Intelligenza Artificiale

Sistemi a regole

Marco Piastra

(Linguaggio del primo ordine in forma ristretta)

Clausole di Horn
Modelli (finiti) di Herbrand
Forward Chaining
Sistemi a regole
Algoritmo Rete

L_{PO} in forma ristretta

- Nelle fbf di L_{PO} considerate da questo punto in poi
  - Non compaiono simboli funzionali (solo costanti e variabili)
  - Compare solo il quantificatore universale, in forma positiva
  - Il linguaggio contiene solo un numero finito di costanti individuali
  - Esempi:
    - $\forall x \, \forall y \, (P(x) \rightarrow (Q(x, y) \wedge R(y)))$
    - $\forall x \, (P(x)) \rightarrow \forall y \, \forall z \, ((Q(y, z) \wedge R(z)))$
    - $\forall x \, (P(x)) \rightarrow \neg \forall z \, (R(z))$ (*NO: equivale a $\forall x \, (P(x)) \rightarrow \exists z \, \neg (R(z))$)
- **Forma normale prenessa**
  - Le fbf di sono traducibili in una forma
    $\forall x_1 \forall x_2 \ldots \forall x_n \varphi$
    dove φ non contiene quantificatori (φ è anche detta *matrice*)
  - Procedura
    - Si spostano i quantificatori all'inizio, eventualmente ridenominando le variabili
    - $\forall x \, (P(x)) \rightarrow \forall x \, (R(x))$ diventa $\forall x \, \forall y \, (P(x) \rightarrow R(y))$

Forma a clausole in L_{PO}

- Molto simile al caso proposizionale …
  - Si parte da fbf in forma prenessa

    $\forall x_1 \forall x_2 \ldots \forall x_n \varphi$ (φ non contiene quantificatori)
    - Essendo tutti universali, i quantificatori si possono semplicemente omettere
    - La matrice φ viene tradotta in FNC (con le stesse regole del caso proposizionale) e quindi in forma a clausole FC (con le stesse regole del caso proposizionale)

  - Esempio:
    - $\forall x \, \forall y \, (P(x) \rightarrow (Q(x, y) \wedge R(y)))$ (fbf di partenza)
    - $P(x) \rightarrow (Q(x, y) \wedge R(y))$ (eliminazione dei quantificatori)
    - $\neg P(x) \vee (Q(x, y) \wedge R(y)$ (equivalenza di $\rightarrow$)
    - $(\neg P(x) \vee Q(x, y)) \wedge (\neg P(x) \vee R(y))$ (FNC, distribuitività di $\vee$)
    - $\{\neg P(x), Q(x, y)\}, \ \{\neg P(x), R(y)\}$ (FC)

Clausole di Horn in L_{PO}

- Definizione quasi identica al caso proposizionale
  - Forma a clausole
  - In ciascuna clausola occorre al massimo un atomo in forma positiva
  - Fatti, regole e goal
    - **Fatti**: clausola con un singolo atomo in forma positiva

      $\{Umano(socrate)\}$, $\{Pyramid(a)\}$, $\{Sorella(alba, paolo)\}$
    - **Regole**: clausola di due o più atomi, uno in forma positiva

      $\{Umano(x), \neg Filosofo(x)\}$,
      $\forall x\ (Filosofo(x) \rightarrow Umano(x))$

      $\{Madre(x,y), \neg Femmina(x), \neg Genitore(x, y)\}$
      $\forall x \forall y\ ((Femmina(x) \wedge Genitore(x, y)) \rightarrow Madre(x, y))$
    - **Goal**: clausola di atomi in forma negativa

      $\{\neg Umano(socrate)\}$

      $\{\neg Sorella(alba, amelia)\}$

Base (finita) di Herbrand

- **Atomi** di Herbrand in L_{PO}
  - Un **atomo** di Herbrand è una fbf *atomica* (*ground atom*) che non contiene variabili (solo costanti individuali)
    - Esempi: $P(a)$, $P(d)$, $P(e)$, $Q(a, b)$, $Q(d, e)$, $R(a, c, e)$, ...
- **Base** (finita) di Herbrand
  - La **base** di Herbrand è l'insieme di tutti gli atomi di Herbrand

    $B_H \equiv \{P(a), P(b), P(c), ..., Q(a, a), Q(a, b), ..., R(a, a, a), R(a, a, b), R(a, a, c), ...\}$
    - Se il linguaggio ha un numero finito di predicati e costanti, la base di Herbrand è finita

(Tramite la base di Herbrand B_H si definiscono interpretazioni di L_{PO} usando universo gli oggetti del linguaggio …)

Modelli (finiti) di Herbrand

- **Struttura** (finita) di Herbrand per L_{PO}
  - Una struttura $<\mathbf{U}_H, v_H>$ tale che
    - $\mathbf{U}_H$ è formato dalle costanti individuali del linguaggio ($\mathrm{Cost}(L_{PO})$)
    - v_H di fatto, è un sottoinsieme della base di Herbrand
- **Interpretazione** v_H di Herbrand
  - $v_H \equiv \{P(a), P(b), P(c), Q(a, b), R(a, c) \ldots\}$ (solo formule *ground atoms*)
  - $v_H \subseteq \mathrm{B}_H$
- **Modello** di Herbrand

$\varphi \in \mathrm{Atomi}(L_{PO}),\ <\mathbf{U}_H, v_H> [s] \models \varphi$ sse $\varphi \in v_H$

$\varphi \in \mathrm{Atomi}(L_{PO}),\ <\mathbf{U}_H, v_H> [s] \models \neg\varphi$ sse $\varphi \notin v_H$

$<\mathbf{U}_H, v_H> [s] \models \neg\varphi$ sse $<\mathbf{U}_H, v_H> [s] \not\models \varphi$

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

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

Teorema di Herbrand (in forma ristretta)

- **Sistema** (finito) di Herbrand per un enunciato
  - Dato un enunciato universale, cioè della forma
    $\forall x_1 \forall x_2 \ldots \forall x_n \varphi$ (φ non contiene quantificatori)
  - è l'insieme (finito) di formule **chiuse** generato per sostituzione
    $\varphi\ [x_1/c_1, x_2/c_2 \ldots x_n/c_n]$
  - con <u>tutte</u> le possibili combinazioni $<c_1, c_2 \ldots c_n>$ di $c_i \in \mathrm{Cost}(L_{PO})$

- **Sistema** (finito) di Herbrand di una teoria
  - Data una teoria Σ di enunciati universali, il sistema (finito) $\mathrm{H}(\Sigma)$ è l'unione di tutti i sistemi di Herbrand generati dagli enunciati Σ

- **Teorema di Herbrand** (in forma ristretta)
  - Data una teoria di enunciati universali Σ, $\mathrm{H}(\Sigma)$ ha un modello sse Σ ha un modello
    - In questa forma (finita), è quasi una procedura effettiva ...

Derivazioni: teorema di Herbrand

– Esempio:

$\Sigma \equiv \{\{Umano(x), \neg Filosofo(x)\}, \{Mortale(y), \neg Umano(y)\}\}$
$\text{Cost}(L_{PO}) \equiv \{socrate, platone, aristotele\}$

- Sistema di Herbrand

$H(\Sigma) \equiv \{\{Umano(socrate), \neg Filosofo(socrate)\}, \{Umano(platone), \neg Filosofo(platone)\},$
$\{Umano(aristotele), \neg Filosofo(aristotele)\}, \{Mortale(socrate), \neg Umano(socrate)\},$
$\{Mortale(platone), \neg Umano(platone)\}, \{Mortale(aristotele), \neg Umano(aristotele)\}\}$

- Si consideri un insieme di fatti

$\Gamma \equiv \{Filosofo(socrate)\}, \{Filosofo(platone)\}, \{Filosofo(aristotele)\}\}$

- Applicando il *Modus Ponens* in modo esaustivo a $H(\Sigma) \cup \Gamma$:

$K \equiv \{\{Umano(socrate)\}, \{Umano(platone)\}, \{Umano(aristotele)\},$
$\{Mortale(socrate)\}, \{Mortale(platone)\}, \{Mortale(aristotele)\}\}$

Regola di istanziazione

- Una regola di inferenza specifica (per L_{PO}):

$$\frac{\forall x_1 \forall x_2 \ldots \forall x_n \varphi}{\varphi\ [x_1/c_1, x_2/c_2 \ldots x_n/c_n]}$$

  - Da un enunciato universalmente quantificato si può derivare un enunciato (non quantificato) in cui tutte le variabili sono sostituite da costanti
  - Esempi:
    - $\forall x\ (Filosofo(x) \rightarrow Umano(x)) \vdash Filosofo(socrate) \rightarrow Umano(socrate)$
    - $\forall x \forall y\ ((Femmina(x) \wedge Genitore(x, y)) \rightarrow Madre(x, y)) \vdash ((Femmina(paola) \wedge Genitore(paola, amelia)) \rightarrow Madre(paola, amelia))$
  - La regola *INST* è **corretta** (in L_{PO})
    - Corollario del teorema di Herbrand

Metodo del forward chaining in L_{PO}

- Descrizione:
  - Per stabilire se $\Gamma \vdash \varphi$
    - (Γ regole e fatti come clausole definite universalmente quantificate, φ fatto)
  - Si applicano le regole *INST* e *GMP* a Γ in modo esaustivo

    INST: $\forall x_1 \forall x_2 \ldots \forall x_n \varphi \vdash \varphi\,[x_1/c_1, x_2/c_2 \ldots x_n/c_n]$

    GMP: $\alpha_1,\ \alpha_2,\ \ldots\ \alpha_n, \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \rightarrow \beta \vdash \beta$

    ($\alpha_1,\ \alpha_2,\ \ldots\ \alpha_n, \beta$ sono fbf *ground*, vale a dire dove non occorrono variabili)
    - Si ottiene un insieme Σ di fatti, $\Sigma = \{\psi : \Gamma \vdash \psi, \textit{per INST e GMP}\}$
    - L'algoritmo termina con successo se $\varphi \in \Sigma$

  - Esempio:
    - *INST*: $\forall x \forall y\ ((Femmina(x) \wedge Genitore(x, y)) \rightarrow Madre(x, y)) \vdash$
      $((Femmina(paola) \wedge Genitore(paola, amelia)) \rightarrow Madre(paola, amelia))$
    - *GMP*: $Femmina(paola), Genitore(paola, amelia),$
      $((Femmina(paola) \wedge Genitore(paola, amelia)) \rightarrow Madre(paola, amelia))$
      $\vdash Madre(paola, amelia)$

Caratteristiche del forward chaining in L_{PO}

- Il metodo del forward chaining in L_{PO}
  - Termina sempre (se le costanti del linguaggio sono in numero finito)
  - E` corretto
  - E` completo per le clausole di Horn
    - Si dimostra utilizzando il Teorema di Herbrand
- Procedura effettiva:
  - Per la regola GMP, non è particolarmente problematica
  - Per la regola $INST$, si dovrebbero generare tutte le possibili *istanziazioni*
    - Altamente inefficiente, comunque: ciascuna fbf quantificata produce n^m fbf *ground* (n numero delle costanti, m numero delle variabili quantificate)
- Miglioramenti per $INST$
  - Se l'obiettivo è applicare GMP a regole e fatti *ground* (privi di variabili)
  - Ci si potrebbe limitare alla generazione delle sole istanziazioni delle regole cui è applicabile GMP, dati i fatti in Γ
  - Ma come?

Sistemi a regole (Jess)

- Un sistema a regole contiene regole e fatti
  - Nelle regole, tutte le variabili sono universalmente quantificate
  - I fatti non contengono variabili
- Ciascuna regola è un'implicazione
  - <LHS - *Left Hand Side*> $\Rightarrow$ <RHS - *Right Hand Side*>
  - LHS e RHS
    - Sono congiunzioni di fbf atomiche in forma positiva
    - Morale: le regole sono traducibili in clausole di Horn

[1] Nelle regole Jess si può usare anche il *NOT* che tuttavia ha un significato speciale (vedi manuale)

Regola "Sorella"

$Madre(m, x)$
$Madre(m, y)$
$Padre(p, x)$
$Padre(p, y)$
$Femmina(x)$
$\Rightarrow$
$Sorella(x, y)$

Regola "Fratello"

$Madre(m, x)$
$Madre(m, y)$
$Padre(p, x)$
$Padre(p, y)$
$Maschio(x)$
$\Rightarrow$
$Fratello(x, y)$

Algoritmo Rete (C. Forgy, 1980)

- Un algoritmo per applicare $INST$ in modo efficace
  - Si traduce l'insieme delle parti sinistre (LHS) delle regole in una struttura a grafo
  - I fatti 'flusicono' attraverso la struttura

- Descrizione
  - La struttura **Rete** è un grafo diretto aciclico
    - Rappresentazione delle LHS delle regole in una struttura 'stratificata'
  - Nodi del grafo
    - *Alpha* (primo livello): clausola atomica che occorre nella LHS di una o più regole (clausole identiche in regole diverse = un solo nodo *alpha*)
    - *Beta* : AND binario di due nodi *alpha* o *beta* di livello precedente (corrispondono ad un *join* delle instanziazioni)
  - Fatti
    - Si 'agganciano' ai nodi *alpha* (come una tabella in un database)
  - Regole
    - Si 'agganciano' ai nodi *alpha* o *beta*

Esempio: algoritmo rete

Memoria di lavoro
(fatti)

Femmina(*amelia*)
Femmina(*alba*)
Femmina(*paola*)
Madre(*paola, amelia*)
Madre(*paola, alba*)
Padre(*mario, amelia*)
Padre(*mario, alba*)
Maschio(*mario*)

Regola "Sorella"

Madre(*m, x*)
Madre(*m, y*)
Padre(*p, x*)
Padre(*p, y*)
Femmina(*x*)
⇒
Sorella(*x, y*)

Regola "Fratello"

Madre(*m, x*)
Madre(*m, y*)
Padre(*p, x*)
Padre(*p, y*)
Maschio(*x*)
⇒
Fratello(*x, y*)

INST

Agenda
(regole istanziate)

Regola "Sorella" (istanza 1)

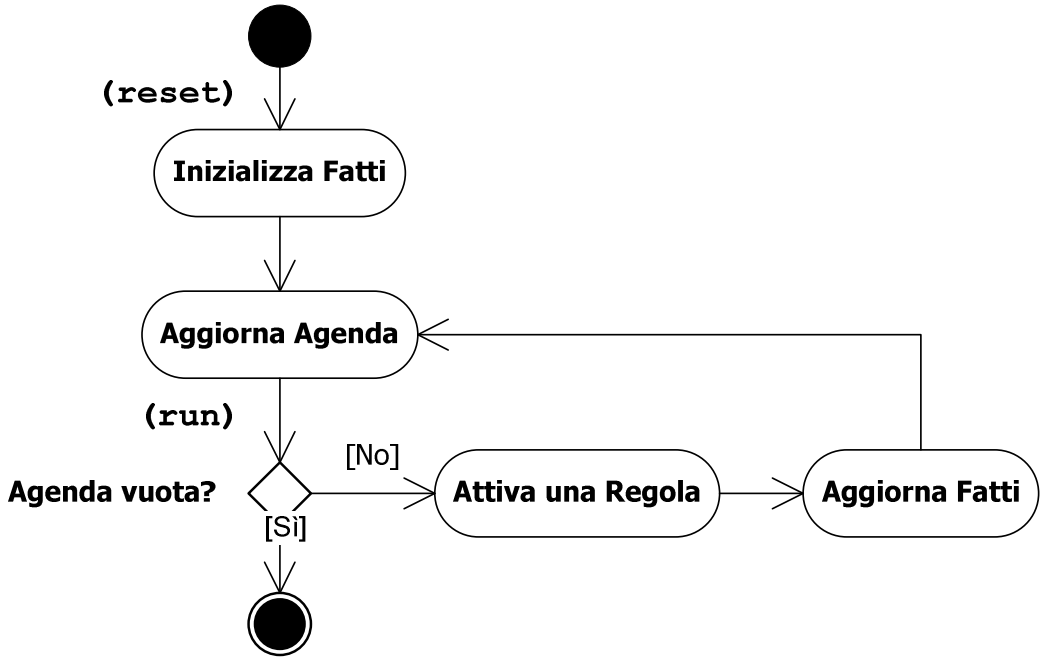
Madre(*paola, amelia*)
Madre(*paola, alba*)
Padre(*mario, amelia*)
Padre(*mario, alba*)
Femmina(*amelia*)
⇒
Sorella(*amelia, alba*)

Regola "Sorella" (istanza 2)

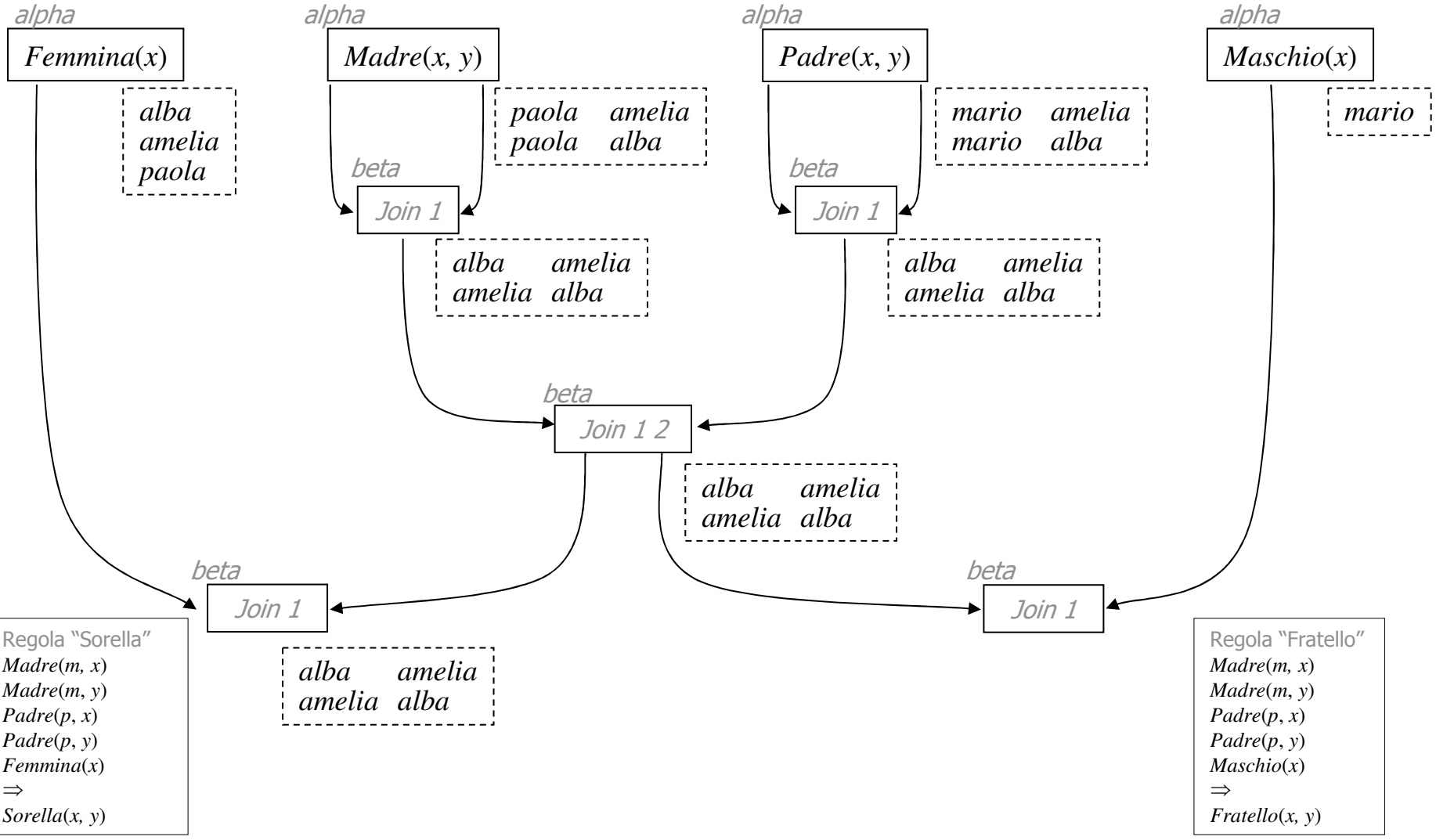
Madre(*paola, alba*)
Madre(*paola, amelia*)
Padre(*mario, amelia*)
Padre(*mario, alba*)
Femmina(*alba*)
⇒
Sorella(*amelia, alba*)

Ciclo di esecuzione (Jess)

- Il sistema a regole Jess procede ciclicamente:
  1) Aggiorna l'agenda (regole istanziate)
  2) Sceglie ed attiva una regola
     (la scelta è "casuale", rispettando le classi di precedenza)
  3) Aggiorna la memoria di lavoro (fatti)



Esempio: struttura rete



Esempio: struttura rete e nuovi fatti