Appunti di Elementi di Logica Matematica

del 02/03/2009

http://www.sugata.eu
sid@camminaresuimonti.org

1. Logica proposizionale

1.1. Lettere proposizionali ($\mathcal{P}$)

Le lettere proposizionali sono le proposizioni atomiche ($p, q, \ldots$).

Il loro insieme viene chiamato $\mathcal{P}$.

1.2. Connettivi proposizionali

I connettivi proposizionali legano proposizioni atomiche o formule fra loro, dando vita a nuove formule e sono: $\overline{}$, $\vee$, $\wedge$, $\rightarrow$. I connettivi hanno la seguente precedenza: $\overline{}$, $\vee$ e $\wedge$, $\rightarrow$.

1.3. Letterali

Un letterale è una lettera proposizionale oppure la negazione di una lettera proposizionale (i.e. $p, \bar{q}, \ldots$).

1.4. Formule

Siano F e G due formule. Le formule sono:

- gli elementi di $\mathcal{P}$, ovvero le lettere proposizionali
- le negazioni: $\bar{F}$
- le congiunzioni: $F \wedge G$
- le disgiunzioni: $F \vee G$
- le implicazioni: $F \rightarrow G$

1.5. Sottoformule

Sia F una formula. G è sottoformula di F se sottostringa, ovvero se, applicate le parentesi sottointese in base alle regole di precedenza fra connettivi G rimane tale.

G si dice sottoformula propria di F se $G \neq F$.

1.6. Grado di una formula

Il grado di una formula F, $\mathrm{g}(F)$, è così definito:

i. $\mathrm{g}(F) = 0$, se F è una lettera proposizionale.
ii. $\mathrm{g}(\bar{F}) = 1 + \mathrm{g}(F)$.
iii. $\mathrm{g}(F \vee G) = \mathrm{g}(F \wedge G) = \mathrm{g}(F \rightarrow G) = 1 + \mathrm{g}(F) + \mathrm{g}(G)$.

Si noti che il grado di una formula corrisponde al numero di connettivi presenti nella formula.

1.7. Valutazione e interpretazione

La funzione v valuta le lettere proposizionali, nel senso che attribuisce ad una lettera proposizionale un valore $\mathbb{V}$ o $\mathbb{F}$.

La funzione $\bar{v}$ interpreta le formule ricorsivamente, usando v per le lettere proposizionali. Per le formule che contengono connettivi proposizionali viene usata la seguente tabella di verità:

p	q	$p \vee q$	$p \wedge q$	$p \to q$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$

Per semplificare la notazione, con v si intenderà sia la funzione v che $\bar{v}$.

1.8. Validità e soddisfacibilità

F è *valida* se è sempre vera.

F è *soddisfacibile* se esiste una combinazione di proposizioni (es. $p = \mathbb{V}, q = \mathbb{F}$) per cui F è vera.

F è *insoddisfacibile* se <u>non</u> esiste quanto sopra.

1.9. Equivalenza logica

Due formule F e G sono logicamente equivalenti ($F \equiv G$) se

$$\forall\, v \text{ interpretazione,} \qquad v(F) = v(G)$$

1.10. Conseguenza logica

Siano F e G formule.

G è conseguenza logica di F ($F \vDash G$) se

$$\forall\, v \text{ interpretazione,} \qquad v(F) \Longrightarrow v(G)$$

Si noti quindi che $\vDash$ è transitiva e riflessiva.

Con $\vDash F$ si intende $\emptyset \vDash F$. Si noti che qualunque interpretazione soddisfa $\emptyset$ e che quindi $\emptyset \vDash F \Leftrightarrow F$ valida.

Si noti che

$$F \text{ valida} \Leftrightarrow \bar{F} \text{ insoddisfacibile}$$
$$\bar{F} \text{ valida} \Leftrightarrow F \text{ insoddisfacibile}$$

Si noti che

$$F \vDash G \Leftrightarrow F \to G \text{ valida}$$
$$F \nvDash G \Leftrightarrow F \wedge \bar{G} \text{ soddisfacibile}$$

Si quindi che $F \equiv G \Leftrightarrow F \vDash G$ e $G \vDash F$, ovvero $F \equiv G \Leftrightarrow F \to G$ è valida e $G \to F$ è valida.

1.11. Insieme di formule

Sia $\Gamma = \{F_1, \dots, F_n | \forall\ i = (1, \dots, n), F_i \text{ formula}\}$ un insieme di formule.

Con Γ, F indichiamo $\Gamma \cup F$.

Un insieme di formule gode delle seguenti proprietà:

- $\Gamma \text{ valido} \Leftrightarrow \forall\ F \in \Gamma, F \text{ valido}$
- $\Gamma \text{ soddisfacibile} \Leftrightarrow \exists v \text{ interpretazione}, \forall F \in \Gamma, v(F) = \mathbb{V}$
- $\Gamma \text{ insoddisfacibile} \Leftrightarrow \forall v \text{ interpretazione}, \exists F \in \Gamma, v(F) = \mathbb{F}$

Si noti che se $\Gamma = (F_1, \dots, F_n)$,

Γ è valido / soddisfacibile / insoddisfacibile $\Leftrightarrow$ $(F_1 \land \dots \land F_n)$ è valido / soddisfacibile / insoddisfacibile.

Si noti che essendo $\Gamma, F = \Gamma \cup F$, vale

$\Gamma, F \text{ valido / soddisfacibile / insoddisfacibile} \Leftrightarrow \Gamma \cup F \text{ valido / soddisfacibile / insoddisfacibile.}$

1.12. Traduzione al linguaggio naturale

La traduzione al linguaggio naturale è abbastanza ovvia, ciononostante notiamo che se

$$p = x \text{ è primo}$$
$$d = x \text{ è dispari}$$

allora

"una condizione <u>sufficiente</u> perché x sia primo è che x sia dispari" $\equiv d \to p$

"una condizione <u>necessaria</u> perché x sia primo è che x sia dispari" $\equiv p \to q$

"una condizione necessaria e sufficiente perché x sia primo è che x sia dispari" $\equiv (d \to p) \land (p \to d)$

1.13. Forma normale congiuntiva e disgiuntiva

Con la notazione $\langle [G_1, \dots, G_n], \dots, [H_1, \dots, H_m] \rangle$, dove ogni G_i e H_i è un letterale, si intende che:

- Ogni elemento fra parentesi quadre è legato da $\lor$.
- Ogni elemento fra parentesi $\langle\ \ \rangle$ è legato da $\land$.

$\langle [G_1, \dots, G_n], \dots, [H_1, \dots, H_m] \rangle$ è una forma normale congiuntiva (i.e. $F_1 \land \dots \land \underbrace{(G_1 \lor \dots \lor G_m)}_{F_i} \land \dots \land F_n$).

$[\langle G_1, \dots, G_n \rangle, \dots, \langle H_1, \dots, H_m \rangle]$ è una forma normale disgiuntiva (i.e. $F_1 \lor \dots \lor \underbrace{(G_1 \land \dots \land G_m)}_{F_i} \lor \dots \lor F_n$).

Si noti che $\lor$ e $\land$ sono associativi e commutativi, quindi è indifferente l'ordine in cui vengono riportati.

Si noti che $\bar{\bar{p}}$ non è un letterale.

1.14. α e β formule

$\boldsymbol{\alpha}$-formule ($\wedge$)		
Formula	Ridotti	
$\boldsymbol{F \wedge G}$	F	G
$\overline{\boldsymbol{F \vee G}}$	$\bar{F}$	$\bar{G}$
$\overline{\boldsymbol{F \to G}}$	F	$\bar{G}$

$\boldsymbol{\beta}$-formule ($\vee$)		
Formula	Ridotti	
$\boldsymbol{F \vee G}$	F	G
$\overline{\boldsymbol{F \wedge G}}$	$\bar{F}$	$\bar{G}$
$\boldsymbol{F \to G}$	$\bar{F}$	G

Ogni α-formula è logicamente equivalente ($\equiv$) alla congiunzione dei suoi ridotti.

Ogni β-formula è logicamente equivalente ($\equiv$) alla disgiunzione dei suoi ridotti.

Una formula è di uno solo dei seguenti tipi: α-formula, β-formula, letterale, doppia negazione.

Si noti che:

1. $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$
2. $(G \vee H) \wedge F \equiv (G \wedge F) \vee (H \wedge F)$
3. $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
4. $(G \wedge H) \vee F \equiv (G \vee F) \wedge (H \vee F)$

1.15. Rango di una formula

Il rango di una formula F è calcolato ricorsivamente come segue:

i. Se F è un letterale, $\operatorname{rg} F = 1$.

ii. Se F è una doppia negazione e, sia F' il suo ridotto, $\operatorname{rg} F = 1 + \operatorname{rg} F'$.

iii. Se F è una α o β-formula, siano F' e F'' i suoi ridotti, $\operatorname{rg} F = 1 + \operatorname{rg} F' + \operatorname{rg} F''$.

1.16. Algoritmo di Fitting (Verificare!)

Ogni formula proposizionale può essere trasformata in forma normale congiuntiva e in forma normale disgiuntiva tramite l'algoritmo di Fitting.

L'algoritmo di Fitting per la <u>trasformazione in forma normale congiuntiva</u> è così definito:

i. Le doppie negazioni si sostituiscono con il loro ridotto.

ii. Con le α-formule, si sostituisce il congiunto con due nuovi congiunti contenenti ognuno un ridotto (i.e. $\langle [A \wedge B, C] \rangle \Longrightarrow \langle [A', C], [B', C] \rangle$).

iii. Con le β-formule, si sostituisce il congiunto con i ridotti (nel congiunto) (i.e. $\langle [A \vee B] \rangle \Longrightarrow \langle [A, B] \rangle$).

L'algoritmo di Fitting per la <u>trasformazione in forma normale disgiuntiva</u> è definito come per la forma normale congiuntiva ad eccezione del fatto il comportamento in caso di α e β-formule è invertito.

Si noti che l'algoritmo non è deterministico ed ha terminazione forte.

Si noti inoltre che gli algoritmi di Fitting non prevedono la sostituzione di una formula con un'altra logicamente equivalente ad essa.

E' dimostrabile che l'algoritmo di Fitting termina al più in $2^{\mathrm{rg}\,F}$ passi, tuttavia spesso l'algoritmo termina molto prima.

1.17. Tableaux

L'algoritmo dei tableaux determina la soddisfacibilità delle formule proposizionali, non è deterministico ed ha terminazione forte.

Sia una coppia complementare una coppia $\{p, \bar{p}\}$ dove p è un letterale.
Al termine dell'algoritmo vale la seguente proprietà:
Un insieme di letterali è soddisfacibile ⟺ non contiene nessuna coppia complementare.

L'algoritmo è così definito: sia F la radice di un albero binario con F formula, scelgo una sua sottoformula G che non sia un letterale.

i. Se G è una doppia negazione, sostituisco G con il suo ridotto nel nodo figlio.
ii. Se G è una α-formula, sostituisco G con i suoi ridotti nel nodo figlio.
iii. Se G è una β-formula, creo due nodi figli. Nel primo sostituisco G con uno dei suoi ridotti, nel secondo con l'altro.

Un tableau si dice chiuso se F è insoddisfacibile (tutti i nodi sono chiusi), ovvero è senza soluzioni.
Un tableau si dice aperto se F è soddisfacibile (o valida) (almeno un ramo è aperto), ovvero c'è almeno una soluzione.
Un ramo si dice aperto se è un ramo che ha almeno una foglia soddisfacibile.

L'altezza massima dell'albero binario è $\leq \mathrm{rg}\,F + 1$.

Si noti che anche l'algoritmo di costruzione dei tableaux non prevede la sostituzione di una formula con un'altra logicamente equivalente ad essa.

E' dimostrabile che se un nodo contiene una coppia complementare, il ramo non è soddisfacibile, pertanto durante lo svolgimento di un tableau potrò fermarmi subito.

Per dimostrare la validità di una formula F è possibile dimostrare l'insoddisfacibilità di $\bar{F}$.

Per stabilire se un insieme di formule $\Gamma = \{F_1, \dots, F_n\}$ è soddisfacibile si costruisce un tableau la cui radice contiene $F_1, \dots, F_n$.

Per stabilire se $F_1, \dots, F_n \vDash G$ si costruisce un tableau la cui radice contiene $F_1, \dots, F_n, \bar{G}$.
Se il tableau risulta chiuso la proposizione è vera, altrimenti è falsa.

1.18. Insieme di Hintikka

Un insieme Γ si dice insieme di Hintikka se:

i. Γ non contiene una coppia complementare di letterali.
ii. Γ contiene il ridotto delle doppie negazioni.
iii. Γ contiene i ridotti delle α-formule.
iv. Γ contiene almeno un ridotto delle β-formule.

Si noti che ogni insieme di Hintikka è soddisfacibile.

Si noti che se r è un ramo aperto di un tableau, l'unione dei nodi di r è un insieme di Hintikka.

2. Logica predicativa

2.1. Introduzione

La logica predicativa lavora con oggetti e per questo introduce i termini.

I termini, al contrario dei letterali della logica proposizionale, non sono mai veri o falsi, le formule invece possono essere valutate.

2.2. Notazione

Ove non indicato, con I si intende una interpretazione di $\mathcal{L}$, con σ si intende uno stato di I.

2.3. Linguaggio predicativo

Il linguaggio predicativo $\mathcal{L}$ contiene i seguenti elementi:

i. Le variabili $(x, y, z, \ldots)$.
ii. I simboli logici $(\underbrace{\vee, \wedge, \rightarrow, \overline{}}_{\text{connettivi}}, \underbrace{\forall, \exists}_{\text{quantificatori}})$.

   I simboli logici hanno precedenza come segue: $(\overline{}\ \text{e}\ \forall\ \text{e}\ \exists), (\vee\ \text{e}\ \wedge), (\rightarrow)$.

Il linguaggio predicativo $\mathcal{L}$ contiene i seguenti insiemi:

i. L'insieme dei simboli di costante (anche nessuno) $[0, 1, \pi, \langle \ldots \rangle, \langle n \rangle]$.
ii. L'insieme dei simboli di funzione (anche nessuno) (ognuno con la sua arietà $n \geq 1$) $[+, \cdot, \ ^n, \{\ \}, \cup, \cap]$.
iii. L'insieme dei simboli di relazione (almeno uno) (ognuno con la sua arietà $n \geq 1$) $[<, >, =, \subset]$.

L'arietà di un simbolo indica a quanti elementi esso si può applicare.

Se un simbolo ha arietà n, esso si dice n-ario.

2.4. Termine

Un termine è una variabile, un simbolo di costante o un simbolo di funzione.

Un termine è chiuso se non ha variabili. i.e. $\underbrace{x}_{\text{aperto}}$; $\underbrace{1}_{\text{chiuso}}$; $\underbrace{+(x,0)}_{\text{aperto}}$.

Insieme dei termini

Sia $(t_1, \dots, t_n)$ l'insieme dei termini di $\mathcal{L}$ è l'unione delle variabili, dei simboli di costante e $\forall\ f$ simbolo di funzione, $f(t_1, \dots, t_n)$.

L'insieme dei termini chiusi di $\mathcal{L}$ è pari all'insieme dei termini di $\mathcal{L}$ senza le variabili.

Se $\mathcal{L}$ non contiene né simboli di costante né simboli di funzione, i termini di $\mathcal{L}$ sono le sole variabili, che ovviamente non sono termini chiusi.

2.5. Letterale e coppia complementare

Un letterale è una formula atomica, oppure la negazione di una formula atomica.

Se F è una formula, $\{F, \bar{F}\}$ è una coppia complementare.

Ovviamente se un insieme di formule contiene una coppia complementare allora è insoddisfacibile.

2.6. Formule

Le formule della logica predicativa sono:

- Le formule atomiche: $t_1 \star \dots \star t_n$, dove t_i è un termine e $\star$ è un simbolo di relazione.
- Una negazione: $\bar{F}$, dove F non è una formula atomica.
- Una congiunzione, disgiunzione o implicazione: $F \vee G, F \wedge G, F \rightarrow G$.
- Una quantificazione esistenziale o universale: $\forall x\ F, \exists x\ F$.

Sia Γ un insieme di formule. Il linguaggio di Γ viene indicato con $\mathcal{L}(\Gamma)$.

2.7. Grado di una formula

Il grado di una formula predicativa si calcola analogamente a quello di una formula proposizionale, eccettuato il fatto che si considerano anche i simboli logici $\forall$ e $\exists$ alla stregua degli altri (peso 1).

2.8. Occorrenze libere e legate di una variabile x

i. Le occorrenze di x sono libere se compaiono in una formula atomica,

ii. Le occorrenze di x sono libere se compaiono in una negazione, congiunzione, disgiunzione, implicazione, quantificatore universale, quantificatore esistenziale di un'altra variabile,

iii. Le occorrenze di x non sono libere se compaiono in un quantificatore universale o esistenziale di x.

Le occorrenze che non sono libere si dicono legate.

2.9. Variabili libere

Le variabili libere di una formula F sono le variabili che hanno almeno un'occorrenza libera in F.

2.10. Enunciato o formula chiusa

Un enunciato o formula chiusa è una formula prima di variabili libere.

2.11. Formula aperta

Una formula aperta è una formula senza alcun quantificatore.

2.12. Sottoformule

Le sottoformule vengono definite come nel caso della logica proposizionale.

2.13. Interpretazione

Per prima cosa vanno interpretati i termini di $\mathcal{L}$.

Sia I è un'interpretazione di $\mathcal{L}$.

I termini interpretati di $\mathcal{L}$ appartengono a D^I, chiamato dominio dell'interpretazione:

- $\forall c$ simbolo di costante chiuso, I associa c con $c^I \in D^I$
- $\forall f$ simbolo di funzione chiuso, $f^I: (D^I)^n \to D^I$
- $\forall p$ simbolo di relazione chiuso, $p^I \subseteq (D^I)^n$

Ad esempio: $D^I = \{0,1,2\}; c^I = 1; p^I = \{1,2\}; q^I = \{0,2\}; f^I(0) = 1; f^I(1) = 2$.

Sia I una interpretazione di $\mathcal{L}$.

I associa ad ogni termine chiuso $t \in \mathcal{L}$ il suo termine interpretato corrispondente $t^I \in D^I$ in questo modo:

- Se t è un simbolo di costante $\Longrightarrow t^I = c^I$
- Se $t = f(t_1, \dots, t_n)$ con $t_1, \dots, t_n$ termini $\Longrightarrow t^I = f^I(t_1^I, \dots, t_n^I)$

Se t è aperto, I non è sufficiente ad interpretarlo e sarà necessario usare gli stati dell'interpretazione I.

2.14. Stato di un'interpretazione

Uno stato di un'interpretazione I è una funzione σ che ad ogni variabile associa un elemento di D^I.

- Se t è una variabile $\Longrightarrow$ sia x questa variabile, $\sigma(t) = \sigma(x)$
- Se t è una costante $\Longrightarrow$ sia c questa costante, $\sigma(t) = c^I$
- Se $t = (t_1, \dots, t_n)$ con $t_1, \dots, t_n$ termini $\Longrightarrow \sigma(t) = f^I(\sigma(t_1), \dots, \sigma(t_n))$

i.e. Uno stato σ di I, è così definito: $\sigma: x \longmapsto 0; y \longmapsto 1; v \longmapsto 2$.

Da questo si deduce che le variabili possono assumere qualsiasi valore del dominio D^I.

Sia I una interpretazione di $\mathcal{L}$, σ uno stato dell'interpretazione I, x una variabile, $d \in D^I$.

$$\sigma[x/d]: x \longmapsto d; \forall y \neq x \longmapsto \sigma(y)$$

Ovvero $\sigma[x/d]$ "assegna" ad x il valore d.

2.15. Soddisfazione

Si noti che $\vDash$ ha sia il significato di "soddisfa" definito qui sotto che quello di conseguenza logica.

La relazione I allo stato σ soddisfa F ($I, \sigma \vDash F$) è così definita:

- $I, \sigma \vDash p(t_1, \dots, t_n)$ con p simbolo di relazione e $t_1, \dots, t_n$ termini $\Leftrightarrow \sigma(t_1), \dots, \sigma(t_n) \in p^I$
- $I, \sigma \vDash \bar{G} \Leftrightarrow I, \sigma \nvDash G$
- $I, \sigma \vDash G \wedge H \Leftrightarrow I, \sigma \vDash G$ e $I, \sigma \vDash H$
- $I, \sigma \vDash G \vee H \Leftrightarrow I, \sigma \vDash G$ o $I, \sigma \vDash H$
- $I, \sigma \vDash G \rightarrow H \Leftrightarrow I, \sigma \nvDash G$ o $I, \sigma \vDash H$
- $I, \sigma \vDash \forall x\, F \Leftrightarrow \forall d \in D,\ I, \sigma[x/d] \vDash F$
- $I, \sigma \vDash \exists x\, F \Leftrightarrow \exists d \in D,\ I, \sigma[x/d] \vDash F$

Si usa scrivere $I \vDash F$ se $\forall \sigma,\ I, \sigma \vDash F$. Si dice anche che <u>$F$ è vera in I</u> o che <u>I è un modello di F</u>.

Sia Γ un insieme di formule,

$I, \sigma \vDash \Gamma$ se $\forall F \in \Gamma,\ I, \sigma \vDash F$.

Si usa scrivere $I \vDash \Gamma$ se $\forall \sigma, \forall F \in \Gamma,\ I, \sigma \vDash F$.

2.16. Equivalenza e conseguenza logica

Il significato di <u>logicamente equivalente</u> ($F \equiv G$) e di conseguenza logica ($F \vDash G$) è identico al caso proposizionale, ovvero:

$$F \equiv G \text{ se } \forall I, \forall \sigma, \quad I, \sigma \vDash F \Leftrightarrow I, \sigma \vDash G$$
$$F \vDash G \text{ se } \forall I, \forall \sigma, \quad I, \sigma \vDash F \Longrightarrow I, \sigma \vDash G$$
$$\Gamma \vDash G \text{ se } \forall I, \forall \sigma, \quad I, \sigma \vDash \Gamma \Longrightarrow I, \sigma \vDash G$$

2.17. Sostituzione nei termini

Sia t un termine, x una variabile, s un termine.

La sostituzione di x con s in t, $t\{x/s\}$ è il termine che si ottiene rimpiazzando ogni x con s all'interno di t:

- $t = s \Longrightarrow t\{x/s\} = s$
- $t \neq s$ oppure t simbolo di costante $\Longrightarrow t\{x/s\} = t$
- $t = f(t_1, \dots, t_n)$ con $t_1, \dots, t_n$ termini e f simbolo di funzione $\Longrightarrow t\{x/s\} = f(t_1\{x/s\}, \dots, t_n\{x/s\})$

2.18. Sostituzione nelle formule

- Se F formula atomica con $F = p(t_1, \dots, t_n)$ con p simbolo di relazione, $t_1, \dots, t_n$ termini $\Longrightarrow$ $F\{x/s\} = p(t_1\{x/s\}, \dots, t_n\{x/s\})$
- Se F non è formula atomica $\Longrightarrow F\{x/s\}$ si ottiene sostituendo ogni A formula atomica $\in$ F con $A\{x/s\}$

Lemma di sostituzione

Sia I interpretazione di $\mathcal{L}$, σ stato di I, t_c un termine chiuso, F una formula. Valgono queste proprietà:

$$I, \sigma \vDash F\{x/t_c\} \Leftrightarrow I, \sigma[x/t_c^I] \vDash F$$

$$\forall x\, F \vDash F\{x/t_c\}$$

$$F\{x/t_c\} \vDash \exists x\, F$$

2.19. Equivalenze logiche notevoli

- Sia F una formula.

$$\overline{\forall x\, F} \equiv \exists x\, \bar{F}$$
$$\overline{\exists x\, F} \equiv \forall x\, \bar{F}$$

- Sia $\circ$ uno qualunque di $\wedge$ e $\vee$.

  Sia $\star$ uno qualunque di $\exists$ e $\forall$.

  Sia F una formula, x una variabile, G una formula senza occorrenze libere di x.

$$\star\, x\, F \circ G \equiv \star\, x(F \circ G)$$
$$G \circ (\star\, x\, F) \equiv \star\, x(G \circ F)$$

- Sia F una formula, x una variabile, G una formula senza occorrenze libere di x.

$$\forall x\, F \to G \equiv \exists x\, (F \to G)$$
$$\exists x\, F \to G \equiv \forall x\, (F \to G)$$

$$G \to \forall x\, F \equiv \forall x\, (G \to F)$$
$$G \to \exists x\, F \equiv \exists x\, (G \to F)$$

- Siano F e G formule.

$$\forall x\, F \wedge \forall x\, G \equiv \forall x(F \wedge G)$$
$$\exists x\, F \vee \exists x\, G \equiv \exists x\, (F \vee G)$$
$$\forall x\, F \to \exists x\, G \equiv \exists x\, (F \to G)$$

2.20. Validità e soddisfacibilità

Sia F una formula.

- F è valida $\Longleftrightarrow$ F è soddisfatta da ogni interpretazione per $\mathcal{L}(F)$.
- F è soddisfacibile $\Longleftrightarrow$ F è soddisfatta da qualche interpretazione per $\mathcal{L}(F)$.
- F è insoddisfacibile $\Longleftrightarrow$ F non è soddisfacibile da qualche interpretazione per $\mathcal{L}(F)$.

Sia Γ un insieme di formule.

- Γ è valida $\Longleftrightarrow$ ogni interpretazione soddisfa ogni $F \in \Gamma$.

  Nota: se ogni $F \in \Gamma$ è valida $\Longrightarrow$ Γ è valida.
- Γ è soddisfacibile $\Longleftrightarrow$ qualche interpretazione soddisfa ogni $F \in \Gamma$.

  Nota: se ogni $F \in \Gamma$ è soddisfacibile $\not\Rightarrow$ Γ è soddisfacibile.
- Γ è insoddisfacibile $\Longleftrightarrow$ nessuna interpretazione soddisfa ogni $F \in \Gamma$.

  Nota: se ogni $F \in \Gamma$ è insoddisfacibile $\not\Rightarrow$ Γ è insoddisfacibile.

2.21. Traduzioni del linguaggio naturale

Spesso all'interno di quantificatori universale si trovano delle implicazioni e in quantificatori esistenziali una o più congiunzioni.

Se dovesse capitare diversamente verificare bene il significato.

2.22. γ e δ formule

$\boldsymbol{\gamma}$-formule ($\forall$)	
Formula	Istanza relativa ad a
$\forall \boldsymbol{x}\, \boldsymbol{F}$	$F\{x/a\}$
$\overline{\exists \boldsymbol{x}\, \boldsymbol{F}}$	$\overline{F\{x/a\}}$

$\boldsymbol{\delta}$-formule ($\exists$)	
Formula	Istanza relativa ad a
$\exists \boldsymbol{x}\, \boldsymbol{F}$	$F\{x/a\}$
$\overline{\forall \boldsymbol{x}\, \boldsymbol{F}}$	$\overline{F\{x/a\}}$

Si noti che una formula è sempre di solo uno dei seguenti tipi: letterale, doppia negazione, $\alpha, \beta, \gamma, \delta$.

Sia G una γ-formula e G' una sua istanza,

$G \vDash G'$.

Sia Γ un insieme di formule,

G una δ-formula,

G' una istanza di G relativa ad una costante che non compare né in Γ né in G.

$$\Gamma, G \text{ soddisfacibile} \Longrightarrow \Gamma, G' \text{ soddisfacibile}$$

2.23. Tableaux

L'algoritmo per la risoluzione dei tableaux nel caso predicativo è <u>non deterministico</u>, <u>non gode della terminazione forte</u> ed è una procedura per la <u>semidecisione della validità o dell'insoddisfacibilità</u>.

Da questo si deduce che se un enunciato F è insoddisfacibile o valido, l'algoritmo si fermerà. Se è soddisfacibile forse si fermerà o forse proseguirà all'infinito.
Church ha dimostrato la non esistenza di una procedura di decisione per l'insoddisfacibilità.

Si noti che anche l'algoritmo di costruzione dei tableaux non prevede la sostituzione di una formula con un'altra logicamente equivalente ad essa.

In questa versione dell'algoritmo escludiamo i simboli di funzione.
Si noti quindi che gli unici termini chiusi sono i simboli di costante.

L'algoritmo è uguale al caso proposizionale, ma si aggiungono i seguenti casi:

i. Se G è una γ-formula, sia a un simbolo di costante relativo alla variabile che compare nel nodo; se esso non esiste, sia a un nuovo simbolo di costante.
   Nel nuovo nodo aggiungo l'istanza di G relativa ad A, senza rimuovere le γ-formule.
ii. Se G è una δ-formula e a una costante che non compare nel nodo. Sia $G^{'}$ una istanza di G relativa ad a.
   Nel nuovo nodo si sostituisce G con $G^{'}$.

Si noti che le γ-formule vanno applicate a tutte le costanti introdotte: anche quelle introdotte successivamente all'aver istanziato la γ-formula.
Per questo le γ-formule non vanno mai cancellate dalle etichette dei nodi.

Sulla base di quest'ultima affermazione si deduce che l'ordine migliore sia $\alpha, \beta, \delta, \gamma$.

I tableaux, nei confronti della conseguenza logica, si comportano come nel caso proposizionale.

2.24. Costruzione sistematica di tableaux

Ad ogni passo,

- Se un nodo contiene una coppia complementare di letterali oppure se un nodo contiene solo letterali e γ-formule senza costanti assegnabili allora il nodo è una foglia.
- Se un nodo contiene solo letterali e γ-formule con costanti assegnabili allora è un γ-nodo.
- Negli altri casi è un nodo ordinario.

Procediamo quindi in questo modo:

i. Se c'è un nodo ordinario lavoro prima su quello, come di consueto.
ii. Poi lavoro sugli δ-nodi: aggiungo tutte le istanze della γ-formula al nodo successivo.

Sia F un enunciato,

$$F \text{ insoddisfacibile} \Longrightarrow \text{ogni tableau sistematico per } F \text{ è chiuso}$$

2.25. Insieme di Hintikka

Un insieme Γ si dice insieme di Hintikka se gode delle proprietà degli insiemi di Hintikka della logica proposizionale e

i. Se una γ-formula appartiene a Γ, allora almeno un simbolo di costante appartiene a Γ.
ii. Se una δ-formula appartiene a Γ, allora almeno una delle sue istanze appartiene a Γ.

Come nella logica proposizionale, ogni insieme di Hintikka è soddisfacibile.