Intelligenza Artificiale I

Logica del primo ordine

Marco Piastra

Introduzione

Limiti del linguaggio proposizionale

- *Esempio*:

  "Ogni essere umano è mortale"
  "Socrate è un essere umano"

  "Socrate è mortale"

  Il legame logico (intuitivo) è evidente

(Essere umano) implica (essere mortale)	$A \to B$	Schema del	$\varphi \to \psi$
Socrate è un essere umano	C	*modus ponens*	φ
Socrate è mortale	D		ψ

  Nella traduzione logico-proposizionale A, B, C e D non hanno alcun legame

- Altri esempi 'intraducibili':

  "Quando capra e cavolo stanno sulla stessa riva, la capra mangia il cavolo"

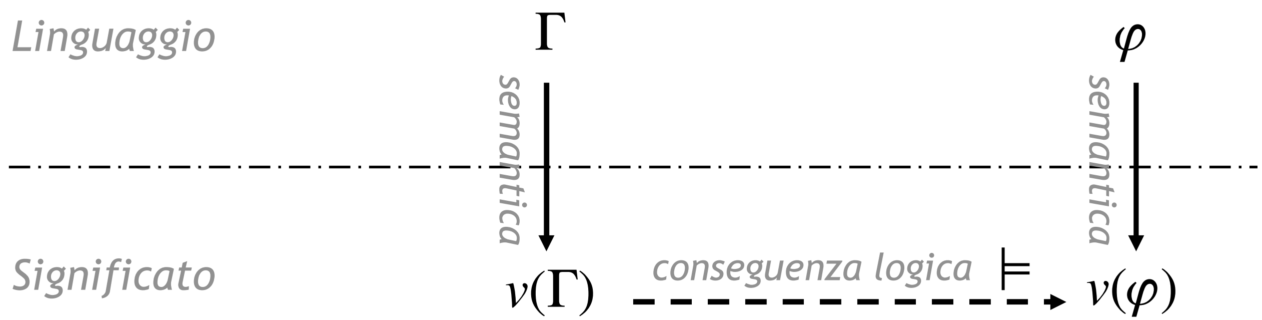
  "Il cacciatore sente una brezza quando si trova sul ciglio di una trappola"

Calcolo simbolico?

- Una fbf φ è **conseguenza logica** di un insieme di fbf Γ sse qualsiasi *modello* di Γ è anche *modello* di φ

  Si scrive anche:

  $\Gamma \models \varphi$

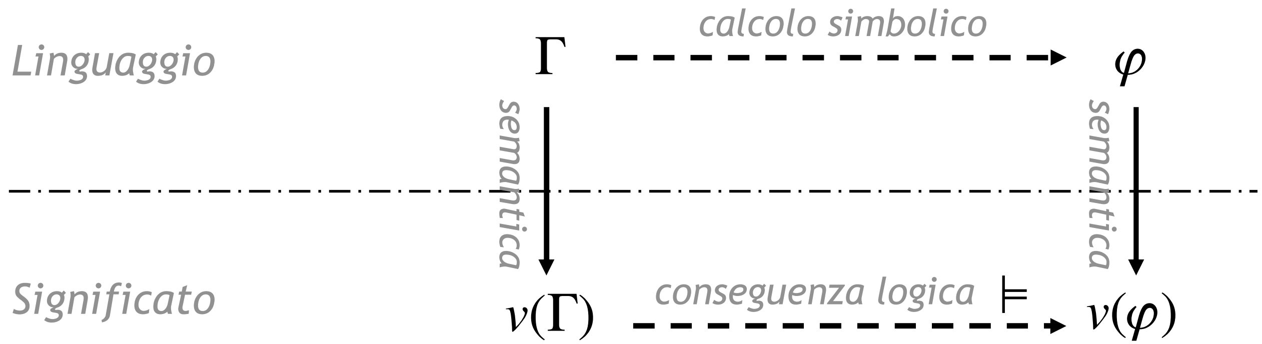


Calcolo simbolico?

- Una fbf φ è **conseguenza logica** di un insieme di fbf Γ sse qualsiasi *modello* di Γ è anche *modello* di φ

  Si scrive anche:

  $\Gamma \models \varphi$



  Si noti che la definizione fa riferimento a tutti i possibili significati (modelli)

- Calcolo simbolico

  Identificazione della relazione di conseguenza logica tramite operazioni (derivazioni) sul linguaggio delle fbf

Strutture proposizionali

Cosa vogliamo rappresentare

Il mondo descritto da una struttura $<\{0,1\}, \boldsymbol{P}, v>$

$\{0,1\}$ è l'insieme dei valori di verità

$\boldsymbol{P}$ è un'insieme di simboli proposizionali (**segnatura**)

v è una *funzione*: $\boldsymbol{P} \rightarrow \{0,1\}$ che assegna valori di verità ai simboli proposizionali

Simboli proposizionali

Ciascuno indica una frase affermativa (**proposizioni**)

Per convezione usiamo i simboli A, B, C, D, …

Mondi possibili

Possiamo considerare diverse strutture:

$<\{0,1\}, \boldsymbol{P}, v>$
$<\{0,1\}, \boldsymbol{P}, v'>$
$<\{0,1\}, \boldsymbol{P}, v''>$
…

Notare che le strutture condividono i simboli $\boldsymbol{P}$ e l'insieme dei valori di verità $\{0,1\}$

Differiscono solo per le funzioni v : i valori di verità assegnati sono in generale diversi

Strutture del primo ordine

Vogliamo rappresentare: un mondo di oggetti e di insiemi

Il mondo descritto da una struttura $<\mathbf{U}, \Sigma, v>$

$\mathbf{U}$ è un insieme di oggetti di base, detto anche ***universo del discorso*** o ***dominio*** (*domain*)

Σ è un'insieme di simboli, detto ***segnatura*** (*signature*)

v è una *funzione* che definisce il significato dei simboli di Σ in relazione al dominio $\mathbf{U}$

Segnatura Σ

- *costanti individuali*: $a, b, c, d, \ldots$
- *simboli funzionali*: $f, g, h, \ldots$

  Ciascuno di essi ha un numero di argomenti prestabilito, detto ***arità*** (*arity*)

  Si indica anche come f/a_f, g/a_g, h/a_h, ...

  Esempio: $f/2$ significa che f ha *arità* 2 (quindi si userà ed esempio come $f(a,b)$)
- *simboli predicativi* (o *relazionali*): $P, Q, R, \ldots$

  Ciascuno di essi ha un'***arità*** prestabilita

  Si indica anche come P/a_P, Q/a_Q, R/a_R, ...

  Esempio: $P/1$ significa che P ha *arità* 1 (quindi si userà ed esempio come $P(f(a,b))$)

Nota: in alcune definizioni le *costanti individuali* sono considerate funzioni ad arità 0

Strutture del primo ordine

Vogliamo rappresentare: un mondo di oggetti e di insiemi

Il mondo descritto da una struttura $<\mathbf{U}, \Sigma, v>$

$\mathbf{U}$ è un insieme di oggetti di base, detto anche ***universo del discorso*** o ***dominio*** (*domain*)

Σ è un'insieme di simboli, detto ***segnatura*** (*signature*)

v è una *funzione* che un significato ai simboli di Σ in relazione al dominio $\mathbf{U}$

Funzione v (interpretazione)

- L'interpretazione di una *costante individuale* è un *oggetto* di $\mathbf{U}$
  $v(a) = obj \in \mathbf{U}$ (a costante individuale)
- L'interpretazione di un *simbolo funzionale* è una *funzione* definita su $\mathbf{U}$
  $v(f/a_f) = fun : \mathbf{U} \rightarrow \mathbf{U}^{a_f}$ (f simbolo funzionale avente arità a_f)
- L'interpretazione di un *simbolo predicativo* è una *relazione* definita su $\mathbf{U}$
  $v(P/a_P) = rel \subseteq \mathbf{U}^{a_P}$ (P simbolo predicativo avente arità a_P)

Σ-atomi

Con i simboli di **Σ** *si possono formare le più piccole fbf del (nuovo) linguaggio:*

- **Σ-atomi**

  Un ***termine*** è una qualsiasi combinazione di costanti e funzioni che rispetti i vincoli di arità

  - Ogni *costante individuale* è un **termine**
  - Se f è un *simbolo funzionale* avente arità n e $t_1, \ldots, t_n$ sono **termini**, allora $f(t_1, \ldots, t_n)$ è un **termine**

    Più precisamente, questi sono i termini ***base*** o ***ground*** (vedi oltre)

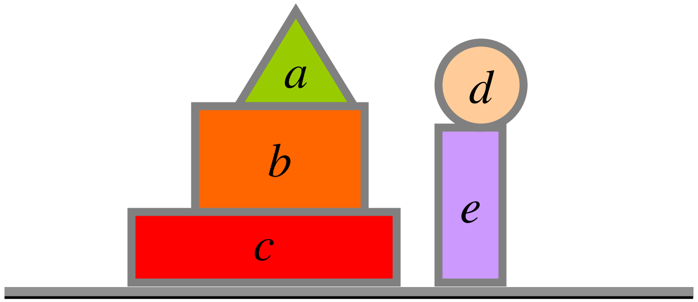
  Un ***atomo*** (o **Σ-atomo)**
  è una fbf formata da un solo simbolo predicativo e termini q.b., in base all'arità

  - Se P è un *simbolo predicativo* avente arità n e $t_1, \ldots, t_n$ sono **termini**, allora $P(t_1, \ldots, t_n)$ è un **atomo**

    Più precisamente, questi sono gli atomi ***base*** o ***ground*** (vedi oltre)

  Un ***atomo*** è la più piccola fbf di un linguaggio del primo ordine (*il nuovo linguaggio*)

Ditelo con gli atomi



- Esempio di struttura <**U**, Σ, *v*>

  Dominio **U**

    Insieme di oggetti: { *a*, *b*, *c*, *d*, *e*, *green*, *orange*, *red*, *rose*, *violet* }

  Segnatura Σ

    Costanti individuali: *a, b, c, d, e, green, orange, red, rose, violet*

    Simboli funzionali: *colorOf*/1

    Simboli predicativi: *Pyramid*/1, *Parallelepiped*/1, *Sphere*/1, *Ontable*/1, *Clear*/1, *Above*/2, =/2

Non confondere gli oggetti di **U** *con i simboli di* Σ

  Un'interpretazione *v* *soddisfa* un insieme di atomi

    Appartenenza a insiemi

      Pyramid(*a*)
      Parallelepiped(*b*), *Parallelepiped*(*c*), *Parallelepiped*(*e*)
      Sphere(*d*)
      Ontable(*c*), *Ontable*(*e*)
      Clear(*a*), *Clear*(*d*)

    Valori di funzioni

      (*colorOf*(*a*) = *green*), (*colorOf*(*b*) = *orange*), (*colorOf*(*c*) = *red*) , (*colorOf*(*d*) = *rose*) , (*colorOf*(*e*) = *violet*)

    Relazioni

      Above(*a*,*b*), *Above*(*b*,*c*), *Above*(*a*,*c*), *Above*(*d*,*e*)

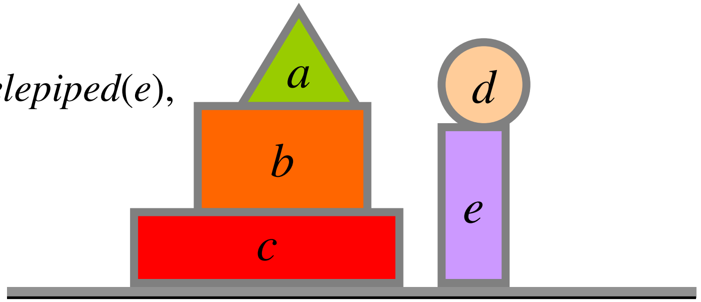
Ditelo con gli atomi

Diverse interpretazioni, stessa segnatura **Σ** *e dominio* **U**

v_1 *Pyramid(a), Parallelepiped(b), Parallelepiped(c), Sphere(d), Parallelepiped(e), Ontable(c), Ontable(e), Clear(a), Clear(d)*

(colorOf(a) = green), (colorOf(b) = orange), (colorOf(c) = red) , (colorOf(d) = rose) , (colorOf(e) = violet)

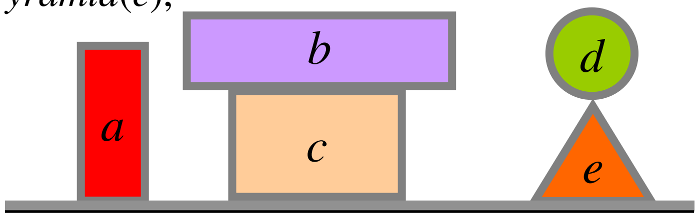
Above(a,b), Above(b,c), Above(a,c), Above(d,e)



v_2 *Parallelepiped(a), Parallelepiped(b), Parallelepiped(c), Sphere(d), Pyramid(e), Ontable(a), Ontable(c), Ontable(e), Clear(a), , Clear(b), Clear(d)*

(colorOf(a) = red), (colorOf(b) = violet), (colorOf(c) = pink) , (colorOf(d) = green) , (colorOf(e) = orange)

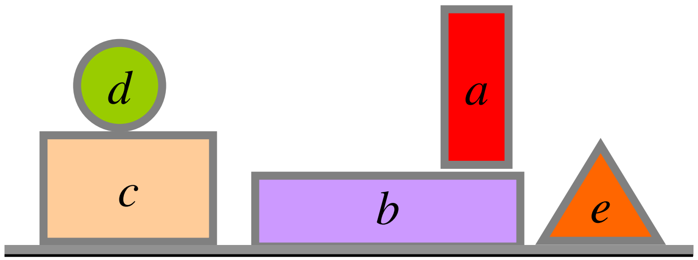
Above(b,c), Above(d,e)



v_3 *Pyramid(a), Parallelepiped(b), Parallelepiped(c), Parallelepiped(e) Sphere(d), Ontable(c), Ontable(e), Clear(a), Clear(d)*

(colorOf(a) = green), (colorOf(b) = orange), (colorOf(c) = red) , (colorOf(d) = rose) , (colorOf(e) = violet)

Above(a,b), Above(b,c), Above(a,c), Above(d,e)



Logica del primo ordine *(prima approssimazione: fbf base - o ground)*

- Linguaggio

  Segnatura Σ + connettivi logici $\neg, \rightarrow, \wedge, \vee, \leftrightarrow$

  Più le parentesi (,)

  Regole di buona formazione: atomi e combinazione di atomi tramite connettivi

- Significato

  Strutture $<\mathbf{U}, \Sigma, v>$

  Notare che $\boldsymbol{\Sigma}$ è fissato (corrisponde al linguaggio), ma $\mathbf{U}$ e $\boldsymbol{v}$ possono essere qualsiasi

  Soddisfacimento

  $<\mathbf{U}, \Sigma, v> \models \varphi$ (φ è una fbf di cui sopra)
  sse φ descrive un fatto *vero* in $<\mathbf{U}, \Sigma, v>$

  Più precisamente:

  Se φ è un atomo $P(t_1, \ldots, t_n)$
  $<\mathbf{U}, \Sigma, v> \models \varphi$ sse $<v(t_1), \ldots, v(t_n)> \in v(P)$

  Se φ e ψ sono fbf qualsiasi

  $<\mathbf{U}, \Sigma, v> \models (\neg\varphi)$ sse $<\mathbf{U}, \Sigma, v> \not\models \varphi$

  $<\mathbf{U}, \Sigma, v> \models (\varphi \wedge \psi)$ sse $<\mathbf{U}, \Sigma, v> \models \varphi$ e $<\mathbf{U}, \Sigma, v> \models \psi$

  $<\mathbf{U}, \Sigma, v> \models (\varphi \vee \psi)$ sse $<\mathbf{U}, \Sigma, v> \models \varphi$ o $<\mathbf{U}, \Sigma, v> \models \psi$

  $<\mathbf{U}, \Sigma, v> \models (\varphi \rightarrow \psi)$ allora non $<\mathbf{U}, \Sigma, v> \models \varphi$ e $<\mathbf{U}, \Sigma, v> \not\models \psi$

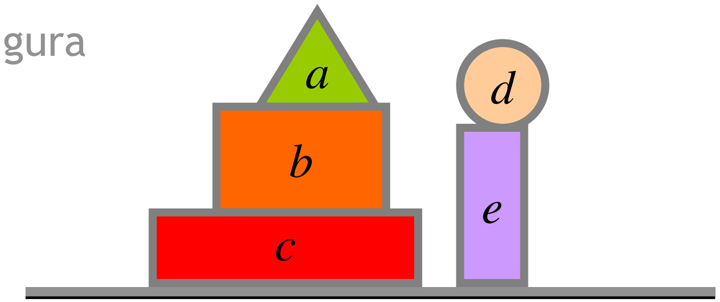
  Regole identiche al caso proposizionale

Esempi

- Struttura $<\mathbf{U}, \Sigma, v>$

  La stessa dell'esempio precedente, v come descritta in figura

  Si assume $v(a) = \underline{a}$, $v(b) = \underline{b}$, $v(c) = \underline{c}$ $\ldots$



- Soddisfacimento

  $<\mathbf{U}, \Sigma, v> \models Pyramid(a)$
  perchè $v(a) \in v(Pyramid/1)$

  $<\mathbf{U}, \Sigma, v> \not\models Parallelepiped(d)$
  perchè $v(d) \notin v(Parallelepiped/1)$

  $<\mathbf{U}, \Sigma, v> \models Above(a, b)$
  perchè $<v(a), v(b)> \in v(Above/2)$

  $<\mathbf{U}, \Sigma, v> \models \neg Above(c,d)$
  perchè $<v(c), v(d)> \notin v(Above/2)$

  $<\mathbf{U}, \Sigma, v> \not\models Pyramid(a) \wedge Sphere(a)$
  perchè $<\mathbf{U}, \Sigma, v> \models Pyramid(a)$ ma $<\mathbf{U}, \Sigma, v> \not\models Sphere(a)$

  $<\mathbf{U}, \Sigma, v> \models Above(a,b) \vee Above(b,a)$
  perchè $<\mathbf{U}, \Sigma, v> \models Above(a,b)$ anche se $<\mathbf{U}, \Sigma, v> \not\models Above(b,a)$

Cosa abbiamo ottenuto *(con questa prima approssimazione)*

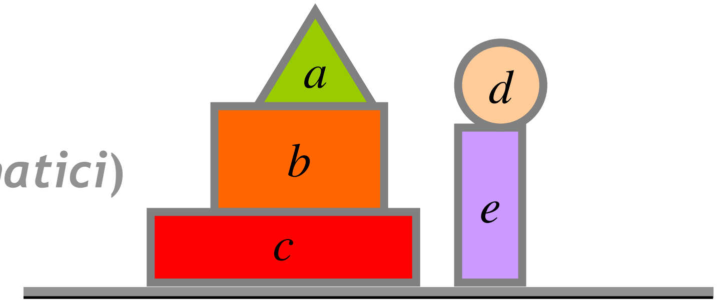
Una logica proposizionale, in forma più elaborata

- Linguaggio
  - Ciascun atomo base può essere descritto da un unico simbolo proposizionale
  - Le fbf hanno le stesse caratteristiche
- Significato
  - $<\mathbf{U}, \Sigma, v> \models \varphi$ in sintesi, sse φ è <u>vera</u> in $<\mathbf{U}, \Sigma, v>$
- Tutto il resto
  - Conseguenza logica
    - $\Gamma \models \varphi$ sse tutte le strutture $<\mathbf{U}, \Sigma, v>$ che soddisfano Γ soddisfano anche φ
      - Notare che sono $\mathbf{U}$ e v a poter variare
  - Derivazioni, derivabilità
    - Come nel caso proposizionale
  - Correttezza, completezza
    - Come nel caso proposizionale
  - Metodi di calcolo automatico
    - Come nel caso proposizionale

Cosa abbiamo ottenuto *(con questa prima approssimazione)*

Una logica proposizionale, in forma più elaborata

- Linguaggio
    - Ciascun atomo base può essere descritto da un unico simbolo proposizionale
    - Le fbf hanno le stesse caratteristiche
- Significato
    - $<\mathbf{U}, \Sigma, v> \models \varphi$ in sintesi, sse φ è vera in $<\mathbf{U}, \Sigma, v>$
- Nota:
    - Esempio di segnatura $\mathbf{\Sigma}$:
        - Costanti: 0, simboli funzionali: $s/1$, simboli predicativi: $+/2$, $\cdot/2$, $=/2$
        - I termini di una segnatura possono essere infiniti: $0, s(0), s(s(0)), s(s(s(0))), \ldots$
            - Basta un solo simbolo funzionale a produrre quest'effetto
        - Il dominio $\mathbf{U}$ può essere infinito
            - Ad esempio $\mathbf{U} \equiv \mathbf{N}$, l'insieme dei numeri naturali
            - *La segnatura di questo esempio è sufficiente per descrivere tutti i fatti specifici dell'aritmetica (teoria dei numeri naturali)*



"*Many-sorted or nil*" (*dedicato agli informatici*)

Tornando alla segnatura dell'esempio mondo dei blocchi

$green$, $colorOf(green)$, $colorOf(colorOf(green))$, $colorOf(colorOf(colorOf(green)))$, …

Sono tutti termini sintatticamente corretti, in base alla definizione.
Peccato che, intuitivamente, non abbiano senso: un colore non è un oggetto …

Per le applicazioni pratiche, le segnature dovrebbero avere un *tipo* (*sort*)

Per descrivere un dominio che contiene oggetti di tipo diverso (segnatura *many-sorted*)

Il tipo si applica alle costanti ed agli argomenti di simboli funzionali e predicativi
Complicazione notevole: si riflette in tutte le definizioni sintattiche e semantiche

Comodità del nil

Una costante particolare: nil
cui corrisponde un interpretazione (canonica) di un non-oggetto

In questo modo, funzioni e relazioni possono essere definite in modo parziale:

$(colorOf(a) = green) \wedge (colorOf(green) = nil)$

$Above(a,b) \wedge Above(c,nil)$

Possono essere fbf vere in una struttura

In questo modo, si evita l'uso esplicito del tipo

Logica del del primo ordine

- La differenza rispetto a L_P
  - Possibilità di definire formule astratte: *variabili* e *quantificatori*
- Linguaggio di L_{PO}
  - Si introducono:
    - Variabili: $x, y, z, \ldots$
    - Quantificatori: ∀ (universale), ∃ (esistenziale)
  - Regole di buona formazione:
    - Estensione della definizione di *termine*
      - Ogni *costante individuale* o *variabile* è un **termine**
      - Se f è un *simbolo funzionale* avente arità n e $t_1, \ldots, t_n$ sono **termini**, allora $f(t_1, \ldots, t_n)$ è un **termine**
    - Nuove regole per le fbf contenenti quantificatori

$$\varphi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\forall x\, \varphi) \in \text{fbf}(L_{PO})$$

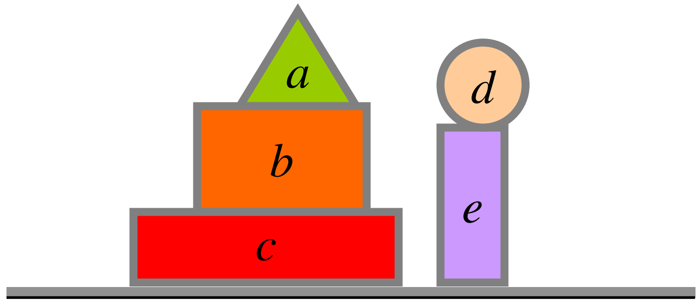
$$\varphi \in \text{fbf}(L_{PO}) \quad (\exists x\, \varphi) \;\Leftrightarrow\; (\neg \forall x\, \neg \varphi)$$

Notare che il quantificatore esistenziale è derivato, basterebbe quello universale

- Semantica di L_{PO}
  - (vedi oltre)

Ditelo con le fbf di L_{PO}

(*semantica intuitiva, per ora*)



- Proprietà di carattere generale

  $\neg\forall x\ \exists y\ (Above(x,y))$

  $\neg\forall y\ \exists x\ (Above(x,y))$

- Definizioni di nuovi predicati

  $\forall x \forall y\ (On(x,y) \leftrightarrow (Above(x,y)\ \wedge\ \neg\exists z\ (Above(x,z)\ \wedge\ Above(z,y)))$

  $\forall x\ (Ontable(x) \leftrightarrow \neg\exists z\ Above(x,z))$

  $\forall x\ (Clear(x) \leftrightarrow \neg\exists z\ Above(z,x))$

Ditelo con le fbf di L_{PO}

- "Essere fratelli significa essere parenti"
  $\forall x \forall y\ (Fratello(x, y) \rightarrow Parente(x, y))$

- "La relazione di parentela è simmetrica"
  $\forall x \forall y\ (Parente(x ,y) \leftrightarrow Parente(y, x))$

- "Una madre è un genitore di sesso femminile"
  $\forall x \forall y\ (Madre(x, y) \leftrightarrow (Genitore(x, y) \land Femmina(x)))$

- "Un cugino è figlio di un fratello o una sorella di uno dei genitori"
  $\forall x \forall y\ (Cugino(x, y) \leftrightarrow \exists z \exists w\ (Genitore(z, x) \land Genitore(w, y) \land (Fratello(z, w) \lor Sorella(z, w))))$

- "Ciascuno ha una madre"
  $\forall x \exists y\ Madre(y, x)$
  Occorre fare attenzione all'ordine dei quantificatori:
  $\exists y \forall x\ Madre(y, x)$
  "Esiste una madre di tutti"
  L'ordine dei quantificatori non può essere modificato senza alterare il significato

Logica del primo ordine

Linguaggio di L_{PO}

- Simboli del linguaggio L_{PO}

  Costanti individuali

  Esempi: *Socrate*, *Sfera1*, *MickeyMouse*, *Amelia*, ...

  In generale: a, b, c, …

  Variabili

  In generale: x, y, z, …

  Simboli funzionali con numero di argomenti prestabilito

  Esempi: *sqrt*/1, *colorOf*/1, *greatestCommonDivisor*/2

  In generale: f/n, g/m, h/p, …

  Simboli predicativi con numero di argomenti prestabilito

  Esempi: *Red*/1, *Large*/1, *GreaterThan*/2, =/2

  In generale: P/n, Q/m, R/p, …

  Connettivi

  Come in logica proposizionale: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$

  Quantificatori

  $\forall$ (universale), $\exists$ (esistenziale)

  *Notazione: ciascun quantificatore si applica **sempre** ad una ed una sola variabile*

  Parentesi e virgola

  ( ) ,

Linguaggio di L_{PO}

- **Termini**

  Ogni *variabile* o *costante individuale* è un **termine**

  Se f è un *simbolo funzionale* a n argomenti e $t_1, \ldots, t_n$ sono **termini**, allora $f(t_1, \ldots, t_n)$ è un **termine**

  Un termine **base** (*ground*) non contiene variabili

- **Atomo o formula atomica**

  Se P è un *simbolo predicativo* a *n* argomenti e $t_1, \ldots, t_n$ sono **termini**, allora $P(t_1, \ldots, t_n)$ è un **atomo** o **formula atomica**

  Un atomo **base** (*ground*) non contiene variabili

Esempi:

$$\underbrace{\underbrace{Sorella}_{\text{predicato}}(\underbrace{\underbrace{Amelia}_{\text{costante}}}_{\text{termine}}, \underbrace{\underbrace{Alba}_{\text{costante}}}_{\text{termine}})}_{\text{formula atomica}}$$

$$\underbrace{\underbrace{shapeOf(\underset{\text{costante}}{b})}_{\text{termine}} \underset{\text{predicato}}{=} \underbrace{shapeOf(\underset{\text{variabile}}{x})}_{\text{termine}}}_{\text{formula atomica}}$$

Linguaggio di L_{PO}

- Regole di buona formazione

Ogni *formula atomica* è una fbf

$$\varphi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\neg\varphi) \in \text{fbf}(L_{PO})$$
$$\varphi, \psi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\varphi \to \psi) \in \text{fbf}(L_{PO})$$
$$\varphi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\forall x\, \varphi) \in \text{fbf}(L_{PO})$$

$$\varphi, \psi \in \text{fbf}(L_{PO}), \qquad (\varphi \lor \psi) \;\Leftrightarrow\; ((\neg\varphi) \to \psi)$$
$$\varphi, \psi \in \text{fbf}(L_{PO}), \qquad (\varphi \land \psi) \;\Leftrightarrow\; (\neg(\varphi \to (\neg\psi)))$$
$$\varphi, \psi \in \text{fbf}(L_{PO}), \qquad (\varphi \leftrightarrow \psi) \;\Leftrightarrow\; ((\varphi \to \psi) \land (\psi \to \varphi))$$
$$\varphi \in \text{fbf}(L_{PO}) \qquad (\exists x\, \varphi) \Leftrightarrow (\neg\forall x\, \neg\varphi)$$

Si dice linguaggio **del primo ordine** in quanto
i quantificatori agiscono solo sugli **oggetti**,
vale a dire sulle variabili x, y, z …, e non sulle **relazioni** e **funzioni**
(*In una logica del secondo ordine si hanno formule del tipo*: $\exists F\, F(a,b)$)

Formule aperte, enunciati

- Variabili **libere** e **vincolate**

  Una variabile (in una fbf) è **vincolata** se si trova nel raggio di azione di un **quantificatore** per quella variabile

  Una variabile è **libera** se non è *vincolata*

  esempi di variabile vincolata:
  $\forall x\ P(x)$
  $\exists x\ (P(x) \rightarrow (A(x) \wedge B(x))$

  esempi di variabile libera:
  $P(x)$
  $\exists y\ (P(y) \rightarrow (A(x,y) \wedge B(y)))$

- **Formule aperte** e **chiuse**

  Una fbf è **aperta** se in essa vi è almeno una variabile libera

  Una fbf è **chiusa** (anche **enunciato** - *sentence*) in caso contrario

  Solo le fbf *chiuse*, cioè gli *enunciati*, hanno un valore di verità (vedi oltre)
  (in quanto rappresentano delle *affermazioni* ...)

Strutture, interpretazioni e assegnazioni

Si omette da ora in poi il riferimento a Σ

- Una **struttura** $<\mathbf{U}, v>$ per L_{PO} contiene:
  - Un insieme di oggetti $\mathbf{U}$ (l'universo del discorso)
  - Un'**interpretazione** v che associa
    - ad ogni **costante** c un **oggetto** di $\mathbf{U}$
      $v(c) \in \mathbf{U}$
    - ad ogni **predicato** P a n argomenti una **relazione** n-aria in $\mathbf{U}^n$
      $v(P) \subseteq \mathbf{U}^n$
    - ad ogni **funzione** f a n argomenti una **funzione** da $\mathbf{U}^n$ a $\mathbf{U}$
      $v(f) \subseteq \mathbf{U}^n \rightarrow \mathbf{U}$

    La funzione v non assegna un significato alle variabili

- Assegnazione
  - Data una struttura $<\mathbf{U}, v>$, un'**assegnazione** (*valuation*) s
    - è una *funzione* che associa ad ogni **variabile** x un **oggetto** di $\mathbf{U}$
      $s(x) \in \mathbf{U}$
    - La combinazione di una $<\mathbf{U}, v>$ e di una s determina univocamente gli oggetti associati a ciascun elemento di L_{PO}

Soddisfacimento

- Data una struttura $<\mathbf{U}, v>$ un'assegnazione s

  Se φ è una formula atomica, $<\mathbf{U}, v> [s] \models \varphi$ sse

  se φ ha la forma $P(t_1, \dots, t_n)$ allora $<v(t_1) [s], \dots, v(t_n) [s]> \in v(P) [s]$

  Se φ e ψ sono fbf qualsiasi

  $<\mathbf{U}, v> [s] \models (\neg\varphi)$ sse $<\mathbf{U}, v> [s] \not\models \varphi$

  $<\mathbf{U}, v> [s] \models (\varphi \wedge \psi)$ sse $<\mathbf{U}, v> [s] \models \varphi$ e $<\mathbf{U}, v> [s] \models \psi$

  $<\mathbf{U}, v> [s] \models (\varphi \vee \psi)$ sse $<\mathbf{U}, v> [s] \models \varphi$ o $<\mathbf{U}, v> [s] \models \psi$

  $<\mathbf{U}, v> [s] \models (\varphi \rightarrow \psi)$ allora non $<\mathbf{U}, v> [s] \models \varphi$ e $<\mathbf{U}, v> [s] \not\models \psi$

  Formule con quantificatori

  $<\mathbf{U}, v> [s] \models \forall x\, \varphi$ sse per ogni $\underline{d} \in \mathbf{U}$ si ha $<\mathbf{U}, v> [s](x:\underline{d}) \models \varphi$

  $<\mathbf{U}, v> [s] \models \exists x\, \varphi$ sse esiste un $\underline{d} \in \mathbf{U}$ per cui si ha $<\mathbf{U}, v> [s](x:\underline{d}) \models \varphi$

L'assegnazione s serve a poter definire una semantica anche per le fbf <u>aperte</u>

Modelli

- **Validità** in un'interpretazione, **modello**

  Una fbf φ tale per cui si ha $<\mathbf{U}, v> [s] \models \varphi$ per qualsiasi assegnazione s è detta **valida** in $<\mathbf{U}, v>$

  Si dice anche che $<\mathbf{U}, v>$ è un **modello** di φ

  si scrive $<\mathbf{U}, v> \models \varphi$ (si elimina il riferimento a s)

  Una struttura $<\mathbf{U}, v>$ è detta **modello** di un *insieme di fbf* Γ sse è un modello di tutte le fbf in Γ

  si scrive allora $<\mathbf{U}, v> \models \Gamma$

- **Verità**

  Un **enunciato** ψ si dice **vero** in $<\mathbf{U}, v>$ se è **valido** in $<\mathbf{U}, v>$

  per un enunciato, basta l'esistenza di un'assegnazione s per cui $<\mathbf{U}, v> [s] \models \psi$

Validità

- Validità e verità logiche

  Una fbf (aperta o chiusa) è **valida** (o **logicamente valida**) se è **valida** in qualsiasi $<\mathbf{U}, v>$

  Esempi:

  $(P(x) \vee \neg P(x))$ (tautologia come formula aperta)

  Un enunciato ψ è **vero** (o **logicamente vero**) se è **vero** in qualsiasi $<\mathbf{U}, v>$

  si scrive allora $\models \psi$ (si elimina il riferimento a $<\mathbf{U}, v>$)

  Esempi:

  $\forall x\, (P(x) \vee \neg P(x))$ (generalizzazione di una tautologia)

  $\forall x \forall y\, (G(x,y) \rightarrow (H(x,y) \rightarrow G(x,y)))$ (generalizzazione di assioma – vedi oltre)

- Inconsistenza

  Una fbf (aperta o chiusa) è **inconsistente** se non è soddisfacibile

  Un enunciato ψ è **inconsistente** se non ha un *modello*

  Esempi:

  $\forall x\, (P(x) \wedge \neg P(x))$ (generalizzazione di una contraddizione)

Conseguenza logica

- Definizione

  Dato un insieme di fbf Γ ed una fbf φ di L_{PO} si ha

  $$\Gamma \models \varphi$$

  sse tutte le $<\mathbf{U}, v>[s]$ che soddisfano Γ soddisfano anche φ

- Osservazioni

  La definizione si estende a tutte le possibili $<\mathbf{U}, v>[s]$

  Quindi, a tutti i possibili insiemi $\mathbf{U}$, alle relazioni e funzioni in $\mathbf{U}$ ed alle associazioni di oggetti di $\mathbf{U}$ a variabili e costanti

  Il calcolo diretto della conseguenza logica in L_{PO} è impossibile anche nelle forme più semplici

FAQ 1

- Funzioni o predicati (i.e. relazioni)?

  I due oggetti semantici sono molto simili, si può fare a meno delle funzioni?

- Le funzioni (come oggetti semantici) si possono *rappresentare* anche tramite predicati

  ad esempio, la validità dell'enunciato:

  $$\forall x \forall y \forall z\ ((\varphi(x,y)\ \wedge\ \varphi(x,z)) \rightarrow (y = z))$$

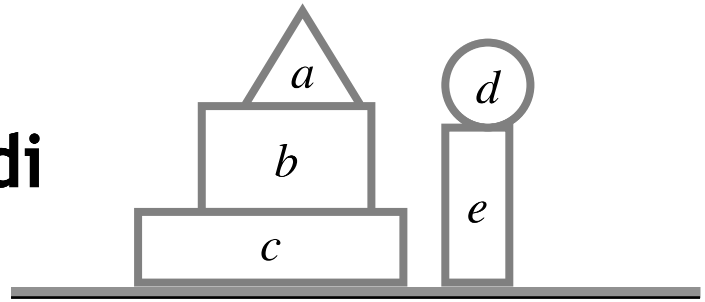
  indica che l'interpretazione di $\varphi(..)$ (in generale, una relazione $v(\varphi) \subseteq \mathbf{U}^2$)
  è anche una funzione $\mathbf{U} \rightarrow \mathbf{U}$

  La presenza delle funzioni arricchisce il linguaggio in modo sostanziale:
  a differenza dei predicati, le funzioni si possono nidificare (nei termini)

  Di conseguenza, viene grandemente aumentata la portata del calcolo logico-simbolico

  (*con un corrispondente aumento della complessità di calcolo* ...)

Assiomatizzare una famiglia di mondi



Il mondo dei blocchi

- Assiomi Ax_{bw} (Cook & Liu, 2002)
  1) $\forall x\, \neg Above(x,x)$
  2) $\forall x \forall y \forall z\, ((Above(x,y) \wedge Above(y,z)) \rightarrow Above(x,z))$
  3) $\forall x \forall y \forall z\, ((Above(x,y) \wedge Above(x,z)) \rightarrow ((y=z) \vee Above(y,z) \vee Above(z,y))$
  4) $\forall x \forall y \forall z\, ((Above(y,x) \wedge Above(z,x)) \rightarrow ((y=z) \vee Above(y,z) \vee Above(z,y))$
  5) $\forall x\, (Ontable(x) \vee \exists y\, (Above(x,y) \wedge Ontable(y)))$
  6) $\forall x\, (Clear(x) \vee \exists y\, (Above(y,x) \wedge Clear(y)))$
  7) $\forall x \forall y\, (Above(x,y) \rightarrow (\exists z\, On(x,z) \wedge \exists w\, On(w,y)))$

  Gli assiomi descrivono le caratteristiche generali del mondo (dei blocchi)
  - Si assumono veri in tutti i mondi (dei blocchi) possibili
    - Indipendentemente dalla disposizione effettiva

Assiomatizzare una famiglia di mondi

Il mondo delle liste di oggetti $[a, b, c, \ldots]$

$cons(s, x)$

funzione, associa ad un oggetto (es. a) ed una lista (es. $[b, c]$) la lista ottenuta inserendo l'oggetto all'inizio (es. $[a, b, c]$)_

$Append(x,y,z)$

predicato, associa alle liste x e y la concatenazione z

nil

costante, indica la lista vuota.

Notazione abbreviata:

$[] \Leftrightarrow nil$
$[a] \Leftrightarrow cons(a,nil)$
$[a,b] \Leftrightarrow cons(a,cons(b,nil))$
$[a|[b,c]] \Leftrightarrow cons(a,[b,c])$

- Assiomi (**AL**)

1) $\forall x\ Append(nil,x,x)$
2) $\forall x \forall y \forall z\ (Append(x,y,z) \rightarrow \forall s\ Append(cons(s,x),y,cons(s,z)))$

Esempi (conseguenze logiche)

$\mathbf{AL} + \exists z\ Append([a],[b,c],z) \models Append([a],[b,c],[a,b,c])$ = $[z/[a,b,c]]$

$\mathbf{AL} + \exists x \exists y\ Append(x,y,[a,b]) \models Append([a],[b],[a,b])$ = $[x/[a], x/[b]]$

$\models Append(nil,[a,b],[a,b])$ = $[x/nil, y/[a,b]]$

$\models Append([a,b],nil,[a,b])$ = $[x/[a,b], y/nil]$

(Semi)decidibilità

Decidibilità ed automazione di L_{PO}

- Indecidibilità di L_{PO}

  Non esiste un algoritmo (di valore generale) in grado di stabilire se $\Gamma \models \varphi$

  Al contrario del caso proposizionale, in L_{PO} non si possono verificare direttamente tutte le possibili interpretazioni

  Qual è quindi la speranza di avere un calcolo automatico?

- In realtà, L_{PO} è **semi-decidibile** (Herbrand, 1930)

  E\` possibile stabilire (in tempo finito) se

  $\Gamma \models \varphi$

  ... ma <u>non</u> se

  $\Gamma \not\models \varphi$

- In altri termini, possono esistere metodi che:

  Di fronte al problema "$\Gamma \models \varphi$?"

  a) Terminano con successo se $\Gamma \models \varphi$

  b) Possono divergere (i.e. girare all'infinito) se $\Gamma \not\models \varphi$

Universo e base di Herbrand

- **Termini** e **atomi** di Herbrand

  Dato un linguaggio L_{PO}

  Un **termine** di Herbrand è un *termine base* (*ground term* = che non contiene variabili)

  Esempi:

  $f(a)$, $g(a,b)$, $g(f(a),b)$, $g(f(a),g(b,c))$, $g(f(a),g(f(b),c))$, …

  Un **atomo** di Herbrand è un *atomo base* (*ground atom* = che non contiene variabili)

  Esempi:

  $P(f(a))$, $P(g(a,b))$, $Q(g(f(a),b), g(f(a),g(b,c)))$, …

- **Universo** e **base** di Herbrand

  L'**universo** di Herbrand è l'insieme di tutti i termini di Herbrand

  Esempio:

  $\mathbf{U}_H \equiv \{f(a), g(a,b), g(f(a),b), g(f(a),g(b,c)), g(f(a),g(f(b),c)), \ldots\}$

  La **base** di Herbrand è l'insieme di tutti gli atomi di Herbrand

  Esempio:

  $B_H \equiv \{P(f(a)), P(g(a,b)), Q(g(f(a),b), g(f(a),g(b,c))), \ldots\}$

Modelli di Herbrand

- Struttura di Herbrand per L_{PO}

  Una struttura $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}>$ tale che

  $\forall c \in \mathrm{Cost}(L_{PO}),\ v_\mathrm{H}(c) = c$

  $\forall t \in \mathbf{U}_\mathrm{H},\ v_\mathrm{H}(t) = t$

- Interpretazione v_H di Herbrand

  Un qualsiasi sottoinsieme della base di Herbrand B_H

  $v_\mathrm{H} \equiv \{P(a), P(f(b)), P(c), Q(a,g(b,c)), Q(b,c) \ldots \}$ (solo formule atomiche chiuse)

  $v_\mathrm{H} \subseteq \mathrm{B}_\mathrm{H}$

- Modello di Herbrand

  $\varphi \in \mathrm{Atom}(L_{PO}),\ <\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \models \varphi$ sse $\varphi \in v_\mathrm{H}$

  $\varphi \in \mathrm{Atom}(L_{PO}),\ <\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \models \neg\varphi$ sse $\varphi \notin v_\mathrm{H}$

  $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \models \neg\varphi$ sse $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \not\models \varphi$

  $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \models \varphi \to \psi$ sse non $(<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \models \varphi$ e $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \not\models \psi)$

  $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s] \models \forall x\, \varphi$ se per ogni $c \in \mathrm{Cost}(L_{PO})$ si ha $<\mathbf{U}_\mathrm{H}, v_\mathrm{H}> [s](x{:}c) \models \varphi$

Sistemi di Herbrand

- **Sistema** di Herbrand di un enunciato universale

  Dato un enunciato universale, della forma

  $\forall x_1 \forall x_2 \ldots \forall x_n \varphi$ (φ non contiene quantificatori)

  il Sistema di Herbrand è l'insieme (anche infinito) di formule chiuse generato per sostituzione

  $\varphi[x_1/t_1, x_2/t_2 \ldots x_n/t_n]$

  con <u>tutte</u> le possibili combinazioni $<t_1, t_2 \ldots t_n>$ di $t_i \in \mathbf{U}_\mathrm{H}$

  Esempi:

  $\mathrm{H}(\forall x\, P(x) \to Q(x))) = \{P(f(a)) \to Q(f(a)), P(g(a, b)) \to Q(g(a, b)), \ldots \}$

  $\mathrm{H}(\forall x\, \forall y\, R(x, y)) = \{R(f(a), f(a)), R(g(a, b), f(a)), R(f(a), g(a, b)), \ldots \}$

- **Sistema** di Herbrand di una teoria

  Data una teoria Σ di enunciati universali
  è l'unione $\mathrm{H}(\Sigma)$ di tutti i sistemi di Herbrand generati dagli enunciati Σ

  Esempio:

  $\Sigma = \{\varphi, \psi, \chi\}$

  $\mathrm{H}(\Sigma) = \mathrm{H}(\psi) \cup \mathrm{H}(\varphi) \cup \mathrm{H}(\chi)$

Teorema di Herbrand

- **Teorema di Herbrand**

  Data una teoria di enunciati universali Σ,
  $H(\Sigma)$ ha un modello sse Σ ha un modello

  ... ma qual'è l'utilità?

  - $H(\Sigma)$ può essere infinito anche quando Σ è finito
  - il teorema si applica solo agli enunciati universali

Forma normale prenessa

- Forma normale **prenessa** (FNP) (= tutti i quantificatori all'inizio)

Una fbf φ qualsiasi può essere trasformata in un enunciato equivalente

$Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \psi$ (ψ è anche detta ***matrice***)

dove Q_i è $\forall$ oppure $\exists$ e ψ non contiene quantificatori

Si ottiene da φ eliminando tutte le negazioni $\neg$ davanti ai quantificatori tramite la definizione $\neg\forall x \equiv \exists x \neg$ e le equivalenze derivanti

$(\alpha \to \forall x\, \beta) \equiv \forall x\, (\alpha \to \beta)$ dato che $(\alpha \to \forall x\, \beta) \equiv (\neg\alpha \vee \forall x\, \beta) \equiv \forall x\, (\neg\alpha \vee \beta)$

$(\forall x\, \alpha \to \beta) \equiv \exists x\, (\alpha \to \beta)$ dato che $(\forall x\, \alpha \to \beta) \equiv (\neg\forall x\, \alpha \vee \beta) \equiv (\exists x\, \neg\alpha \vee \beta) \equiv \exists x\, (\neg\alpha \vee \beta)$

E\` necessario *ridenominare* le variabili per evitare sovrapposizioni

Esempio: $\exists y\, (P(y) \to \forall x\, P(x))$

$\exists y \forall x\, (P(y) \to P(x))$ (FNP, usando $(\varphi \to \forall x\, \psi) \leftrightarrow \forall x\, (\varphi \to \psi)$)

Esempio: $\exists y\, (\forall x\, P(x) \to P(y))$

$\exists y \exists x\, (P(x) \to P(y))$ (FNP, usando $(\forall x\, \varphi \to \psi) \leftrightarrow \exists x\, (\varphi \to \psi)$)

Esempio: $\forall x \exists y\, (Q(x,y) \to P(y)) \wedge \neg\forall x\, P(x)$

$\forall x \exists y\, (Q(x,y) \to P(y)) \wedge \exists x\, \neg P(x)$ (definizione $\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$)

$\forall x \exists y\, (Q(x,y) \to P(y)) \wedge \exists z\, \neg P(z)$ (*ridenominazione* di x in z)

$\forall x \exists y \exists z\, ((Q(x,y) \to P(y)) \wedge \neg P(z))$ (FNP)

Forma di Skolem

- Forma di Skolem

  Si eliminano i quantificatori esistenziali in una forma normale prenessa sostituendo ciascuna variabile esistenzialmente quantificata con una (nuova) costante o una (nuova) funzione:

  Si opera sui quantificatori della formula $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \psi$ partendo da sinistra

  Per ogni $Q_i x_i$ della forma $\exists x_i$:

  - si applica a ψ la sostituzione $[x_i/k_i(x_1, \ldots, x_{i-1})]$ (nuova funzione di Skolem) oppure $[x_i/k_i]$ (nuova costante di Skolem), se $i = 1$
  - si elimina $\exists x_i$ dalla formula

  Esempi:

  $\exists y \forall x\, (P(y) \rightarrow P(x))$

  $\forall x\, (P(k) \rightarrow P(x))$ (k costante di Skolem: si espande il linguaggio)

  $\forall x \exists y \exists z\, ((Q(x,y) \rightarrow P(y)) \land \neg P(z))$

  $\forall x\, ((Q(x, k(x)) \rightarrow P(k(x))) \land \neg P(j(x)))$ ($k/1$ e $j/1$ funzioni di Skolem)

- Teorema

  Per qualsiasi enunciato φ,

  φ ha un modello sse $sko(\varphi)$ (forma di Skolem di φ) ha un modello

Semi-decidibilità di L_{PO}

- Corollario del teorema di Herbrand

  Le seguenti affermazioni sono equivalenti
  - $\Gamma \models \varphi$
  - $\Gamma \cup \{\neg\varphi\}$ non è soddisfacibile (= non ha un modello) (= è inconsistente)
  - Esiste un sottoinsieme **finito** di $\mathrm{H}(sko(\Gamma \cup \{\neg\varphi\}))$ (sistema di Herbrand della forma di Skolem) che è **contraddittorio**

  Quindi:

  Una procedura che enumera *tutti* i sottoinsiemi finiti di $\mathrm{H}(sko(\Gamma \cup \{\neg\varphi\}))$ prima o poi (*in un tempo finito*) trova una contraddizione, sse $\Gamma \models \varphi$

Teorema di Herbrand e clausole di Horn

- **Teorema di Herbrand** **(in forma generale)**

  Data una teoria di enunciati universali Σ,
  $H(\Sigma)$ ha un modello sse Σ ha un modello

- Corollario (forma a clausole di Horn)

  Sia Γ un insieme di clausole di Horn, le seguenti affermazioni sono equivalenti:
  - Γ è soddisfacibile
  - Γ ha un modello di Herbrand

    (Notare: si afferma che Γ ha un modello di Herbrand, non $H(\Gamma)$)

  <u>Non</u> vale in generale: solo se Γ è un insieme clausole di Horn

  In questa forma (finita), è quasi una procedura effettiva ...