Rappresentazione della Conoscenza

Lezione 2

Logica come linguaggio di rappresentazione della conoscenza

Sommario

- ♢ richiami sintassi e semantica
- ♢ Risoluzione: richiamo RN 9-5
- ♢ Completezza e correttezza della risoluzione
- ♢ Strategie di risoluzione
- ♢ Restrizioni del linguaggio

Il linguaggio: i simboli logici

Un *linguaggio del primo ordine* $\mathcal{L}$ è costruito sui seguenti insiemi di simboli:

Simboli Logici

- I connettivi proposizionali: $\neg, \wedge, \vee, \implies$ e $\Leftrightarrow$;
- Le costanti proposizionali $\top$ e $\bot$;
- Il simbolo di uguaglianza $=$, eventualmente assente;
- I simboli separatori '(', ')' e ',';
- Un'infinità numerabile di simboli di *variabile individuale* $x_1, x_2, \ldots$;
- Il simbolo di *quantificazione universale* $\forall$;
- Il simbolo di *quantificazione esistenziale* $\exists$.

Il linguaggio: i parametri

Parametri

- Un insieme finito o numerabile di *simboli di predicato*, ognuno dei quali ha associato un intero positivo n detto arità. Un predicato di arità n è detto n-ario;
- Un insieme finito o numerabile di *simboli di funzione*, ognuno dei quali ha associato un intero positivo detto arità. Una funzione di arità n è detta n-aria;
- Un insieme finito o numerabile di *simboli di costante*.

Formule atomiche

L'insieme Atom degli *atomi* o *formule atomiche* è definito induttivamente come segue:

1. $\perp$ e $\top$ sono atomi;
2. Se t_1 e t_2 sono termini allora $t_1 = t_2$ è un atomo;
3. Se $t_1, \ldots, t_n$ sono termini e P è un simbolo di predicato n-ario $P(t_1, \ldots, t_n)$ è un atomo.

Formule

L'insieme delle *formule* di $\mathcal{L}$ è l'insieme induttivo definito come segue:

- Ogni atomo è una formula;
- Se A è una formula $\neg A$ è una formula;
- Se $\circ$ è un connettivo binario, A e B due formule, $A \circ B$ è una formula;
- Se A è una formula, x una variabile, $\forall x A$ e $\exists x A$ sono formule.

Interpretazioni e modelli

Una *struttura per il linguaggio* $\mathcal{L}$ è una coppia $\mathfrak{A} = \langle D, I \rangle$ dove:

- D è un insieme non vuoto chiamato *dominio* di $\mathfrak{A}$;
- I è una funzione chiamata *interpretazione*. I associa:
  - a ogni simbolo di costante c un elemento $c^I \in D$;
  - a ogni simbolo di funzione n-aria f una funzione $f^I : D^n \rightarrow D$;
  - a ogni simbolo di predicato n-ario P una relazione n-aria $P^I \subseteq D^n$.

Soddisfacibilità di formule

Consideriamo solo formule chiuse, per le formula aperte le definizioni l'interpretazione viene estesa sostituendo le variabili libere con un'assegnazione.

La soddisfacibilità di una formula chiusa ϕ, in una struttura $\mathfrak{A}$ è denotato con:

$$\mathfrak{A} \models \phi$$

che, intuitivamente, dice che la struttura $\mathfrak{A}$ soddisfa ϕ sse è vera l'interpretazione di ϕ determinata da $\mathfrak{A}$.

Soddisfacibilità: definizione 1

Sia $\mathfrak{A} = \langle D, I \rangle$ una struttura per il linguaggio $\mathcal{L}$

1. $\mathfrak{A} \models \top$ e $\mathfrak{A} \not\models \bot$;
2. Se A è una formula atomica del tipo $P(t_1, \ldots, t_n)$, allora

$$\mathfrak{A} \models P(t_1, \ldots, t_n) \text{ sse } \langle t_1^I \ldots t_n^I \rangle \in P^I;$$

3. se A è una formula atomica del tipo $t_1 = t_2$ allora

$$\mathfrak{A} \models t_1 = t_2 \text{ sse } t_1^I = t_2^I;$$

4. $\mathfrak{A} \models \neg A$ sse $\mathfrak{A} \not\models A$;
5. $\mathfrak{A} \models A \wedge B$ sse $\mathfrak{A} \models A$ e $\mathfrak{A} \models B$;
6. $\mathfrak{A} \models A \vee B$ sse $\mathfrak{A} \models A$ oppure $\mathfrak{A} \models B$;
7. $\mathfrak{A} \models (A \implies B)$ sse $\mathfrak{A} \models A$ implica $\mathfrak{A} \models B$;

Soddisfacibilità: definizione 2

8. $\mathfrak{A} \models (A \Leftrightarrow B)$ sse $\mathfrak{A} \models A$ e $\mathfrak{A} \models B$ oppure $\mathfrak{A}, \not\models A$ e $\mathfrak{A} \not\models B$;
9. $\mathfrak{A} \models \forall x A$ sse **per ogni** $d \in D$ è verificato che $\mathfrak{A} \models A\{d = x\}$);
10. $\mathfrak{A} \models \exists x A$ sse **esiste un** $d \in D$ per cui è verificato che $\mathfrak{A} \models A\{d = x\}$.

$A\{d = x\}$ denota la sostituzione delle occorrenze di x con d.

Modelli, validità, soddisfacibilità

Sia A una formula chiusa. Se $\mathfrak{A} \models A$ diciamo che $\mathfrak{A}$ è un *modello* di A, ovvero che A è *vera* in $\mathfrak{A}$.

Una formula $A \in \mathcal{L}$ è *valida* sse è vera in tutte le strutture di $\mathcal{L}$ e lo scriviamo $\models A$.

Un insieme di formule Γ è *soddisfacibile* se esiste una struttura $\mathfrak{A}$, tale che $\mathfrak{A} \models A$ per ogni $A \in \Gamma$.

La validità e la soddisfacilità non possono essere verificate facilmente costruendo la tabella di verità!

Conseguenza logica, Equivalenza

Sia Γ un insieme di formule e A una formula chiusa. Allora Γ *implica logicamente* A, scritto $\Gamma \models A$, sse per ogni struttura $\mathfrak{A}$ del linguaggio $\mathfrak{A} \models B$ per ogni $B \in \Gamma$ è verificato che $\mathfrak{A} \models A$.

Possiamo dire in modo equivalente che $\Gamma \cup \neg A$ è insoddisfacibile.

Risoluzione

- $\diamondsuit$ riconduce $\Gamma \models A$ alla verifica che $\Gamma \cup \neg A$ è insoddisfacibile

- $\diamondsuit$ richiede la costruzione della **conjunctive normal form** (CNF) spostando i quatificatori universali all'esterno ed eliminando i quantificatori esistenziali con la skolemizzazione

Conversione in CNF

Chi ama tutti gli animali ha qualcuno che lo ama:

$$\forall\, x\ \ [\forall\, y\ \ Animal(y) \implies Loves(x,y)] \implies [\exists\, y\ \ Loves(y,x)]$$

1. Eliminazione delle implicazioni semplici e doppie

$$\forall\, x\ \ [\neg\forall\, y\ \ \neg Animal(y) \vee Loves(x,y)] \vee [\exists\, y\ \ Loves(y,x)]$$

2. Spingere $\neg$ all'interno: $\neg\forall\, x, p \quad \equiv \exists\, x \quad \neg p$, $\quad \neg\exists\, x, p \quad \equiv$ $\forall\, x\ \ \neg p$:

$$\forall\, x\ \ [\exists\, y\ \ \neg(\neg Animal(y) \vee Loves(x,y))] \vee [\exists\, y\ \ Loves(y,x)]$$
$$\forall\, x\ \ [\exists\, y\ \ \neg\neg Animal(y) \wedge \neg Loves(x,y)] \vee [\exists\, y\ \ Loves(y,x)]$$
$$\forall\, x\ \ [\exists\, y\ \ Animal(y) \wedge \neg Loves(x,y)] \vee [\exists\, y\ \ Loves(y,x)]$$

Conversione in CNF (2)

3. Standardizzare le variabili: ogni quantificatore con una variabile diversa

$$\forall x \; [\exists y \; Animal(y) \land \neg Loves(x,y)] \lor [\exists z \; Loves(z,x)]$$

4. Skolemizzazione: le variabili quantificate esistenzialmente sono sostituite da funzioni di Skolem delle variabii unversali più esterne:

$$\forall x \; [Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x)$$

5. Si eliminano i quantificatori universali:

$$[Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x)$$

6. Si distribuiscono $\land$ su $\lor$:

$$[Animal(F(x)) \lor Loves(G(x), x)] \land [\neg Loves(x, F(x)) \lor Loves(G(x), x)]$$

Proprietà della trasformazione

Tutti i passaggi preservano l'equivalenza tranne la Skolemizzazione che preserva la soddisfacibilità.

Sia ϕ e ϕ^{sko} la sua forma normale di Skolem

Tutti i modelli di ϕ^{sko} sono anche modelli di ϕ, ma non il viceversa. Quindi:

- ♢ ϕ soddisfacibile <u>sse</u> ϕ^{sko} soddisfacibile
- ♢ ϕ non soddisfacibile <u>sse</u> ϕ^{sko} non soddisfacibile
- ♢ ma ϕ valida <u>NON implica</u> ϕ^{sko} valida

Risoluzione

Binaria:

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{(\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)\theta}$$

dove:

- ◇ al numeratore compaiono 2 clausole dette **genitrici**
- ◇ al denominatore compare una clausola detta **risolvente**
- ◇ Unify$(\ell_i, \neg m_j) = \theta$

Fattorizzazione

Risoluzione binaria, o ground:

- ◇ elimina un solo letterale per volta
- ◇ è completa per la logica proposizionale
- ◇ non è completa invece per la logica del primo ordine.

Occorre definire una versione della regola di risoluzione che incorpora la **fattorizzazione**, ossia la possibilità di eliminare più di 2 letterali attraverso un solo passo di risoluzione.

Dimostrazioni tramite la risoluzione

Sia $\mathcal{S}$ un insieme finito di clausole, e C una clausola.

C *si dimostra da* $\mathcal{S}$ *per G-risoluzione* sse
$\diamondsuit$ esiste un albero di risoluzione la cui radice è C,
$\diamondsuit$ tutte le foglie dell'albero sono clausole di $\mathcal{S}$
$\diamondsuit$ ogni nodo è ottenuto applicando unicamente la regola di risoluzione.

$\mathcal{S} \vdash_{GLR} C$ indica che $\mathcal{S}$ dimostra C per G-risoluzione.

$$\mathcal{S} \text{ insoddisfacibile implica } \mathcal{S} \vdash_{GLR} \{\}$$

Completezza della risoluzione

<u>Teorema</u>

Sia $\phi(x_1, \ldots, x_n)$ un insieme di clausole. $\forall x_1 \ldots x_n \phi$ è insoddisfacibile sse $\phi \vdash_{GLR} \{\}$.

La risoluzione è **completa per la refutazione**:

> se S è un insieme insoddisfacibile di formule in forma di clausole, allora l'applicazione a S di un numero finito di passi di risoluzione porterà ad una contraddizione.

Schema di dimostrazione

1. Se S è insoddisfacibile, allora esiste un particolare insieme di *istanze ground* della formula S, tali che questo insieme è anche insoddisfacibile (teorema di Herbrand).
2. La risoluzione è completa per formule ground (Facile poiché l'insieme di conseguenze di una teoria proposizionale è sempre finito.)
3. **Lemma di lifting**: per ogni dimostrazione di risoluzione che usa un insieme di formule ground, esiste una dimostrazione corrispondente che usa formule della logica del primo ordine da cui sono state ottenute le formule ground.

Schema di dimostrazione

Qualsiasi insieme di frasi S è rappresentabile in forma clausale

Si assume che S sia insoddisfacibile ed in forma clausale

Teorema di Herbrand

Qualche insieme S’ di istanze ground è insoddisfacibile

Teorema di risoluzione ground

La risoluzione può trovare una contraddizione in S’

Lemma di lifting

Vi è una dimostrazione per risoluzione per la contraddizione in S

Significato dei quantificatori

Su un dominio finito:
un quantificatore universale $\forall$ si comporta come un $\wedge$
un quantificatore esistenziale $\exists$ si comporta come un $\vee$.

$\forall x P(x)$ si può scrivere come: $P(x_1) \wedge \ldots \wedge P(x_n)$

$\exists x P(x)$ si può scrivere come: $P(x_1) \vee \ldots \vee P(x_n)$

dove n è la cardinalità di D

IMPORTANTE: È possibile ridursi sempre a domini finiti, cioè come ricondurre la valutazione della verità per le formule del primo ordine a un'analisi delle sole strutture finite.

Universo di Herbrand

Se S è un insieme di clausole, allora H_S, l'**Universo di Herbrand** di S, è l'insieme di tutti i termini ground costruibili come segue:

a. I simboli di funzione in S, se ve ne sono.

b. I simboli di costante in S, se ve ne sono; se non ve ne sono allora il simbolo di costante A.

Esempio

Se S contiene solo la clausola

$$P(x, F(x, A)) \wedge Q(x, A) \implies R(x, B)$$

allora H_S è il seguente insieme infinito di formule ground:

$$\{A, B, F(A, A), F(A, B), F(B, A), F(B, B),$$
$$F(A, F(A, A)), F(A, F(A, B)), \ldots\}$$

Base di Herbrand

Saturazione: se S è un insieme di clausole e P un insieme di termini ground, allora $P(S)$, la saturazione di S rispetto a P, è l'insieme di tutte le clausole ground ottenute applicando tutte le possibili sostituzioni consistenti per le variabili in S con termini ground in P.

Base di Herbrand: la saturazione di un insieme di clausole S in relazione al proprio universo di Herbrand è chiamata base di Herbrand di S, scritta come $H_S(S)$.

Esempio

$H_S(S) = \{P(A, F(A, A)) \wedge Q(A, A) \implies R(A, B),$
$P(B, F(B, A)) \wedge Q(B, A) \implies R(B, B),$
$P(F(A, A), F(F(A, A), A)) \wedge Q(F(A, A), A) \implies R(F(A, A), B),$
$P(F(A, B), F(F(A, B), A)) \wedge Q(F(A, B), A) \implies R(F(A, B), B),$
$\ldots\}$

Si noti che sia l'universo di Herbrand che la base di Herbrand possono essere infiniti, anche se l'insieme originale di formule S è finito.

Teorema di Herbrand

Teorema di Herbrand(1930):

> Se un insieme di clausole S è insoddisfacibile, allora esiste un sottinsieme finito di $H_S(S)$ che è a sua volta insoddisfacibile

Il sottoinsieme finito è l'espansione di Herbrand.

Chiusura della risoluzione

Sia S' un sottinsieme finito di $H_S(S)$. La **chiusura della risoluzione** di S', che è l'insieme di tutte le clausole derivabili dall'applicazione ripetuta del passo di inferenza di risoluzione alle clausole in S' o alle loro derivate.

Sia T la chiusura di risoluzione di S', sia $A_{S'} = \{A_1, A_2, \ldots, A_k\}$ l'insieme delle formule atomiche che compaiono in S'. Si noti che siccome S' è finito, anche $A_{S'}$ deve essere finito.

$$\begin{aligned}
S' &= \{\ P(A),\ P(A) \Longrightarrow Q(A),\ Q(A) \Longrightarrow Falso\ \} \\
A_{S'} &= \{P(A),\ Q(A),\ Falso\} \\
T &= \{\ P(A),\ P(A) \Longrightarrow Q(A),\ Q(A) \Longrightarrow Falso, \\
&\qquad Q(A),\ P(A) \Longrightarrow Falso, Falso\ \}
\end{aligned}$$

Completezza della risoluzione ground

Teorema di risoluzione ground:

Se un insieme di clausole ground è insoddisfacibile, allora la chiusura per risoluzione di quelle clausole contiene la clausola $Falso$.

Dimostrazione: principio

Si dimostra il contrappositivo: se la chiusura T *non* contiene $Falso$, allora S' è soddisfacibile. Costruiamo un'assegnazione per le formule atomiche in S' prendendo un'assegnazione ($Vero$ o $Falso$) per ogni formula atomica in $A_{S'}$ secondo un qualche ordine fissato $A_1, \ldots, A_k$:

- Se vi è una clausola in T contenente il letterale $\neg A_i$, tale che tutti gli altri letterali sono falsi secondo l'assegnazione scelta per $A_1, \ldots, A_{i-1}$, allora si assegni ad A_i $Falso$.
- Altrimenti si assegni ad A_i $Vero$.

L'assegnazione così costruita soddisferà S', assodato che T è chiuso secondo la risoluzione e che non contiene la clausola $Falso$.

Lemma di Lifting

Cè sempre una dimostrazione con risoluzione che coinvolge qualche sottinsieme finito della base di Herbrand S.

> Siano C_1 e C_2 due clausole senza alcuna variabile condivisa, siano inoltre C_1' e C_2' istanze ground di C_1 e C_2. Se C' è un risolvente di C_1' e C_2', allora esiste una clausola C tale che (1) C è un risolvente di C_1 e C_2, (2) C' è un'istanza ground di C.

Robinson ha inventato l'unificazione e mostrato tutte le proprietà degli unificatori più generali che consentono la prova di questo lemma.

Esempio

$$
\begin{aligned}
C_1 &= P(x, F(x, A)) \wedge Q(x, A) \implies R(x, B) \\
C_2 &= N(G(y), z) \implies P(H(y), z) \\
C_1' &= P(H(B), F(H(B), A)) \wedge Q(H(B), A) \implies R(H(B), B) \\
C_2' &= N(G(B), F(H(B), A)) \implies P(H(B), F(H(B), A)) \\
C' &= N(G(B), F(H(B), A)) \wedge Q(H(B), A) \implies R(H(B), B) \\
C &= N(G(y), F(H(y), A)) \wedge Q(H(y), A) \implies R(H(y), B)
\end{aligned}
$$

C' è un'istanza ground di C

C_1' e C_2' hanno dei risolventi, se ottenuti applicando a C_1 e C_2 l'mgu di una coppia di letterali complementari in C_1 e C_2.

Dal lemma di lifting alla risoluzione

Per qualsiasi clausola C' nella chiusura per risoluzione di S', vi è una clausola C nella chiusura per risoluzione di S, tale che C' è un'istanza ground di C e che la derivazione di C è della stessa lunghezza della derivazione di C'.

Ricapitolando: abbiamo mostrato che se S è insoddisfacibile, allora vi è una derivazione finita della clausola $Falso$ usando la regola di risoluzione.

Risoluzione con l'uguaglianza

paramodulazione:

$$\frac{\ell_1 \vee \cdots \vee \ell_k \vee x = y, \qquad m_1 \vee \cdots \vee m_n[z]}{(\ell_1 \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_n[y])\theta}$$

dove Unify$(x, z) = \theta$

in alternativa:

$\diamondsuit$ si estende l'unificazione in modo da unificare tenendo conto delle proprietà dell'uguaglianza: **Unificazione equazionale**

Dimostrazioni non costruttive

$\neg P(x,a)$
$P(b,a) \vee P(c,a)$

Applicando la risoluzione si ottiene:
$P(c,a)$ e poi la clausola vuota.

Si introduce, un predicato $Answer$ con le variabili dell'interrogazione:
$\neg P(x,a) \vee Answer(x)$

che restituisce solo le dimostrazioni ottenute assegnando un solo valore alle variabili.

Strategie di risoluzione

La risoluzione è un metodo di prova molto inefficiente: il numero di clausole generate durante una prova può infatti crescere a dismisura.

◇ Strategie:

<u>unitaria (unit)</u>
Si cerca di usare sempre una clausola con un solo letterale. Completa per clausole definite, incompleta in generale.

<u>insieme di supporto (set of support)</u>
Almeno una delle due clausole genitrici proviene dal negato del goal. La completezza dipende dall'insieme di supporto.

input

Una delle due clausole è l'ultima generata l'altra proviene dall'insieme delle clausole iniziali

Non è completa per le clausole generali, ma solo per quelle definite.

lineare

Una delle due clausole è l'ultima generata