Linguaggi

16: Risoluzione proposizionale

Claudio Sacerdoti Coen

<sacerdot@cs.unibo.it>

Universitá di Bologna

06/04/2011

Outline

1 Risoluzione proposizionale

Risoluzione proposizionale

Wikipedia: "*...*"

Premessa e motivazioni

La prova costruttiva del teorema di completezza per la deduzione naturale classica fornisce un algoritmo automatico di ricerca di prove.

Tuttavia l'algoritmo è inefficiente (complessità esponenziale) e trova prove complesse (di taglia esponenziale) e innaturali.

Pertanto, a tale algoritmo si preferiscono "banali" ricerche esaustive (che tengano presente l'invertibilità delle regole ed evitino la divergenza quando una prova esiste).

Lo spazio della ricerca esaustiva è molto ampio: troppe regole applicabili in ogni momento e troppe regole che richiedono di individuare le giuste premesse.

Premessa e motivazioni

PROBLEMA: esistono sistemi deduttivi alternativi caratterizzati da uno spazio di ricerca più limitato?

IDEA: il numero di regole applicabili può essere probabilmente limitato considerando solamente premesse e conclusioni in forme particolari (p.e. forme normali congiuntive, disgiuntive, di Shannon).

Cerchiamo ora di individuare la forma normale più appropriata fra quelle a noi note.

Premessa e motivazioni

Osservazioni:

1. Per via della regola $\wedge_e$, una dimostrazione di $F \wedge G, \Gamma \vdash H$ può essere ridotta a una dimostrazione di $F, G, \Gamma \vdash H$ decomponendo la premessa $F \wedge G$ nei suoi fattori F e G.
2. Poichè conosciamo un algoritmo efficiente per calcolare la forma normale congiuntiva di una formula data, possiamo ridurci a alberi di derivazione $\Gamma \vdash F$ dove tutte le premesse sono in CNF, ovvero sono o $\top$ oppure congiunzioni di disgiunzioni di letterali.
3. Le premesse $\top$ sono inutili e possono essere ignorate.
4. Le premesse che sono congiunzioni di disgiunzioni possono essere decomposte come spiegato al punto 1 in fattori che sono tutti disgiunzioni di letterali.

Quindi classicamente possiamo efficientemente ridurci al caso $\Gamma \vdash F$ dove tutte le premesse sono disgiunzioni di letterali.

Premessa e motivazioni

Ulteriori osservazioni:

1. Usando la RAA, ogni dimostrazione di $\Gamma \vdash F$ può essere ridotta a una dimostrazione di $\Gamma, \neg F \vdash \bot$.
2. Effettuando questa operazione prima di mettere le premesse in CNF e decomporle, ci si può sempre ridurre al caso $\Gamma \vdash \bot$ dove tutte le premesse sono disgiunzioni di letterali.
3. Una disgiunzione di letterali $L_1 \vee (L_2 \vee (\ldots \vee L_n) \ldots)$ è caratterizzata da un ordine che non è importante in quanto sappiamo che $\vee$ gode delle proprietà associativa, commutativa e idempotente.
4. Pertanto, possiamo ridurci a pensare a una digiunzione di letterali come a un insieme $\{L_1, \ldots, L_n\}$ (l'insieme è vuoto nel caso $\bot$). Indichiamo l'insieme vuoto con $\square$.

Risoluzione proposizionale

Definizione: un letterale è una variabile proposizionale A o una variabile proposizionale negata $\neg A$.

Definizione: una clausola è o la clausola vuota $\square$ o un insieme non vuoto di letterali che non contiene sia una variabile proposizionale che la sua negata.

Algoritmo: per ogni Γ e F possiamo costruire un insieme di clausole $C_1, \ldots, C_k$ ottenute da Γ e F aggiungendo $\neg F$ a Γ, mettendo tutte le formule in Γ in CNF e trasformandole in clausole decomponendo le congiunzioni $\neq \top$ ed eliminando le disgiunzioni che contengono variabili proposizionali negate o meno.

Vogliamo cercare di costruire alberi di derivazione la cui radice sia $\square$ e le cui foglie siano alcune delle clausole $C_1, \ldots, C_k$.

Risoluzione proposizionale

Domanda: esistono regole di derivazione localmente corrette e complete le cui premesse e conclusione siano tutte clausole?

Si', BASTA UNA SOLA REGOLA (regola di risoluzione):

$$\frac{C_1 \quad C_2}{(C_1 \setminus \{A\}) \cup (C_2 \setminus \{\neg A\})} \qquad \text{(RES)}$$

dove $A \in C_1$ e $\neg A \in C_2$.

Lettura bottom up: non intuitiva. Vedi teorema di correttezza locale.

Le prove per risoluzione sono sempre costruite in maniera bottom-up. Pertanto non siamo interessati all'invertibilità.

Risoluzione proposizionale

Esempio: dimostrare per risoluzione proposizionale

$$A \Rightarrow B \vee C, \neg B \Rightarrow A \vdash \neg C \wedge \neg D \Rightarrow B$$

Passo 1: per deduzione semantica sposto le premesse della conclusione fra le ipotesi: $A \Rightarrow B \vee C, \neg B \Rightarrow A, \neg C \wedge \neg D \vdash B$
Passo 2: sposto la conclusione negata fra le premesse:
$A \Rightarrow B \vee C, \neg B \Rightarrow A, \neg C \wedge \neg D, \neg B$
Passo 3: metto le formule in CNF:
$\neg A \vee B \vee C, \neg\neg B \vee A, \neg C \wedge \neg D, \neg B$
Passo 4: formo le clausole decomponendo le congiunzioni:
$\{\neg A, B, C\}, \{B, A\}, \{\neg C\}, \{\neg D\}, \{\neg B\}$
Passo 5: costruisco l'albero di derivazione per risoluzione

Risoluzione proposizionale

Esempio: dimostrare per risoluzione proposizionale

$$A \Rightarrow B \vee C, \neg B \Rightarrow A \vdash \neg C \wedge \neg D \Rightarrow B$$

Passo 5: costruisco l'albero di derivazione per risoluzione a partire da $\{\neg A, B, C\}, \{B, A\}, \{\neg C\}, \{\neg D\}, \{\neg B\}$

$$\dfrac{\dfrac{\dfrac{\{\neg A, B, C\} \quad \{B, A\}}{\{B, C\}} \quad \{\neg C\}}{\{B\}} \quad \{\neg B\}}{\square}$$

Risoluzione proposizionale

Esercizio: dimostrare per risoluzione proposizionale

$$\vdash (\neg A \Rightarrow B \Rightarrow A) \Rightarrow ((A \Rightarrow C \wedge \neg A) \Rightarrow \neg B \vee D)$$

Teorema di correttezza

Teorema di correttezza locale della regola di risoluzione:

$$C_1, C_2 \Vdash (C_1 \setminus \{A\}) \vee (C_2 \setminus \{\neg A\})$$

dove $A \in C_1$ e $\neg A \in C_2$.

Dimostrazione:
sia $D_1 = C_1 \setminus \{A\}$ e $D_2 = C_2 \setminus \{\neg A\}$. Dimostriamo $A \vee D_1, \neg A \vee D_2 \vdash D_1 \vee D_2$ usando la deduzione naturale. Il teorema segue per correttezza della deduzione naturale.

$$\dfrac{A \vee D_1 \qquad \dfrac{\neg A \vee D_2 \qquad \dfrac{\dfrac{[\neg A] \quad [A]}{\bot}\,(\neg_e)}{D_2}\,(\bot_e) \qquad [D_2]}{D_2}\,(\vee_e)}{D_1 \vee D_2}\,(\vee_{i_2}) \qquad \dfrac{[D_1]}{D_1 \vee D_2}}{D_1 \vee D_2}$$

Teorema di correttezza

Teorema di corretta per la risoluzione proposizionale: se $C_1, \ldots, C_k \vdash C$ usando solamente la regola di risoluzione, allora $C_1, \ldots, C_k \Vdash C$.

Dimostrazione: per induzione strutturale sull'albero di prova.

- Caso C: si ha $C \in \{C_1, \ldots, C_k\}$ e $C_1, \ldots, C_k \Vdash C$.
- Caso $D_1, D_2 \vdash (D_1 \setminus \{A\}) \cup (D_2 \setminus \{\neg A\})$:
  per ipotesi induttiva si ha
  $C_1, \ldots, C_k \Vdash D_1$ e $C_1, \ldots, C_k \Vdash D_2$
  e, per il teorema di correttezza locale, si ha
  $D_1, D_2 \Vdash (D_1 \setminus \{A\}) \cup (D_2 \setminus \{\neg A\})$.
  Per transitività della conseguenza logica si ha quindi
  $C_1, \ldots, C_k \Vdash (D_1 \setminus \{A\}) \cup (D_2 \setminus \{\neg A\})$.

Ricerca automatica di dimostrazioni

Come si può cercare automaticamente una derivazione $C_1, \ldots, C_k \vdash \Box$?

Algoritmo brute search:
siano

$$S_0 = \{C_1, \ldots, C_k\}$$
$$S_{i+1} = S_i \cup \{(D_1 \setminus \{A\}) \cup (D_2 \setminus \{\neg A\}) \mid D_1, D_2 \in S_i \text{ e } A \in Var\}$$

Terminazione:
Poichè le variabili proposizionali contenuti in ogni clausola di S_i per ogni i sono già contenuti nelle clausole di S_0 e sono finite, esistono solamente un numero finito di clausole distinte. Poichè $S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots$ esiste un $\bar{k}$ tale che $S_{\bar{k}} = S_{\bar{k}+1}$. Questo è il caso di terminazione dell'algoritmo, che ha generato tutte le possibili clausole a partire da S_0.

Teorema di completezza

Teorema di completezza: se $C_1, \ldots, C_k \Vdash \bot$ allora $C_1, \ldots, C_k \vdash \Box$ usando il metodo di risoluzione.

Dimostrazione: è sufficiente dimostrare che $\Box \in S_{\bar{k}}$ definito come nel lucido precedente. Procediamo per induzione sul numero n di variabili proposizionali che occorrono in S_0.

Caso $n = 0$: $S_0 = \{\Box\}$ e $\bar{k} = 0$ e trivialmente $\Box \in S_0$.

Caso $n + 1$:
per ipotesi induttiva tutti gli insiemi S_0 di clausole insoddisfacibili in cui occorrano solo le variabili proposizionali $\{A_1, \ldots, A_n\}$ sono dimostrabili essere tali per risoluzione. Consideriamo un insieme insoddisfacibile S_0 in cui occorrano $\{A_1, \ldots, A_n, A_{n+1}\}$ e sia $S_{\bar{k}}$ il relativo insieme massimale di clausole deducibili per risoluzione.

Teorema di completezza

Definiamo:

$$S_f = \{C \setminus \{A_{n+1}\} \mid \neg A_{n+1} \notin C \wedge C \in S_{\bar{k}}\}$$
$$S_t = \{C \setminus \{\neg A_{n+1}\} \mid A_{n+1} \notin C \wedge C \in S_{\bar{k}}\}$$

Si ha $S_f \Vdash \bot$. Supponiamo infatti per assurdo che $v \Vdash S_f$. Allora per $v' = v[A \mapsto 0]$ si ha $v' \Vdash S_{\bar{k}} \supseteq S_0$ (il che è assurdo). Infatti $v' \Vdash S_{\bar{k}}$ in quanto $v' \Vdash C$ t.c. $C \in S_{\bar{k}}$ e $\neg A_{n+1} \in C$ e inoltre se $v \Vdash C \setminus \{A_{n+1}\}$ ovvero $\max [\![F]\!]^v = 1$ per $F \in C \setminus \{A_{n+1}\}$ si ha anche $v' \Vdash C$ (si aggiunge una formula a una disgiunzione già vera).

Similmente si dimostra $S_t \Vdash \bot$.

Poichè sia in S_f che in S_t occorrono solo le variabili $\{A_1, \ldots, A_n\}$, per ipotesi induttiva $S_f \vdash \Box$ e $S_t \vdash \Box$.

Teorema di completezza

Considero ora
$S_f^* = \{C \cup \{A_{n+1}\} \mid C \in S_f\} \cap S_0 \subseteq S_0$
$S_t^* = \{C \cup \{\neg A_{n+1}\} \mid C \in S_t\} \cap S_0 \subseteq S_0$.

Si ha $S_0 = S_f^* \cup S_t^*$.

L'albero di derivazione $S_f \vdash \Box$ in cui aggiungo A_{n+1} in ogni clausola dalla quale era stata cancellata è un albero di derivazione per $S_f^* \vdash \Box$ (e in questo caso ho finito) oppure per $S_f^* \vdash \{A_{n+1}\}$ (lo si dimostra trivialmente per induzione strutturale sull'albero di derivazione) e quindi anche per $S_0 \vdash \{A_{n+1}\}$.

Similmente, l'albero di derivazione $S_t \vdash \Box$ in cui aggiungo $\neg A_{n+1}$ in ogni clausola dalla quale era stata cancellata è un albero di derivazione per $S_t^* \vdash \Box$ (e in questo caso ho finito) oppure per $S_t^* \vdash \{\neg A_{n+1}\}$ e quindi anche per $S_0 \vdash \{\neg A_{n+1}\}$.

Infine, con un ultimo passo di risoluzione, dimostro $S_0 \vdash \Box$.