Intelligenza Artificiale II

Logica del Primo Ordine

Parte 1.2 - Teoria

Marco Piastra

2

Linguaggio e semantica

Linguaggio del primo ordine

- Simboli di un **linguaggio predicativo** L_{PO}:
  - **Costanti** individuali
    - a, b, c, … (Esempi: $socrate$, $sfera1$, $mickeyMouse$, $amelia$, $alba$)
  - **Predicati**
    - $P(.)$, $Q(.)$, $R(..)$, =, … (Esempi: $Red(.)$, $Large(.)$, $GreaterThan(..)$, >)
  - **Funzioni**
    - $f(.)$, $g(.)$, $h(..)$, … (Esempi: $sqrt(.)$, $colorOf(.)$, $shapeOf(.)$)
  - **Variabili**
    - x, y, z, …
  - **Connettivi**
    - $\neg$, $\rightarrow$ e connettivi derivati $\wedge$, $\vee$, $\leftrightarrow$
  - **Parentesi e virgola** (per separare gli argomenti di funzioni e predicati)
    - (,) , ,
  - **Quantificatori**
    - $\forall$ ed il quantificatore derivato $\exists$ ($\exists x \Leftrightarrow \neg \forall x \neg$)

Termini e formule atomiche

- **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 **chiuso** (*ground*) non contiene variabili
- **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 una **formula atomica**
  - Esempi:

$$\underbrace{\underbrace{Sorella}_{\textbf{predicato}}(\underbrace{\underbrace{amelia}_{\textbf{costante}}}_{\textbf{termine}},\ \underbrace{\underbrace{alba}_{\textbf{costante}}}_{\textbf{termine}})}_{\textbf{formula atomica}}$$

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

Regole di buona formazione

- **Formule ben formate** (**fbf**)
  - regole sintattiche:
    - l'insieme di tutte le fbf di L_{PO} si indica con $\mathrm{fbf}(L_{PO})$
    - ogni *formula atomica* è una fbf
    - $\varphi \in \mathrm{fbf}(L_{PO}) \;\Rightarrow\; (\neg\varphi) \in \mathrm{fbf}(L_{PO})$
    - $\varphi, \psi \in \mathrm{fbf}(L_{PO}) \;\Rightarrow\; (\varphi \to \psi) \in \mathrm{fbf}(L_{PO})$
    - $\varphi \in \mathrm{fbf}(L_{PO}) \;\Rightarrow\; (\forall x\, \varphi) \in \mathrm{fbf}(L_{PO})$ (aggiuntiva rispetto ad L_P)

    - $\varphi, \psi \in \mathrm{fbf}(L_{PO}),\quad (\varphi \vee \psi) \;\Leftrightarrow\; ((\neg\varphi) \to \psi)$
    - $\varphi, \psi \in \mathrm{fbf}(L_{PO}),\quad (\varphi \wedge \psi) \;\Leftrightarrow\; (\neg(\varphi \to (\neg\psi)))$
    - $\varphi, \psi \in \mathrm{fbf}(L_{PO}),\quad (\varphi \leftrightarrow \psi) \;\Leftrightarrow\; ((\varphi \to \psi) \wedge (\psi \to \varphi))$
    - $\varphi \in \mathrm{fbf}(L_{PO}) \quad (\exists x\, \varphi) \Leftrightarrow (\neg\forall x\, \neg\varphi)$ (aggiuntiva rispetto ad L_P)

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:
      $P(\boldsymbol{x})$
      $\exists y\, (P(y) \rightarrow (A(\boldsymbol{x},y) \wedge B(y)))$

- **Formule aperte** e **chiuse**
  - si dice **aperta** una fbf in cui occorre almeno una variabile libera
  - si dice **chiusa** o anche **enunciato** (*sentence*) in caso contrario
  - solo le fbf *chiuse*, cioè gli *enunciati*, hanno un valore di verità
    (in quanto rappresentano delle *affermazioni* ...)

Strutture e interpretazioni

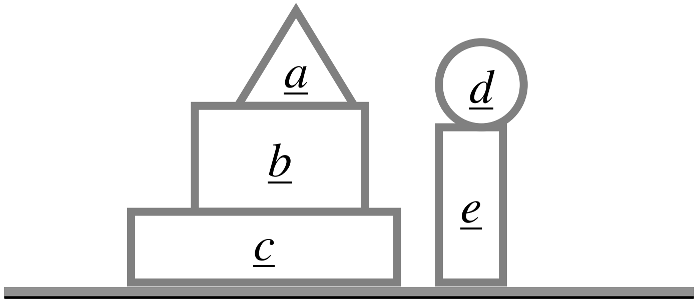
- Una struttura $<\mathbf{U}, v>$ per un linguaggio L_{PO} contiene:
  - un **insieme di oggetti** $\mathbf{U}$ (l'universo del discorso)
  - un'**interpretazione** v, cioè una *funzione*
    - per ogni **costante** c
      $v(c) \in \mathbf{U}$
      il valore assegnato è un **oggetto** di $\mathbf{U}$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

Ai *simboli predicativi* unari sono associati *sottoinsiemi* di $\mathbf{U}$

Assegnazioni

- (ed il significato delle variabili?)
- Per le *variabili*
  - data una struttura $<\mathbf{U},v>$, un'**assegnazione** (*valuation*) s è una funzione che associa ad ogni *variabile* un elemento di $\mathbf{U}$
  - (in altre parole, data $<\mathbf{U},v>$, un'assegnazione s *trasforma le variabili in costanti individuali*) *Aaaargh!*
- *La distinzione tra interpretazione ed assegnazione è un tecnicismo tipico della logica del primo ordine*
  - *serve per definire la semantica di tutte le fbf, aperte e chiuse*
- Confronto tra assegnazioni
  - date due assegnazioni s_1 e s_2
    - se $\forall v \in \mathrm{Var}(L_{PO})$ si ha $s_1(v) \equiv s_2(v)$ allora $s_1 = s_2$
    - per $\forall \underline{d} \in \mathbf{U}$, l'assegnazione $s_2 = [s_1](x:\underline{d})$ ($\underline{d}$ è un oggetto di $\mathbf{U}$, non del linguaggio) si costruisce imponendo $s_2(v) \equiv s_1(v)$ per $\forall v$, $v \neq x$ e $s_2(x) = \underline{d}$

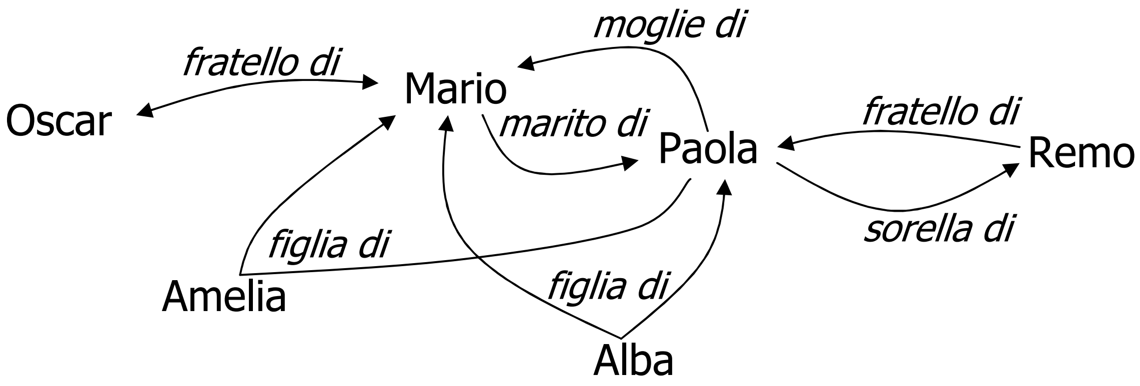
Esempio 1



- Linguaggio
  - simboli predicativi: $Ontable(.)$, $Above(..)$, $On(..)$, $Between(\ldots)$
  - variabili: x, y, z, ...
  - costanti individuali: a, b, c, d, e
- Interpretazione
  - Universo del discorso
    $\mathbf{U} = \{\underline{a}, \underline{b}, \underline{c}, \underline{d}, \underline{e}\}$
  - Predicati
    $v(Ontable) = \{\underline{c}, \underline{e}\}$
    $v(Above) = \{<\underline{a},\underline{b}>, <a,c>, <b,c>, <d,e>\}$
    $v(On) = \{<\underline{a},\underline{b}>, <\underline{b},\underline{c}>, <\underline{d},\mathrm{e}>\}$
    $\mathrm{v}(Between) = \{<\underline{b},\underline{a},\underline{c}>\}$
  - Costanti individuali
    $v(a) = \underline{a}$, $\mathrm{v}(b) = \underline{b}$, $v(c) = \underline{c}$, $v(d) = \underline{d}$, $v(e) = \underline{e}$
- Assegnazione
  - esempio: $s = \{(x:\underline{a}), (y:\underline{b}), (z:\underline{a}) \ldots\}$ (per *tutte* le variabili del linguaggio)

Si usa a per indicare la costante ed $\underline{a}$ per indicare l'oggetto (*solo per comodità)

Esempio 2



- Linguaggio
  - simboli predicativi: $Maschio(.)$, $Femmina(.)$, $Fratello(..)$, $Sorella(..)$
  - simboli funzionali: $madre(.)$, $padre(.)$
  - variabili: $x, y, z, \ldots$
  - costanti individuali: $mario$, $paola$, $remo$, $oscar$, $amelia$, $alba$
- Semantica
  - universo del discorso $\mathbf{U} = \{\underline{mario}, \underline{paola}, \underline{remo}, \underline{oscar}, \underline{amelia}, \underline{alba}\}$
  - interpretazione v:
    - costanti: $v(mario) = \underline{mario}$, $v(paola) = \underline{paola}$, ...
    - predicati:

$$v(Maschio(.)) = \{\underline{mario}, \underline{remo}, \underline{oscar}\}$$
$$v(Femmina(.)) = \{\underline{paola}, \underline{amelia}, \underline{alba}\}$$
$$v(Fratello(..)) = \{<\underline{oscar},\underline{mario}>, <\underline{mario},\underline{oscar}>, <\underline{remo},\underline{paola}>\}$$
$$v(Sorella(..)) = \{<\underline{paola},\underline{remo}>, <\underline{alba},\underline{amelia}>, <\underline{amelia},\underline{alba}>\}$$

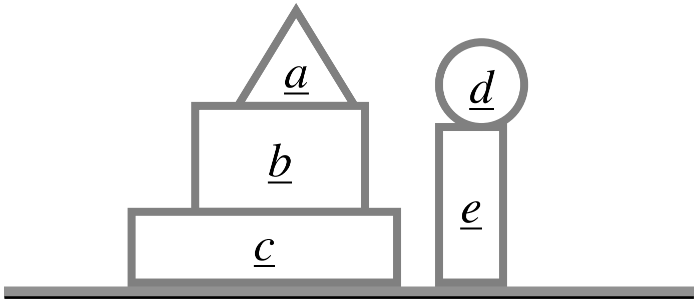
    - funzioni:

$$v(madre(.)) = \{<\underline{alba},\underline{paola}>, <\underline{amelia},\underline{paola}>\}$$
$$v(padre(.)) = \{<\underline{alba},\underline{mario},>, <\underline{amelia},\underline{mario}>\}$$

Soddisfacimento (forma intuitiva)

- Una fbf φ è soddisfatta da una terna $<\mathbf{U},v>$ $[s]$ sse φ afferma una cosa vera in $<\mathbf{U},v>$ $[s]$

- Esempi (mondo degli oggetti)



$Pyramid(a)$

- è vera perchè: $(v(a) = \underline{a}) \in v(Pyramid(.)) = \{\underline{a}\}$

$Parallelepiped(d)$

- non è vera perchè: $(v(d) = \underline{d}) \notin v(Parallelepiped(.)) = \{\underline{b}, \underline{c}, \underline{e}\}$

$\neg\exists x\ (Parallelepiped(x) \wedge Sphere(x))$

- è vera perché: $((v(Parallelepiped(.)) = \{\underline{b}, \underline{c}, \underline{e}\}) \cap (v(Sphere(.)) = \{\underline{d}\})) \equiv \varnothing$

$\forall x\ (Pyramid(x) \vee Parallelepiped(x) \vee Sphere(x))$

- è vera perché:
  $((v(Pyramid(.)) = \{\underline{a}\}) \cup (v(Parallelepiped(.)) = \{\underline{b}, \underline{c}, \underline{e}\}) \cup (v(Sphere(.)) = \{\underline{d}\})) \equiv \mathbf{U}$

Soddisfacimento

- Formule atomiche
  - data una struttura $<\mathbf{U},v>$, un'assegnazione s ed una formula atomica φ
  - si ha che $<\mathbf{U},v>[s] \models \varphi$ 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 ha la forma $t_1 = t_2$ allora $v(t_1)[s] \equiv v(t_2)[s]$ (se si usa l'identità)

- Fbf qualsiasi
  - si ha che $<\mathbf{U},v>[s] \models \varphi$ sse
    - se φ è una formula atomica, vedi sopra

Come per LP

- se $\neg\varphi$ allora $<\mathbf{U},v>[s] \not\models \varphi$
- se $\varphi \wedge \psi$ allora $<\mathbf{U},v>[s] \models \varphi$ e $<\mathbf{U},v>[s] \models \psi$
- se $\varphi \vee \psi$ allora $<\mathbf{U},v>[s] \models \varphi$ o $<\mathbf{U},v>[s] \models \psi$
- se $\varphi \rightarrow \psi$ allora non $<\mathbf{U},v>[s] \models \varphi$ e $<\mathbf{U},v>[s] \not\models \psi$

Esempio 3

- (in riferimento alla interpretazione dell'Esempio 2)
- Soddisfacimento

  $<\mathbf{U},v>[s] \models Maschio(mario)$
  - in quanto $\underline{mario} \in v(Maschio(.))$

  $<\mathbf{U},v>[s] \models Maschio(padre(alba))$
  - in quanto $<\underline{mario},\underline{alba}> \in v(padre)$ e $\underline{mario} \in v(Maschio(.))$

  $<\mathbf{U},v>[s] \models \neg Maschio(paola)$
  - in quanto $\underline{paola} \notin v(Maschio(.))$

  $<\mathbf{U},v>[s] \models Maschio(mario) \wedge Fratello(remo,paola)$
  - in quanto $\underline{remo} \in v(Maschio(.))$ e $<\underline{remo},\underline{paola}> \in v(Fratello(..))$
- Soddisfacimento ed assegnazione

  $<\mathbf{U},v>[(x:\underline{paola}), ...] \models Femmina(x)$

  $<\mathbf{U},v>[(x:\underline{mario}), ...] \not\models Femmina(x)$

  – Il valore delle formule chiuse non dipende dalle assegnazioni …

Soddisfacimento e quantificatori

- Fbf con quantificatori
  - si ha che $<\mathbf{U},v>[s] \models \forall x\ \varphi$ sse
    - se per ogni d $\in \mathbf{U}$ si ha $<\mathbf{U},v>[s](x{:}\underline{d}) \models \varphi$
  - per definizione $\exists x\ \varphi \equiv \neg\forall x\ \neg\varphi$
  - quindi: $<\mathbf{U},v>[s] \models \exists x\ \varphi$
    - se esiste un $\underline{d} \in \mathbf{U}$ per cui si ha $<\mathbf{U},v>[s](x{:}\underline{d}) \models \varphi$

Esempio 4

- (in riferimento alla interpretazione dell'Esempio 2)
- Soddisfacimento

  $<\mathbf{U},v>[s] \models \forall x\,(Maschio(x) \lor Femmina(x))$
  - in quanto per ogni $d \in \mathbf{U}$ si ha che $<\mathbf{U},v>[s](x:\underline{d}) \models (Maschio(x) \lor Femmina(x))$
  - cioè $\underline{d} \in v(Maschio(.))$ oppure $d \in v(Femmina(.))$

  $<\mathbf{U},v>[s] \not\models \forall x\,(Maschio(x))$
  - in quanto non per ogni $\underline{d} \in \mathbf{U}$ si ha che $<\mathbf{U},v>[s](x:\underline{d}) \models Maschio(x)$

  $<\mathbf{U},v>[s](x:\underline{mario}) \models Maschio(x)$
  - in quanto $\underline{mario} \in v(Maschio(.))$

  $<\mathbf{U},v>[s] \models \forall x \forall y\,(Fratello(x,y) \rightarrow Maschio(x))$
  - in quanto in ogni tupla $<\underline{d}_1,\underline{d}_2> \in v(Fratello(..))$, si ha che $\underline{d}_1 \in v(Maschio(.))$

  $<\mathbf{U},v>[s] \models \forall x \forall y\,(Fratello(x,y) \rightarrow Fratello(y,x))$
  - in quanto per ogni tupla $<\underline{d}_1,\underline{d}_2> \in v(Fratello(..))$,
    si ha che $<\underline{d}_2,\underline{d}_1> \in v(Fratello(..))$

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 logica

- Validità e verità logiche
  - una fbf (aperta o chiusa) è **valida** (o **logicamente valida**) se è **valida** in qualsiasi $<\mathbf{U},v>$
    - Esempi:

      $(P(x) \lor \neg P(x))$ (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:

      $\forall x\ (P(x) \lor \neg P(x))$ (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:

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

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 $v(\varphi) : \mathbf{U} \rightarrow \mathbf{U}$
- La presenza delle funzioni arricchisce il linguaggio in modo sostanziale: a differenza dei predicati, le funzioni si possono nidificare (nei termini)
  - Esempio: il mondo delle liste di oggetti $[\underline{a}, \underline{b}, \underline{c}, \ldots]$

Esempio 5

- Esempio: il mondo delle liste di oggetti $[\underline{a}, \underline{b}, \underline{c}, \ldots]$

  $cons(s, x)$
  funzione, associa ad un oggetto $s = \underline{a}$ ed una lista $x = [\underline{b}, \underline{c}]$ la lista $[\underline{a}, \underline{b}, \underline{c}]$

  $Append(x,y,z)$
  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 ($\mathbf{AL}$)

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

  – Applicazioni

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

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

Linguaggio e funzioni

- Ricchezza espressiva
  - Basta una sola funzione in L_{PO} per creare un'infinità (numerabile) di termini
    - Una sola funzione $s(.)$ (=*successore*) è infatti sufficiente per definire l'aritmetica (*la teoria delle proprietà dei numeri naturali*)
- Assiomi ricorsivi
  - Esempio: 'postulati di Peano' (in Mendelson, 1972)

S1 $\forall x \forall y \forall z\ ((x = y) \rightarrow ((x = z) \rightarrow (y = z)))$
S2 $\forall x \forall y\ ((x = y) \rightarrow (s(x) = s(y)))$
S3 $\neg \exists x\ (x = s(0))$
S4 $\forall x \forall y\ ((s(x) = s(y)) \rightarrow (x = y))$
S5 $\forall x\ (x + 0 = x)$
S6 $\forall x \forall y\ ((x + s(y)) = s(x + y))$
S7 $\forall x\ (x \cdot 1 = x)$
S8 $\forall x \forall y\ ((x \cdot s(y)) = ((x \cdot y) + x))$
S9 Per qualsiasi fbf $\varphi(x)$: $\varphi(0) \rightarrow (\forall x\ (\varphi(x) \rightarrow \varphi(s(x))) \rightarrow \forall x\ \varphi(x)$
(*Principio di induzione matematica*)

- Complessità di calcolo
  - Intuitivamente, a ricchezza espressiva corrisponde maggiore complessità
    - Si possono generare termini all'infinito ...

FAQ 2

- Per quale motivo si vogliono definire in modo tanto preciso le corrispondenze tra linguaggio e significato?
  - Ad esempio, nel mondo degli oggetti, avendo a disposizione $\mathbf{U}$ e le relazioni che corrispondono a predicati e funzioni, le verità potrebbero essere derivate direttamente
  - In realtà, in logica formale, si usa il linguaggio come unico strumento allo scopo di non dover mai definire $\mathbf{U}$ e le relazioni in modo esplicito

- Anticipando l'argomento della programmazione logica, si studi l'esempio delle *n* regine in Prolog per ulteriori spunti
  - L'universo del problema viene descritto tramite le fbf il linguaggio
  - Le soluzioni vengono calcolate come fbf
  - L'algoritmo per trovare la soluzione non è descritto affatto … Algorithm = Logic + Control (Kowalski, 1979)