'applicazione di una regola elementare.*

Enunciamo ora un importante **teorema**:

Teorema 2.4.2 *α é conseguenza semantica di Γ se e solo se ne é conseguenza sintattica per deduzione naturale:*

$$\Gamma \models \alpha \Longleftrightarrow \Gamma \overset{\mathcal{DN}}{\longmapsto} \alpha$$

Mediante l'applicazione della regola di riduzione ad assurdo al secondo membro del teorema, é possibile ricavare che: $\Gamma \cup \{\neg a\} \overset{\mathcal{DN}}{\longmapsto} \perp$ e questa condizione verrà utilizzata nella pratica per decidere della soddisfacibilità dell'insieme.

Osservazioni. Verificare per deduzione naturale che:

$$\Gamma = \{a \wedge \neg b \to c, c \to b \vee d, \neg d \to a, \neg b\} \models d$$

Ciò equivale a dimostrare che:

$$\{a \wedge \neg b \to c, c \to b \vee d, \neg d \to a, \neg b\} \cup \{\neg d\} \overset{\mathcal{DN}}{\longmapsto} \perp$$

Evidenziamo i passaggi:

frasi utilizzate[3]	**frase dedotta**	**regola utilizzata**
$\{\neg d\}^*, \{\neg d \to a\}^*$	a	$(\to, e)$
$\{a\}, \{\neg b\}^*$	$a \wedge \neg b$	$(\wedge, i)$
$\{a \wedge \neg b\}, \{a \wedge \neg b \to c\}^*$	c	$(\to, e)$
$\{c\}, \{c \to (b \vee d)\}^*$	$b \vee d$	$(\to, e)$
$\{b \vee d\}, \{\neg b\}^*$	d	$(\vee, e)$
$\{d\}, \{\neg d\}^*$	$\perp$	$(\vee, e)$

◀

Osservazioni. Dimostrare che

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

Ciò equivale a dimostrare, per la prima regola condizionale, che:

$$\{a \to (b \vee c), (b \vee c) \to \neg d, a\} \overset{\mathcal{DN}}{\longmapsto} \neg d$$

Evidenziamo i passaggi:

frasi utilizzate[4]	**frase dedotta**	**regola utilizzata**
$\{a \to (b \vee c)^*, a^*\}$	$b \vee c$	$(\to, e)$
$\{b \vee c, (b \vee c) \to \neg d)^*\}$	$\neg d$	$(\to, e)$

◀

2.5 Soddisfacibilità di insieme non finiti.

2.5.1 Compattezza semantica

Sia dato un linguaggio proposizionale $\mathcal{L}_{\mathcal{A}}$, ed un sottoinsieme Γ di frasi in esso; se Γ é infinito, per esso vale il seguente:

Teorema 2.5.1 Γ *é soddisfacibile se e soltanto se ogni sottoinsieme finito di* Γ *lo é.*

$\bigtriangledown$ *Dimostrazione.*
Ipotizziamo l'esistenza di un insieme $\Gamma' \subset \Gamma$, tale che Γ' sia finito e soddisfacibile; di seguito vedremo una buona scelta dell'insieme che ci permetterà di concludere la dimostrazione.

Per dimostrare dunque che Γ é soddisfacibile, si supponga per assurdo che esso non lo sia o, il che é equivalente, che $\Gamma \overset{\mathcal{DN}}{\longmapsto} \bot$, cioé esiste un albero di derivazione *finito* che ha come radice l'assurdo e i nodi terminali in Γ. Sia Γ' l'insieme delle foglie di questo albero; ma allora Γ' é un sottoinsieme di Γ insoddisfacibile. $\bigtriangleup$

Capitolo 3

Linguaggi del I ordine

Si osservino le seguenti frasi:

1. Antonio ama Lucia
2. *Tutti* i ragazzi amano Lucia
3. Antonio é *un* ragazzo

Dalle (2,3) sembra del tutto ragionevole *dedurre* la (1), tuttavia nel contesto dei linguaggi proposizionali, né (2), né (3) sono frasi atomiche; al contrario, (1) é una frase atomica. Non solo, ma (2),(3) non appartengono neppure a nessun linguaggio proposizionale, poiché contengono elementi che non sono stati definiti (sono evidenziati in corsivo).

Si suppongano tuttavia (2), (3) come frasi atomiche, e si generi il linguaggio $\mathcal{L}_{\{2,3\}}$: non é possibile in alcun modo derivare la (1) con alcuno dei metodi proposti.

Si consideri ancora il seguente esempio:

Antonio dice che Marco *lo* ha danneggiato

Il significato che si può attribuire al pronome *lo* é ambiguo: certamente esso non si riferisce a Marco, ma non é altresí detto che esso si riferisca ad Antonio, piuttosto che ad una terza persona introdotta in precedenza nel discorso.

Queste osservazioni dimostrano che i linguaggi proposizionali non possono in alcun modo trattare un'ampia classe di preposizioni del linguaggio comune, ed é perciò necessario introdurre nuovi elementi che permettano di manipolare le piú complesse strutture che qui abbiamo esemplificato. A tale scopo si introducono i **linguaggi del primo ordine (o predicativi)**.

3.1 Alfabeto dei linguaggi predicativi.

Esso é costituito da:

1. Un insieme (eventualmente vuoto o infinito) di *costanti*: $\mathcal{C}$, che verranno indicate con le prime lettere dell'alfabeto minuscolo: {a, b, c, ...}
2. Un insieme infinito di *variabili*: $\mathcal{V}$, che verranno indicate con le ultime lettere dell'alfabeto minuscolo: {x, y, z, ...}

3. Un insieme (non vuoto, eventualmente infinito, ma generalmente finito) di *predicati*: $\mathcal{P}$, che verranno indicati con le prime lettere dell'alfabeto maiuscolo: {A, B, C,...}. Ogni predicato é caratterizzato da una propria *arietà*, che consiste nel numero di elementi su cui agisce. Tra i predicati di $\mathcal{P}$ ce n'é sempre almeno uno binario, detto *predicato di uguaglianza*.

4. Un insieme (eventualmente vuoto, generalmente finito), di *simboli funzionali*: $\mathcal{F}$, che verranno indicati con le lettere intermedie dell'alfabeto minuscolo {f, g, h, ...}. Ogni simbolo funzionale é caratterizzato da una propria *arietà*.

5. Un insieme di connettivi: $\neg, \quad \wedge, \quad \vee, \quad \rightarrow$

6. Il simbolo di assurdo: $\perp$

7. I simboli ausiliari: (,).

8. Due nuovi simboli, detti *quantificatori*:

   - $\forall$, detto *quantificatore universale*, che significa letteralmente per tutti...
   - $\exists$, detto *quantificatore esistenziale*, che significa letteralmente esiste...

I primi quattro elementi dell'alfabeto sono anche detti **lessico**, e caratterizzano in maniera univoca un 'certo' linguaggio predicativo. Da esso si ricava l'insieme *Ter* dei **termini** del linguaggio, cosí definito:

Definizione 3.1 *insieme dei termini Ter é il piú piccolo insieme tale che:*

1. *L'insieme delle costanti é in Ter:* $\mathcal{C} \subset Ter$
2. *L'insieme delle variabili é in Ter:* $\mathcal{V} \subset Ter$
3. *Se* $f \in \mathcal{F}$ *é un simbolo funzionale di arietà k, (ove* $k \geq 1$*), e un certo insieme di termini* $\{t_1, t_2, ..., t_k\} \subset Ter$*, allora* $f(t_1, t_2, ..., t_k) \in Ter$

Osservazioni. Si osservi che l'insieme dei termini ha cardinalità infinita, poiché le variabili sono sempre in quantità infinita. Tuttavia, se l'insieme dei simboli funzionali non é vuoto, l'insieme *Ter* contiene infiniti termini che *non* sono variabili. ◄

Osservazioni. Sia dato un linguaggio del primo ordine il cui alfabeto é: $\mathcal{A} = (\mathcal{C} = \{a, b\}, \mathcal{F} = \{f(*)\}, \mathcal{V} = \emptyset, \mathcal{P} = \emptyset)$. Allora:

$$Ter(\mathcal{A}) = \{a, b, f(a), f(b), f(f(a)), ...\}$$

◄

Definizione 3.2 *frasi atomiche Le* ***frasi atomiche*** *del linguaggio sono il piú piccolo insieme Atom tale che:*

1. $\perp \in Atom$

2. *Se $A \in \mathcal{P}$ é un predicato di arietà k, e $(t_1, t_2, ..., t_k) \subset Ter$ é un qualche insieme di termini, allora $A(t_1, t_2, ..., t_k)$ é una frase atomica del linguaggio*

Definizione 3.3 *linguaggio predicativo Un linguaggio proposizionale $\mathcal{L}_{(\mathcal{C},\mathcal{V},\mathcal{P},\mathcal{F})}$ é il minimo insieme tale che:*

- *Le frasi atomiche sono in esso: $Atom \subset \mathcal{L}$*
- *Se $\alpha \in \mathcal{L}$, allora $\neg\alpha \in \mathcal{L}$*
- *Se $\{\alpha, \beta\} \subset \mathcal{L}$, allora $\{\alpha \wedge \beta, \alpha \vee \beta, \alpha \to \beta\} \subset \mathcal{L}$*
- *Se $\alpha \in \mathcal{L}$, $x \in \mathcal{V}$, allora $\{(\forall x)\alpha, (\exists x)\alpha\} \subset \mathcal{L}$*

Osservazioni. Siano dati ad esempio due predicati, un insieme di costanti, un insieme di funzionali:

- $\mathcal{P} = \{A(*), B(*, *)\}$, ove A é un predicato ad 1 posto (ad esempio 'correre') e B un predicato a 2 posti (ad esempio 'amare')
- $\mathcal{C} = \{a, b\}$
- $\mathcal{F} = \{f\}$; ad esempio il funzionale 'il padre di'

Allora una frase atomica é ad esempio la seguente:

$$\forall x \neg(A(a) \vee B(x, f(a)))$$

che si può tradurre in linguaggio naturale come:

Per ogni x non é vero che a corre o x odia il padre di a

◀

3.1.1 Variabili libere.

Si dice che una variabile x *occorre* in una frase α se x figura in almeno una delle frasi atomiche che si trovano nei nodi terminali dell'albero sintagmatico della frase stessa.

Osservazioni. Nella frase $(\forall x)A(a, y)$ l'unica variabile é y. ◀

Definizione 3.4 *variabili libere Sia ora x una variabile che occorre in α: si dirà che tale variabile è vincolata se essa é C-comandata da uno dei due quantificatori $(\forall x), (\exists x)$. In caso contrario essa sarà detta libera.*

Osservazioni. Nell'esempio precedente nessuna variabile é vincolata. ◀

Osservazioni. Nella frase $(\exists x)(A(x, y) \to \neg B(x, y))$ sono presenti due variabili: x, y. Entrambe le occorrenze di x sono vincolate, mentre entrambe quelle di y sono libere. ◀

Osservazioni. Nella frase $(\forall x)A(x,y) \rightarrow (\neg\exists y)B(x,y)$ sono presenti due variabili: x, y. La prima occorrenza della variabile x é vincolata, mentre la seconda é libera; la prima occorrenza di y é libera, mentre la seconda é vincolata. ◀

Data una frase α, indicheremo con $\mathcal{F}_{\mathcal{V}}(\alpha)$ l'insieme delle variabili in $\mathcal{V}$ tali che hanno almeno una occorrenza libera in α. Tale insieme può anche essere vuoto: ciò ovviamente accade quando tutte le variabili sono vincolate; in questo caso la frase si dice *chiusa*.

3.2 Semantica di un linguaggio del primo ordine.

Definizione 3.5 *struttura Si definisce* ***struttura*** *(o universo del discorso) una coppia $\mathcal{M}$ formata da un certo insieme M, non vuoto, detto dominio del discorso, e una certa legge g dagli insiemi $\mathcal{C}, \mathcal{F}, \mathcal{P}$ a M:*

$$\mathcal{M} = (M, g) \qquad g : \mathcal{C}, \mathcal{F}, \mathcal{P} \rightarrow M^k$$

Osservazioni.

- sia c una costante. La funzione $g : c \rightarrow g(c)$ é una funzione $g : \mathcal{C} \rightarrow \mathcal{C}_M \subset M$ che associa ad una costante dell'alfabeto iniziale, una costante nel dominio del discorso M.

- sia $A \in \mathcal{P}$ un predicato unario (ad esempio 'essere sciocchi'); allora $g : A \rightarrow A_{\mathcal{M}} \subseteq \mathcal{M}$. Sia invece $B \in \mathcal{P}$ un predicato binario (ad esempio 'amare'); allora $g : B \rightarrow B_{\mathcal{M}} \subseteq \mathcal{M} \times \mathcal{M}$.

  In questo caso la funzione associa ad un predicato, un sottoinsieme di M^k costituito dagli elementi di M che soddisfano la preposizione. Esisterà dunque un sottoinsieme di $\mathcal{M}$ composto da tutti quegli elementi che soddisfano il predicato essere sciocchi; esso può essere al limite un insieme vuoto, o l'insieme stesso.

- Sia invece $f \in \mathcal{F}$ un funzionale unario (ad esempio: 'é il padre di'): allora $g : f \rightarrow f_M$ ove $f_M : \mathcal{M} \rightarrow \mathcal{M}$. Sia invece $f \in \mathcal{F}$ un funzionale binario (ad esempio: 'é figlio di'[1] allora $g : f \rightarrow f_M$ ove $f_M : \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{M}$.

  In questo caso la funzione g associa a un elemento di M^k un valore in M che corrisponde al 'risultato' del funzionale.

◀

[1] si sottointende, supposto il funzionale binario, che sia data la coppia di informazione (madre, padre) o la commutata

Definizione 3.6 *assegnazione Si definisce* ***assegnazione*** *(talora detto ambiente) una funzione dall'insieme delle variabili a* $\mathcal{M}$*:*

$$\eta : \mathcal{V} \to \mathcal{M}$$

Osservazioni. Una assegnazione é una funzione un po' *speciale*: essa svolge un compito analogo a quello della funzione g di una qualche struttura, poiché 'collega' l'insieme delle variabili col dominio del discorso. Ma effettua questa operazione su un insieme molto particolare - come si vedrà in seguito - quello delle variabili. ◀

Per ogni $t \in Ter(\mathcal{L})$, indicheremo con $[t]_{\mathcal{M},\eta}$ l'elemento corrispondente nell'insieme $\mathcal{M}$.

Osservazioni.

termine	**elemento in** $\mathcal{M}$
$t_1 = a \in \mathcal{C}$	$[t_1]_{(\mathcal{M},\eta)} = g(a)$
$t_2 = x \in \mathcal{V}$	$[t_2]_{(\mathcal{M},\eta)} = \eta(x)$
$t = f(a,x)$, con $f \in \mathcal{F}$	$[t]_{(\mathcal{M},\eta)} = f_M(g(a), \eta(x)) = f_M([t_1]_{(\mathcal{M},\eta)}, [t_2]_{(\mathcal{M},\eta)})$

◀

Definizione 3.7 *interpretazione Date una struttura ed una assegnazione, si può definire una* ***interpretazione*** *che attribuisca un valore di verità alle frasi del linguaggio. Essa é una funzione del tipo:*

$$v_{(\mathcal{M},\eta)} : \mathcal{L} \to \{0,1\}$$

Enunciamo inoltre il *procedimento* di sostituzione di una variabile quantificata, necessario per attribuire un valore di verità a frasi che contengono quantificatori. Sia dunque data una struttura (M,η) e un qualche elemento $p \in M$. Si definisce la funzione $\eta_{(p|x)} : \mathcal{V} \to M$ come segue:

$$\begin{cases} \eta_{(p|x)}(y) = \eta(y) & x \neq y \\ \eta_{(p|x)}(y) = p & x = y \end{cases}$$

3.2.1 Semantica del linguaggio.

Definiamo semantica di un linguaggio del primo ordine l'insieme delle seguenti regole formali:

1. Il valore di verità dell'assurdo é zero: $v_{(\mathcal{M},\eta)}(\perp) = 0$

2. Definisce il valore di verità di un predicato:

   $$v_{(\mathcal{M},\eta)}(A(t_1,...,t_n)) = 1 \Leftrightarrow [t]_{(\mathcal{M},\eta)} = ([t_1]_{(\mathcal{M},\eta)}, ..., [t_n]_{(\mathcal{M},\eta)}) \in A_{\mathcal{M}}$$

   Cioé un predicato é vero se e soltanto se l'n-pla dei termini che esso elabora appartiene al sottoinsieme del dominio del discorso che soddisfano tale predicato.

3. Regole per i connettivi[2]:
   - $v_{(\mathcal{M},\eta)}(\neg\alpha) = 1 - v_{(\mathcal{M},\eta)}(\neg\alpha)$
   - $v_{(\mathcal{M},\eta)}(\alpha \wedge \beta) = min\{v_{(\mathcal{M},\eta)}(\alpha), v_{(\mathcal{M},\eta)}(\beta)\}$
   - $v_{(\mathcal{M},\eta)}(\alpha \vee \beta) = max\{v_{(\mathcal{M},\eta)}(\alpha), v_{(\mathcal{M},\eta)}(\beta)\}$
   - $v_{(\mathcal{M},\eta)}(\alpha \rightarrow \beta) = max\{1 - v_{(\mathcal{M},\eta)}(\alpha), v_{(\mathcal{M},\eta)}(\beta)\}$
4. Regole per i quantificatori:
   - $v_{(\mathcal{M},\eta)}(\forall x)\alpha = 1 \iff v_{(\mathcal{M},\eta_{(p|x)})}\alpha = 1 \quad \text{per ogni } p \in M$
   - $v_{(\mathcal{M},\eta)}(\exists x)\alpha = 1 \iff v_{(\mathcal{M},\eta_{(p|x)})}\alpha = 1 \quad \text{per almeno un } p \in M$

Osservazioni. Si osservi che il quantificatore universale può essere interpretato come una *congiunzione iterata* su tutti gli elementi dell'universo del discorso.

Analogamente, il quantificatore esistenziale può essere visto come una *disgiunzione iterata* su tutti gli elementi dell'universo del discorso. ◀

Enunciamo in seguito alcune definizioni che trovano una corrispondenza con quelle enunciate nel contesto dei linguaggi proposizionali.

Definizione 3.8 *insieme soddisfacibile Un insieme di frasi $\Gamma \subset \mathcal{L}_{\mathcal{A}}$ é **soddisfacibile** se e solo se esiste una struttura $\mathcal{M}$ e una assegnazione η per cui*

$$v_{(\mathcal{M},\eta)}(\alpha) = 1 \qquad \forall\alpha \in \Gamma$$

*In caso contrario l'insieme si dice **insoddisfacibile**, cioé se per ogni struttura ed ogni interpretazione esso contiene almeno una frase falsa.*

Definizione 3.9 *conseguenza semantica Una frase α é **conseguenza semantica** di un insieme di frasi Γ, e lo si indica con $\Gamma \models \alpha$ se e solo se esiste una interpretazione che soddisfa l'insieme Γ, e per tale interpretazione la frase α é vera. Analogamente al caso proposizionale, si può verificare che l'insieme $\Gamma \cup \{\neg\alpha\}$ é insoddisfacibile.*

Definizione 3.10 *tautologia, contraddizione Si definisce **tautologia** una frase α tale che*

$$v_{(\mathcal{M},\eta)}(\alpha) = 1 \quad \forall\mathcal{M}, \eta$$

cioé una frase vera per qualunque interpretazione o, equivalentemente, vera in ogni struttura.

*Si definisce **contraddizione** una frase α tale che*

$$v_{(\mathcal{M},\eta)}(\alpha) = 0 \quad \forall\mathcal{M}, \eta$$

cioé una frase falsa per ogni interpretazione o, equivalentemente, falsa in ogni struttura.

Osservazioni. La frase $(\forall x)A(x) \vee (\exists x)\neg A(x)$ é una tautologia.
La frase $(\forall x)A(x) \wedge (\exists x)\neg A(x)$ é una contraddizione. ◀

[2] ricordiamo che i connettivi agiscono sulle *frasi atomiche* del linguaggio

Una frase vera in un linguaggio del primo ordine: che significa?

Vediamo ora di spiegare che significa *calcolare il valore di verità di una frase* in un linguaggio del primo ordine attraverso alcuni semplici esempi: Sia data una struttura $\mathcal{M} = (M, g)$.

- Sia data la frase: $(\forall x)A(x)$. In questo caso la funzione $g : A \longrightarrow A_M \subset M$ individua gli elementi dell'universo M tali da soddisfare la preposizione A. Se ad esempio A : *é un numero naturale*, é definita su la funzione individua tutti gli elementi di M che soddisfano tale condizione.

  Ora: $v_{(\mathcal{M},\eta)}(\forall x)A(x) = 1 \Longleftrightarrow v_{(\mathcal{M},\eta(x|p))}A(x)$ per ogni $p \in M$.

  Ma $p \in A_M \Longleftrightarrow A_M \equiv M$

- Sia data la frase: $(\exists x)B(x)$.

  Ora: $v_{(\mathcal{M},\eta)}(\exists x)A(x) = 1 \Longleftrightarrow v_{(\mathcal{M},\eta(x|p))}A(x)$ per almeno un $p \in M$.

  Ma $p \in A_M \Longleftrightarrow A_M \neq \emptyset$

- Sia data la frase: $(\forall x)(A(x) \rightarrow B(x))$.

  $v_{(\mathcal{M},\eta)}(\forall x)(A(x) \rightarrow B(x)) = 1 \Longleftrightarrow v_{(\mathcal{M},\eta_{(x|p)})}(A(x) \rightarrow B(x)) = 1$ per ogni $p \in M$. Ricordando un'equivalenza semantica introdotta già per i linguaggi proposizionali, vale che: $v_{(\mathcal{M},\eta_{(x|p)})}(\neg A(x) \wedge B(x)) = 1 \Longleftrightarrow p \in M - A_M \cup B_M \Leftrightarrow A_M \subset B_M$

3.2.2 Equivalenze semantiche.

Definizione 3.11 *equivalenza semantica Due frasi α, β si dicono* ***(semanticamente) equivalenti*** *e lo si indica con $\alpha \equiv \beta$ se:*

$$v_{(\mathcal{M},\eta)}(\alpha) = v_{(\mathcal{M},\eta)}(\beta) \qquad \forall(\mathcal{M}, \eta)$$

Teorema 3.2.1 *Sia data una frase α del linguaggio. Si fissino in seguito una struttura $\mathcal{M}$, e due assegnazioni $\xi, \eta : \mathcal{V} \to \mathcal{M}$. Se le due assegnazioni coincidono sull'insieme delle variabili libere $\mathcal{F}_v(\alpha)$, allora:*

$$v_{(\mathcal{M},\xi)} = v_{(\mathcal{M},\eta)}$$

$\triangledown$ *Dimostrazione.*
Sia $A(t_1, ..., t_k)$ un predicato dell'alfabeto di linguaggio. Si considerino i due casi:

1. A **non contiene variabili libere**: allora, o le A non contiene variabili quantificate, e le due interpretazioni banalmente coincidono poichè sono definite sullo stesso dominio, oppure A contiene variabili quantificate, per cui esiste certamente una sostituzione $\xi_{(x|p')}$ ed una sostituzione $\eta_{(x|p'')}$ opportunamente scelte per cui valga $(p' = p'')$; questo é possibile per la congruenza del dominio.

2. A **contiene variabili libere**: in questo caso, si considerino i termini su cui agisce il predicato: se il termine t_k non contiene variabili, vale banalmente che $[t_k]_{(\mathcal{M},\eta)} = [t_k]_{(\mathcal{M},\xi)}$; se il termine t_k contiene variabili quantificate, per il punto sopra vale ancora l'uguaglianza; se poi il termine t_k contiene variabili libere, per l'ipotesi del teorema vale ancora l'uguaglianza.

La composizione di frasi atomiche mediante connettivi ha una dimostrazione induttiva, lasciata per esercizio. △

Osservazioni.

- Questo teorema evidenzia l'importante delle variabili libere rispetto a quelle condizionate.
- Se l'insieme delle variabili libere é vuoto, l'interpretazione del linguaggio é indipendente dall'assegnazione

◀

Osservazioni. Sia data la frase $\alpha : \quad (\forall x)A(x) \to (\exists y)B(y)$. Questa frase contiene due variabili x, y, entrambe vincolate. In questo caso $\mathcal{F}_v(\alpha) = 0$, e qualunque interpretazione non dipende dall'assegnazione. ◀

Si considerino ora le due frasi:

$$\alpha = (\forall x)(\exists y)A(x,y) \text{ e } \beta = (\exists y)(\forall x)A(x,y)$$

Esse contengono le stesse variabili, quantificate allo stesso modo. Eppure, il loro significato é profondamente diverso: se si attribuisce il significato *amare* alla preposizione $A(*,*)$, esse si possono tradurre come:

- α : per ogni x c'é almeno un y che é amato da lui. Cioé ogni x ama qualcuno, non necessariamente lo stesso y per x differenti.
- β : esiste un y che é amato da ciascun x. Cioé per ogni x, lo stesso y é amato da questo; cioé c'é un y amato da tutti.

Si osservi come il significato di β sia piú restittivo di quello di α: nel secondo caso, infatti, ogni x si trova ad amare la stessa persona. Cioé c'é una relazione di consequenzialità semantica tra le due frasi, ma in un solo verso: $\beta \models \alpha$ ma $\alpha \not\models \beta$

Dal punto di vista 'semantico', la frase α significa che per ogni $p_i \in \mathcal{M}$ esiste almeno un $q_i \in \mathcal{M}$ tale che la coppia $(p_i, q_i) \in A_{\mathcal{M}}$; mentre la frase β significa che esiste un q costante tale che la coppia $(p_i, q) \in A_{\mathcal{M}}$. Se si visualizzasse tale situazione su di un piano, il dominio della frase α sarebbe un sottonsieme del piano (x, y), mentre il dominio della frase β sarebbe una retta all'interno dello stesso piano.

Ora, valgono per i linguaggi predicativi tutte le equivalente introdotte nella discussione dei linguaggi proposizionali. Ma a queste vanno aggiunte le eventuali equivalenze che si generano con l'introduzione dei quantificatori: essi infatti non possono essere scambiati tra loro indiscriminatamente, quando sono di natura differente.

1. $\alpha \in \mathcal{L}_{\mathcal{A}},\ x \notin \mathcal{F}_{\mathcal{V}}(\alpha) : \quad \alpha = (\exists x)\alpha = (\forall x)\alpha$. Se cioé una variabile é vincolata, o non é presente, la sua quantificazione non produce alcun cambiamento nel valore di verità della frase.

2. Interazioni quantificatori - negazione
   - $\neg(\forall x)(\alpha) \equiv \exists x(\neg\alpha)$
   - $\neg(\exists x)(\alpha) \equiv \forall x(\neg\alpha)$
3. Interazione quantificatori - congiunzione/disgiunzione
   - $(\forall x)(\alpha \wedge \beta) \equiv (\forall x)(\alpha) \wedge (\forall x)(\beta)$
   - $(\exists x)(\alpha \vee \beta) \equiv (\exists x)(\alpha) \vee (\exists x)(\beta)$
     Per le successive uguaglianze bisogna considerare le condizioni in cui vanno applicate:
   - $\left.\begin{array}{l}(\forall x)(\alpha \vee \beta) \equiv (\forall x)(\alpha) \vee (\forall x)(\beta)\\(\exists x)(\alpha \wedge \beta) \equiv (\exists x)(\alpha) \wedge (\exists x)(\beta)\end{array}\right\}(\alpha,\beta) \in \mathcal{L}_{\mathcal{A}}, x_i \notin F_{\mathcal{V}}(\beta)$
4. Interazione quantificatori - implicazione
   - $\left.\begin{array}{l}(\forall x)(\alpha) \to \beta \equiv (\exists x)(\alpha \to \beta)\\(\exists x)(\alpha) \to \beta \equiv (\forall x)(\alpha \to \beta)\end{array}\right\}(\alpha,\beta) \in \mathcal{L}_{\mathcal{A}}, x \notin F_{\mathcal{V}}(\beta)$
   - $\left.\begin{array}{l}\alpha \to (\forall x)(\beta) \equiv (\forall x)(\alpha \to \beta)\\\alpha \to (\exists x)(\beta) \equiv (\exists x)(\alpha \to \beta)\end{array}\right\}(\alpha,\beta) \in \mathcal{L}_{\mathcal{A}}, x \notin F_{\mathcal{V}}(\alpha)$

Osservazioni. $\neg(\exists x)(\forall y)A(x,y) \equiv (\forall x)\neg(\forall y)A(x,y) \equiv (\forall x)(\exists y)\neg A(x,y)$ ◀

Varianti alfabetiche.

Siano date le due frasi:

$$\alpha = (\forall x)A(x) \qquad \beta = (\forall y)B(y)$$

Ci si può domandare se le due frasi siano equivalenti, una volta che - ad esempio nella prima - si sia sostituita ogni occorrenza di x con una occorrenza di y: indicheremo con la scrittura $\alpha[y|x]$ tale sostituzione Nel caso considerato, la risposta é affermativa, ma in generale essa non lo sarà: si consideri infatti la seguente equivalenza (errata!): data $\alpha = (\forall x)((\exists x)A(y,x))$ si consideri:

$$(\forall y)\alpha[y|x] = (\forall y)((\exists y)A(y,y))$$

é evidente come tale uguaglianza generi una frase che non ha piú il significato che aveva in precedenza! Nel caso piú generale é *dunque necessario valutare le occorrenze libere di una variabile che sono C - comandate dal quantificatore*, e sostituirle con una nuova variabile (che cioé non occorre in precedenza in α):

$$\left.\begin{array}{c}\forall z\\\exists z\end{array}\right\}\alpha[z|x] \equiv \left.\begin{array}{c}\forall x\\\exists x\end{array}\right\}\alpha \qquad \text{ove } z \text{ non occorre in } \alpha.$$

Esempio.
$(\forall x)A(x) \vee (\forall x)B(x) \equiv (\forall x)(A(x) \vee (\forall x)B(x)) \equiv$
$\equiv (\forall x)(A(x) \vee (\forall z)B(z)) \equiv (\forall x)(\forall z)(A(x) \vee B(z))$

3.2.3 Enunciati e variabili libere.

Definizione 3.12 *enunciato Si definisce **enunciato** una frase che non contiene variabili libere.*

Si consideri ora una frase che, al contrario, contiene k variabili libere: $\alpha \Leftrightarrow F_{\mathcal{V}}(\alpha) = \{x_1, x_2, ..., x_k\}$. Essa può essere ricondotta in forma di enunciato vincolando tutte le k - occorrenze con altrettanti quantificatori. Cosí, per una frase con una sola variabile libera, sarà possibile creare due enunciati:

$$\alpha : F_{\mathcal{V}}(\alpha) = \{x\} \Longleftrightarrow \begin{cases} (\exists x)A(x) \\ (\forall x)A(x) \end{cases}$$

In generale, per k variabili libere, il numero di possibili enunciati é dato dall'espressione $2^k k!$, che rapidamente diventa molto grande.

Definizione 3.13 *chiusura esistenziale, chiusura universale Sia data $\alpha \in \mathcal{L}_{\mathcal{A}}$; si definiscono:*

- ***chiusura universale** di α, indicata dalla scrittura $\bar{\alpha}^{\forall}$ l'enunciato che si ottiene vincolando con un quantificatore universale tutte le variabili libere in α*
- ***chiusura esistenziale** di α, indicata dalla scrittura $\bar{\alpha}^{\exists}$ l'enunciato che si ottiene vincolando con un quantificatore esistenziale ogni variabili libere in α*

Valgono i due seguenti teoremi:

Teorema 3.2.2

α é soddisfacibile se e soltanto se lo é la sua chiusura esistenziale

Teorema 3.2.3

α é una tautologia se e soltanto se lo é la sua chiusura universale

$\triangledown$ *Dimostrazione.*
Diamo un *esempio dimostrativo* del primo teorema enunciato: sia data una frase α per cui $\mathcal{F}_{\mathcal{V}}(\alpha) = \{x, y\}$, e dimostriamo la doppia implicazione:

$$\alpha \text{ é soddisfacibile } \Longleftrightarrow \bar{\alpha}^{\exists} \text{ é soddisfacibile}$$

Verifichiamo entrambe le direzioni:

- $v_{\mathcal{M},\eta}(\alpha) = 1 \Longrightarrow v_{\mathcal{M},\eta}(\exists x)(\exists y)(\alpha) = 1$ é banalmente verificata. Infatti, per l'ipotesi, esiste certamente almeno una coppia $(p, q) \in M$ tale che $v_{\mathcal{M},\eta(x|p)(y|q)}(\alpha) = 1$. In questo caso, é sufficiente porre: $p = \eta(x), q = \eta(y)$.
- Nel secondo caso, $v_{\mathcal{M},\eta}(\exists x)(\exists y)(\alpha) = 1$ significa che esiste una qualche assegnazione ξ per cui $v_{\mathcal{M},\xi(x|p)(y|q)}(\alpha) = 1$. Può accadere che $\xi \neq \eta$, cioé l'implicazione vale ugualmente, ma non necessariamente attraverso la stessa assegnazione; esiste però certamente una struttura $\mathcal{M}'$ ove l'uguaglianza é verificata.

$\triangle$

Teorema 3.2.4 *Dato un insieme di frasi Γ, tale insieme é soddisfacibile se e soltanto se lo é l'insieme $\bar{\Gamma} = \{\bar{\alpha}^{\exists} | \alpha \in \Gamma\}$.*

$\triangledown$ *Dimostrazione.*
Segue direttamente dalla definizione di soddisfacibilità per un insieme di frasi, e dal teorema dimostrato appena sopra. $\triangle$

3.3 Algoritmo di Skölem

3.3.1 Forma normale prenessa.

Definizione 3.14 *forma normale prenessa Data una frase α si dice **forma normale prenessa** una frase equivalente alla data dove i quantificatori sono all'inizio della stessa.*

Ogni frase può essere riportata in forma normale prenessa, mediante successive applicazioni delle equivalenze semantiche esposte in precedenza, a patto di prestare la dovuta attenzione alle situazioni in cui si rende necessario l'utilizzo di una variante alfabetica.

Illustriamo la procedura con un esempio: sia data la frase

$$(\underbrace{\forall x}_{3} A(x,y) \to \underbrace{\exists y}_{4} B(x,y)) \vee \underbrace{\forall x}_{1} (\underbrace{\exists z}_{2} C(x,z))$$

Si portano ora in testa i quantificatori nell'ordine segnato. Per estrarre il primo quantificatore, bisogna osservare che una variabile x compare nei predicati A, B, ma non ovviamente la 'stessa' x, poiché il quantificatore (1) non C - comanda sui predicati A, B e pertanto la sua estrazione comporta l'uso di una variante alfabetica, che permette di non vincolare anche queste occorrenze. Sia ad esempio la variabile t:

$$\forall t \left\{ (\underbrace{\forall x}_{3} A(x,y) \to \underbrace{\exists y}_{4} B(x,y)) \vee (\underbrace{\exists z}_{2} C(t,z)) \right\}$$

Allo stesso modo si può estrarre z, ma dal momento che in questo caso l'unica occorrenza di una qualche z é solo nella frase C - comandata dal quantificatore da estrarre, questo può essere estratto semplicemente:

$$\forall t \exists z \left\{ (\underbrace{\forall x}_{3} A(x,y) \to \underbrace{\exists y}_{4} B(x,y)) \vee C(t,z) \right\}$$

L'estrazione di (3), poiché esso non C - comanda la x in B, va realizzato mediante l'utilizzo di una variante alfabetica, ad esempio la variabile s:

$$\forall t \exists z \exists s \left\{ (A(s,y) \rightarrow \underbrace{\exists y}_{4} B(x,y)) \vee C(t,z) \right\}$$

L'estrazione di (4) va fatta in due tempi: esso va estratto progressivamente da tutte le frasi in cui é contenuto, verificando di volta in volta la necessità dell'utilizzo di una variante. Nel primo caso si renderà necessaria (r)...

$$\forall t \exists z \exists s \left\{ \exists r (A(s,y) \rightarrow B(x,r)) \vee C(t,z) \right\}$$

...nel secondo caso invece no:

$$\forall t \exists z \exists s \exists r \left\{ (A(s,y) \rightarrow B(x,r)) \vee C(t,z) \right\}$$

Quest'ultima espressione é la forma normale prenessa della frase iniziale: si osservino i quantificatori tutti sulla sinistra.

$\triangleright$ `Esercizio.`
Riportare i forma normale prenessa la seguente frase:

$$((\forall t)A(t,y) \rightarrow (\exists y)B(x,y)) \vee (\forall x)(\exists z)C(x,z)$$

$\triangleleft$

3.3.2 Algoritmo di Skölem

L'algoritmo di Skölem é una procedura che permette di cancellare tutti i quantificatori esistenziali di una frase in forma normale prenessa.

La procedura funziona nel modo seguente: si scorrono tutti i quantificatori:

- quando si incontra un quantificatore universale lo si tralascia
- quando si incontra un quantificatore esistenziale lo si cancella e si sostituiscono tutte le occorrenze della variabile quantificata con una nuova funzione delle variabili quantificate dai quantificatori a sinistra di quello cancellato

Un'applicazione, sull'esempio introdotto in precedenza, é la seguente:

$$\forall t \exists z \exists s \exists r \left\{ (A(s,y) \rightarrow B(x,r)) \vee C(t,z) \right\}$$

Eliminando la variabile quantificata z:

$$\forall t \exists s \exists r \left\{ (A(s,y) \rightarrow B(x,r)) \vee C(t,f(t)) \right\}$$

Eliminando la variabile quantificata s:

$$\forall t \exists r \left\{ (A(g(t),y) \rightarrow B(x,r)) \vee C(t,f(t)) \right\}$$

Eliminando la variabile quantificata r:

$$\forall t \left\{ (A(f(t),y) \rightarrow B(x,h(t))) \vee C(t,f(t)) \right\}$$

Quest'ultima espressione corrisponde alla frase α *skölemizzata*, anche indicata con α^S

Vediamo un altro esempio:

$$\forall x \forall y \exists z \forall t \exists s A(x, y, t, z, s)$$

Eliminando la variabile quantificata z:

$$\forall x \forall y \forall t \exists s A(x, y, t, f(x, y), s)$$

Eliminando la variabile quantificata s:

$$\forall x \forall y \forall t A(x, y, t, f(x, y), g(x, y, t))$$

L'ultima espressione é in forma di Skölem.

Osservazioni.

- Ogni funzione introdotta in sostituzione delle variabili quantificate deve essere una funzione che non appartiene già al vocabolario; per cui il vocabolario delle frasi in forma skölemizzata può essere piú ampio di quello delle frasi originarie.
- Una frase skölemizzata é in forma di enunciato universale. Si può ottenere un enunciato universale da qualunque frase di un linguaggio del primo ordine applicando di seguito la procedura di normalizzazione e di skölemizzazione.

◀

Teorema 3.3.1 *Se una frase α é soddisfacibile, allora lo é anche la sua corrispondente skölemizzata.*

▽ *Dimostrazione.*
Si consideri l'enunciato $(\exists x)A(x)$ e i due casi:

- a é una costante del linguaggio $\mathcal{L}_{\mathcal{A}}$
- a non é una costante del linguaggio

(primo caso):

Sia $A(a)$ una frase soddisfacibile. Significa che esiste una struttura $\mathcal{M} = (M, g)$ nella quale $a_M \in A_M \subset M$. Esiste certamente una seconda struttura $\mathcal{M}' = (M' = M, g' = g)$ ed una seconda assegnazione $\eta : \mathcal{V} \to M$ tale che, per un qualche $p \in M'$, valga che $\eta(p) = a_M$. In questo caso, allora, $v_{(\mathcal{M}',\eta)}(\exists x)A(x) = 1$.

Viceversa, sia soddisfacibile la frase quantificata. Esiste certamente un elemento dell'universo del discorso $p \in M$ per cui $A(p) = 1$. Ponendo allora $a = p$ mediante una funzione $\tilde{g} = \begin{cases} g' & \text{per costanti } \neq a \\ p & \text{per costanti } = a \end{cases}$ si ottiene una struttura $(\mathcal{M}, \tilde{g})$ nella quale l'uguaglianza é verificata.

(secondo caso): Ora a non é una costante del linguaggio. Ciò significa che il procedimento di skölemizzazione introduce nuove costanti all'interno dell'alfabeto del linguaggio. Si supponga per semplicità che $\mathcal{C} = \emptyset$. Il vocabolario di $A(a)$ sarà allora del tipo: $\mathcal{A} = (\mathcal{C} = \{a\}, \mathcal{V}, \mathcal{P}, \mathcal{F})$.

Sia ora soddisfacibile la frase quantificata. Allora esiste una struttura $\mathcal{M}$ con $g : A \to A_M \subset M$ ed una assegnazione η tale che $v_{(\mathcal{M},\eta)}(\exists x)A(x) = 1$, cioé esiste un p tale che $p \in M \iff p \in A_M$. In questo caso esiste certamente una seconda struttura $\mathcal{M} = (M, \tilde{g})$ (differente dalla prima) ove $\tilde{g} = \begin{cases} g & \text{su } \mathcal{P}, \mathcal{F} \\ \tilde{g}(a) = p \end{cases}$, per la quale $A(a)$ é soddisfacibile.

Viceversa, sia soddisfacibile la frase non quantificata. Allora esiste certamente NON LO SO COSA. DA FARE. $\triangle$

Osservazioni. Sia data la frase: $\alpha = (\forall x)(\exists y)A(x, y)$. La relativa frase skölemizzata é: $\alpha^S = (\forall x)A(x, f(x))$, ove la funzione $f \in \mathcal{F}$ se e soltanto se l'insieme $\mathcal{F}$ contiene funzionali ad un posto, in caso contrario f é una nuova funzione. Segue un secondo esempio di dimostrazione. ◀

3.3.3 Teorema di Herbrandt - Skölem

Questo importante teorema riconduce il problema della soddisfacibilità di un insieme di frasi del primo ordine alla soddisfacibilità di un corrispondente insieme di frasi del linguaggio proposizionale, che può essere poi elaborato sintatticamente con i metodi introdotti nel precedente capitolo.

Sia dato un linguaggio del primo ordine $\mathcal{L}_\mathcal{A}$, con un vocabolario che contiene un insieme di costanti non vuoto.

- Si definisce l'insieme $\mathcal{H}_\mathcal{A}$, insieme dei termini del linguaggio dato che non contengono variabili. Vale di conseguenza che $\mathcal{H}_\mathcal{A} \subset Ter(\mathcal{L}_\mathcal{A})$.
- Si definisce poi un insieme $\mathcal{B}_\mathcal{A}$, insieme delle frasi atomiche del linguaggio dato che non contengono variabili.

Osservazioni. Sia dato un alfabeto $\mathcal{A}$, ove: $\mathcal{C} = \{a, b\}$, $\mathcal{F} = \{f(*)\}$ cioé un funzionale unario, e $\mathcal{P} = \{A(*, *)\}$, cioé un predicato a due posti. Allora:

$$\mathcal{H}_\mathcal{A} = \{a, b, f(a), f(b), f(f(a)), f(f(b), ...\}$$

In generale, quando l'insieme dei funzionali non é vuoto, l'insieme $\mathcal{H}_\mathcal{A}$ sarà infinito. Poi:

$$\mathcal{B}_\mathcal{A} = \{A(a, b), A(b, a), A(f(a), b), A(a, f(b)), ...\}$$

Anche in questo caso, l'insieme sarà infinito quando esiste almeno un funzionale nell'alfabeto. ◀

Osservazioni. Detto $\mathcal{L}_{\mathcal{B}_\mathcal{A}}$ il linguaggio proposizionale generato dall'insieme $\mathcal{B}_\mathcal{A}$, vale che $\mathcal{L}_{\mathcal{B}_\mathcal{A}} \subset \mathcal{L}_\mathcal{A}$. ◀

Definizione 3.15 *espansione di Herbrandt Sia ora Λ un insieme di enunciati universali (in forma prenessa) di $\mathcal{L}_\mathcal{A}$. Si dice **espansione di Herbrandt** di un enunciato universale α l'insieme definito ricorsivamente come segue:*

- *se in α non figurano quantificatori, l'espansione coincide con α stesso: $Her(\alpha) = \{\alpha\}$*
- *se α contiene quantificatori, sarà nella forma: $\alpha = \forall x_1 \forall x_2 ... \forall x_k \beta$. In questo caso $Her(\alpha) = B[t_1|x_1, t_2|x_2, ..., t_n|x_n]$, con $(t_1, ..., t_k) \in \mathcal{H}_\mathcal{A}$*

Osservazioni. Sia data la frase: $\alpha = (\forall x)(\forall y)\,[(x = y) \vee \neg A(x, y)]$ L'espansione di Herbrandt di questa frase é allora:

$$Her(a) = \{(a = b) \vee \neg A(a, b); (a = a) \vee \neg A(a, a); a = f(b) \vee \neg A(a, f(b)); ...\}$$

◀

Per l'insieme dato Λ di frasi, si pone: $Her(\Lambda) = \bigcup_{\alpha \in \Lambda} Her(\alpha)$

Teorema 3.3.2 (di Herbrandt - Skölem) *Un insieme di frasi Λ é soddisfacibile in $\mathcal{L}_\mathcal{A}$ se e soltanto se lo é la corrispondente espansione di Herbrandt $Her(\Lambda)$*

$\triangledown$ *Dimostrazione.*
Sia Γ un insieme di frasi di un linguaggio del primo ordine. Si ponga ogni frase dell'insieme in forma normale prenessa, ottenendo l'insieme $\Gamma_{f.n.p.}$. Si realizzi la chiusura esistenziale di tutte le frasi di questo nuovo insieme, ottenendo un terzo insieme $\bar{\Gamma}^{\exists}_{f.n.p.}$. L'insieme finora ottenuto é soddisfacibile se e solo se lo é l'insieme iniziale, per quanto visto in precedenza. Si skölemizzi ogni frase dell'ultimo insieme. Si ottiene un nuovo insieme $\tilde{\Gamma} = (\bar{\Gamma}^{\exists}_{f.n.p.})^S$; tale insieme é composto di soli enunciati universali.

Ora, ogni frase di questo nuovo insieme deve essere espansa in forma di Herbrandt. É immediato intuire per la definizione di soddisfacibilità applicata ad una frase quantificata universalmente, che tale frase é soddisfacibile se e soltanto se lo é la relativa espansione di Herbrandt: basta infatti porre $M = \mathcal{H}_\mathcal{A}$ per cui ogni assegnazione η avrà valori in $\mathcal{H}_\mathcal{A}$. Tuttavia, come dimostrato in precedenza, tale insieme é generalmente infinito, per cui la valutazione della soddisfacibilità di una frase quantificata universalmente é di conseguenza impossibile in un tempo finito.

Siano dati un insieme di enunciati universati $\hat{S}$: ad esso si può associare l'insieme $Her(\hat{S})$, ottenuto mediante espansione di Herbandt. All'enunciato universale $\forall x_1 ... \forall x_k \beta$ corrisponde $\beta[t_1|x_1, t_2|x_2, ..., t_n|x_n]$ ove $t_1, ...t_k \in Ter(\mathcal{L}_\mathcal{A})$ non contengono variabili. A cui corrisponde una clausola di un linguaggio proposizionale. $\triangle$

3.4 Metodi sintattici al primo ordine.

I metodi sintattici enunciati nel contesto dei linguaggi proposizionali hanno un corrispettivo all'interno dei linguaggi predicativi. Prima di enunciare in dettaglio come questi metodi vengono realizzati al primo ordine, enunciamo un teorema e alcune definizioni che saranno utili in seguito.

3.4.1 Compattezza semantica.

Il seguente teorema enuncia un criterio per valutare la soddisfacibilità di un insieme infinito.

Teorema 3.4.1 (di compattezza semantica) *Sia* Γ *un insieme infinito di frasi.* Γ *é soddisfacibile se e soltanto se ogni sottoinsieme finito* $\Gamma' \subset \Gamma$ *lo é.*

$\triangledown$ *Dimostrazione.*
L'implicazione verso destra é immediata. Sia ora $\Gamma' \subset \Gamma$ un sottoinsieme finito di Γ. Dimostreremo che, se ogni sottoinsieme finito é soddisfacibile allora anche Γ é soddisfacibile. O, il che é equivalente, che se Γ é insoddisfacibile allora esiste almeno un sottoinsieme Γ' che lo é.

Per assurdo, sia dunque Γ un insieme insoddisfacibile. Allora $\Gamma \xmapsto{\mathcal{DN}} \perp$ cioé esiste un albero finito che ha come radice l'assurdo e nodi terminali in Γ. Sia Γ' l'insieme delle foglie di questo albero: allora $\Gamma' \xmapsto{\mathcal{DN}} \perp \Longrightarrow \Gamma' \models \perp$, cioé Γ' é insoddisfacibile. $\triangle$

3.4.2 Sostituzioni.

Sia dato un insieme di variabili $\mathcal{V}' \subset \mathcal{V}$. Si definisce **sostituzione** il seguente *operatore*:

$$\sigma = \{t_1|x_1, t_2|x_2, ..., t_n|x_n\}$$

ove:

- $x_j \in \mathcal{V}', i \neq j \Leftrightarrow x_i \neq x_j$
- $t_j \in Ter(\mathcal{L}_\mathcal{A})$

Osservazioni. É definita anche la sostituzione nulla ϵ: essa é tale per cui $t_j = x_j \quad \forall j$. ◀

Sia ora data un'espressione E del linguaggio $\mathcal{L}_\mathcal{A}$ (cioé un termine, o una frase atomica o una sua negazione), ed una qualche sostituzione σ; allora $E\sigma$ é una nuova espressione ottenuta sostituendo ogni occorrenza della variabile x_j con il corrispondente termine t_j.

Osservazioni. Sia data l'espressione $E = A(x, y)$ e la sostituzione $\sigma = \{u|x, g(z)|y\}$. L'espressione $E\sigma = A(u, g(z))$. Su una differente espressione $E' = B(f(x), y, t)$ la stessa sostituzione agisce come segue: $E'\sigma = B(f(u), g(z), t)$. ◀

É possibile effettuare piú sostituzioni: date due sostituzioni σ, η si definisce **sostituzione prodotto** $\tau = \sigma\eta$:

- se una delle due sostituzioni é la sostituzione nulla, la sostituzione prodotto é uguale all'altra.

- se entrambe le trasformazioni sono non nulle, siano esse allora $\sigma = \{t_1|x_1, t_2|x_2, ..., t_k|x_k\}$ e $\eta = \{s_1|y_1, s_2|y_2, ..., s_h|y_h\}$
  1. si applica la seconda trasformazione ad ogni termine della prima; si inserisce in fondo la seconda trasformazione:
  $$\{t_1\eta|x_1, t_2\eta|x_2, ..., t_k\eta|x_k, s_1|y_1, s_k|y_k\}$$
  2. si cancellano i termini del tipo $s_i|y_i$ se y_i é uguale ad un qualche x_j: ad essi corrispondono 'sostituzioni doppie' per uno stesso termine x_j
  3. si cancellano i termini $t_j\eta|x_j$ se $t_j\eta = x_j$: ad essi corrispondono 'sostituzioni nulle'.

  i termini rimasti costituiscono la sostituzione prodotto.

Per come é definita, la sostituzione prodotto gode delle seguenti proprietà:

1. $(E\sigma)\eta = E(\sigma\eta)$
2. proprietà associativa: $\sigma(\eta\tau) = (\sigma\eta)\tau$

3.4.3 Unificazione.

Dato un insieme di espressioni $E = \{E_1, ..., E_k\}$, esso si dice **unificabile** se esiste almeno una sostituzione σ tale che:

$$E_1\sigma = E_2\sigma = ... = E_k\sigma$$

σ é detto **unificatore** dell'insieme E.

Osservazioni. Siano date le espressioni: $E_1 = A(x), E_2 = A(f(y))$. Per $\sigma = \{f(y)|x\}$ vale che $E_1\sigma = E_2\sigma = A(f(y))$.

Siano date invece le espressioni: $E_1 = A(x), E_2 = A(f(x))$. Qualunque sostituzione del tipo $\sigma^* = \{t|x\}$ si ottengono le espressioni: $E_1\sigma^* = A(t), E_2\sigma^* = A(f(t))$ che non sono unificabili. ◄

Teorema 3.4.2 *Se $\{E_1, ..., E_k\}$ sono espressioni unificabili, allora tra gli unificatori di queste espressioni ne esiste uno piú generale degli altri, nel senso che se η é un altro unificatore di queste espressioni allora $\eta = \sigma\tau$ per qualche τ.*

3.5 Metodo di risoluzione al primo ordine.

In un linguaggio $\mathcal{L}$ del primo ordine il vocabolo **letterale** identifica una frase atomica o la sua negazione, come nel caso proposizionale. Una **clausola** é analogamente un insieme finito e non vuoto di letterali. L'aspetto al tempo stesso critico e fondamentale é nella nuova definizione di clausola ottenuta per risoluzione.

Definizione 3.16 *risoluzione per clausole Siano date due clausole C_1, C_2 non vuote, ed una terza clausola C. Si dice che C é deducibile per risoluzione da C_1, C_2 quando:*

1. *(separazione delle variabili): esistono due sostituzioni* σ_1, σ_2 *tali che* $C_1\sigma_1$ *e* $C_2\sigma_2$ *non hanno variabili in comune*

2. *C'é un numero finito di letterali* $\{l_1^{'}, l_2^{'}, ..., l_k^{'}\} \subset C_1\sigma_1$ *con* $k \geq 1$ *ed un numero finito di letterali* $\{l_1^{''}, l_2^{''}, ..., l_h^{''}\} \subset C_2\sigma_2$ *con* $h \geq 1$ *tali che l'insieme* $\{\bar{l_1^{'}}, \bar{l_2^{'}}, ..., \bar{l_k^{'}}, l_1^{''}, l_2^{''}, ..., l_h^{''}\}$ *é unificabile mediante una qualche sostituzione* σ*; sia* η *l'unificatore piú generale per l'insieme di letterali considerato*

3. *La clausola* C *ha il seguente aspetto:*

$$C \equiv [(C_1\sigma_1 \setminus \{l_1^{'}, ..., l_k^{'}\}) \cup (C_2\sigma_2 \setminus \{l_1^{''}, ..., l_h^{''}\})]\eta$$

Sia dato ora un insieme di clausole $\Gamma \subset \mathcal{L}_{\mathcal{A}}$ e C una clausola. La scrittura $S \overset{\mathcal{R}}{\longmapsto} C$ (che si legge 'C é derivabile per risoluzione da Γ') significa:

- $C \in \Gamma$
- $C \notin \Gamma$ ed esiste un albero di risoluzione con radice C e foglie in Γ

Si osservi in primo luogo che, ad ogni clausola, corrisponde un enunciato universale. Dato infatti un insieme Γ di frasi, la sua chiusura esistenziale é ugualmente soddisfacibile o insoddisfacibile. Da questa, attraverso la procedura di normalizzazione (forma normale prenessa) e di seguito la procedura di skölemizzazione ottengo un insieme $\hat{\Gamma}$ di enunciati universali; anch'esso soddisfacibile se e soltanto se lo é l'insieme iniziale.

Ognuna delle frasi di questo nuovo insieme sarà del tipo:

$$(\forall x_1)(\forall x_2)...(\forall x_k)\beta$$

Si può riportare β in forma normale congiuntiva, per cui:

$$(\forall x_1)(\forall x_2)...(\forall x_k)(\alpha_1 \wedge \alpha_2...\alpha_p)$$

ove il singolo a_j é una disgiunzione finita di letterali al primo ordine.

Per una delle equivalenze semantiche introdotte in precedenza, vale che:

$$(\forall x_1)...(\forall x_k)\beta = (\forall x_1)...(\forall x_k)\alpha_1 \wedge ... \wedge (\forall x_1)...(\forall x_k)\alpha_p =$$

ove ad ogni α_j corrisponde una clausola: $\alpha_j \Longleftrightarrow \{l_1, ..., l_k\}$

Teorema 3.5.1 *Sia* Γ *un insieme di clausole. Se* Γ *é insoddisfacibile, allora* $S \overset{\mathcal{R}}{\longmapsto} \square$*, e viceversa.*

$\triangledown$ *Dimostrazione.*

Il teorema precedente ha un duplice significato, per ciascuno dei due versi dell'implicazione:

- Ciò che é deducibile sintatticamente lo é anche semanticamente (facile)
- Ciò che é deducibile semanticamente si deduce anche sintatticamente (difficile, lo dimostriamo di seguito)

Il passaggio critico consiste nel dimostrare che esiste un albero di prova del linguaggio proposizionale che si pu 'rimontare' al primo ordine. Per ottenere questo importante risultato, dobbiamo prima enunciare il seguente:

Lemma 3.1 (Lifting - lemma) *Siano dati $C'_1, C'_2 \in \tilde{S}, C' \in \mathcal{L}_{\mathcal{A}}$ e sia possibile la derivazione $C'_1, C'_2 \longmapsto C'$ mediante risoluzione [proposizionale]; allora esiste una clausola $C \in \mathcal{L}_{\mathcal{A}}$ di cui C' é espansione di Herbrandt (cioé $C' = C\delta$) C che é ottenuta per risoluzione al primo ordine da C_1, C_2*

Questo lemma afferma perciò che, date tre clausole di un linguaggio proposizionale, delle quali la prima sia deducibile per risoluzione dalle altre due, se esistono due trasformazioni δ_1, δ_2 tali che valgano le uguaglianze $C'_1 = C_1\delta_1, C'_2 = C_2\delta_2$ allora esiste certamente una trasformazione δ che fa corrispondere alla clausola derivata del linguaggio proposizionale una clausola derivata al primo ordine; questo permette di 'ricostruire' al primo ordine un albero di risoluzione ottenuto nel corrispondente linguaggio proposizionale.

$\triangledown$ *Dimostrazione.*
Si può supporre, senza perdere di generalità, che C_1, C_2 non abbiano variabili in comune. Allora, posto $\delta = \delta_1\delta_2$ valgono le seguenti uguaglianze: $C'_1 = C_1\delta$, $C'_2 = C_2\delta$. Cioé: esistono unsieme di letterali in C_1 che possono essere unificati tramite δ, e analogamente per C_2:

$$\exists\{l'_1, l'_2, ..., l'_m\} \in C_1 \Longleftrightarrow l'_j\delta = \bar{l} \quad j = 1, ..., m$$
$$\exists\{l''_1, l''_2, ..., l''_n\} \in C_2 \Longleftrightarrow l''_i\delta = l \quad i = 1, ..., n$$

allora $\{l'_1, ..., l'_m, l''_1, ..., l''_n\}$ è unificabile e δ é un unificatore.

Sia ora θ un unificatore generale, tale cioé che $\delta = \theta\rho$ Allora:

$$C' = (C'_1 \setminus \{\bar{l}\}) \cup (C'_2 \setminus \{l\}) = (C_1\delta \setminus \{\bar{l}\}) \cup (C_2\delta \setminus \{l\}) =$$
$$= (C_1\delta \setminus \{l'_1, ..., l'_m\})\delta \cup (C_2\delta \setminus \{l''_1, ..., l''_m\})\delta =$$
$$= (C_1\delta \setminus \{l'_1, ..., l'_m\})\delta \cup (C_2\delta \setminus \{l''_1, ..., l''_m\})\theta\rho = C\rho$$

ove

$$C = (C_1\delta \setminus \{l'_1, ..., l'_m\})\delta \cup (C_2\delta \setminus \{l''_1, ..., l''_m\})\theta$$

é una clausola al primo ordine. $\triangle$

É molto più semplice dimostrare la seconda parte del teorema, e cioé che: $S \overset{\mathcal{R}}{\longmapsto} \square \Longrightarrow S$ é insoddisfacibile.

Infatti, se la clausola vuota può essere dedotta dall'insieme S, allora esiste un albero di risoluzione che, ad ogni terna di nodi (C, C_1, C_2), - ove ad ogni nodo corrisponde una clausola - per cui valga che $C_1, C_2 \longmapsto C$, allora esistono certamente $\{\alpha^C, \alpha^{C_1}, \alpha^{C_2}\}$ tali che $\{\alpha^{C_1}, \alpha^{C_2} \models \alpha^C\}$. $\triangle$

Lemma 3.2 (di correttezza) *Siano date C, C_1, C_2 clausole al primo ordine: se $C_1, C_2 \overset{\mathcal{R}}{\longmapsto} C$ allora $\{\alpha^{C_1}, \alpha^{C_2} \models \alpha^C\}$*

$\triangledown$ *Dimostrazione.*
dimostrare per esercizio. $\triangle$

Osservazioni.

Dato il seguente insieme di frasi:

1. α: Ogni uomo é stupido
2. β: Qualche uomo é disonesto
3. γ: Qualche uomo é stupido e disonesto

dimostrare che $\alpha, \beta \models \gamma$

Questo significa dimostrare che l'insieme $\{\alpha, \beta, \neg\gamma\}$ é insoddisfacibile. Assumeremo i predicati:

- U(*): '... é un uomo'
- S(*): '... é stupido'
- D(*): '... é disonesto'

Per cui le frasi si possono tradurre in logica del primo ordine come segue:

1. $(\forall x)(U(x) \to S(x))$
2. $(\exists y)(U(x) \wedge D(x))$
3. $(\exists z)(U(x) \wedge S(x) \wedge D(x))$

Si riconducano queste tre frasi in forma di enunciato universale:
La prima é già in forma di enunciato universale; da essa si ricava la clausola $C_1 = \{\neg U(x), S(x)\}$.
La seconda deve prima essere skölemizzata, e dà origine a due clausole: $C_2 = \{U(a)\}, C_3 = \{D(a)\}$.
La terza deve essere prima posta in forma negata, e dà origine ad una clausola: $C_4 = \{\neg U(x), \neg S(x), \neg D(x)\}$
Dall'insieme di clausole $\{C_1, C_2, C_3, C_4\}$ si deve ora dedurre per risoluzione la clausola vuota.
Si prendano le clausole C_1, C_4. Per la regola di derivazione:

- si deve trovare una coppia di trasformazioni (σ_1, σ_2) tali che la coppia $C_1\sigma_1, C_2\sigma_2$ non abbia variabili in comune. Ad esempio, si può porre: $\sigma_1 = \{y|x\}, \sigma_2 = \epsilon$. In questo modo si ottiene la coppia di clausole: $C_1\sigma_1 = \{\neg U(y), S(y)\}, C_4\sigma_2 = \{\neg U(x), \neg S(x), \neg D(x)\}$
- Si considerino i letterali candidati alla cancellazione: nella prima clausola il letterale $S(y)$, nella seconda clausola $\neg S(x)$. Bisogna trovare un unificatore generale η per l'insieme $\neg S(y), \neg S(x)$. Un possibile unificatore generale è ad esempio: $\eta = \{z|x, z|y\}$
- Si ottiene in seguito la clausola: $C_5 = [C_1\sigma_1 \setminus \{S(y)\}, C_2\sigma_2 \setminus \{\neg S(x)\}]\, \eta = \{\neg U(z), \neg D(z)\}$

Si prendano ora le clausole C_2, C_5; in questo caso:

- Le due clausole sono già a variabili separate; in questo caso assumeremo dunque che le trasformazioni siano nulle: $\sigma_1 = \sigma_2 = \epsilon$
- I letterali candidati alla cancellazione sono rispettivamente: $U(a), \neg U(z)$. Un unificatore generale é allora: $\eta = a|z$

- Si ottiene in seguito la clausola: $C_6 = [\neg D(z)]\,\eta = \neg D(a)$

Si prendano infine le clausole C_3, C_6

- Le due clausole non contengono variabili. Assumeremo ancora $\sigma_1 = \sigma_2 = \epsilon$
- Le due clausole sono già unificate. Per cui assumeremo anche che $\eta = \epsilon$
- Si ottiene la clausola vuota.

◄

Osservazioni.

1. Lucia ama solo ragazzi sportivi
2. Lucia ama qualche ragazzo
3. Qualche ragazzo é sportivo

Si dimostri che $\{1, 2\} \models 3$, ovvero che $\{1, 2, \neg 3\} \longmapsto \square$. Si assumono i seguenti predicati:

1. R(*): '... é un ragazzo'
2. S(*): '... é sportivo'
3. A(*,*): '... ama ...'

e la costante L che rappresenta 'Lucia'.

Tradurremo le frasi come segue:

1. $[(\exists x)(R(x) \wedge \neg S(x))] \rightarrow \neg A(L, x)$
2. $(\exists x)A(L, x)$
3. $(\exists x)\,[R(x) \wedge S(x)]$

Si osservi che la prima frase poteva essere tradotta anche come:

$$(\forall x)\,[A(L, x) \rightarrow (R(x) \wedge S(x))]$$

Tuttavia questa traduzione é più 'forte' della precedente; mentre la prima individuava un sottoinsieme all'interno dell'insieme dei ragazzi, in particolare quello dei ragazzi sportivi, quest'ultima traduzione individua il sottoinsieme dei ragazzi sportivi all'interno dell'universo del discorso, affermando che solo essi sono amati da Lucia; da questa seconda traduzione non é possibile che Lucia ami *animali, cose, situazioni, ecc...*, mentre tutto ciò non era implicito nella prima versione. Una terza versione che tiene conto di questa osservazione, e che é perciò equivalente alla prima é:

$$(\forall x)\,[A(L, x) \rightarrow (\neg R(x) \vee S(x))]$$

La prima frase origina una clausola $C_1 = \{\neg R(a), S(a), \neg A(L, a)\}$ La seconda frase origina due clausole: $C_2 = \{R(a)\}, C_3 = \{A(L, a)\}$ La terza frase origina una clausola: $C_4 = \{\neg R(x), \neg S(x)\}$

Dalle clausole C_1, C_4, mediante trasformazioni $\sigma_1 = \sigma_2 = \epsilon$ e $\eta = \{a|x\}$ si ottiene la clausola $C_5 = \{\neg R(a), \neg A(L, a)\}$ Dalle clausole C_2, C_5, mediante le trasformazioni $\sigma_1 = \sigma_2 = \eta = \epsilon$ si ottiene la clausola $C_6 = \{\neg A(L, a)\}$. Da quest'ultima e dalla clausola C_3 si ottiene, mediante trasformazioni nulle, la clausola vuota.

◀

▷ `Esercizio.`
Da ogni gruppo di frasi, derivare l'ultima del gruppo mediante risoluzione al primo ordine.

1. - Ogni barbiere rade tutti coloro che non si radono da soli
   - Nessun barbiere rade chi si rade da solo
   - Non esistono barbieri
2. - Antonio non passa l'esame, a meno che non risolva tutti i problemi
   - Se Mario si impegna più di Antonio allora Mario risolve qualche problema che Antonio non risolve
   - Chiunque é diverso da Antonio si impegna più di Antonio
   - Antonio non passa l'esame
3. - Ogni passeggero che non é membro dell'equipaggio viene perquisito da almeno un poliziotto
   - Un passeggero o un membro dell'equipaggio é un trafficante di droga
   - Ogni poliziotto non corrotto che perquisisce un trafficante di droga lo arresta
   - Nessun passeggero viene arrestato da alcun poliziotto
   - Qualche poliziotto è corrotto o qualche membro dell'equipaggio é un trafficante di droga

◁

3.6 Deduzione naturale al I ordine.

Analogamente al caso proposizionale, esiste anche al primo ordine il metodo sintattico di deduzione naturale. Esso é composto di un numero molto superiore di regole rispetto al metodo di risoluzione per clausole, ma a suo vantaggio non ha nessuna procedura di *inizializzazione* dei dati iniziali, poiché opera direttamente sulle frasi date.

In seguito, scambieremo il simbolo $\overset{\mathcal{DN}}{\longmapsto}$ con $\longmapsto$, intendendo che la derivazione é ottenuta mediante deduzione naturale.

Analogamente al caso proposizionale, le regole della deduzione naturale si suddividono in *elementari* e *condizionali*.

3.6.1 Regole elementari.

Connettivo di *congiunzione*:

1. *eliminazione* della congiunzione: data una congiunzione, ne posso dedurre ciascuna delle frasi che la compongono.

$$(\wedge, e): \qquad \alpha \wedge \beta \longmapsto \begin{cases} \alpha \\ \beta \end{cases}$$

2. *inserimento* della congiunzione: date due frasi α, β, ne posso dedurre ciascuna delle loro congiunzioni.

$$(\wedge, i): \qquad \{\alpha, \beta\} \longmapsto \begin{cases} \alpha \wedge \beta \\ \beta \wedge \alpha \end{cases}$$

Connettivo di *disgiunzione*:

1. *eliminazione* della disgiunzione: data una disgiunzione, e la negazione di uno dei disgiunti, ne posso dedurre l'altro.

$$(\vee, e): \qquad \begin{array}{l} \{\alpha \vee \beta, \neg\alpha\} \overset{\mathcal{DN}}{\longmapsto} \beta \\ \{\alpha \vee \beta, \neg\beta\} \overset{\mathcal{DN}}{\longmapsto} \alpha \end{array}$$

2. *inserimento* della disgiunzione: date una frasi α, é sempre possibile congiungerla ad una qualunque frase β:

$$(\vee, i): \qquad \{\alpha\} \overset{\mathcal{DN}}{\longmapsto} \begin{cases} \alpha \vee \beta \\ \beta \vee \alpha \end{cases}$$

Connettivo di *implicazione*:

1. *eliminazione* dell'implicazione: data una implicazione $\alpha \to \beta$, e l'implicando α, é sempre possibile dedurne l'implicato; tale regola va sotto il nome di **modus ponens**:

$$(\to, i.1): \qquad \{\alpha \to \beta\}, \{\alpha\} \overset{\mathcal{DN}}{\longmapsto} \beta$$

Ancora, data una implicazione, e la negazione dell'implicato, é sempre possibile dedurne la negazione dell'implicato; tale regola va sotto il nome di **modus tollens**:

$$(\to, i.2): \qquad \{\alpha \to \beta\}, \{\neg\beta\} \overset{\mathcal{DN}}{\longmapsto} \neg\alpha$$

2. *inserimento* dell'implicazione: é una regola condizionale, e verrà esposta nel paragrafo successivo.

Connettivo di *negazione*:

1. *eliminazione* della negazione: data una frase in forma vera e la stessa in forma negata, é possibile dedurne l'assurdo:

$$(\neg, e): \qquad \{\alpha\}, \{\neg\alpha\} \overset{\mathcal{DN}}{\longmapsto} \perp$$

2. *inserimento* della negazione: dato l'assurdo, se ne può dedurre qualunque frase (e la propria negazione):

$$(\neg, i): \qquad \{\perp\} \overset{\mathcal{DN}}{\longmapsto} \neg\alpha$$

Quantificatori:

1. *eliminazione* del quantificatore universale: da una frase universalmente quantificata posso dedurre una frase che al posto della variabile contenga un qualunque termine del linguaggio:

$$(\forall, e): \qquad \{(\forall x)\alpha\} \longmapsto \alpha[t|x] \text{ per qualunque termine } t \in Ter(\mathcal{L}_{\mathcal{A}})$$

2. *introduzione* del quantificatore esistenziale: da una frase ove un termine del linguaggio sostituisce una variabile, posso dedurre che esiste almeno un termine del linguaggio che rende vera la stessa frase ove non sia stata effettuata la sostituzione:

$$(\exists, i): \qquad \{\alpha[t|x] \longmapsto (\exists x)\alpha\} \text{ ove } t \in Ter(\mathcal{L}_{\mathcal{A}})$$

3.6.2 Regole condizionali.

1. Regola di *introduzione dell'implicazione*. Dato un insieme Γ di frasi del linguaggio:

$$(\rightarrow, i): \qquad \Gamma \longmapsto (\alpha \rightarrow \beta) \Longleftrightarrow \Gamma \cup \{\alpha\} \longmapsto \beta$$

2. Regola di *riduzione ad assurdo*: da un insieme di frasi posso derivare una qualunque frase se aggiungendo la negazione di questa ne viene l'assurdo, cioé:

$$(R.A.A.): \qquad \Gamma \longmapsto \alpha \Longleftrightarrow \Gamma \cup \{\neg\alpha\} \overset{\mathcal{DN}}{\longmapsto} \perp$$

3. Regola di *introduzione del quantificatore universale*: da un insieme di frasi posso introdurre il quantificatore universale su una frase se dallo stesso insieme posso dedurre la frase ove alla variabile quantificata si sostituisca una qualche variabile che non compare libera in Γ

$$(\forall, i): \qquad \Gamma \longmapsto (\forall x)\alpha \Longleftrightarrow \Gamma \longmapsto \alpha[y|x]$$

per ogni y che non occorre libera in Γ

4. Regola di *eliminazione del quantificatore esistenziale*: da un insieme di frasi ove ne sia presente una quantificata dal quantificatore esistenziale posso dedurre una nuova frase, eliminando il quantificatore, se dallo stesso insieme posso ricavare la nuova frase sostituendo la variabile quantificata nella frase iniziale per una qualche variabile libera che non occorre libera né nell'insieme dato, né nella frase dedotta:

$$(\exists, e): \qquad \Gamma \cup (\exists x)\alpha \longmapsto \beta \Longleftrightarrow \Gamma \cup \alpha[y|x] \longmapsto \beta$$

per ogni y che non occorre libera in Γ, β

Definizione 3.17 *deduzione naturale Sia dato un insieme Γ ed una frase α. α é deducibile (per deduzione naturale) da Γ se:*

- $\alpha \in \Gamma$ *oppure*
- *esiste un albero finito non banale di radice α i cui nodi terminali sono in Γ ed ogni nodo non terminale é ottenuto dai due figli mediante una qualche regola di Deduzione Naturale.*

Teorema 3.6.1 (di correttezza/completezza della deduzione naturale.) *Sia dato Γ insieme di frasi ed una frase α. Allora α é deducibile sintatticamente mediante deduzione naturale dall'insieme Γ se e soltanto se α é conseguenza semantica di Γ, e viceversa. Cioé:* $\Gamma \overset{\mathcal{DN}}{\longmapsto} \alpha \Longleftrightarrow \Gamma \models \alpha$

Il significato di questo teorema é duplice: La scrittura

$$\Gamma \overset{\mathcal{DN}}{\longmapsto} \alpha \Longrightarrow \Gamma \models \alpha$$

significa che se α é conseguenza semantica di Γ allora essa é anche conseguenza sintattica dello stesso insieme; questa caratteristica é anche detta *proprietà di **correttezza** del metodo di deduzione naturale.*

La scrittura

$$\Gamma \overset{\mathcal{DN}}{\longmapsto} \alpha \Longleftarrow \Gamma \models \alpha$$

significa che, se α é conseguenza semantica di Γ, allora esiste un albero di prova (in Deduzione Naturale) che lo dimostra sintatticamente, a partire dall'insieme Γ; questa proprietà é anche detta *proprietà di **completezza** del metodo di deduzione naturale.*

La proprietà di completezza ha una dimostrazione molto più complessa rispetto alla prima.

Appendice A

Esercizi d'esame.

A.1 Esercizi di risoluzione semantica.

▷ `Esercizio.`
Si dica se il seguente insieme soddisfacibile:

$$\{\forall x \neg A(x,x), \forall x \forall y \forall z [A(x,y) \wedge (y,z) \to A(x,z)], \forall x A(x, f(x))\}$$

◁

▽ *Svolgimento.*
L'insieme é soddisfacibile se e soltanto se esiste una struttura ed una interpretazione ove le frasi che lo compongono siano verificate (cioe' *vere*).

L'insieme é composto da tre frasi, di cui analizziamo di seguito il significato singolarmente; osserviamo prima di tutto che le tre frasi sono tre enunciati universali inerenti al solo predicato $A(*,*)$:

- $\forall x \neg A(x,x)$: significa che il predicato A individua una relazione non riflessiva.
- $\forall x \forall y \forall z [A(x,y) \wedge (y,z) \to A(x,z)]$: significa che il predicato individua una relazione transitiva.
- $\forall x A(x, f(x))$: significa che, per ogni $p \in \mathcal{M}$, la coppia $(p, f_M(p)) \in \mathcal{A}_{\mathcal{M}}$

Una struttura che soddisfa l'insieme di frasi sopra é ad esempio l'insieme dei numeri naturali, ove il predicato A sia la relazione d'ordine ($<$, ma non $\leq$) e dove la funzione $f(x)$ sia ad esempio $f : x \to x+1$. É sufficiente individuare una sola struttura per dichiarare soddisfacibile l'insieme. Si badi, però: l'insieme dato non é una tautologia: é infatti possibile individuare una struttura ove qualche frase non é verificata: ad esempio il predicato di $\leq$, relazione d'ordine che può soddisfare le ultime due frasi, ma non la prima. △

▷ `Esercizio.`
Sia T un connettivo ternario per cui:

$$v(T(a,b,c)) = 1 \longleftrightarrow v(a) + 2v(b) - v(c) \geq 1$$

Scrivere $T(a, b, c)$ in una qualsiasi forma normale. ◁

▽ *Svolgimento.*
Si costruisce la tabella di verità del connettivo:

v(a)	**v(b)**	**v(c)**	**T**
0	*0*	*0*	**0**
0	*0*	*1*	**0**
0	1	0	1
0	1	1	1
1	0	0	1
1	*0*	*1*	**0**
1	1	0	1
1	1	1	1

É preferibile la forma normale congiuntiva, per costruire la quale si osservano le righe ove T ha un valore 0. Da ciascuna riga riga si ottiene un disgiunto di frasi atomiche, in forma vera se la frase atomica é a 0, in forma negata se la frase atomica é ad 1. Perciò si ottiene la seguente espressione per T:

$$T = (a \vee b \vee c) \wedge (a \vee b \vee \neg c) \wedge (\neg a \vee b \vee \neg c)$$

△

▷ `Esercizio.`
Sia T un connettivo ternario per cui:

$$v(T(a, b, c)) = 1 \longleftrightarrow [v(a) - v(b)][-v(a) + v(b) - v(c)] \geq 1$$

Riscrivere $T(a, b, c)$ usando i connettivi $\rightarrow, \neg$. ◁

▽ *Svolgimento.*
Si costruisce la tabella di verità del connettivo:

v(a)	**v(b)**	**v(c)**	**T**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Come si può osservare, il connettivo produce sempre valore 'falso'. A questo punto non é difficile creare una frase nella quale siano utilizzati solo i connettivi indicati, che sia sempre falsa. Ad esempio la frase $\neg(a \rightarrow a)$. Oppure, se si vuole costruire una frase che contenga tutte e tre le frasi atomiche considerate, basta costruirne una ove un implicando vero (del tipo $a \rightarrow a$)) implica una frase falsa: ad esempio:

$$(a \rightarrow a) \rightarrow \neg[(b \rightarrow b) \rightarrow (c \rightarrow c)]$$

△

▷ `Esercizio.`
Descrivere tutti i modelli dell'enunciato:

$$\forall x \forall y [A(x,y) \to \neg A(y,x)]$$

◁

▽ *Svolgimento.*
Si osservi che la frase data é un enunciato universale: entrambe le variabili sono quantificate in tutte le loro occorrenze. Per cui l'interpretazione non dipende dall'assegnazione, ma solo dal dominio del discorso.

Per cui la frase afferma che, data una qualunque coppia $(p,q) \in M$, se la coppia $(p,q) \in A_M$ allora $(q,p) \notin A_M$. Cioé la proposizione A individua una relazione non simmetrica sull'insieme M. Si osservi inoltre che, qualora si verificasse il caso $[a|x] = [b|y]$, cioé si scegliessero le coppie di elementi uguali in M, si otterrebbe la frase $A(x,x) \to \neg A(x,x)$, che risulta vera solo quando l'implicando é falso, cioé quando $(p,p) \notin A_M$. Perciò l'insieme $A_M \subset M$ non contiene le coppie di elementi uguali: A individua allora una relazione non riflessiva. In definitiva, i modelli che soddisfano la frase sono quelli ove la preposizione A individua una relazione non riflessiva e non simmetrica sul dominio del discorso M: ad esempio, si pensi all'insieme dei numeri naturali ove A sia l'usuale relazione di 'minore'. △

▷ `Esercizio.`
Descrivere tutti i modelli dell'enunciato seguente:

$$\forall x \forall y [f(x) = f(y) \to (x = y)] \wedge \neg \forall x \exists y (f(y) = x)$$

◁

▽ *Svolgimento.*
Si osservi primariamente la struttura della frase: essa contiene le variabili x, y, entrambe quantificate da un quantificatore universale, per cui la frase é un enunciato. Questo significa che l'interpretazione non dipende dall'assegnazione, ma soltanto dal dominio del discorso M. All'interno della frase c'é un simbolo funzionale f ad un posto.

La frase é strutturata come congiunto di due sottofrasi, delle quali esaminiamo rispettivamente il significato:

$$f(x) = f(y) \to (x = y)$$

significa che $\forall (p,q) \in M$ se $F_M(p) = F_M(q)$ allora $p = q$. Cioé la funzione f_M é iniettiva. Si osservi la seconda sottofrase privata della negazione:

$$\forall x \exists y (f(y) = x)$$

Essa afferma che per ogni $p \in M$ c'é un $q \in M$ tale che $f_M(q) = p$, cioé la funzione f_M é suriettiva. Poichè la frase é negata, la funzione f_M non é suriettiva.

In definitiva, si vuole che la funzione f_M sia iniettiva, ma non suriettiva. Affinché questo si possa verificare, si deve ammettere l'esistenza di un dominio del discorso infinito. △

▷ `Esercizio.`
Trasformare la frase seguente in un enunciato universale:

$$\forall y^{(1)}[\exists x^{(2)} A(x, y) \to B(y, x)] \wedge^{(*)} \exists y^{(3)}[\forall x^{(4)} C(x, y) \wedge B(x, y)]$$

◁

▽ *Svolgimento.*
Si osservi che nella frase c'é una variabile libera: la variabile x in entrambe le occorrenze del predicato B. Se poi si costruisce l'albero sintagmatico della frase, si osserva che il connettivo piú esterno é una congiunzione $(*)$. Si estraggono per primi i connettivi interni (2,4); poiché essi quantificano rispettivamente solo le frasi atomiche $A(x, x)$ e $C(x, y)$ e poiché la variabile x é presente in entrambe le occorrenze del predicato B, si rende necessario l'uso di una variante alfabetica, per cui:

$$\forall y^{(1)} \forall t[A(t, y) \to B(y, x)] \wedge^{(*)} \exists y^{(3)} \forall s[C(s, y) \wedge B(x, y)]$$

A questo punto si devono estrarre uno per volta i vari quantificatori dalle due frasi congiunte da $(*)$; per i primi due l'estrazione é immediata, poiché non ci sono variabili libere y, t nel secondo congiunto. Si ottiene:

$$\forall y^{(1)} \forall t\{[A(t, y) \to B(y, x)] \wedge^{(*)} \exists y^{(3)} \forall s[C(s, y) \wedge B(x, y)]\}$$

L'estrazione dei due quantificatori richiede più attenzione, per quel che riguarda il quantificatore esistenziale (3); infatti esistono ora occorrenze libere della variabile y nel primo congiunto, che verrebbero erroneamente vincolate se non si utilizzasse una variante alfabetica. Infine si ottiene:

$$\forall y \forall t \exists z \forall s\{[A(t, y) \to B(y, x)] \wedge^{(*)} [C(s, z) \wedge B(x, z)]\}$$

Sfruttando note equivalente semantiche si ottiene poi la frase:

$$\forall y \forall t \exists z \forall s\{[\neg A(t, y) \vee B(y, x)] \wedge^{(*)} C(s, z) \wedge B(x, z)\}$$

A questo punto, prima di trasformare la frase in un enunciato universale é necessario vincolare l'occorrenza libera della variabile x; per un teorema noto questo verrá fatto anteponendo a monte della frase un quantificatore universale su quella variabile:

$$\exists x \forall y \forall t \exists z \forall s\{[\neg A(t, y) \vee B(y, x)] \wedge^{(*)} C(s, z) \wedge B(x, z)\}$$

A questo punto, la frase é già un enunciato: per ottenere un enunciato universale bisogna ora applicare la procedura di skölemizzazione: si ottiene infine l'enunciato universale:

$$\forall y \forall t \forall s\{[\neg A(t, y) \vee B(y, a)] \wedge^{(*)} C(s, f(y, t)) \wedge B(a, f(y, t))\}$$

Si osservi che il vocabolario iniziale della frase era composto come segue:

$$\mathcal{A} = \{\mathcal{C} = \{\emptyset\}, \mathcal{F} = \{\emptyset\}, \mathcal{P} = \{A, B, C\}\}$$

mentre il vocabolario della skölemizzata é piú ampio:

$$\mathcal{A}^S = \{\mathcal{C} = \{a\}, \mathcal{F} = \{f(*, *)\}, \mathcal{P} = \{A, B, C\}\}$$

Se si costruisce l'insieme $\mathcal{H}_\mathcal{A}$ per la skölemizzata, cioé l'insieme di tutti i termini di Ter che non contengono variabili, si ottiene:

$$\mathcal{H}_\mathcal{A} = \{a, f(a, a), f(f(a, a), f(a, a)), ...\}$$

L'espansione di Herbrandt della frase si ottiene sostituendo ad ogni variabile un termine $t \in \mathcal{H}_\mathcal{A}$. △

▷ `Esercizio.`
Si consideri l'enunciato seguente:

$$\alpha = (\forall x(A(x) \rightarrow B(x)) \rightarrow (\exists x A(x) \rightarrow \exists x B(x))$$

1. Dire se esso é una tautologia.

2. Detta $\tilde{\alpha}$ la skölemizzata della frase, dire se essa é una tautologia.

◁

▽ *Svolgimento.*
Si osservi che l'enunciato in questione é composto da due sottofrasi legate tra di loro da una implicazione. In particolare, la sottofrase $(\forall x(A(x) \rightarrow B(x))$ implica la sottofrase: $(\exists x A(x) \rightarrow \exists x B(x))$ Questo significa, per definizione di implicazione, che:

1. Se l'implicando é falso, allora l'implicato può essere vero o falso.

2. Se l'implicando é vero, bisogna considerare il valore di verità dell'implicato

Dunque, se l'implicando é vero, ciò significa che, per ogni $p \in M$, se $p \in A_M$ allora $p \in B_M$, cioé vale che $A_M \subset B_M$. Quando l'implicato é vero, del resto, significa che, se esiste un $p \in M$ tale che $p \in A_M$ allora esiste un $q \in M$ tale che $q \in B_M$, cioé $A_M \neq \emptyset \rightarrow B_M \neq \emptyset$, ma questo é sempre vero, poiché per l'implicando $A_M \subset B_M$.

Ora si porta l'enunciato in forma normale prenessa:

$$\alpha' = \exists x \forall r \exists y[\neg(\neg A(x) \vee B(x)) \vee (\neg A(r) \vee B(y)) =$$

$$= \exists x \forall r \exists y[(A(x) \wedge \neg B(x)) \vee (\neg A(r) \vee B(y))$$

e di seguito lo si skölemizza:

$$\tilde{(\alpha)} = \forall r[(A(a) \wedge \neg B(a)) \vee (\neg A(r) \vee B(b))$$

Come si vede immediatamente, é possibile effettuare una scelta delle costanti a, b e della variabile x in modo che la frase non sia sempre vera; per cui essa non é una tautologia. $\triangle$

$\triangleright$ `Esercizio.`
Si consideri l'enunciato:

$$[\forall x(A(x) \rightarrow B(x))] \rightarrow [\forall x A(x) \rightarrow \forall x B(x)]$$

Dire se esso é una tautologia. $\triangleleft$

$\bigtriangledown$ *Svolgimento.*
Si consideri, ugualmente all'esercizio precedente, il connettivo piú esterno della frase, che in questo caso é l'implicazione centrale. Se l'implicando é falso, la frase é vera qualunque sia il valore di verità dell'implicato. Se al contrario esso é vero, allora anche l'implicando deve essere necessariamente vero.

Se dunque l'implicando é vero, ciò significa che esiste un valore $p \in M$ tale che, se $p \in A_M$ allora $p \in B_M$, ovvero $A_M \subset B_M$. Si consideri allora l'implicato: esso é a sua volta una implicazione: affinché sia vera, a parte il caso in cui la premessa sia falsa, essa afferma che, dato qualunque $p \in M$, se $p \in A_M$ allora per ogni $q \in M$ vale che $q \in B_M$. Cioé essa afferma che, se $A_M \equiv M$ allora $B_M \equiv M$, cioé $A_M \equiv B_M \equiv M$.

La domanda é dunque: $A_M = B_M = M$ é condizione necessaria affinché valga che $A_M \subset B_M$? La risposta é ovviamente no: la seconda é infatti più restrittiva, rispetto alla prima. Infatti la si potrebbe tradurre nella domanda: 'se un primo insieme é incluso in un secondo, allora i due insiemi sono uguali?', la cui risposta é negativa. Si osservi però che l'implicazione al contrario ha invece valore vero: infatti, se due insiemi sono uguali, allora uno di essi é contenuto nell'altro (e viceversa). $\triangle$

A.2 Esercizi di risoluzione per clausole.

$\triangleright$ `Esercizio.`
Si sa che:

1. Ogni gazzella teme chiunque la insegue
2. Ogni leone insegue qualche gazzella o qualche zebra
3. Non ci sono zebre

É possibile dedurne che

qualche gazzella teme qualcuno?

Se no, quale ulteriore informazione aggiunta alle frasi date permette di dedurre l'ultima? $\triangleleft$

$\bigtriangledown$ *Svolgimento.*
Si utilizzeranno i predicati ad un posto:

1. $G(x)$: 'x é una gazzella'
2. $L(x)$: 'x é un leone'
3. $Z(x)$: 'x é una zebra'

e quelli a due posti:

1. $I(x, y)$: 'x insegue y'
2. $T(x, y)$: 'x teme y'

Tradurremo le quattro frasi come segue:

- $\{\forall x \forall y[G(x) \wedge I(x, y)]\} \rightarrow T(y, x)$
- $(\forall x L(x)) \rightarrow [\exists y(G(y) \vee Z(y)) \wedge I(x, y)]$
- $\neg \exists x Z(x)$
- $\exists x \exists y(G(x) \wedge T(x, y))$

Ora, se dalle tre frasi iniziali é possibile dedurre la quarta, questo equivale a dire che l'insieme composto dalle quattro frasi, di cui l'ultima negata, é insoddisfacibile. Benchè tale procedura sia una procedura finita, é comunque di notevole impegno, poiché bisognerebbe individuare l'insieme risoluzione, l'insieme che cioé costituisce il limite superiore alla risoluzione per clausole, e verificare se al suo interno é contenuta la clausola vuota.

Intuitivamente, si può vedere che non è possibile dall'insieme dato ottenere la clausola vuota, e bisogna supporre ulteriormente l'esistenza di almeno un leone, cioé: $\exists x L(x)$

$\triangle$

A.3 Esercizi di deduzione naturale.

$\triangleright$ `Esercizio.`
Dimostrare che:

$$\{\neg C \rightarrow (\neg A \vee D), C \rightarrow (B \vee D), \neg D \rightarrow A\} \overset{\mathcal{DN}}{\longmapsto} B \vee D$$

$\triangleleft$

$\triangledown$ *Svolgimento.*
Ciò equivale a dimostrare che

$$\{\neg C \rightarrow (\neg A \vee D), C \rightarrow (B \vee D), \neg D \rightarrow A, \neg(B \vee D)\} \overset{\mathcal{DN}}{\longmapsto} \perp$$

frasi utilizzate	**frase dedotta**	**regola utilizzata**
$\neg B \wedge \neg D^*$	$\neg D, \neg B$	$(\wedge, e)$
$\neg D, \neg D \rightarrow A^*$	A	modus ponens
$A, \neg D$	$A \wedge \neg D$	$(\wedge, i)$
$A \wedge \neg D$	$\neg(\neg A \vee D)$	equiv. semantica
$\neg(\neg A \vee D), \neg C \rightarrow (\neg A \vee D)^*$	C	modus tollens
$C, C \rightarrow (B \vee D)^*$	$B \vee D$	modus ponens
$\neg B \wedge \neg D^*$	$\neg(B \vee D)$	equiv. semantica
$\neg(B \vee D), (B \vee D)$	$\bot$	$(\wedge, i)$

△

▷ `Esercizio.`
Dimostrare per deduzione naturale che l'insieme seguente é insoddisfacibile:

$$\{\forall x(M(x) \rightarrow B(x)), \exists x(A(x) \wedge M(x)), \neg\exists x(A(x) \wedge B(x))\}$$

◁

▽ *Svolgimento.*
Per note equivalenze semantiche, l'insieme puó essere riscritto come segue:

$$\{\forall x(M(x) \rightarrow B(x)), \exists x(A(x) \wedge M(x)), \forall x(\neg A(x) \vee \neg B(x))\}$$

Dimostrare che esso é insoddisfacibile equivale a dimostrare che:

$$\{\forall x(M(x) \rightarrow B(x)), \exists x(A(x) \wedge M(x)), \forall x(\neg A(x) \vee \neg B(x))\} \overset{\mathcal{DN}}{\longmapsto} \bot$$

La **prima operazione** da fare é quella di *eliminare il quantificatore esistenziale*. Secondo la regola condizionale di eliminazione, esso puó essere sostituito da una qualsiasi variabile che non compaia libera nelle altre due frasi; ad esempio sia questa la variabile y, per cui la seconda frase, per $(\exists, e)$ diventerà:

$$(A[y|x] \wedge M[y|x])$$

L'eliminazione dei quantificatori universali é più immediata, e produce le due frasi:

$$(M(x) \rightarrow B(x))[t|x], (\neg A(x) \vee \neg B(x))[s|x]$$

Ora, nulla vieta di porre $t = s = y$; infatti, se i quantificatori universali rendono valida una frase per ogni termine t o s, lo faranno tanto più per il generico elemento y del dominio del discorso che renda vera la frase quantificata dal quantificatore esistenziale.

Si ottiene in definitiva l'insieme di frasi:

$$\{(\neg M(y) \vee B(y)), (A(y) \wedge M(y)), (\neg A(y) \vee \neg B(y))\}\}$$

Si osservi che le tre frasi sono ora enunciati. Mediante le usuali regole di Deduzione Naturale, si ha successivamente:

frasi utilizzate	**frase dedotta**	**regola utilizzata**
$(A(y) \wedge M(y))^*$	$A(y), M(y)$	$(\wedge, e)$
$M(y), (\neg M(y) \vee B(y))^*$	$B(y)$	$(\vee, e)$
$B(y), (\neg A(y) \vee \neg B(y))^*$	$\neg A(y)$	$(\vee, e)$
$\neg A(y), A(y)$	$\perp$	$(\perp, i)$

$\triangle$

$\triangleright$ `Esercizio.`
Dimostrare che:

$$\{\neg C \rightarrow (\neg A \vee B), (\neg D \wedge \neg B) \rightarrow C, \neg D \rightarrow A\} \overset{\mathcal{DN}}{\longmapsto} \neg B \rightarrow D$$

$\triangleleft$

$\triangledown$ *Svolgimento.*
Per quanto noto dalla Deduzione Naturale, in due passaggi, equivale a dimostrare che

$$\{\neg C \rightarrow (\neg A \vee B), (\neg D \wedge \neg B) \rightarrow C, \neg D \rightarrow A, \neg B, \neg D\} \overset{\mathcal{DN}}{\longmapsto} \perp$$

frasi utilizzate	**frase dedotta**	**regola utilizzata**
$\neg B^*, \neg D^*$	$\neg B \wedge \neg D$	$(\wedge, i)$
$\neg B \wedge \neg D, (\neg B \wedge \neg D) \rightarrow \neg C^*$	$\neg C$	$(\rightarrow, e)$
$\neg C, \neg C \rightarrow (\neg A \vee B)^*$	$\neg A \vee B$	$(\rightarrow, e)$
$\neg B^*, \neg A \vee B$	$\neg A$	$(\vee, e)$
$\neg D, \neg D \rightarrow A^*$	A	$(\rightarrow, e)$
$\neg A, A$	$\neg A \wedge A$	$(\wedge, i)$
$A \wedge \neg A$	$\perp$	$(\perp, i)$

$\triangle$