KER(LOGICA)

Disclaimer:

Questo documento contiene il nucleo delle lezioni di logica matematica del corso del prof.se S. Aguzzoli.
Ogni lezione rappresenta quindi il contenuto principale della teoria necessaria, per eventuali esempi consultare le slide del professore e/o le esercitazioni di laboratorio.
Il corso è diviso in due parti, logica proposizionale e logica del primo ordine, è consigliato seguire le lezioni in ordine per comprendere gli argomenti trattati in modo migliore e con meno ambiguità.

Legenda:

- **ROSSO** = elementi importanti che servono per comprendere altri argomenti e anche gli esercizi
- **VERDE** = Teoremi
- **GIALLO** = Regole generali dei connettivi e quantificatori
- **AZZURRO** = Regole Con di fitch e assiomi di Peano
- **VIOLA** = appunti, note personali o chiarimenti

Indice

I PARTE

II PARTE

Lezione 2

- **Logica proposizionale:** logica dei connettivi e delle lettere e logica degli enunciati completamente istanziati

- **Enunciato:** frase interpretata come vera o falsa in una data circostanza

- **Enunciato semplice:** enunciato non composto da altri enunciati usando i connettivi. Formalizzato con:
    - Lettere proposizionali S,P,Q,R
    - Proposizioni Atomiche completamente istanziate

- **Proposizioni atomiche:** sono le più semplici frasi con senso compiuto interpretabili come vere o false in una data circostanza di un contesto. Si ottengono riempiendo i posti dei <u>predicati</u> del linguaggio con delle <u>costanti</u>

- un **contesto** sottende:
    - un **Linguaggio** = caratterizzato da Vocabolario e Sintassi
    - una **Semantica** = Circostanze e Interpretazione (True / False)

- **Sintassi:** le proposizioni atomiche si ottengono riempiendo i posti dei predicati del linguaggio con dei nomi o più in generale dei sintagmi nominali.

- FOL (First Order <u>Language</u>) il linguaggio di un contesto comprende:
    - **Costanti:** nomi per indicare oggetti
    - **Predicati (n-ari):** per definire proprietà o mettere in relazione oggetti
    - **Funzioni (n-arie):** per indicare oggetti indirettamente

- **Contesto =** Insieme di circostanze a cui si associa un linguaggio
    - nessun contesto è specificato → si sta ragionando nella logica pura (FOL)

- **Linguaggio =** dato da costanti + predicati + funzioni

- **Proposizioni atomiche =** predicati completamente istanziati (in ogni posto) con costanti

- **Ragionamento:** serie di frasi riferite ad un <u>contesto</u>, in cui una, detta <u>conseguenza</u>, segue dalle altre, dette <u>premesse</u>
    - Si dice **logicamente valido in un contesto** sse la conseguenza è vera in tutte le circostanze del contesto in cui sono vere le premesse. La conseguenza in questo caso è <u>conseguenza logica delle premesse</u>
    - Si dice **fondato in una circostanza** sse è valido e le premesse sono vere in essa
    - Si dimostra tramite prove o controesempi

- **Prova:** dimostrazione di una conseguenza a partire dalle premesse attraverso una successione di passaggi**.** Sono di 2 tipi:
  - **Formali:** utilizzo di linguaggio formale e regole di inferenza
  - **Informali:** linguaggio naturale o gergo specialistico

Regole formali dell' IDENTITA' (=)

- a = b è vero in una circostanza sse in quella circostanza a, b sono nomi dello stesso individuo

Identity Introduction (= Intro):

$$\triangleright \;|\; n = n$$

Identity Elimination (= Elim):

$$\begin{array}{l|l} & P(n) \\ & \vdots \\ & n = m \\ & \vdots \\ \triangleright & P(m) \end{array}$$

Lezione 3

- **Connettivo:** costruisce enunciati composti a partire da enunciati più semplici chiamati componenti

- **Connettivo vero funzionale:** il valore di verità di un enunciato è definibile mediante una tavola di verità.
    - Sono AND,OR,NOT

- **fbf** può essere:

    1) **BASE:** una proposizione atomica del linguaggio
    2) **PASSO:** si ottiene riempiendo con fbf del linguaggio i posti dei costrutti

- Il valore di verità di una fbf si ottiene:

    1) si sostituiscono le atomiche con i rispettivi valori di verità
    2) si calcola il valore dell'espressione booleana ottenuta applicando le tavole di verità

- **Interpretazione:** fissato l'universo del discorso A si definiscono:
    - **Costanti:** elementi dell'universo del discorso → $\mathbf{I(c)\in A}$
    - **Predicati:** relazioni (n-arie) sull'universo del discorso → $\mathbf{I(P)\in A^n}$
    - **Proposizione atomica:** $P(c_1,\dots,C_n)$ è vera sse $\mathbf{(I(C_1),\dots,I(C_n))\in I(P)}$

- **P è logicamente vera** sse P è VERA in tutte le circostanze del contesto
- **P è tautologicamente vera** sse P è VERA in tutte le righe della tavola di verità
- **P è logicamente possibile** sse ESISTE una circostanza di contesto in cui P è vera
- **P è tautologicamente possibile** sse ESISTE una interpretazione booleana in cui P è vera
- **P è logicamente impossibile** sse P è FALSE in tutte le circostanza del contesto
- **P è tautologicamente impossibile** sse P è FALSE in tutte le righe della tavola di verità

- Le tautologie rappresentano leggi di ragionamento generali poiché sono vere in ogni circostanza e contesto. Alcune tautologie importanti:

    - **Terzo Escluso:** $\mathbf{P \vee \neg P}$
    - **Principio di non contraddizione:** $\mathbf{\neg(P \wedge \neg P)}$

- **TEOREMA: P è logicamente impossibile in un contesto C sse ¬P è logicamente vera in C**

- **TEOREMA: P è proposizionalmente impossibile sse ¬P è tautologicamente vera.**

- Alcune proprietà importanti sono:

    - P Tautologia → P logicamente vera
    - P Possibile in un contest → proposizionalmente possibile

Lezione 4

- Due proposizioni sono

  - **Tautologicamente equivalenti ($P \Leftrightarrow_T Q$)**: sse hanno lo stesso valore di verità in tutte le interpretazioni booleane.

  - **Logicamente equivalenti in un contesto C ($P \Leftrightarrow_C Q$):** sse hanno lo stesso valore di verità in tutte le interpretazioni possibili nel contesto

- **Rimpiazzamento:** se $P \Leftrightarrow Q$ è possibile rimpiazzare P con Q in una formula F ottenendo $F \Leftrightarrow G$

- **Conseguenza logica:** Q segue logicamente da $P_1,\ldots,P_n$ in un contesto C sse Q è vera in tutte le interpretazioni nel contesto in cui $P_1,\ldots,P_n$ sono vere → $P_1,\ldots,P_n \vDash_C Q$

- **Conseguenza Tautologica:** Q segue tautologicamente da $P_1,\ldots,P_n$ sse Q è vera in ogni interpretazione booleana in cui $P_1,\ldots,P_n$ sono vere → $P_1,\ldots,P_n \vDash_T Q$

Regole formali dell' AND (∧)

Conjunction Introduction (∧ Intro):

$$\begin{array}{l|l} & P_1 \\ & \Downarrow \\ & P_n \\ & \vdots \\ \triangleright & P_1 \wedge \ldots \wedge P_n \end{array}$$

Conjunction Elimination (∧ Elim):

$$\begin{array}{l|l} & P_1 \wedge \ldots \wedge P_i \wedge \ldots \wedge P_n \\ & \vdots \\ \triangleright & P_i \end{array}$$

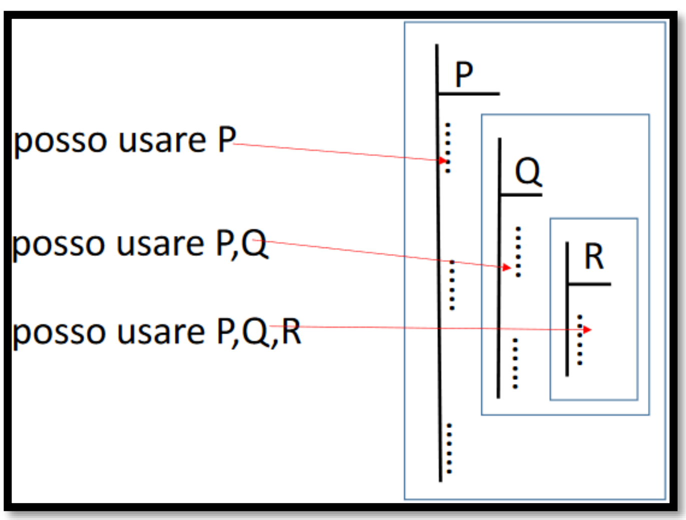
Regole Formali dell'OR ($\vee$)

Disjunction Introduction ($\vee$ Intro):

$$\begin{array}{l|l} & P_i \\ & \vdots \\ \triangleright & P_1 \vee \ldots \vee P_i \vee \ldots \vee P_n \end{array}$$

- Per quanto riguarda la **regola di eliminazione** bisogna lavorare per casi: cosa posso inferire nel caso P e cosa nel caso Q; se in entrambi i casi posso inferire C, allora **C segue da «P o Q»**

- le <u>prove per casi</u> introducono l'utilizzo di dimostrazioni con <u>assunzioni</u>:

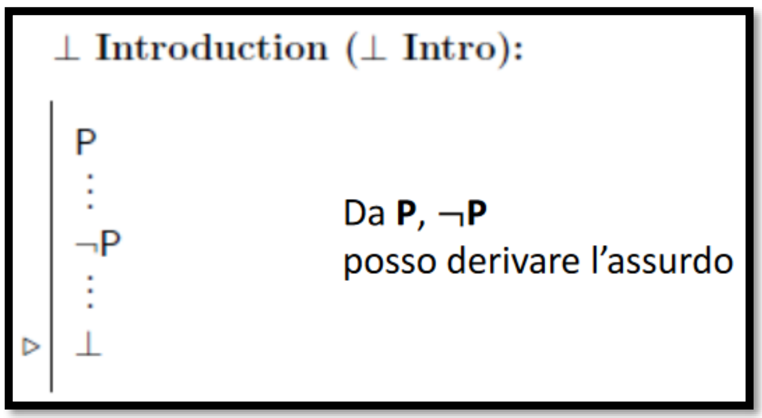


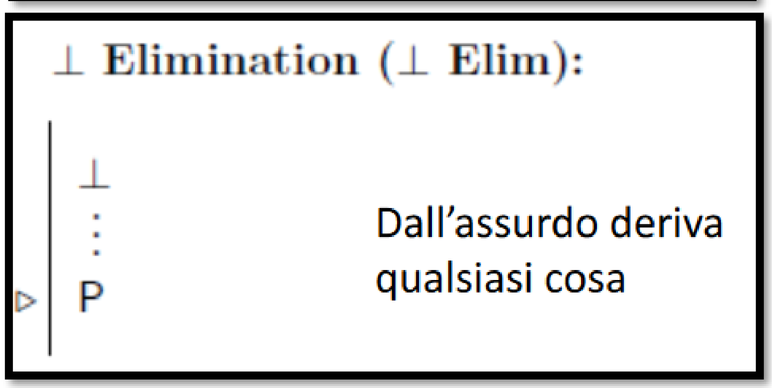
- vengono assunte proprietà non note che vengono usate come premesse di sottoprove
- una assunzione può essere usata solo nella sua sottoprova

Lezione 5

- **Assurdo** = proprietà logicamente impossibile ovvero falsa in ogni circostanza del contesto
- **Assurdo proposizionale:** **$P \wedge \neg P$**

Regole Formali dell'ASSURDO (⊥)





- **TAUT CON:** Sia C1,...,Cn $\vDash_T$ D una conseguenza tautologica. Se fra le premesse o le conseguenze intermedie di una prova abbiamo C1,...,Cn, allora nella prova possiamo inferire D. è una regola **valida in ogni contesto**

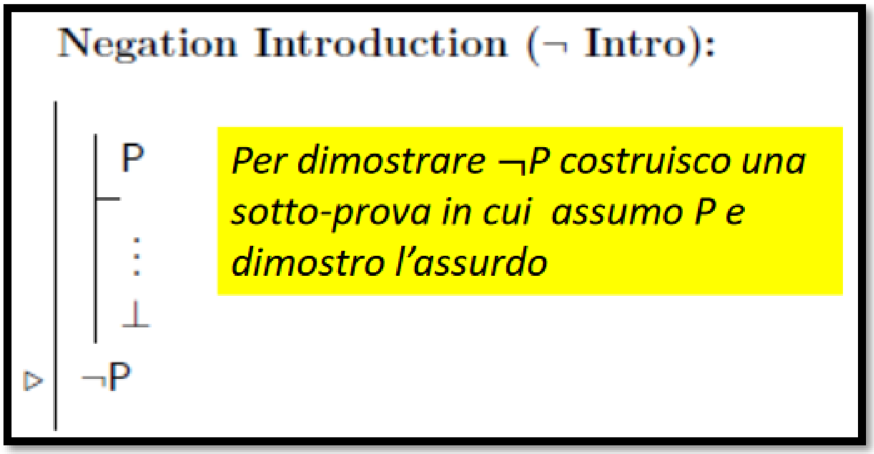
  → permette di inferire un enunciato che segue dagli enunciati selezionati unicamente in virtù della semantica dei connettivi vero-funzionali.

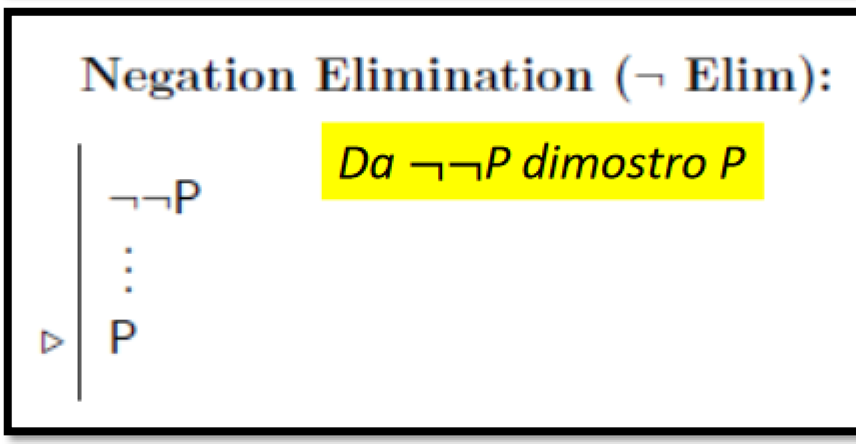
- **ANA CON:** Sia C1,...,Cn $\vDash_{Tw}$ D una conseguenza logica nel **mondo dei blocchi**. Se fra le premesse o le conseguenze intermedie di una prova abbiamo C1,...,Cn, allora nella prova possiamo inferire D. Tale prova è **valida solo nel contesto TW**.

  → permette di inferire un enunciato che segue dagli enunciati selezionati in virtù della semantica dei connettivi vero-funzionali, dei quantificatori, del predicato di identità e dei predicati di TW

- **Validità di una regola:** Una regola è **valida in un contesto** sse le nuove conseguenze che introduce in una prova sono conseguenze logiche (contestuali) delle premesse della prova stessa.

Regole Formali della NEGAZIONE (¬)





Lezione 6

- L'implicazione permette di costruire asserzioni condizionali
- Esistono "due tipi" di implicazione:

  1) **Materiale**: è quella verofunzionale ovvero che si definisce tramite una tabella della verità → quella che verrà usata nel corso

  2) **Causale**: è quella usata nel parlare e quindi che non si basa su una tavola di verità

- Implicazione si legge "Ogni qualvolta P è vera, anche Q è vera"

  → $P \rightarrow \ \Leftrightarrow_T (\neg P) \lor Q$

- Da ciò derivano le seguenti tavole di verità:

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

P	Q	$Q \rightarrow P$
T	T	T
T	F	T
F	T	F
F	F	T

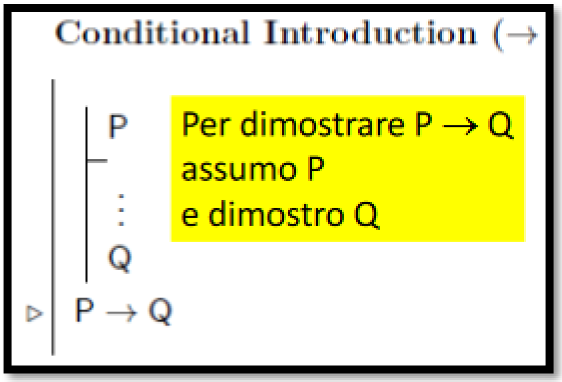
- P vera è **sufficiente** affinché Q sia vera
- Q vera è **necessaria** affinché P sia vera

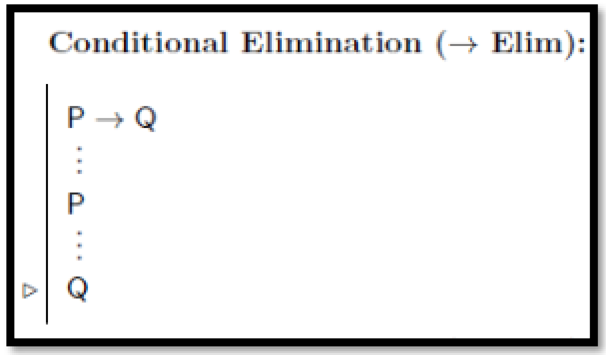
TEOREMA DI DEDUZIONE:

$$\mathbf{(P1, ..., Pn) \vDash_c Q \text{ sse } \vDash_c (P1 \land ... \land Pn) \rightarrow Q}$$

Ovvero:
l'implicazione «cattura» la conseguenza (tauto)logica: Q è conseguenza (tauto)logica di P1, ..., Pn, se e solo se(P1∧... ∧Pn) →Q è (tauto)logicamente vera

Regole Formali della IMPLICAZIONE(→)

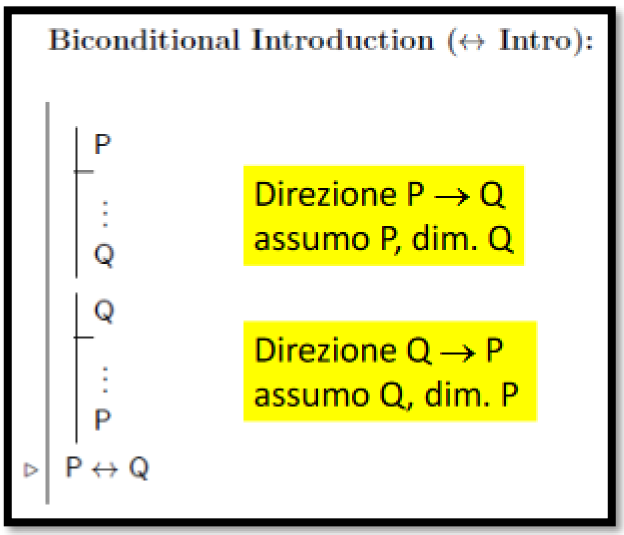


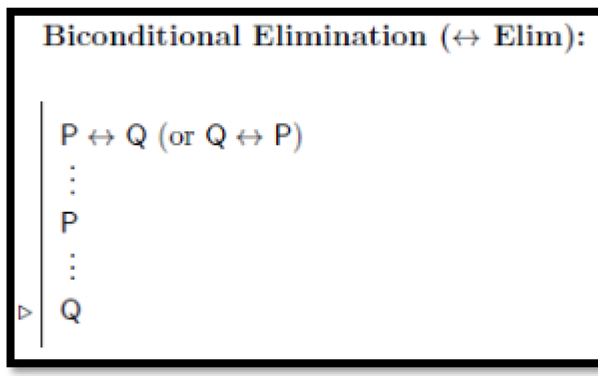


MODUS PONENS

L'introduzione si basa sul teorema di Deduzione

Regole Formali della CO-IMPLICAZIONE(↔)





- Conseguenze logiche notevoli: Contrapposizione e Modus Tollens:
  - **Modus Tollens per →:** $(\neg Q, P \to Q) \vDash_T (\neg P)$
  - **Modus Tollens per ↔:** $(\neg Q, P \leftrightarrow Q) \vDash_T (\neg P)$
    $(\neg Q, Q \leftrightarrow P) \vDash_T (\neg P)$
  - **Contrapposizione:** $\vDash_T (P \to Q) \leftrightarrow (\neg Q \to \neg P)$

Lezione 7

- Si ricordi che:
  - **($P1,...,Pn$)** $\vdash_T$ **(Q)** = Esiste una prova in fitch con premesse p1-pn e conseguenza Q
  - $\vDash_T$ = Tautologia

- **Teorema di validità:** Se $(P_1,...,P_n) \vdash_T (Q)$, allora $(P_1,...,P_n) \vDash_T (Q)$

- **Teorema di completezza:** Se $(P_1,...,P_n) \vDash_T (Q)$, allora $(P_1,...,P_n) \vdash_T (Q)$

- **Teoria:** un insieme Γ di enunciati eventualmente anche infinito

- **Teorema di deduzione:** $\Gamma \cup \{P\} \vDash_T Q$ **sse** $\Gamma \vDash_T P \rightarrow Q$.
  $\Gamma \cup \{P\} \vdash_T Q$ **sse** $\Gamma \vdash_T P \rightarrow Q$.

Lezione 9

- i linguaggi del primo ordine sono infiniti
- **Variabili:** insieme finito di simboli fissato arbitrariamente (x,y,z,x1,y2,z2…)
- **Connettivi:** $\wedge, \rightarrow, \neg, \perp, \leftrightarrow$
- **Quantificatori:** $\forall$ e $\exists$: hanno bisogno di una variabile e di una formula
- **Costanti:** denotano oggetti dell'universo del discorso
- **Predicati:** denotano relazioni tra oggetti del discorso
    - si definiscono completamente istanziati sono paragonabili a proposizioni atomiche
- **Funzioni:** si usano per denotare oggetti in maniera indiretta
- Ogni linguaggio del FOL è diviso in due livelli:

1) **Termini:** Denotano oggetti dell'universo e sono costruiti in questo modo*:*
    - variabili
    - costanti
    - simboli di funzione

2) **Formule:** Denotano frasi che possono essere T/F in una data circostanza. Definiti da:
    - termini
    - predicati
    - connettivi
    - quantificatori

- si definisce ora **T(L)** dei termini di L induttivamente:
    - **ogni variabile x è un termine di L**
    - **ogni costante x in C(L) è termine di L**
    - **se f $\in$ F(L) e {t1,t2…} sono termini di L allora f(t1,t2,t3…) è termine di L**

- si definisce **FBF(L)** delle fbf di L è definito induttivamente come segue:
    - Se **P $\in$ P(L)** **e {t1,t2,...,tn} $\in$ T(L)** allora **P(t1,t2,…,tn) è fbf atomica**
    - Se **P,Q $\in$ FBF(L)** allora anche **(P,Q), (P.Q), (P$\rightarrow$Q), (P$\leftrightarrow$Q),$\neg$P, $\perp$ lo sono**
    - Se **P $\in$ FBF(L)** e **x una variabile** allora **($\forall$x P)** e **($\exists$x P)** sono fbf di L

- definizione induttiva di **Variabili libere** e **vincolate**

    - **BASE:**
        - le occorrenze di variabili libere in una atomica P(t1,…,tn) sono tutte le occorrenze di variabili in t1,t2,…,tn.
        - ⊥ non ha occorrenze di variabili libere

    - **PASSO:**
        - le occorrenze di variabili libere in (¬P) , (P ∧Q), (P ∨Q), (P →Q), (P ↔Q) sono tutte le occorrenze di variabili libere in P e Q
        - Le occorrenze di variabili libere in ∀**v P** sono tutte le occorrenze di variabili libere in P. Ogni occorrenza di v in P è vincolata in ∀**v P.**
        - Le occorrenze di variabili libere in ∃**v P** sono tutte le occorrenze di variabili libere in P. Ogni occorrenza di v in P è vincolata in ∃**v P.**

- una variabile può avere occorrenze libere e vincolate nella stessa fbf

- **Formula chiusa:** fbf che non contiene occorrenze di variabili libere

- **Formula aperta:** fbf che contiene occorrenze di variabili libere

- Una **fbf** è detta **proposizion/enunciato** sse è chiusa

- Sia A(x1,x2,…,xn) una fbf in cui occorrono libere le variabili x1,x2,…,xn

    - allora per ogni n-pla di oggetti (a1,…,an) dell'universo del discorso U, si dice che **(a1,a2,…,an) soddisfa A** sse **A(a1,a2,…,an) True in U**
    - A(a1,a2,…,an) è ottenuta da A(x1,x2,…,xn) rimpiazzando ogni occorrenza libera di x1 con a1,x2 con a2,…,xn con an

- Problema: ogni ai è un oggetto «semantico», non un «nome» sintattico. Non è detto che ogni oggetto di U abbia un nome sintattico che lo denoti

    - → per ogni n-pla di oggetti (a1,a2,…,an) dell'universo U si dice che **(a1,a2,…,an) soddisfa A** sse **A(c1,c2,…,cn) è True in U**

- **L STRUTTURA:** Sia L un linguaggio (dato da **C(L)**, **F(L)**, **P(L)**). Una L struttura è una coppia (U,I) dove:

    - **U** = universo del discorso
    - **I** = funzione interpretazione definita come segue:
        - per ogni c ∈ C(L), **I(c) ∈ U**
        - per ogni f ∈ F(L), **I(f) è funzione n-aria su U**
        - per ogni p ∈ P(L), **I(p) è una relazione n-aria su U**

Lezione 10

- Dato un linguaggio L sia S = (U,I) una L-struttura, **l'insieme GT(L) dei termini chiusi** è definito induttivamente:
  - Ogni costante c in C(L) = C(L) ∪ {ca: a ∈U} <u>è termine chiuso</u> di Lu
  - Se f è simbolo di funzione n-ario in F(L) e (t1,t2,…,tn) termini chiusi di Lu, allora anche f(t1,t2,…,tn) è <u>termine chiuso</u>

- Dato un linguaggio L sia S = (U,I) una L-struttura, **l'interpretazione I(t) di ogni termine** t∈ GT(Lu) è data induttivamente:
  - per ogni c∈C(L), I(C) è già definita
  - per ogni a∈U, si pone I(Ca):=a
  - Se s ∈ F(L) e (t1,t2,…,tn) ∈ GT(Lu) allora I(f(t1,t2,…,tn)):=(I(f))(I(t1),I(t2),…,I(tn))

→ garantisce che per ogni termine t ∈GT(LU) **I(t) ∈ U** quindi **t denota l'oggetto I(t)**

- Dato un linguaggio L e S=(U,I) una L-struttura allora **E(LU) = {A ∈FBF(LU) : libere(A) = ∅}**
  - se A,B ∈E(LU) allora anche { ⊥, ¬A, A∧B,A∨B,A →B, A ↔B } ∈ E(LU).

- Dato un linguaggio L e S=(U,I) una L-struttura allora **I(A) di ogni enunciato A∈E(LU)** è definita induttivamente: Se P∈ P(L), e (t1,t2,…,tn) ∈GT(LU) allora:
  - I(P(t1, t2, ..., tn)) := T se (I(t1),I(t2),..., I(tn)) ∈I(P)
  - I(P(t1, t2, ..., tn)) := F se (I(t1),I(t2),..., I(tn)) ∉I(P)

- Dato un linguaggio L e sia S=(U,I) una L-struttura
  - Sia A ∈FBF(LU)
  - Sia x una variabile
  - Sia c ∈C(LU)

  → **A[x:c]** è la scrittura per intendere la formula ottenuta <u>rimpiazzando in A ogni occorrenza libera di x con c</u> Inoltre:

    - A[x:c] ∈FBF(LU)
    - x non occorre libera in A[x:c]
    - se libere(A) ⊆{x}, allora A[x:c]∈E(LU)

- Dato un linguaggio L e sia S=(U,I) una L-struttura

  1) Se **∀x A è un enunciato** allora
    - I(∀x A) := **True** se per ogni a∈U, I(A[x:ca]) = **True**
    - I(∀x A) := **False** altrimenti

  2) Se **∃x A è un enunciato** allora
    - I(∃x A) := **True** se per almeno un a∈U, I(A[x:ca]) = **True**
    - I(∃x A) := **False** altrimenti

- Dato un linguaggio L e S=(U,I) una L-struttura allora per ogni enunciato A $\in$E(L_U): **I(A) $\in${ F, T }**
  Nb: *Lu non è altro che la dicitura per identificare il linguaggio esteso*

- Dato un linguaggio L e S=(U,I) una L-struttura allora per ogni enunciato A $\in$E(L): **I(A) $\in${ F, T }**

- In ogni fissata L-struttura S ogni L-enunciato ha un preciso valore di verità

Nb: Cos'è un L-enunciato?

"Un enunciato è una fbf priva di occorrenze di variabili libere.
Quando si vuole sottolineare a che linguaggio L esso appartiene si dice L-enunciato, intendendo per l'appunto un enunciato che usa solo i simboli di costante, funzione e predicato che definiscono L. Un enunciato sul linguaggio ampliato Lu allora sara' un Lu -enunciato. Se il linguaggio L e' chiaro dal contesto, spesso si dira' solo "enunciato", al posto di L-enunciato. Inoltre, a volte, impropriamente, un Lu-enunciato sara' chiamato L-enunciato o semplicemente enunciato: dato che l'ampliamento di L a Lu e' solo strumentale per dare la semantica degli L-enunciati (proprio quelli sul linguaggio originale L), puo' capitare di chiamare L-enunciato un Lu-enunciato, anche se cio' e' improprio (gli L_U-enunciati sono oggetti "temporanei", usati solo per valutare gli L-enunciati in una L-struttura S = (U,I) con universo per l'appunto U)."

- Dato un linguaggio L e S=(U,I) una L-struttura se si volesse attribuire un valore di verità ad una formula aperta A$\in$FBF(L) si utilizza la **chiusura universale di A:**

  - I(A) := I($\forall x_1$ $\forall x_2$,...,$\forall x_n$ A) dove libere(A) = {x1, x2,..., xn}

- Uno strumento molto utile per la traduzione dal linguaggio naturale sono le **Formule Aristoteliche:**

1)	**" Ogni P è Q"**	**$\leftrightarrow$**	**$\forall x\ (P(x) \to Q(x))$**
2)	**"Qualche P è Q"**	**$\leftrightarrow$**	**$\exists x\ (P(x) \wedge Q(x))$**
3)	**"Nessun P è Q"**	**$\leftrightarrow$**	**$\forall x\ (P(x) \to \neg Q(x))$**
4)	**Qualche P non è Q**	**$\leftrightarrow$**	**$\exists x\ (P(x) \vee \neg Q(x))$**

Lezione 11

- **Modello:** Una struttura S=(U,I) è un <u>modello</u> di un L-enunciato A sse A è vero in S ovvero I(A)=True
  → Si Dice **S ⊨A**

- **L-Teoria:** Un insieme di L-Enunciati Γ

- **Modello di una teoria:** Una struttura S=(U,I) è un <u>modello di una teoria</u> Γ sse A è vera in S per ogni $A \in \Gamma$ → Si Dice **S⊨Γ**

- **Contromodello di una teoria:** Una struttura S=(U,I) è un <u>contromodello di una teoria</u> Γ sse A è falsa per almeno un $A \in \Gamma$

- **Verità logica (FO):** Dato un L-enunciato A, A è <u>verità logica</u> sse **S ⊨A** per ogni L-Struttura S=(U,I)

  → Si Dice $\models_{FO} A$

- **Enunciato vero in una teoria Γ (FO):** Un L-enunciato A è vero in una L-teoria Γ sse per ogni L-struttura S=(U,I) vale che Se **S⊨Γ** allora **S⊨A**

  → Si Dice $\Gamma \models_{FO} A$

- <u>**Conseguenza Logica:**</u>
  - Q si dice conseguenza logica delle premesse P1,…,Pn, scritto <u>**P1 ,..., Pn$\models_{FO}$Q**</u> sse Q è vera nella teoria {P1 ,..., Pn}.
  - = Per ogni L-struttura S=(U,I) Se **I(P)=I(P2)=…=I(Pn)=True** allora **I(Q)=True**

- <u>**Verità Logica:**</u> P è logicamente Vera in C sse P è vera in ogni L-struttura che appartiene a C

  → Si Dice $\models_C P$

- <u>**Equivalenza Logica**</u>: P e Q sono logicamente equivalent in C sse I valori di verità di P e Q coincidono in ogni L-struttura che appartiene a C
  → Si Dice $P \Leftrightarrow_c Q$

Lezione 12

- **Atomica generalizzata:** Formula che al primo passo non può essere scomposta con un connettivo
    - Si dice che una fbf A è una atomica generalizzata sse A non è nelle forme A = ¬B, A = BvC, A = B∧C, A = B→C, A = B↔C, A = ⊥
    - la formula **verofunzionale** fvf(A) di un enunciato (o una fbf) A si ottiene rimpiazzando ogni atomica generalizzata di A con una lettera proposizionale

      → la fvf di ∀x Tet(x) ∨¬∀xTet(x) è P∨¬P

      Nb: *con la formula vero funzionale posso vedere se una formula è anche taut con oppure no. Sostituendo ogni atomica generalizzata con una lettera proposizionale posso verificare se la formula è una tautologia oppure no.*

- **FO CON:** permette di inferire un enunciato che segue dagli enunciati selezionati unicamente in virtù della semantica dei connettivi vero-funzionali, dei quantificatori e del predicato di identità.

Regole Formali del PER OGNI(∀)

Universal Introduction (∀ Intro):

$\boxed{c}$

⋮

P(c)

▷ ∀x P(x)

Where c does not occur outside the subproof where it is introduced.

Universal Elimination (∀ Elim):

∀x S(x)

⋮

▷ S(c)

da ∀x S(x)
possiamo inferire S(c)

Dove c è un termine chiuso (costante o composto)

Regole Formali del ESISTE(∃)

Existential Introduction (∃ Intro):

$S(c)$
$\vdots$
$\triangleright\ \exists x\, S(x)$

da S(c)
possiamo inferire ∃x S(x)

Dove c è un termine chiuso
(costante o composto)

Existential Elimination (∃ Elim):

$\exists x\, S(x)$
$\vdots$
$\boxed{c}\ S(c)$
$\vdots$
Q
$\triangleright\ Q$

Where c does not occur outside the subproof where it is introduced.

***Nb Eliminazione di* ∃ e l'intro di ∀:** fanno uso di sotto-prove nelle quali si usa un nuovo nome generico per denotare nel corso del ragionamento un generico elemento del dominio:

- in **elim di ∃,** questo elemento esiste ma non lo si conosce. eliminando ∃xP(x), assumiamo P(generico); il ragionamento deve andar bene per qualunque elemento, incluso quello che assumiamo esistere (e che abbiamo chiamato «generico»)
- in **intro di ∀,** dimostriamo P(generico) e nel ragionamento «generico» deve essere usato come sinonimo di «chiunque», in modo da poter inferire ∀x P(x)

MACRO REGOLA ∀→Intro

utilizzata per ragionare su forme aristoteliche che combinano ∀e →.

General Conditional Proof (∀ Intro):

| $\boxed{c}$ P(c)
|---
| ⋮
| Q(c)

Where c does not occur outside the subproof where it is introduced.

▷ ∀x (P(x) → Q(x))

Nb: questa macro regola non è altro che l'unione delle regole di introduzione per l'implica e l'esiste

Lezione 13

- sia A un enunciato e si consideri la sua forma verofunzionale B=fvf(A). Se B è tautologicamente equivalente a C, posso usare l'equivalenza $B \Leftrightarrow_T C$ per riscrivere A in una forma logicamente equivalente.

  → Se **fvf(P_1),..., fvf(P_n) $\vDash_T$ fvf(Q)**, allora **P_1 ,..., $P_n \vDash_T$ Q**

  Nb: è la stessa cosa del rimpiazzamento visto nelle prime slide

- formule di **de Morgan** per i **quantificatori**:

  1) $\neg\forall x\, P(x) \Leftrightarrow_{FO} \exists x\, \neg P(x)$
  2) $\neg\exists x\, P(x) \Leftrightarrow_{FO} \forall x \neg P(x)$
  3) $\forall x\, P(x) \Leftrightarrow_{FO} \neg\exists x\, \neg P(x)$
  4) $\exists x\, P(x) \Leftrightarrow_{FO} \neg\forall x\, \neg P(x)$

- bisogna fare molta attenzione quando si opera con I quantificatori perchè non si possono spostare a "caso". Stessa cosa capita con i quantificatori multipli, dire $\forall x\, \exists y$ è diverso da dire $\exists y \forall x$

- quando vengono inseriti quantificatori ridondanti si dice che è una **quantificazione vacua** Se x non occorre libera in P si ha**:**
  - $\forall x \forall x P(x) \Leftrightarrow_{FO} \exists x \forall x P(x) \Leftrightarrow_{FO} \forall x\, P(x)$
  - $\exists x\, \exists x\, P(x) \Leftrightarrow_{FO} \forall x\, \exists x\, P(x) \Leftrightarrow_{FO} \exists x\, P(x)$
  - $\forall x\, (P \vee Q(x)) \Leftrightarrow_{FO} P \vee \forall x\, Q(x)$
  - $\exists x\, (P \wedge Q(x)) \Leftrightarrow_{FO} P \wedge \exists x\, Q(x)$

- è possibile anche **rinominare le variabili** se quella che vogliamo introdurre **non occorre neanche vincolata in P(X).** Quindi si può dire che $\forall x\, P(x) \Leftrightarrow_{FO} \forall y\, P(y)$ e $\exists x\, P(x) \Leftrightarrow_{FO} \exists y\, P(y)$

- con i quantificatori è possibile esprimere proposizioni che parlano di quantità ben definite:

  1) **"esiste esattamente un individuo per cui vale P"**

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

  2) **"esiste esattamente n individui per cui vale P"**

     → $\exists x1 \exists x2\ \exists xn \forall xn+1\ ([x1 \neq x2] \wedge [xn-1 \neq xn] \wedge P(xn+1) \leftrightarrow ([xn+1= x1] \vee [xn+1= xn])$

  3) **"esistono almeno n individui per cui vale P"**

     → $\exists x1\ \exists x2\ \exists xn\ (P(x1) \wedge P(x2) \wedge P(xn) \wedge [x1 \neq x2] \wedge [x1 \neq x3] \wedge [x1 \neq xn] \wedge [x2 \neq x3] \wedge [x2 \neq xn] \wedge [xn-1 \neq xn])$

  4) **"esistono al massimo n individui per cui vale P"**

     → $\forall x1 \forall x2\ \forall xn\ \forall xn+1\ ((P(x1) \wedge P(x2) \wedge P(xn+1)) \rightarrow ([x1= x2] \vee [x1= x3] \vee [x1= xn+1] \vee [x2= x3] \vee [x2= xn+1] \vee [xn= xn+1]))$

Lezione 14

- Quando si progetta un linguaggio L per parlare di una classe di L-strutture, alcuni simboli di L (predicati, funzioni) sono definiti a partire da altri simboli di L tramite enunciati. Queste definizioni possono essere:
  - **ricorsive**
  - **esplicite**

- accenni: i linguaggi multisortati servono a definire una tipizzazione più esplicita e coincisa. Si definiscono le sorte e gli elementi del linguaggio avranno un tipo da specificare e per esempio i predicati non saranno piu P(x,y,z) ma P(tipo,tipo,tipo)

- **Assiomatizzare un insieme di circostanze:** Fornire una teoria Γ **tale che tutti i modelli di Γ(**vale a dire, le L-strutture S =(U,I) per cui S $\models\Gamma$) **siano tutte e sole le L-strutture che descrivono**, astrattamente, **l'insieme desiderato di circostanze**.

- **Nb:** *questa lezione è abbastanza astratta, nel caso si volessero esempi basta guardare la lezione 14 del prof per comprendere meglio i concetti di assiomatizzazione e tipizzazione*

Lezione 15

- le definizioni induttive sono un modo per definire linguaggi, tipi di dati ricorsivi e altre cose
- la definizione induttiva di un insieme S sottende:

  1) **Principio di induzione**: consente di dimostrare proprietà che valgono per tutti gli elementi di S

  2) **Ricorsione**: definire le funzioni totali e predicati su S

- il principio di induzione assume la formula di una regola che si basa su base, passo e chiusura

- Si consideri un certo insieme S, si avrà la seguente **definizione induttiva**

  1) **Base:** ad S appartengono b1,b2,…,bk elementi

  2) **Passo:** Se (e1,e2,…,en) $\in$ S allora anche p(e1,e2,…,en) $\in$S.

     Nb: ci possono essere più passi.

  3) **Chiusura:** nient'altro $\in$S

  → Da ciò possiamo definire il relativo **principio induttivo** che è:

  Per ogni fbf H(x) (con libere(H) = {x}), se valgono:

  1) **base:** H(b1), H(b2),..., H(bk)
  2) **passo**: $\forall x1 \forall x2 \ldots \forall xn((H(x1) \wedge H(x2) \wedge \ldots \wedge H(xn)) \rightarrow H(p(x1,\ldots,xn)))$

  allora vale $\forall$**x H(x)**

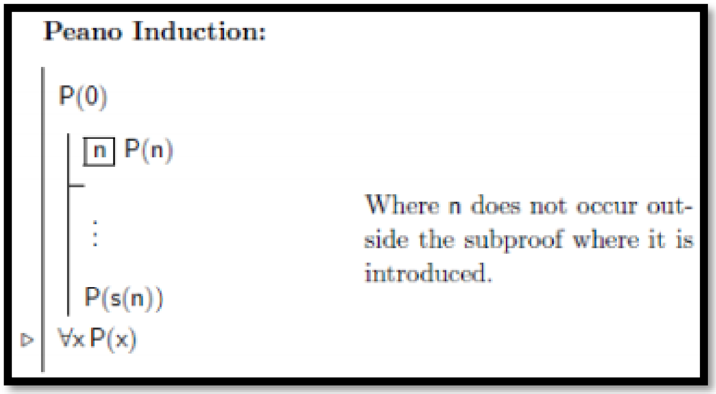
- si ricordi anche per esempio che la definizione di termine è stata data induttivamente

Lezione 16

- la logica di **Peano** si occupa di formalizzare l'aritmetica
- Peano lavora tramite gli assiomi che ci permettono di dimostrare una base e un passo

  N.b: *Durante lo svolgimento di un esercizio con peano, si usano molto le condizioni di eliminazione dei quantificatori e spesso queste regole vengono "ripetute" diverse volte durante la prova per giungere alle dimostrazioni richieste. Questo concetto comunque viene spiegato meglio nella foto di esempio sotto*

- Fitch ha una regola per l'induzione di peano:



- si definisce ora il linguaggio L(PA) dell'Aritmetica di Peano:
  - C(L(PA)) = {0}. = **unica costante è 0**
  - s/1 ∈ F(L(PA)). = **contiene solo una funzione che è detta successore** es *(s(2)=3)*
  - P(L(PA)) = {=/2} = **predicato di identità**

  → il modello standard è una L(PA)-struttura **N = (ℕ, I)** dove:
  - **I(0)** = **0**.
  - **I(s)** = {(**n,n + 1**) : n ∈ ℕ }.
  - I termini 0, s(0), s(s(0)), sono detti **numerali**

- come detto, Peano si basa su assiomatizzazioni, di seguito ne vengono definite alcune:
  - **PA1:** ∀**x** ¬**(s(x) = 0)** ovvero *0 non è il successore di alcun naturale.*
  - **PA2:** ∀**x** ∀**y (s(x) = s(y) → x = y)** ovvero *s è una funzione iniettiva.*

    →*PA1 e PA2 assicurano che in ogni loro modello, le interpretazioni dei numerali 0, s(0), s(s(0)), …, formano una sequenza infinita di elementi distinti*

    →*Non possiamo dimostrare, solo con PA1 e PA2, che il successore di un elemento è sempre diverso dall'elemento stesso.*

  - (**PA3**,**PA4**,**PA5**,**PA6**)
  - **PA7:** **(P(0) ∧∀x(P(x) ⇒P(s(x)))) ⇒ ∀x P(x)**

- Una variante equivalente e utile del principio di induzione è il principio di **induzione forte:**

  "per ogni fbf. P(n) tale che libere(P) = {n}:
  Se per ogni naturale n vale che
  se P(k) vale per ogni k<n, allora vale P(n):
  allora P(n) vale per ogni naturale n"

  → **$\forall n \, ((\forall k \, (k < n \Rightarrow P(k))) \rightarrow P(n)) \Rightarrow \forall n \, P(n)$.**

Strong Induction:

$\boxed{n}$ $\forall x \, (x < n \rightarrow P(x))$

⋮

$P(n)$

▷ $\forall x \, P(x)$

Where n does not occur outside the subproof where it is introduced.

Nb: Solo a scopo di conoscenza generale

EQUIVALENZE LOGICHE

⇔ FO	⇔ FO	Nome legge
$P \rightarrow Q$	$\neg Q \rightarrow \neg P$	Legge di contrapposizione
$\neg(P \land Q)$	$\neg P \lor \neg Q$	Legge di De Morgan
$\neg(P \lor Q)$	$\neg P \land \neg Q$	Legge di De Morgan
$\neg \forall x\, P(x)$	$\exists x\, \neg P(x)$	Legge di De Morgan per i quantificatori
$\neg \exists x\, P(x)$	$\forall x\, \neg P(x)$	Legge di De Morgan per i quantificatori
$\forall x\, P(x)$	$\neg \exists x\, \neg P(x)$	Legge di De Morgan per i quantificatori
$\exists x\, P(x)$	$\neg \forall x\, \neg P(x)$	Legge di De Morgan per i quantificatori
$P \rightarrow Q$	$\neg P \lor Q$	Implicazione materiale
$\forall x\, (P(x) \land Q(x))$	$\forall x\, P(x) \land \forall x\, Q(x)$	Spostamento dei quantificatori
$\exists x\, (P(x) \lor Q(x))$	$\exists x\, P(x) \lor \exists x\, Q(x)$	Spostamento dei quantificatori
$\forall x \forall y\, P(x,y)$	$\forall y \forall x\, P(x,y)$	Scambio dei quantificatori
$\exists x\, \exists y\, P(x,y)$	$\exists y\, \exists x\, P(x,y)$	Scambio dei quantificatori
$\forall x\, (P \lor Q(x))$	$P \lor \forall x\, Q(x)$	Quantificazione vacua (*x occorre libera in P*)
$\exists x\, (P \land Q(x))$	$P \land \exists x\, Q(x)$	Quantificazione vacua (*x occorre libera in P*)
$\forall x P(x)$	$\forall y P(y)$	Rinomina di variabili
$\exists x P(x)$	$\exists y P(y)$	Rinomina di variabili

ERRORI

$\forall x\, \exists y\, P(x,y)$	$\exists y \forall x\, P(x,y)$
$\forall x\, (P(x) \lor Q(x))$	$\forall x\, P(x) \lor \forall x\, Q(x)$
$\exists x\, (P(x) \land Q(x))$	$\exists x\, P(x) \land \exists x\, Q(x)$
$\forall x\, \exists y\, Q(x,y)$	$\forall y\, \exists y\, Q(y,y)$