Riduzione

Affermazione: una sentenza ground è conseguenza logica della nuova KB sse è conseguenza logica della KB originaria

Affermazione: ogni FOL KB può essere proposizionalizzata in modo da preservarne le conseguenze logiche

Idea: proposizionalizzare sia KB che la query, applicare la risoluzione, restituire il risultato

Problema: con i simboli di funzione, c'è un numero infinito di termini ground,
e.g., $Padre(Padre(Padre(Giovanni)))$

Teorema: Herbrand (1930). Se una sentenza α è conseguenza logica di una FOL KB,
essa è conseguenza logica di un sottoinsieme *finito* della KB proposizionale

Idea: For $n = 0$ to ∞ do
creare una KB proposizionale con termini di profondità n
controllare se α è conseguenza logica della KB considerata

Problema: funziona solo se α è conseguenza logica, cicla se α non è conseguenza logica

Teorema: Turing (1936), Church (1936), "entailment" in FOL è semidecidible

Problemi con la proposizionalizzazione

La proposizionalizzazione sembra generare tante sentenze irrilevanti
P.e.,

$$\forall x \;\; Re(x) \land Ingordo(x) \;\Rightarrow\; Diavolo(x)$$
$$Re(Giovanni)$$
$$\forall y \;\; Ingordo(y)$$
$$Fratello(Riccardo, Giovanni)$$

ok per $Diavolo(Giovanni)$, ma la proposizionalizzazione produce tanti fatti come $Ingordo(Riccardo)$ che sono irrilevanti

Con p predicati k-ari e n constanti, ci sono $p \cdot n^k$ istanziazioni!

Unificazione

Si può ottenere l'inferenza immediatamente se possiamo trovare una sostituzione θ tale che $Re(x)$ e $Ingordo(x)$ corrispondano a $Re(Giovanni)$ e $Ingordo(y)$

$\theta = \{x/Giovanni, y/Giovanni\}$ va bene

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

p	q	θ
$Conosce(Giovanni, x)$	$Conosce(Giovanni, Giovanna)$	
$Conosce(Giovanni, x)$	$Conosce(y, OJ)$	
$Conosce(Giovanni, x)$	$Conosce(y, Madre(y))$	
$Conosce(Giovanni, x)$	$Conosce(x, OJ)$	

Unificazione

Si può ottenere l'inferenza immediatamente se possiamo trovare una sostituzione θ tale che $Re(x)$ e $Ingordo(x)$ corrispondano a $Re(Giovanni)$ e $Ingordo(y)$

$\theta = \{x/Giovanni, y/Giovanni\}$ va bene

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

p	q	θ
$Conosce(Giovanni, x)$	$Conosce(Giovanni, Giovanna)$	$\{x/Giovanna\}$
$Conosce(Giovanni, x)$	$Conosce(y, OJ)$	
$Conosce(Giovanni, x)$	$Conosce(y, Madre(y))$	
$Conosce(Giovanni, x)$	$Conosce(x, OJ)$	

Unificazione

Si può ottenere l'inferenza immediatamente se possiamo trovare una sostituzione θ tale che $Re(x)$ e $Ingordo(x)$ corrispondano a $Re(Giovanni)$ e $Ingordo(y)$

$\theta = \{x/Giovanni, y/Giovanni\}$ va bene

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

p	q	θ
$Conosce(Giovanni, x)$	$Conosce(Giovanni, Giovanna)$	$\{x/Giovanna\}$
$Conosce(Giovanni, x)$	$Conosce(y, OJ)$	$\{x/OJ, y/Giovanni\}$
$Conosce(Giovanni, x)$	$Conosce(y, Madre(y))$	
$Conosce(Giovanni, x)$	$Conosce(x, OJ)$	

Unificazione

Si può ottenere l'inferenza immediatamente se possiamo trovare una sostituzione θ tale che $Re(x)$ e $Ingordo(x)$ corrispondano a $Re(Giovanni)$ e $Ingordo(y)$

$\theta = \{x/Giovanni, y/Giovanni\}$ va bene

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

p	q	θ
$Conosce(Giovanni, x)$	$Conosce(Giovanni, Giovanna)$	$\{x/Giovanna\}$
$Conosce(Giovanni, x)$	$Conosce(y, OJ)$	$\{x/OJ, y/Giovanni\}$
$Conosce(Giovanni, x)$	$Conosce(y, Madre(y))$	$\{y/Giovanni, x/Madre(Giovanni)\}$
$Conosce(Giovanni, x)$	$Conosce(x, OJ)$	

Unificazione

Si può ottenere l'inferenza immediatamente se possiamo trovare una sostituzione θ tale che $Re(x)$ e $Ingordo(x)$ corrispondano a $Re(Giovanni)$ e $Ingordo(y)$

$\theta = \{x/Giovanni, y/Giovanni\}$ va bene

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

p	q	θ
$Conosce(Giovanni, x)$	$Conosce(Giovanni, Giovanna)$	$\{x/Giovanna\}$
$Conosce(Giovanni, x)$	$Conosce(y, OJ)$	$\{x/OJ, y/Giovanni\}$
$Conosce(Giovanni, x)$	$Conosce(y, Madre(y))$	$\{y/Giovanni, x/Madre(Giovanni)\}$
$Conosce(Giovanni, x)$	$Conosce(x, OJ)$	$fallimento$

Standardizing apart elimina sovrapposizioni di variabili, p.e., $Conosce(z_{17}, OJ)$

Unificazione

Si può ottenere l'inferenza immediatamente se possiamo trovare una sostituzione θ tale che $Re(x)$ e $Ingordo(x)$ corrispondano a $Re(Giovanni)$ e $Ingordo(y)$

$\theta = \{x/Giovanni, y/Giovanni\}$ va bene

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

p	q	θ
$Conosce(Giovanni, x)$	$Conosce(Giovanni, Giovanna)$	$\{x/Giovanna\}$
$Conosce(Giovanni, x)$	$Conosce(y, OJ)$	$\{x/OJ, y/Giovanni\}$
$Conosce(Giovanni, x)$	$Conosce(y, Madre(y))$	$\{y/Giovanni, x/Madre(Giovanni)\}$
$Conosce(Giovanni, x)$	$Conosce(x, OJ)$	$fallimento$

Standardizing apart elimina sovrapposizioni di variabili, p.e., $Conosce(z_{17}, OJ)$

$Conosce(Giovanni, x)$	$Conosce(z_{17}, OJ)$	$\{z_{17}/Giovanni, x/OJ\}$

Algoritmo di Unificazione

function $\text{Unify}(x, y, \theta)$ **returns** una sostituzione che rende x e y identici (dato θ)
  inputs: x, una variabile, costante, lista, o espressione composta
    y, una variabile, costante, lista, o espressione composta
    θ, la sostituzione corrente, inizialmente vuota

  if $\theta =$ fallimento **then return** fallimento
  else if $x = y$ **then return** θ
  else if $\text{Variabile?}(x)$ **then return** $\text{Unify-Var}(x, y, \theta)$
  else if $\text{Variabile?}(y)$ **then return** $\text{Unify-Var}(y, x, \theta)$
  else if $\text{Composta?}(x)$ **and** $\text{Composta?}(y)$ **then**
    return $\text{Unify}(\text{Args}[x], \text{Args}[y], \text{Unify}(\text{Op}[x], \text{Op}[y], \theta))$
  else if $\text{Lista?}(x)$ **and** $\text{Lista?}(y)$ **then**
    return $\text{Unify}(\text{Resto}[x], \text{Resto}[y], \text{Unify}(\text{Primo}[x], \text{Primo}[y], \theta))$
  else return fallimento

---

function $\text{Unify-Var}(var, x, \theta)$ **returns** una sostituzione
  inputs: var, una variabile, x una qualsiasi espressione, θ la sostituzione corrente

  if $\{var/val\} \in \theta$ **then return** $\text{Unify}(val, x, \theta)$
  else if $\{x/val\} \in \theta$ **then return** $\text{Unify}(var, val, \theta)$
  else if $\text{Controlla-Occorrenza?}(var, x)$ **then return** fallimento
  else return aggiungi $\{var/x\}$ a θ

Algoritmo di Unificazione Modificato

function UNIFY(x, y, θ) **returns** una sostituzione che rende x e y identici (dato θ)
inputs: x, una variabile, costante, lista, o espressione composta
y, una variabile, costante, lista, o espressione composta
θ, la sostituzione corrente, inizialmente vuota

if $\theta =$ fallimento **then return** fallimento
else if $x = y$ **then return** θ
else if VARIABILE?(x) **then return** UNIFY-VAR(x, y, θ)
else if VARIABILE?(y) **then return** UNIFY-VAR(y, x, θ)
else if COMPOSTA?(x) **and** COMPOSTA?(y) **then**
return UNIFY(ARGS[x], ARGS[y], UNIFY(OP[x], OP[y], θ))
else if LISTA?(x) **and** LISTA?(y) **then**
return UNIFY(SUBST(θ,RESTO[x]), SUBST(θ,RESTO[y]), UNIFY(SUBST(θ,PRIMO[x]), SUBST(θ,PRIMO[y]), θ))
else return fallimento

---

function UNIFY-VAR(var, x, θ) **returns** una sostituzione
inputs: var, una variabile, x una qualsiasi espressione, θ la sostituzione corrente

if $\{var/val\} \in \theta$ **then return** UNIFY(val, x, θ)
else if $\{x/val\} \in \theta$ **then return** UNIFY(var, val, θ)
else if CONTROLLA-OCCORRENZA?(var, x) **then return** fallimento
else return COMPOSE($\{var/x\}$,θ)

Modus Ponens Generalizzato (GMP)

$$\frac{p_1{}',\ \ p_2{}',\ \ldots,\ p_n{}',\ \ (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q)}{q\theta}$$

dove $p_i{}'\theta = p_i\theta$ per tutte le i

$p_1{}'$ è $Re(Giovanni)$ $\qquad$ p_1 è $Re(x)$
$p_2{}'$ è $Ingordo(y)$ $\qquad$ p_2 è $Ingordo(x)$
θ è $\{x/Giovanni, y/Giovanni\}$ $\qquad$ q è $Diavolo(x)$
$q\theta$ è $Diavolo(Giovanni)$

GMP usato con KB composto di clausole definite (*esattamente* un letterale positivo)
Si assume che tutte le variabili siano quantificate universalmente