Risoluzione e unificazione

Lezione 18 di Fondamenti di informatica

Docente: Giuseppe Scollo

Università di Catania
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica, I livello, AA 2008-09

Indice

risoluzione di clausole disgiuntive

come si è visto in una lezione precedente:

- ogni teoria predicativa Φ è equisoddisfacibile alla teoria Skolem(Φ) da essa ottenuta per Skolemizzazione
- rendendo implicita la quantificazione universale delle variabili, le matrici delle formule in Skolem(Φ) ammettono equivalenti forme clausali CNF
- l'unione di tali insiemi di clausole disgiuntive è ancora una forma clausale CNF, equivalente a Skolem(Φ) e pertanto equisoddisfacibile a Φ

gli elementi di ciascuna clausola disgiuntiva, detti **letterali**, possono essere:

- **positivi:** formule atomiche
- **negativi:** negazioni di formule atomiche

il **principio di risoluzione** (J.A. Robinson, 1963) è una regola di inferenza che permette di dedurre la clausola $\Gamma \cup \Delta$, detta **risolvente**, dalle clausole $\Gamma \cup \{\alpha\}$ e $\Delta \cup \{\neg\alpha\}$

N.B. variabili possono ben occorrere nei letterali in considerazione; cioè, qui **non** si assume di dedurre nell'espansione di Herbrand della teoria data

deduzione automatica per risoluzione

similmente ai metodi Herbrand-Skolem e al metodo dei tableau, i metodi deduttivi basati sul principio di risoluzione sono di tipo **refutativo**:

si dimostra $\Phi \models \psi$ per deduzione della clausola vuota dalla CNF di Skolem($\Phi \cup \{\neg\psi\}$)

a differenza dei metodi Herbrand-Skolem, però, la presenza delle variabili richiede la considerazione non solo delle clausole date, o via via dedotte, ma anche delle loro **istanze**, ottenute per **sostituzione** di variabili con termini

come stiamo per vedere, il "segreto algoritmico" dell'efficienza della deduzione automatica per risoluzione sta proprio nell'impiego accorto di metodi di reperimento di sostituzioni appropriate

indipendentemente dall'efficienza, una proprietà fondamentale del principio di risoluzione, supplementato da una regola di sostituzione (che può anche essere espressa assieme ad esso in un'unica regola di inferenza) è che ci fornisce un sistema deduttivo **completo** per la logica predicativa

sostituzioni, unificatori, sussunzione

dato un insieme di variabili $\mathbf{V}$ e una segnatura algebrica Σ, una **sostituzione** è una funzione $\sigma : \mathbf{V} \to T_\Sigma(\mathbf{V})$, che manda variabili in termini

una sostituzione σ viene naturalmente estesa ad una funzione $\sigma : T_\Sigma(V) \to T_\Sigma(V)$ per induzione strutturale:

$\sigma(\omega(t_1,\ldots,t_n)) = \omega(\sigma(t_1),\ldots,\sigma(t_n))$

$\sigma(\kappa) = \kappa$ se il termine è un simbolo di costante κ

similmente una sostituzione viene estesa a una funzione sulle $\Sigma(V)$-formule

l'**esistenza** di sostituzioni determina il **preordine di sussunzione** dei termini:

$t' \preccurlyeq t \Leftrightarrow \exists\sigma\ t' = \sigma(t)$

analogamente si definisce la sussunzione di formule; la transitività di tali relazioni consegue dalla componibilità delle sostituzioni (sul dominio esteso)

quali termini sono **minimali** nel preordine di sussunzione? quali sono massimali?

un **unificatore** di due termini è una sostituzione che ne dà immagini uguali

due termini sono **unificabili** se ne esiste un unificatore

teorema di unificazione

l'unificabilità di termini si estende in modo ovvio a quella di formule

più generalmente, un unificatore di un **insieme** S, di termini o di formule, è una sostituzione σ tale che $\sigma(S)$ ha un solo elemento; S è unificabile se ha un unificatore

un preordine di sussunzione è definito anche sulle sostituzioni, dalla condizione

$\sigma' \preccurlyeq \sigma \Leftrightarrow \exists\tau\ \ \sigma' = \tau\circ\sigma$

quali sostituzioni sono massimali in tale preordine?

non tutti gli insiemi di termini o formule sono unificabili: ad es. $\{x, \omega(x)\}$ non lo è

vale tuttavia un fatto notevole:

se S è unificabile, allora il suo insieme di unificatori ha un elemento massimale

il Teorema di Unificazione appena enunciato è dimostrato dall'esistenza di un algoritmo che, dato in input S, termina sempre, producendo in output tale elemento, detto **unificatore di massima generalità (m.g.u.)**, quando S è unificabile, o altrimenti determina che S non lo è

semineutralità locale: $\sigma=\sigma\circ\mu$ se μ è un m.g.u. di S e σ è un unificatore di S

strutture astratte per l'unificazione

esistono diverse varianti dell'algoritmo di unificazione, qui ci si limita all'idea essenziale, considerando per semplicità solo l'unificabilità di due termini ***t,t'***

si definisce, per induzione strutturale, l'insieme ***diff(t,t')*** di coppie non ordinate di sottotermini dei rispettivi termini, essendo ciascuna coppia relativa a posti corrispondenti nelle strutture ad albero (radicato, ordinato) dei due termini

la coppia è l'insieme vuoto se i due sottotermini sono identici, altrimenti è l'insieme dei due sottotermini se le rispettive radici sono simboli (operatori o variabili) diversi

N.B. se le radici sono operatori uguali, si aggiungono all'insieme ***diff(t,t')*** le coppie relative a posti corrispondenti dei loro sottotermini

l'insieme delle **riduzioni di *diff(t,t')*** è il suo sottoinsieme costituito dalle coppie {u, v} che soddisfano la condizione di **negoziabilità**, costituita da entrambe le condizioni seguenti:

almeno uno dei due elementi è una variabile

nessuno dei due elementi coincide con un sottotermine dell'altro

diff(t,t') è **negoziabile** se è non vuoto e tutte le sue coppie soddisfano tale condizione

un algoritmo di unificazione

un algoritmo di unificazione, dovuto a A.J.Robinson (1963) può essere concisamente descritto come segue, con l'impiego della seguente notazione:

ε designa la sostituzione identica

← designa l'assegnamento (imperativo)

σ ← ε

while *diff*(σ(***t***), σ(***t'***)) negoziabile **do**

scegli una riduzione μ di ***diff***(σ(***t***), σ(***t'***))

σ ← μ∘σ

end while

if ***diff***(σ(***t***), σ(***t'***)) vuoto

then output (YES, σ)

else output NO