

Logica Matematica

Gabriele Fioco

a.a. 2023-2024

Contents

1	Logica proposizionale	3
1.1	Definizione	3
1.2	Formule ben formate	3
2	Verità, conseguenze ed equivalenze tautologiche	4
2.1	Proposizioni tautologicamente vere, possibili e impossibili	4
2.2	Conseguenze tautologiche	4
2.3	Equivalenze tautologiche	4
3	Connettivi vero funzionali	4
3.1	Negazione	5
3.2	Congiunzione	5
3.3	Disgiunzione	5
3.4	Assurdo	5
3.5	Condizionali: implicazione e coimplicazione	5
3.5.1	Teorema di deduzione	6
3.5.2	Regole di implicazione e coimplicazione	6
4	Conseguenze ed equivalenze tautologiche notevoli	6
5	Completezza della logica proposizionale	6
5.1	Il calcolo F_T , notazione esistenza di una prova	6
5.2	Teorema di validità	7
5.3	Teorema di completezza	7
5.4	Teorema di validità e completezza per teorie	7
6	Logica del primo ordine	7
6.1	Definizione	7
6.2	Termini	8
6.3	Formule ben formate	8
6.4	Variabili libere e vincolate	8
6.5	Formule ben formata chiuse e aperte	9
6.6	Enunciati	9
6.7	Forma vero funzionale di un enunciato	9
6.8	Predicato identità	9

7	Quantificatori	10
7.1	Quantificatore esistenziale	10
7.2	Quantificatore universale	10
7.3	Forme aristoteliche	10
7.4	Quantificazione numerica	11
8	Strutture	11
8.1	Perché formalizzare le strutture	11
8.2	Definizione	11
8.3	Strutture isomorfe	12
8.4	Strutture elementarmente equivalenti	12
8.5	Teorema di compattezza	12
8.6	Teorema di Lowenheim-Skolem	12
9	Interpretare gli enunciati	12
9.1	Interpretazione dei termini chiusi	12
9.2	Interpretazione degli enunciati	13
10	Modelli e contromodelli	14
11	Verità, conseguenze ed equivalenze logiche in FO e in un contesto	14
11.1	Verità logiche in FO e in un contesto	14
11.2	Conseguenze logiche in FO e in un contesto	14
11.3	Equivalenze logiche in FO e in un contesto	14
11.4	Equisoddisfacibilità	14
11.5	Quadro complessivo: verità, conseguenze ed equivalenze in TAUT, FO e in un contesto	15
12	Conseguenze ed equivalenze logiche notevoli	15
12.1	Rimpiazzamento	15
12.2	De Morgan per i quantificatori e le formule aristoteliche	16
12.3	Spostare i quantificatori	16
12.4	Quantificazione vacua	17
12.5	Rinomina di variabili	17
13	Elementi utili ad arricchire un linguaggio	17
13.1	Tipizzazione	17
13.2	Assiomatizzazione	17
14	Induzione e ricorsione	18
14.1	Definizioni induttive	18
14.2	Principio di induzione	18
15	Aritmetica di Peano	18
15.1	Definizione induttiva dei naturali	18
15.2	Principio di induzione sui naturali	18
15.3	Definizione induttiva di funzioni sui naturali	18
15.4	Il modello standard e il linguaggio di Peano	19
15.5	Assiomi di Peano	19

15.6 Ordinare i naturali in PA	19
15.7 Principio di induzione forte	20
16 Deduzione automatica e forme normali	20
16.1 Forma normale prenessa	20
16.2 Forma normale universale (o di Skolem)	21
16.3 Forma normale congiuntiva	21
17 Completezza della logica del primo ordine	22
17.1 Calcolo F , notazione esistenza di una prova	22
17.2 Teorema di validità in FO	22
17.3 Teorema di completezza in FO	22
17.4 Teorie formalmente complete	22
18 Decidibilità della logica proposizionale e semidecidibilità della logica del primo ordine	22
18.1 Semidecidibilità della logica del primo ordine	22
18.2 Decidibilità della logica proposizionale	23
19 Metodi di prova	23
19.1 Prova per assurdo	23
19.2 Prova per casi	23

1 Logica proposizionale

1.1 Definizione

La logica proposizionale è un linguaggio formato da:

- lettere proposizionali;
- connettivi logici.

Combinati formano una proposizione (o enunciato). Un enunciato è una formula che può essere interpretata come vera o falsa. Una proposizione atomica è una proposizione che non può essere scissa in proposizioni più semplici.

Un'insieme finito o infinito Γ di enunciati è detto teoria.

Le proposizioni sono trattate in modo astratto e vanno trattati tutti i casi veri/falsi (es. tabella di verità).

1.2 Formule ben formate

Una formula ben formata (fbf, in inglese wff = well formatted formula) del linguaggio si definisce induttivamente come:

- Base: una qualsiasi proposizione atomica.
- Passo: si ottiene riempiendo con fbf i posti dei costrutti.

Il valore di verità di una fbf si ottiene sostituendo le atomiche con i rispettivi valori di verità. Applicando le tavole di verità dei costrutti e procedendo per strati si ottiene il valore finale

della fbf per ogni interpretazione booleana delle atomiche.

2 Verità, conseguenze ed equivalenze tautologiche

2.1 Proposizioni tautologicamente vere, possibili e impossibili

Sia P una proposizione:

- P è tautologicamente vera (o tautologica) se e solo se P è vera in tutte le interpretazioni booleane.
- P è tautologicamente possibile (o soddisfacibile) se e solo se esiste almeno un'interpretazione booleana in cui P è vera.
- P è tautologicamente impossibile (o insoddisfacibile) se e solo se P è falsa in tutte le interpretazioni booleane.

Si osserva che P è tautologicamente impossibile se e solo se P non è tautologicamente vera.

Proporzionalmente è sinonimo di tautologicamente.

2.2 Conseguenze tautologiche

Siano Q e P_1, \dots, P_n delle proposizioni.

Si dice che Q è conseguenza tautologica di P_1, \dots, P_n se e solo se Q è vera in ogni interpretazione booleana in cui P_1, \dots, P_n sono vere. Si denota con $P_1, \dots, P_n \models_T Q$

Si osserva che se P è tautologicamente vera allora $\models_T P$.

2.3 Equivalenze tautologiche

Siano P e Q delle proposizioni.

Esse si dicono tautologicamente equivalenti se e solo se hanno lo stesso valore di verità in tutte le interpretazioni booleane. Si denota con $P \Leftrightarrow_T Q$.

Se vale $P \Leftrightarrow_T Q$ allora posso effettuare il rimpiazzamento di P con Q in una formula F .

3 Connettivi vero funzionali

La logica proposizionale decide la verità degli enunciati composti attraverso i connettivi vero-funzionali. Il valore di verità (la semantica) di un enunciato viene determinato a partire dai valori di verità dei componenti mediante una tavola di verità.

I connettivi vero-funzionali e i relativi simboli sono:

- Negazione: \neg
- Congiunzione: \wedge
- Disgiunzione: \vee
- Implicazione: \rightarrow
- Coimplicazione: \leftrightarrow
- Assurdo: \perp

Ogni connettivo ha delle regole dette di introduzione e di eliminazione.

3.1 Negazione

La negazione in italiano assume il significato di “non P ”. Utilizza il simbolo \neg e ammette le regole:

- Introduzione: $\perp \models_T \neg P$
- Eliminazione: $\neg\neg P$

3.2 Congiunzione

La congiunzione in italiano assume il significato di “ P e Q ”. Utilizza il simbolo \wedge e ammette le regole:

- Introduzione: $P, Q \models_T P \wedge Q$
- Eliminazione: $P \wedge Q \models_T P, Q$

3.3 Disgiunzione

La disgiunzione in italiano assume il significato di “ P o Q ”. Utilizza il simbolo \vee e ammette le seguenti regole:

- Introduzione: $P \models_T P \vee Q$ e $Q \models_T Q \vee P$
- Eliminazione: $(P \vee R), (P \rightarrow Q), (R \rightarrow Q) \models_T Q$. Ovvero se P implica Q ed anche R implica Q , e sono vere P o R allora Q deve essere vera. Modella il ragionamento per casi.

3.4 Assurdo

Assurdo non è un connettivo ma una proprietà logicamente impossibile dunque falsa in ogni circostanza di un contesto. L’assurdo proposizionale è la contraddizione $P \wedge \neg P$ falsa in tutte le interpretazioni booleane.

Il simbolo logico di assurdo è \perp . Ammette le regole:

- Introduzione: $P, \neg P \models_T \perp$
- Eliminazione: $\perp \models_T P$. Dall’assurdo derivo qualsiasi cosa.

3.5 Condizionali: implicazione e coimplicazione

Un’asserzione del tipo “se P allora Q ” è detta condizionale. In logica proposizionale corrisponde al connettivo implica e si scrive $P \rightarrow Q$. La contrapposta è $\neg Q \rightarrow \neg P$.

Viceversa un’asserzione “ P se Q ” vuol dire $Q \rightarrow P$, “ Q solo se P ”.

Le due asserzioni unite formano la coimplicazione che si può leggere come “ P se e solo se Q ” e si denota col simbolo $P \leftrightarrow Q$.

Si nota che l’implicazione $P \rightarrow Q$ vuol dire “ Q è vera ogni qualvolta P è vera”. Dunque può accadere che Q sia vera e P falsa, ma non il viceversa. P è infatti una condizione necessaria ma non sufficiente purché Q sia vera; nella coimplicazione invece P è condizione necessario e anche sufficiente affinché Q sia vera e viceversa.

Le tabelle di verità corrispondenti sono le seguenti.

P	Q	“Se P allora Q ” $P \rightarrow Q$	“ P se Q ” $Q \rightarrow P$	“ P se e solo se Q ” $P \leftrightarrow Q$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

3.5.1 Teorema di deduzione

Il teorema di deduzione afferma che $(P_1, \dots, P_n \models Q) \leftrightarrow (\models (P_1 \wedge \dots \wedge P_n) \rightarrow Q)$

L'implicazione infatti in un certo senso cattura la conseguenza logica: dalle premesse P_1, \dots, P_n ottengo Q corrisponde a scrivere $(P_1 \wedge \dots \wedge P_n) \rightarrow Q$. E di fatti la regola di introduzione di implica richiamerà questo teorema.

Corollario: $P \leftrightarrow_C Q \leftrightarrow \vdash_C (P \leftrightarrow Q)$ dove C è un contesto (si vede nella seconda parte del corso).

3.5.2 Regole di implicazione e coimplicazione

Implicazione:

- Introduzione: $P \vdash_T Q \leftrightarrow P \rightarrow Q$
- Eliminazione: $P, P \rightarrow Q \vdash_T Q$

Coimplicazione:

- Introduzione: $P \rightarrow Q, Q \rightarrow P \vdash_T P \leftrightarrow Q$
- Eliminazione: $P, P \leftrightarrow Q \vdash_T Q$ e $Q, P \leftrightarrow Q \vdash_T P$.

4 Conseguenze ed equivalenze tautologiche notevoli

Si danno alcune conseguenze ed equivalenze tautologiche notevoli:

- Principio del terzo escluso: $\vdash_T P \vee \neg P$
- Contrapposizione: $\vdash_T (P \rightarrow Q) \leftrightarrow (\neg Q \rightarrow \neg P)$
- Modus Ponens: $P, P \rightarrow Q \vdash_T Q$
- Modus Tollens: $\neg Q, P \rightarrow Q \vdash_T \neg P$
- Modus Tollens Esteso: $\neg Q, P \leftrightarrow Q \vdash_T \neg P$ e $\neg Q, Q \leftrightarrow P \vdash_T \neg P$
- Principio di non contraddizione: $\vdash_T \neg(P \wedge \neg P)$

5 Completezza della logica proposizionale

5.1 Il calcolo F_T , notazione esistenza di una prova

Il calcolo logico F_T consiste nelle argomentazioni provabili usando tutte e solo le regole di introduzione ed eliminazione dei connettivi.

La notazione $P_1, \dots, P_n \vdash_T Q$ indica che esiste una prova in F_T per Q con premesse P_1, \dots, P_n .

5.2 Teorema di validità

Il teorema di validità afferma: se $P_1, \dots, P_n \vdash_T Q$ allora $P_1, \dots, P_n \models_T Q$

Ovvero se esiste una prova per Q con premesse P_1, \dots, P_n allora Q è conseguenza delle premesse P_1, \dots, P_n .

Contrapposta: se Q non è conseguenza logica delle premesse P_1, \dots, P_n allora non esiste una prova di Q con premesse P_1, \dots, P_n .

5.3 Teorema di completezza

Il teorema di completezza afferma: se $P_1, \dots, P_n \models_T Q$ allora $P_1, \dots, P_n \vdash_T Q$

Ovvero se Q è conseguenza delle premesse P_1, \dots, P_n allora esiste una prova per Q con premesse P_1, \dots, P_n .

Contrapposta: se non esiste una prova per Q con premesse P_1, \dots, P_n allora Q non è conseguenza logica di P_1, \dots, P_n .

5.4 Teorema di validità e completezza per teorie

Il teorema di validità e completezza vale anche per le teorie infinite Γ e afferma: $\Gamma \vdash_T Q \leftrightarrow \Gamma \models_T Q$.

Inoltre:

- $(\Gamma \cup \{P\} \models_T Q) \leftrightarrow (\Gamma \models_T P \rightarrow Q)$
- $(\Gamma \cup \{P\} \vdash_T Q) \leftrightarrow (\Gamma \vdash_T P \rightarrow Q)$

6 Logica del primo ordine

6.1 Definizione

La logica del primo ordine è un linguaggio che estende la logica proposizionale introducendo alcuni elementi:

- Variabili: insieme infinito di simboli scelti arbitrariamente.
- Costanti: denotano oggetti dell'universo.
- Quantificatori esistenziali e universali.
- Predicati: denotano relazioni tra oggetti. I predicati istanziati con costanti sono le proposizioni atomiche della logica proposizionale.
- Funzioni: denotano oggetti in maniera indiretta.

Si ricorda che gli altri elementi utilizzati ereditati dalla logica proposizionale sono:

- Lettere proposizionali.
- Connettivi logici.

Variabili, connettivi e quantificatori sono comuni a tutti i linguaggi; viceversa costanti, predicati e funzioni sono caratteristici di ogni linguaggio e lo caratterizzano. Dato un linguaggio del primo ordine L si denota con:

- $C(L) = \{c_1, \dots, c_h\}$ l'insieme delle h costanti di L .

- $P(L) = \{p_1/n, \dots, p_h/m\}$ l'insieme degli h predicati di L . Gli interi n e m indicano l'arietà del predicato p_i .
- $F(L) = \{f_1/n, \dots, f_h/m\}$ l'insieme delle h funzioni di L . Gli interi n e m indicano l'arietà della funzione f_i .

Ogni linguaggio del primo ordine è stratificato in due livelli:

- Termini: servono a denotare gli oggetti dell'universo. Sono costruiti da variabili, costanti e simboli di funzioni.
- Formule: denotano frasi che possono essere vere o false. Sono costruite a partire dai termini usano connettivi, predicati e funzioni.

6.2 Termini

L'insieme dei termini $T(L)$ di un linguaggio del primo ordine L si definisce induttivamente come:

- Ogni variabile x appartiene a $T(L)$.
- Ogni costante c appartiene a $T(L)$.
- Sia f un simbolo di funzione n -ario di L , siano t_1, t_2, \dots, t_n termini di L allora anche $f(t_1, t_2, \dots, t_n)$ appartiene a $T(L)$.
- Nient'altro appartiene a $T(L)$.

6.3 Formule ben formate

Dati un linguaggio del primo ordine L , $T(L)$ l'insieme dei suoi termini e $P(L)$ l'insieme dei suoi predicati, si definisce $FBF(L)$ l'insieme delle formule ben formate del linguaggio in modo induttivo:

- Siano $P \in P(L)$ e $t_1, t_2, \dots, t_n \in T(L)$ allora $P(t_1, t_2, \dots, t_n) \in FBF(L)$ ed è detta atomica.
- Siano $P, Q \in FBF(L)$ allora anche $(P \wedge Q), (P \vee Q), (P \rightarrow Q), (P \leftrightarrow Q), (\neg P), \perp \in FBF(L)$
- Siano $P \in FBF(L)$ e x una variabile allora $(\forall x P), (\exists x P) \in FBF(L)$
- Nient'altro è in $FBF(L)$

Si noti che una funzione in FOL non dà un valore di verità ma restituisce soltanto un altro elemento del discorso. Per tanto una formula del tipo $\neg f(a)$ non è ben formata.

6.4 Variabili libere e vincolate

Le variabili libere e vincolate si definiscono induttivamente come:

- Base: le occorrenze di variabili libere in $P(t_1, t_2, \dots, t_n)$ sono tutte le occorrenze di variabili in t_1, t_2, \dots, t_n . \perp non ha variabili libere.
- Passo:
 - Le occorrenze di variabili libere in $(P \wedge Q), (P \vee Q), (P \rightarrow Q), (P \leftrightarrow Q), (\neg P)$ sono tutte le occorrenze di variabili libere in P e Q .
 - Le occorrenze di variabili libere in $\forall v P$ sono tutte le occorrenze di variabili libere in P ad eccezione di v ; ogni occorrenza di v in P è una variabile vincolata in $\forall v P$

- Le occorrenze di variabili libere in $\exists v P$ sono tutte le occorrenze di variabili libere in P ad eccezione di v ; ogni occorrenza di v in P è una variabile vincolata in $\exists v P$

6.5 Formule ben formata chiuse e aperte

Una formula ben formata si dice:

- Chiusa se non contiene occorrenze di variabili libere.
- Aperta se contiene almeno un'occorrenza di variabile libera.

6.6 Enunciati

Una fbf è detta enunciato (o proposizione) se e solo se è chiusa. Infatti a una fbf chiusa è possibile attribuire un valore vero o falso in una data circostanza; a una fbf aperta non è possibile attribuire un valore di verità.

Inoltre sia L un linguaggio, si chiamano:

- L-enunciato: un enunciato in L .
- L-teoria: un insieme Γ di L-enunciati.

6.7 Forma vero funzionale di un enunciato

Sia L un linguaggio e $A \in FBF(L)$.

A si dice forma atomica generalizzata se e solo se non è in una delle seguenti forme: $A = \neg B, A = B \vee C, A = B \wedge C, A = B \rightarrow C, A = B \leftrightarrow C, A = \perp$. Dunque A è forma atomica generalizzata se è un'atomica $P(t_1, \dots, t_n)$ o è nella forma $A = \exists x B$ o $A = \forall x B$.

La forma vero funzionale di A , denotata con $fvf(A)$, si ottiene rimpiazzando ogni forma atomica generalizzata con una lettera proposizionale introdotta e associata a una e una sola forma atomica generalizzata. Sia Q una lettera proposizionale:

- $fvf(P(t_1, \dots, t_n)) = Q$
- $fvf(\forall x B) = Q$
- $fvf(\exists B) = Q$
- $fvf(\perp) = \perp$
- $fvf(A \vee B) = fvf(A) \vee fvf(B)$
- $fvf(A \wedge B) = fvf(A) \wedge fvf(B)$
- $fvf(A \rightarrow B) = fvf(A) \rightarrow fvf(B)$
- $fvf(A \leftrightarrow B) = fvf(A) \leftrightarrow fvf(B)$

6.8 Predicato identità

In FOL il predicato di identità $=$ è l'unico con un'interpretazione prefissata: $a = b$ in una circostanza è vero se e solo se in quella circostanza a e b sono nomi dello stesso oggetti. Ha le seguenti regole:

- Introduzione (Riflessiva dell'identità): $n = n$.
- Eliminazione (Identità degli indiscernibili): $P(n), n = m \vdash_{FO} P(m)$.

7 Quantificatori

La logica del primo ordine introduce due espressioni dette quantificatori:

- Quantificatore esistenziale \exists
- Quantificatore universale \forall

Anch'essi hanno le loro regole in maniera simile ai connettivi proposizionali.

7.1 Quantificatore esistenziale

Il quantificatore esistenziale in FOL può essere tradotto in linguaggio naturale come “esiste”, “almeno un”, “qualche”. Una proposizione $\exists x(P(x))$ è vera se e solo se esiste almeno un elemento che soddisfa P .

Ammette le regole:

- Introduzione (Generalizzazione esistenziale): $S(c) \models_{FO} \exists x S(x)$
- Eliminazione (Particolarizzazione esistenziale): $\exists x S(x), ([c]S(c) \models Q) \models_{FO} Q$

7.2 Quantificatore universale

Il quantificatore universale in FOL si traduce in linguaggio naturale come “per ogni”, “comunque io prenda”. Una proposizione $\forall x(P(x))$ è vera se e solo se preso un qualunque oggetto x questo soddisfa P .

Ammette le regole:

- Introduzione (Generalizzazione universale):
 - $([c] \models_{FO} P(c)) \models_{FO} \forall x P(x)$
 - $([c]P(c) \models_{FO} Q(c)) \models_{FO} \forall x(Q(x) \rightarrow P(x))$
- Eliminazione (Particolarizzazione universale): $\forall x S(x) \models_{FO} S(c)$

7.3 Forme aristoteliche

Le quattro formule aristoteliche permettono di costruire delle formule a partire da frasi del linguaggio naturale comuni. Sono le seguenti:

- Ogni P è Q :

$$\forall x(P(x) \rightarrow Q(x))$$
- Qualche P è Q :

$$\exists x(P(x) \wedge Q(x))$$
- Nessun P è Q :

$$\forall x(P(x) \rightarrow \neg Q(x))$$
- Qualche P non è Q :

$$\exists x(P(x) \wedge \neg Q(x))$$

7.4 Quantificazione numerica

In logica del primo ordine si possono esprimere proposizioni che parlano di quantità numeriche:

- Esiste uno e uno solo per cui vale P (notazione: $\exists!xP(x)$):

$$\exists x(P(x) \wedge \forall y(P(y) \rightarrow x = y))$$

- Esistono al minimo n per cui vale P (notazione: $\exists^{\leq n}xP(x)$):

$$\exists x_1, \dots, \exists x_n (P(x_1) \wedge \dots \wedge P(x_n) \wedge x_1 \neq x_2, \dots \wedge x_{n-1} \neq x_n)$$

- Esistono al massimo n per cui vale P (notazione: $\exists^{\geq n}xP(x)$):

$$\forall x_1, \dots, \forall x_n, \forall x_{n+1} ((P(x_1) \wedge \dots \wedge P(x_n) \wedge P(x_{n+1}) \rightarrow (x_1 = x_2 \vee \dots \vee x_n = x_{n+1}))$$

- Esistono esattamente n per cui vale P (notazione: $\exists^n xP(x)$):

$$\exists^{\leq n}xP(x) \wedge \exists^{\geq n}xP(x)$$

$$\Leftrightarrow_{FO}$$

$$\exists x_1, \dots, \exists x_n \forall x_{n+1} ((x_1 \neq x_2 \wedge \dots \wedge x_{n-1} \neq x_n \wedge P(x_{n+1})) \leftrightarrow (x_{n+1} = x_1 \vee \dots \vee x_{n+1} = x_n))$$

8 Strutture

8.1 Perché formalizzare le strutture

Si è fissata in modo formale la sintassi dei linguaggi del primo ordine, si vuole formalizzare la modalità con cui assegnare un valore di verità agli enunciati. Nella logica proposizionale basta assegnare un valore di verità alle atomiche, in logica del primo ordine questo non è possibile.

Al fine di assegnare un valore di verità a un enunciato di un linguaggio del primo ordine L bisogna:

- Formalizzare i mondi in cui interpretare ogni elemento di L come un concetto insiemistico sull'universo del discorso U . In questo modo si fa una “fotografia” del mondo stesso
- Amplificare L con un nuovo linguaggio L_U che assegna ad ogni oggetto del mondo un nome. Questo permette di far fronte a variabili libere e oggetti senza nome.

8.2 Definizione

Sia L un linguaggio del primo ordine dato da $C(L), F(L), P(L)$, si definisce L -struttura una coppia (U, I) dove:

- U è un insieme non vuoto detto universo del discorso.
- I è la funzione interpretazione definita come:
 - Per ogni costante $c \in C(L)$, $I(c) \in U$.
 - Per ogni simbolo di funzione n -ario $f \in F(L)$, $I(f) : U^n \rightarrow U$ ($I(f)$ è una funzione n -aria su U).

- Per ogni predicato n -ario $P \in P(L)$, $I(P) \subseteq U^n$ ($I(P)$ è un predicato n -ario su U).

Un contesto C è un insieme di L-strutture.

Una struttura si può vedere come una riga della tabella di verità.

8.3 Strutture isomorfe

Due strutture $S = (U, I)$ e $R = (V, J)$ sono isomorfe se e solo se esiste una funzione biettiva $\sigma : U \rightarrow V$ tale che:

- $\sigma(I(c)) = J(c) \quad \forall c \in C(L)$
- $\sigma((I(f))(a_1, \dots, a_n)) = (J(f))(\sigma(a_1), \dots, \sigma(a_n)) \quad \forall f \in F(L), \forall a_1, \dots, a_n \in U$
- $(a_1, \dots, a_n) \in I(P) \leftrightarrow (\sigma(a_1), \dots, \sigma(a_n)) \in J(P) \quad \forall P \in P(L), \forall a_1, \dots, a_n \in U$

Informalmente, due strutture sono isomorfe se e solo se rappresentano la stessa struttura.

8.4 Strutture elementarmente equivalenti

Sia L un linguaggio, due L-strutture $S = (U, I)$ e $R = (V, J)$ si dicono elementarmente equivalenti se e solo se non sono distinguibili tramite enunciati di L (ovvero se modellano gli stessi L-enunciati).

Due L-strutture isomorfe sono elementarmente equivalenti, ma non necessariamente viceversa.

8.5 Teorema di compattezza

Il teorema di compattezza enuncia: sia L un linguaggio, sia Γ una L-teoria, sia S una L-struttura, allora se ogni sottoinsieme di Γ è modellato da S , Γ è modellato da S .

Il teorema di compattezza trova applicazione:

- Nella deduzione automatica, data una teoria infinita Γ ci si chiede se $\Gamma \models A$. Posso ridurre il problema a stabilire $\Gamma' \vdash A$ con Γ' sottoinsieme finito di Γ .
- Per provare che esistono modelli non standard di PA elementarmente equivalenti a N con stessa cardinalità del modello standard (si veda aritmetica di Peano).

8.6 Teorema di Lowenheim-Skolem

Il teorema di Lowenheim-Skolem enuncia: se Γ ha almeno un modello infinito allora Γ ha modelli di ogni cardinalità infinita elementarmente equivalenti al primo.

9 Interpretare gli enunciati

9.1 Interpretazione dei termini chiusi

Dato un linguaggio del primo ordine L si vuole dare una semantica ai suoi enunciati. Al fine di dare un nome ad ogni oggetto del mondo, si espande L con L_U dato da $C(L_U) = C(L) \cup \{c_a : a \in U\}$, $F(L_U) = F(L)$, $P(L_U) = P(L)$.

Si definisce induttivamente $GT(L_U)$ l'insieme dei termini chiusi di L_U come:

- Ogni costante $c \in C(L_U)$ è in $GT(L_U)$.
- Sia $f \in F(L_U)$ n -aria, siano $t_1, \dots, t_n \in GT(L_U)$, allora $f(t_1, \dots, t_n) \in GT(L_U)$.
- Nient'altro è termine chiuso di L_U .

L'interpretazione $I(t)$ per ogni $t \in GT(L_U)$ è data induttivamente come:

- Per ogni $c \in C(L)$, $I(c)$ è già definita.
- Per ogni $a \in U$, $I(c_a) = a$.
- Sia $f \in F(L)$, siano $t_1, \dots, t_n \in GT(L_U)$ allora $I(f(t_1, \dots, t_n)) = (I(f)(I(t_1), \dots, I(t_n)))$

Questa definizione di $I(t)$ garantisce che per ogni termine $t \in GT(L_U)$ valga $I(t) \in U$.

9.2 Interpretazione degli enunciati

Sia $E(L_U) = \{A \in FBF(L_U) : \text{libere}(A) = \emptyset\}$ l'insieme delle fbf su L_U prive di variabili libere, ovvero gli enunciati di L_U . Si denota con $A[x : c]$ una fbf ottenuta sostituendo ad ogni occorrenza libera di x la costante c . Si osserva che se $\text{libere}(A) \subseteq \{x\}$ allora $A[x : c] \in E(L_U)$.

Siano $A, B, Q \in E(L_U)$ allora l'interpretazione $I(Q)$ è definita induttivamente come:

- Siano $P \in P(L)$ e $t_1, \dots, t_n \in GT(L_U)$ allora:
 - $I(P(t_1, \dots, t_n)) = T$ se $(I(t_1), \dots, I(t_n)) \in I(P)$
 - $I(P(t_1, \dots, t_n)) = F$ se $(I(t_1), \dots, I(t_n)) \notin I(P)$
- $I(\perp) = F$
- $I(\neg A) =$
 - T se $I(A) = F$
 - F se $I(A) = T$
- $I(A \wedge B) =$
 - T se $I(A) = T$ e $I(B) = T$
 - F altrimenti
- $I(A \vee B) =$
 - T se $I(A) = T$ o $I(B) = T$
 - F altrimenti
- $I(A \rightarrow B) =$
 - T se $I(A) = F$ o $I(B) = T$
 - F altrimenti
- $I(A \leftrightarrow B) =$
 - T se $I(A) = I(B)$
 - F altrimenti
- $I(\forall x A) =$
 - T se per ogni $a \in U$ vale $I(A[x : c_a]) = T$
 - F altrimenti
- $I(\exists x A) =$
 - T se per qualche $a \in U$ vale $I(A[x : c_a]) = T$
 - F altrimenti

Q è vero nella struttura S se e solo se $I(Q) = T$.

10 Modelli e contromodelli

Sia L un linguaggio e sia $S = (U, I)$ una L -struttura. Si danno le seguenti definizioni:

- S è modello di un L -enunciato A se e solo se A è vero in S , ovvero $I(A) = T$. Si denota con $S \models A$.
- S è modello di una L -teoria Γ se e solo se per ogni $A \in \Gamma$ allora A è vero in S , ovvero $I(A) = T \forall A \in \Gamma$. Si denota con $S \models \Gamma$.
- S è contromodello di una L -teoria Γ se e solo se esiste un'enunciato $A \in \Gamma$ falso in S .

11 Verità, conseguenze ed equivalenze logiche in FO e in un contesto

11.1 Verità logiche in FO e in un contesto

Sia L un linguaggio e sia A un L -enunciato, si dice che:

- A è logicamente vero in FO se e solo se $S \models A$ per ogni L -struttura $S = (U, I)$. Si denota con $\models_{FO} A$.
- A è logicamente vero in un contesto C se e solo se $S \models A$ per ogni L -struttura $S = (U, I)$ in C . Si denota con $\models_C A$.

Inoltre si dice che A è vero in una L -teoria Γ in FO se e solo se per ogni L -struttura $S = (U, I)$ vale $(S \models \Gamma) \rightarrow (S \models A)$. Si denota con $\Gamma \models_{FO} A$.

11.2 Conseguenze logiche in FO e in un contesto

Sia L un linguaggio, si dice che:

- Q è conseguenza logica in FO delle premesse P_1, \dots, P_n se e solo se Q è vera nella L -teoria $\{P_1, \dots, P_n\}$. Si denota con $P_1, \dots, P_n \models_{FO} Q$.
- Q è una conseguenza logica in un contesto C delle premesse P_1, \dots, P_n se e solo se Q è vera in ogni L -struttura che appartiene a C che modella $\{P_1, \dots, P_n\}$. Si denota con $P_1, \dots, P_n \models_C Q$.

11.3 Equivalenze logiche in FO e in un contesto

Sia L un linguaggio, si dice che:

- P e Q sono logicamente equivalenti in FO se e solo se P e Q coincidono in ogni interpretazione. Si denota con $P \Leftrightarrow_{FO} Q$.
- P e Q sono logicamente equivalenti in un contesto C se e solo se le interpretazioni di P e Q coincidono in ogni L -struttura che appartiene a C . Si denota con $P \Leftrightarrow_C Q$.

11.4 Equisoddisfacibilità

Due enunciati A e B sono detti equisoddisfacili se e solo se sono entrambi soddisfacibili o entrambi insoddisfacibili. Si nota che se A e B sono logicamente equivalenti allora sono equisoddisfacibili (ma non viceversa).

11.5 Quadro complessivo: verità, conseguenze ed equivalenze in TAUT, FO e in un contesto

Sono appena state introdotte le nozioni di verità, conseguenze ed equivalente logiche in FO e in un contesto. Nella prima parte del corso si sono date nozioni simili per le tautologie.

Si osserva che:

- Se Q è tautologicamente vero allora Q è logicamente vero in FO; se Q è logicamente vero in FO allora Q è logicamente vero in un qualsiasi contesto.
- Se Q è conseguenza tautologica delle P_1, \dots, P_n allora Q è conseguenza logica in FO di P_1, \dots, P_n ; se Q è conseguenza logica in FO di P_1, \dots, P_n allora Q è conseguenza logica di P_1, \dots, P_n in qualsiasi contesto.
- Se P e Q sono tautologicamente equivalenti allora P e Q sono logicamente equivalenti in FO; se P e Q sono logicamente equivalenti in FO allora sono logicamente equivalenti in qualsiasi contesto.

In pratica da tautologie (logica proposizionale) si va giù a cascata; viceversa se qualcosa non è logicamente vero/conseguenza logica/logicamente equivalente in un contesto, allora non lo è neanche ai livelli superiori su FO e in TAUT.

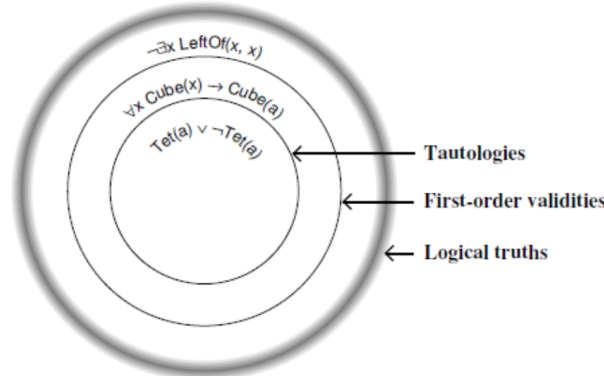


Figure 1: Quadro complessivo di verità, conseguenze ed equivalenze in TAUT, FO e in un contesto

12 Conseguenze ed equivalenze logiche notevoli

12.1 Rimpiazzamento

Data una proposizione P tautologicamente equivalente a Q ($P \Leftrightarrow_T Q$) allora posso rimpiazzare P e Q con enunciati del primo ordine.

In generale siano P_1, \dots, P_n, Q enunciati di un linguaggio L se vale $fvf(P_1), \dots, fvf(P_n) \models_T fvf(Q)$ allora vale $P_1, \dots, P_n \models_T Q$.

Ad esempio sappiamo che vale $P \rightarrow Q \Leftrightarrow_T \neg P \vee Q$ quindi siano $P = \forall x Cube(x), Q = \exists y Tet(y)$ ottengo $\forall x Cube(x) \rightarrow \exists y Tet(y) \Leftrightarrow_T \neg \forall x Cube(x) \vee \exists y Tet(y)$.

12.2 De Morgan per i quantificatori e le formule aristoteliche

Valgono le seguenti leggi di De Morgan per i quantificatori.

$$\neg \forall x P(x) \Leftrightarrow_{FO} \exists x \neg P(x)$$

$$\neg \exists x P(x) \Leftrightarrow_{FO} \forall x \neg P(x)$$

$$\forall x P(x) \Leftrightarrow_{FO} \neg \exists x \neg P(x)$$

$$\exists x P(x) \Leftrightarrow_{FO} \neg \forall x \neg P(x)$$

Applicando alle formule aristoteliche:

- Nessun P è Q :

$$\forall x (P(x) \rightarrow \neg Q(x)) \Leftrightarrow_{FO} \forall x (\neg P(x) \vee \neg Q(x)) \Leftrightarrow_{FO} \forall x \neg (P(x) \wedge Q(x)) \Leftrightarrow_{FO} \neg \exists x (P(x) \wedge Q(x))$$

- Qualche P non è Q :

$$\exists x (P(x) \wedge \neg Q(x)) \Leftrightarrow_{FO} \exists x \neg (P(x) \vee Q(x)) \Leftrightarrow_{FO} \exists x \neg (P(x) \rightarrow Q(x)) \Leftrightarrow_{FO} \neg \forall x (P(x) \rightarrow Q(x))$$

12.3 Spostare i quantificatori

Si possono spostare i quantificatori. Valgono i seguenti:

- Quantificatore universale con \wedge :

$$\forall x (P(x) \wedge Q(x)) \Leftrightarrow_{FO} \forall x P(x) \wedge \forall x Q(x)$$

- Quantificatore esistenziale con \vee :

$$\exists x (P(x) \vee Q(x)) \Leftrightarrow_{FO} \exists x P(x) \vee \exists x Q(x)$$

Inoltre:

- Doppia quantificazione universale:

$$\forall x \forall y P(x, y) \Leftrightarrow_{FO} \forall y \forall x P(x, y)$$

- Doppia quantificazione esistenziale:

$$\exists x \exists y P(x, y) \Leftrightarrow_{FO} \exists y \exists x P(x, y)$$

12.4 Quantificazione vacua

Se x non appare libera in P allora

$$\forall xP \Leftrightarrow_{FO} P \Leftrightarrow_{FO} \exists xP$$

.

Ne consegue:

- $\forall x\forall xP(x) \Leftrightarrow_{FO} \forall xP(x) \Leftrightarrow_{FO} \exists x\forall xP(x)$
- $\exists x\exists xP(x) \Leftrightarrow_{FO} \exists xP(x) \Leftrightarrow_{FO} \forall x\exists xP(x)$
- $\forall x(P \vee Q(x)) \Leftrightarrow_{FO} P \vee \forall xQ(x)$
- $\exists x(P \wedge Q(x)) \Leftrightarrow_{FO} P \wedge \exists xQ(x)$

12.5 Rinomina di variabili

Se y non appare in P allora:

- $\forall xP(x) \Leftrightarrow_{FO} \forall yP(y)$
- $\exists xP(x) \Leftrightarrow_{FO} \exists yP(y)$

13 Elementi utili ad arricchire un linguaggio

13.1 Tipizzazione

In alcuni casi nasce la necessità di tipizzare il linguaggio. Si supponga di voler parlare degli animali domestici e i loro proprietari, allora assumiamo un linguaggio L tale che $P(L)$ contenga *Appartiene/2* con $I(\text{Appartiene}) = \{a, b \in U : a \text{ appartiene a } b\}$. Sia poi $S = (U, I)$ una L-struttura con U insieme degli oggetti rappresentanti gli animali domestici e le persone.

Per dire “ogni animale domestico ha un proprietario” si potrebbe scrivere $\forall x\exists y(\text{Appartiene}(x, y))$. Ma questo non escluderebbe che un animale domestico possenga un umano in quanto x può essere sia animale che persona, e lo stesso per y .

Per tipizzare il predicato precedente si può estendere $P(L)$ con *Persona/1* e *Animale/1* tali che $I(\text{Persona}) = \{a \in U : a \text{ è una persona}\}$ e $I(\text{Animale}) = \{a \in U : a \text{ è un animale}\}$. Il precedente enunciato viene ora scritto $\forall x(\text{Animale}(x) \rightarrow \exists y(\text{Persona}(y) \wedge \text{Appartiene}(x, y)))$.

Al fine di rendere più succinta la scrittura si utilizza la notazione $\forall x : \text{Animale} \exists y : \text{Persona} \text{Appartiene}(x, y)$ che tipizza $C(L), F(L) e P(L)$.

Un linguaggio tipizzato è anche detto multisortato.

13.2 Assiomatizzazione

Assiomatizzare un insieme di circostanze vuol dire fornire una teoria Γ tale che tutti i modelli di Γ (ovvero le strutture tali che $S \models \Gamma$) siano tutte e sole le L-Strutture che descrivono astrattamente l'insieme desiderato di circostanze.

14 Induzione e ricorsione

14.1 Definizioni induttive

Dato un linguaggio L si possono definire i suoi elementi usando gli elementi stessi. Una definizione induttiva di un insieme S avviene definendo i seguenti:

- Base: qualche elemento b_1, \dots, b_k appartiene a S .
- Passo: se gli elementi e_1, \dots, e_n appartengono a S allora anche $p(e_1, \dots, e_n)$ appartiene a S .
- ...
- Passo: se gli elementi e_1, \dots, e_m appartengono a S allora anche $r(e_1, \dots, e_m)$ appartiene a S .
- Chiusura: nient'altro appartiene a S .

14.2 Principio di induzione

Dalla definizione induttiva degli elementi di L vale la seguente.

Sia $H(x)$ una fbf con $\text{libere}(H) = \{x\}$. Se valgono:

- Base: $H(b_1), \dots, H(b_k)$
- Passo: $\forall x_1, \dots, x_n ((H(x_1) \wedge \dots \wedge H(x_n) \rightarrow (H(p(x_1, \dots, x_n))))$
- ...
- Passo: $\forall x_1, \dots, x_m ((H(x_1) \wedge \dots \wedge H(x_m) \rightarrow (H(r(x_1, \dots, x_m))))$

Allora vale $\forall x H(x)$

15 Aritmetica di Peano

15.1 Definizione induttiva dei naturali

Un esempio di utilizzo dell'induzione è sui numeri naturali, definiti come:

- Base: $0 \in \mathbb{N}$
- Passo: $n \in \mathbb{N} \rightarrow n + 1 \in \mathbb{N}$
- Chiusura: nient'altro appartiene a \mathbb{N} .

15.2 Principio di induzione sui naturali

Sia $P(n)$ una fbf tale che $\text{libere}(P) = \{n\}$. Se valgono:

- Base: $P(0)$
- Passo: $\forall n (P(n) \rightarrow P(n + 1))$

Allora vale $\forall n P(n)$.

15.3 Definizione induttiva di funzioni sui naturali

Sia $f : \mathbb{N}^n \rightarrow \mathbb{N}$ una funzione, essa si può definire per induzione:

- Base: $\forall y_1, \dots, \forall y_{n-1} f(0, y_1, \dots, y_{n-1}) = g(y_1, \dots, y_{n-1})$

- Passo: $\forall y_1, \dots, \forall y_{n-1} \forall n f(n+1, y_1, \dots, y_{n-1}) = h(n, y_1, \dots, y_{n-1}, f(n, y_1, \dots, y_{n-1}))$

Con $g : \mathbb{N}^{n-1} \rightarrow \mathbb{N}$ e $h : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ funzioni ausiliarie.

15.4 Il modello standard e il linguaggio di Peano

Si vuole ora assiomatizzare una struttura dei naturali in primo ordine. Il modello creato dovrebbe essere un modello unico (o modello standard), ma si vedrà che ciò non è possibile in quanto in primo ordine vi sono modelli non standard ineliminabili.

Gli elementi del linguaggio del primo ordine $L(PA)$ dell'Aritmetica di Peano sono:

- $C(L(PA)) = \{0\}$
- $F(L(PA)) = \{s/1 + /2, \times /2\}$ (s è la funzione successore)
- $P(L(PA)) = \{= /2\}$

I termini $0, s(0), s(s(0)), \dots$ sono detti numerali.

Definiamo ora la struttura $N = (\mathbb{N}, I)$ dove:

- \mathbb{N} è l'insieme dei numeri naturali
- $I(0) = 0$
- $I(s(\dots(s(0))\dots)) = n$ (n volte s è uguale a n)

N ha cardinalità infinita (U contiene infiniti elementi) dunque per il teorema di Lowenheim-Skolem esistono altri modelli elementarmente equivalenti.

15.5 Assiomi di Peano

Si danno i seguenti assiomi:

- PA1: $\forall x \neg(s(x) = 0)$ (0 non è il successore di nessun naturale)
- PA2: $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$ (s è una funzione iniettiva)
- PA3: $\forall x (x + 0 = x)$
- PA4: $\forall x \forall y (x + s(y) = s(x + y))$
- PA5: $\forall x (x \times 0 = 0)$
- PA6: $\forall x \forall y (x \times s(y) = (x \times y) + x)$
- PA7: $(\forall y_1 \dots \forall y_n ((P(0, y_1, \dots, y_n) \wedge \forall x P(x, y_1, \dots, y_n)) \rightarrow \forall x (P(s(x), y_1, \dots, y_n))) \rightarrow \forall x P(x, y_1, \dots, y_n))$

PA1 e PA2 assicurano che nel linguaggio di peano i numerali formino una sequenza infinita di elementi distinti.

PA3-6 implementano somma e prodotto.

PA7 implementa il principio di induzione, non generalizzato (induzione di Peano): $(P(0) \wedge \forall x (P(x) \rightarrow P(s(x))) \rightarrow \forall x P(x)$

15.6 Ordinare i naturali in PA

Nel modello standard N si può aggiungere l'ordinamento stretto $< /2$ definito esplicitamente come: $\forall x \forall y (x < y \leftrightarrow \exists z (x + s(z) = y))$. PA prova che $<$ è una relazione di ordine, infatti valgono:

- Anti-riflessività: $\vdash_{PA} \forall x \neg(x < x)$
- Transitività: $\vdash_{PA} \forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow (x < z))$
- Tricotomia: $\forall x \forall y (x < y \vee x = y \vee y < x)$

15.7 Principio di induzione forte

Un altro principio di induzione molto utile in PA è il principio di induzione forte. Sia $P(n)$ una fbf tale che $\text{libere}(P) = \{n\}$ allora se per ogni n vale che se $P(k)$ vale per ogni $k < n$ allora vale $P(n)$. Formalmente: $\forall n ((\forall k (k < n \rightarrow P(k))) \rightarrow \forall n P(n))$

16 Deduzione automatica e forme normali

Data una teoria Γ e un enunciato A ci si chiede se A consegua da Γ e se determinarlo si vuole usare un algoritmo.

Vale il teorema: $A \models \Gamma$ se e solo se $\Gamma \cup \{\neg A\}$ è insoddisfacibile.

Un algoritmo di tipo refutazionale lavora su questo teorema, riducendo il problema di decidere se $A \models \Gamma$ ad un problema di insoddisfacibilità. L'algoritmo segue una serie di passi di preprocessing su ogni enunciato di $\Gamma \cup \{\neg A\}$:

1. Trasforma gli enunciati in forma normale prenessa
2. Trasforma gli enunciati in forma normale prenessa in forma normale di Skol
3. Trasforma gli enunciati in forma normale di Skol in forma normale congiuntiva.

Le trasformazioni vengono effettuate da algoritmi efficienti, ovvero che lavorano in tempo polinomiale sulla lunghezza dell'input.

16.1 Forma normale prenessa

Il primo passo di preprocessing trasforma ogni enunciato in forma normale prenessa.

Un enunciato si dice in forma normale prenessa se e solo se ha la forma $Q_1 x_1 \dots Q_n x_n B$ con Q quantificatore (\forall o \exists) e B fbf tale che sia priva di quantificatori e $\text{libere}(B) = \{x_1, \dots, x_n\}$.

Vale il teorema: per ogni L-enunciato esiste un L-enunciato logicamente equivalente in forma normale prenessa.

L'algoritmo trasforma un enunciato in uno logicamente equivalente in forma normale prenessa applicando ripetutamente le equivalenze logiche notevoli:

- $\neg \forall x P \Leftrightarrow \exists x \neg P$
- $\neg \exists x P \Leftrightarrow \forall x \neg P$
- $\exists x P \vee Q \Leftrightarrow \exists x (P \vee Q)$
- $\forall x P \wedge Q \Leftrightarrow \forall x (P \wedge Q)$
- $P \rightarrow \exists x Q \Leftrightarrow \exists y (P[x : y] \rightarrow Q)$
- $P \rightarrow \forall x Q \Leftrightarrow \forall y (P[x : y] \rightarrow Q)$

Se la variabile vincolata x occorresse libera in Q bisogna rinominarla con y :

- $\exists x P \wedge Q \Leftrightarrow \exists y (P[x : y] \wedge Q)$
- $\exists x P \rightarrow Q \Leftrightarrow \forall y (P[x : y] \rightarrow Q)$
- $\forall x P \vee Q \Leftrightarrow \forall y (P[x : y] \vee Q)$

- $\forall x P \rightarrow Q \Leftrightarrow \exists y (P[x : y] \rightarrow Q)$

16.2 Forma normale universale (o di Skolem)

Dopo aver trasformato un enunciato nella sua forma normale prenessa si trasforma quest'ultima in una forma normale di Skolem.

Un enunciato è detto in forma normale di Skolem quando ha la forma $\forall x_1 \dots \forall x_n C$ dove C è una fbf tale che sia priva di quantificatori e $\text{libere}(C) = \{x_1, \dots, x_n\}$.

Vale il teorema: per ogni enunciato esiste un enunciato equisoddisfacibile in forma normale di Skolem.

Un algoritmo trasforma un enunciato in forma normale prenessa in uno equisoddisfacibile in forma normale di Skolem applicando ripetutamente la sostituzione $\forall x_1 \dots \forall x_n \exists y D$ diventa $\forall x_1 \dots \forall x_n D[y : f(x_1, \dots, x_n)]$ con f nuovo simbolo di funzione con arità n .

16.3 Forma normale congiuntiva

L'ultimo passo di preprocessing trasforma un enunciato in forma normale di Skolem in forma normale congiuntiva. Si danno le definizioni:

- Un enunciato forma normale universale $\forall x_1 \dots \forall x_n C$ si dice in forma normale congiuntiva se e solo se $f v f(C)$ è in forma normale congiuntiva proposizionale.
- Una proposizione è in forma normale congiuntiva (CNF) se e solo se ha la forma $C_1 \wedge \dots \wedge C_n$ dove ogni C_i ha la forma $L_{i1} \vee \dots \vee L_{ik_i}$ con L_{ij} lettera proposizionale o sua negata.
- Un enunciato è in forma normale disgiuntiva (DNF) se e solo se ha la forma $C_1 \vee \dots \vee C_n$ dove ogni C_i ha la forma $L_{i1} \wedge \dots \wedge L_{ik_i}$ con L_{ij} lettera proposizionale o sua negata.

Vale il teorema: per ogni formula proposizionale esistono una forma normale congiuntiva e una forma normale disgiuntiva logicamente equivalenti.

Per ottenere una CNF o una DNF equivalente a una formula in input si applicano ripetutamente le conseguenze logiche notevoli:

- $\neg \neg C \Leftrightarrow C$
- $A \rightarrow B \Leftrightarrow \neg A \vee B$
- $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$
- $\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$
- $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$
- $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
- Associatività e commutatività di \wedge e \vee

L'utilizzo della distributività causa una dilatazione esponenziale degli algoritmi ma vale il teorema: esiste un algoritmo efficiente che data in input una formula proposizionale produce una CNF equisoddisfacibile. (Non si conoscono algoritmi efficienti per produrre DNF equisoddisfacibili).

17 Completezza della logica del primo ordine

17.1 Calcolo F , notazione esistenza di una prova

Il calcolo logico F indica tutte le argomentazioni che si possono dimostrare usando tutte e solo le regole di connettivi, quantificatori e identità.

La notazione $P_1, \dots, P_n \vdash_{FO} Q$ indica che esiste una prova in F per Q con premesse P_1, \dots, P_n .

17.2 Teorema di validità in FO

Il teorema di validità in FO afferma: se $P_1, \dots, P_n \vdash_{FO} Q$ allora $P_1, \dots, P_n \models_{FO} Q$.

Ovvero se esiste una prova per Q con premesse P_1, \dots, P_n allora Q è conseguenza logica delle premesse P_1, \dots, P_n .

17.3 Teorema di completezza in FO

Il teorema di completezza in FO afferma: se $P_1, \dots, P_n \models_{FO} Q$ allora $P_1, \dots, P_n \vdash_{FO} Q$.

Ovvero se Q è conseguenza logica delle premesse P_1, \dots, P_n allora esiste una prova per Q con premesse P_1, \dots, P_n .

17.4 Teorie formalmente complete

Una teoria Γ si dice formalmente completa se e solo se per ogni enunciato $Q \in \Gamma$ $\Gamma \vdash_{FO} Q$ o $\Gamma \vdash_{FO} \neg Q$. Equivalentemente per il teorema di validità e completezza $\Gamma \models_{FO} Q$ o $\Gamma \models_{FO} \neg Q$.

Dunque una teoria Γ non è formalmente completa se qualche enunciato $Q \in \Gamma$ è tale che fra le strutture che modellano Γ

L'insieme di tutti gli enunciati veri in una struttura è una teoria formalmente completa.

18 Decidibilità della logica proposizionale e semidecidibilità della logica del primo ordine

18.1 Semidecidibilità della logica del primo ordine

Per il teorema di validità e completezza se Q è una verità logica ($\models_{FO} Q$) allora esiste una prova per Q ($\vdash_{FO} Q$). Elencando le prove una a una prima o poi trovo quella per $\models_{FO} Q$.

Il teorema di Church sancisce che invece non esiste un algoritmo per elencare gli enunciati Q che non sono verità logiche. Se questo esistesse potrei decidere per ogni enunciato Q se questo è una verità logica o meno.

Pertanto esiste un algoritmo che su input Q : se questi è una verità logica termina; se questi non è una verità logica non da risposta.

18.2 Decidibilità della logica proposizionale

Si nota che la logica proposizionale è invece banalmente decidibile. Dato un input $\Gamma \models_T Q$ con Γ teoria finita, basta costruire la tabella di verità per deciderne la verità in un tempo finito. L'unico problema è che ad oggi non si conoscono algoritmi efficienti (polinomiali) per ogni input $\Gamma \models_T Q$.

19 Metodi di prova

19.1 Prova per assurdo

Le regole di negazione e di assurdo permettono di effettuare la prova per assurdo.

Si supponga di voler dimostrare Q . Allora posso assumere $\neg Q$ ed arrivare a Q . Successivamente per introduzione di assurdo ottengo l'assurdo. Infine per introduzione di negazione arrivo a $\neg\neg Q \models_T Q$.

19.2 Prova per casi

L'eliminazione della disgiunzione permette di effettuare una prova per casi. Ad esempio si supponga la seguente situazione.

P = "Ugo deve andare in farmacia" oppure Q = "Ugo deve andare al supermercato". Sia la farmacia che il supermercato sono in via Roma quindi $P \rightarrow$ "Ugo deve andare in via Roma" e anche $Q \rightarrow$ "Ugo deve andare in via Roma". Segue quindi che "Ugo deve andare in via Roma".