Bollettino
Unione Matematica Italiana

Sezione A – La Matematica nella Società e nella Cultura

Renato Bruni

Elementi di programmazione logica

Bollettino dell'Unione Matematica Italiana, Serie 8, Vol. **4-A**—*La Matematica nella Società e nella Cultura* (2001), n.3 (Fascicolo Tesi di Dottorato), p. 415–418.

Unione Matematica Italiana

<http://www.bdim.eu/item?id=BUMI_2001_8_4A_3_415_0>

Bollettino dell'Unione Matematica Italiana, Unione Matematica Italiana, 2001.

Bollettino U. M. I.
La Matematica nella Società e nella Cultura
Serie VIII, Vol. IV-A, Dicembre 2001, 415-418

Elementi di programmazione logica.

Renato Bruni

1. – Introduzione.

Qualsiasi formula logica proposizionale può essere posta in *forma normale congiuntiva* (CNF), ovvero in forma $\mathcal{F}$ di congiunzione di m *clausole* (C_j), dove ogni clausola è una disgiunzione di *letterali*, ed ogni letterale è una *proposizione diretta* (x_i) o *negata* ($\neg x_i$). Indicando con l_j la cardinalità di C_j, e con $[\neg]$ la possibile presenza di $\neg$, si ha cioè

$$\mathcal{F} = \bigwedge_{j=1\dots m} \left(\bigvee_{i=1\dots l_j} [\neg]\, x_i \right) \tag{1}$$

Un'assegnazione dei valori di verità $\{Vero, Falso\}$ per le proposizioni fornisce un valore di verità per la formula. Una formula $\mathcal{F}$ è *soddisfacibile* se esiste un'assegnazione di verità tale che renda $\mathcal{F}$ *Vera*. Se tale assegnazione non esiste, $\mathcal{F}$ è *insoddisfacibile*. Il problema di determinare se una formula logica proposizionale sia soddisfacibile o meno, detto problema della *soddisfacibilità* (SAT), è di fondamentale importanza sia teorica che pratica (cfr. bibliografia della tesi). Il problema dell'*implicazione logica* può essere rappresentato come problema di soddisfacibilità: dato un insieme di affermazioni S, ed una singola affermazione s, $S \Rightarrow s$ se e solo se $S \cup \neg s$ costituisce una formula insoddisfacibile [3]. Inoltre, SAT è il prototipo di problema NP-completo, ed esiste tutta una serie di problemi la cui soluzione può essere ricondotta alla risoluzione di un problema di SAT. Infine, molti problemi pratici (*intelligenza artificiale*, *verifica dei circuiti logici*, etc.) vengono formulati come problemi di SAT.

Nei tre seguenti paragrafi vengono brevemente illustrati i principali contributi della tesi. I primi due sono di tipo algoritmico computazionale, mentre il terzo di tipo modellistico applicativo.

2. – Un Algoritmo Completo Adattativo per il Problema della Soddisfacibilità.

La maggior parte degli algoritmi moderni completi utilizza tecniche di enumerazione implicita generando un albero di ricerca, secondo uno schema generale detto «Davis-Putnam-Loveland» [3]. L'algoritmo proposto fa uso di una variante di albero di ricerca basato sulle clausole. Ad ogni iterazione, viene selezionata una clausola C_s per essere soddisfatta, e vengono quindi scelte le attribuzioni di verità corrispondenti ai letterali presenti in C_s, tali perciò da soddisfare quella clausola.

Nel corso di questa enumerazione, l'analisi dell'esplorazione dell'albero di ricerca viene utilizzata per fornire informazioni riguardo alla struttura della for-

mula, ed in particolare per valutare la difficoltà di soddisfare le singole clausole nel contesto della formula. Diremo che una clausola C_j è stata *visitata* durante l'esplorazione dell'albero di ricerca se viene fatta un'assegnazione di verità tesa a soddisfare C_j. Diremo che sulla clausola C_j si è avuto un *fallimento* sia quando un'assegnazione di verità tesa a soddisfare C_j produce una clausola vuota, sia quando C_j diventa una clausola vuota a causa di altre attribuzioni di verità.

VALUTAZIONE ADATTATIVA DELLA DIFFICOLTÀ DI UNA CLAUSOLA. – *Siano v_j ed f_j rispettivamente numero di visite e di fallimenti per C_j, p una penalità considerata per i fallimenti, e l_j la lunghezza di C_j. Una valutazione della difficoltà di C_j in $\mathcal{F}$ è*

$$\varphi(C_j) = (v_j + pf_j)/l_j. \tag{2}$$

Tale valutazione ha l'importante proprietà di richiedere un *overhead* computazionale molto ridotto, e la sua qualità migliora al procedere dell'esplorazione dell'albero.

Poichè è generalmente riconosciuto che esplorare l'albero di ricerca partendo dalle clausole più difficili riduce notevolmente i tempi di calcolo, dopo il soddisfacimento di tutte le clausole unitarie, vengono soddisfatte le clausole con i valori massimi di φ. L'algoritmo inoltre lavora solo sul sottoinsieme corrente delle clausole più difficili, detto *core*, in modo da ridurre i tempi necessari alla propagazione delle assegnazioni di verità. Tale *core* viene periodicamente aggiornato, variandone la composizione e la dimensione, in modo da selezionare progressivamente la parte più difficile della formula. L'uso di un *core* permette inoltre di dimostrare l'insoddisfacibilità della formula non appena il *core* diventa insoddisfacibile. Si noti che, in molte applicazioni pratiche (tra cui quella di cui al par. 4), l'individuazione di questo sottoinsieme insoddisfacibile di clausole ha grande importanza.

Viene quindi presentato un algoritmo completo ed esatto per SAT, denominato *adaptive core search*. In base alla scelta dei parametri, tale algoritmo risolve efficientemente SAT, ed in tal caso i risultati computazionali mostrano che la procedura è competitiva con i migliori algoritmi attualmente esistenti per SAT, oppure individua sottoformule insoddisfacibili di dimensione ridotta [1].

3. – Ortogonalizzazione di una Formula Logica.

Due clausole C_i e C_j sono *ortogonali* tra loro se almeno una variabile logica x_k compare diretta in una (ad esempio C_i) e negata nell'altra (ad esempio C_j):

$$C_i = (\ldots \vee x_k \vee \ldots), \quad C_j = (\ldots \vee \neg x_k \vee \ldots) \tag{3}$$

Una forma normale *congiuntiva* è *ortogonale* se tutte le clausole che la costituiscono sono ortogonali a coppie. La forma normale *disgiuntiva ortogonale* è definita analogamente, e tutti i seguenti risultati sono immediatamente estendibili. Questa proprietà è di grande importanza in diverse applicazioni, ad esempio nel campo della teoria della *reliability*.

Ogni clausola C_i ha un insieme V_i di punti (i.e. attribuzioni di verità) dell'*ipercubo booleano* B^n dove vale *Vero*, ed un insieme F_i di punti di B^n dove vale *Falso*. Similmente, la formula $\mathcal{F}$ ha un insieme di *punti veri* V ed un insieme di *punti fal-*

si F. Per formule in CNF si ha $V = \bigcap_i V_i$ e $F = \bigcup_i F_i$. In generale, gli insiemi V_i non sono necessariamente disgiunti. Lo stesso accade per gli insiemi F_i. Per tale motivo, molti problemi pratici riconducibili al calcolo della cardinalità degli insiemi V ed F sono difficili, perché occorre identificare rispettivamente tutti i V_i e tutti gli F_i. Al contrario, le cardinalità di tali V_i ed F_i sono facilmente calcolabili. Sia infatti l_i il numero di letterali presenti nella clausola C_i, si ha $Card\{F_i\} = 2^{n-l_i}$, e $Card\{V_i\} = 2^n - Card\{F_i\}$.

Per una formula CNF ortogonale vale l'importante proprietà che gli insiemi F_i sono mutuamente disgiunti.

$$F_i \cap F_j = \phi \qquad \forall\, i, j \in \{1 \ldots m\}. \tag{4}$$

Risulta allora elementarmente $Card\{F\} = \sum_i Card\{F_i\}$, e $Card\{V\} = 2^n - Card\{F\}$.

Viene presentata una procedura per trasformare un'arbitraria CNF in forma ortogonale. Viene inizialmente presentata una operazione di ortogonalizzazione tra coppie di clausole. Applicando iterativamente tale operazione è teoricamente possibile ortogonalizzare una qualsiasi formula CNF. Dal punto di vista pratico, però, questo richiede tempi di calcolo proibitivi, e soprattutto un aumento esponenziale delle dimensioni della formula. Viene allora proposta una procedura che, sfruttando la presenza di insiemi di clausole già ortogonali ed alcune operazioni di semplificazione, è in grado di ridurre l'aumento delle dimensioni della formula. Risultati computazionali mostrano il comportamento favorevole della procedura proposta.

4. – Logica ed Ottimizzazione per Individuazione e Correzione di Errori in Basi di Dati di Grandi Dimensioni.

In tutti i casi di trattamento di dati che possano essere affetti da errori (indagini statistiche, analisi di mercato, misure sperimentali, etc.) sorge il problema di calcolare i risultati considerando dati corretti. Viene in particolare affrontato il caso di un censimento della popolazione. Un *record* di dati, cioè un insieme di valori v_i per un insieme di campi f_i, è in questo caso un questionario Q.

$$Q = \{f_1 = v_1,\ f_2 = v_2,\ \ldots,\ f_p = v_p\}. \tag{5}$$

Gli errori, o, più precisamente, le risposte non ammissibili o le inconsistenze tra le risposte, possono essere dovuti alla compilazione originale del questionario, o introdotti in una qualsiasi fase successiva di immissione o conversione dei dati. Il problema dell'*individuazione* degli errori è generalmente affrontato formulando delle condizioni di errore (dette *edit* [2]) che i *record corretti* devono non verificare. Un esempio di record errato è il seguente: stato civile = sposato e età = 10 anni. L'*edit* corrispondente potrebbe essere:

$$(\text{stato civile} = \text{sposato}) \wedge (\text{età} < 14). \tag{6}$$

L'insieme degli *edit* non deve ovviamente contenere inconsistenze tra *edit* o *edit* ridondanti. Nei casi reali, gli *edit* sono molto numerosi, e tale verifica diventa un problema computazionalmente oneroso. Esistono diversi sistemi software per la correzione automatica, ma nella pratica soffrono di pesanti limitazioni.

Viene proposta una soluzione mediante la codifica degli *edit* attraverso la logica proposizionale. Il problema del controllo della consistenza e non ridondanza dell'insieme degli *edit* si trasforma allora in una sequenza di problemi di soddisfacibilità, che vengono efficacemente risolti col solutore di cui al par.1. Il problema di individuare i record errati diventa il banale problema di valutare se l'attribuzione di verità corrispondente ad un questionario soddisfa la formula logica ottenuta dall'insieme degli *edit*. Dato che generalmente la raccolta dati ha un costo, si desidera utilizzare, per quanto possibile, anche i *record* errati, effettuando la *correzione* degli errori. Tali problemi sono modellati come problemi di ottimizzazione intera di *set covering* (o ricopertura di insiemi). Dato un questionario *errato*, il processo di *imputazione* consiste nel cambiare un insieme di costo minimo dei suoi valori in modo da ottenere un questionario *corretto*, cioè che non verifichi nessuna condizione di errore. Utilizzando n variabili binarie y_i corrispondenti ai singoli cambiamenti effettuabili, e definendo n costi c_i, uno per ogni singolo cambiamento, si ha una funzione obiettivo del tipo

$$\min_{y_i \in \{0,1\}} \sum_{i=1}^{n} c_i y_i. \tag{7}$$

Letterali diretti e negati (x_i, $\neg x_i$) divengono variabili binarie e loro complementi (x_i, $\overline{x}_i$). Una clausola $C_j = (x_i \vee \ldots \vee x_j \vee \neg x_k \vee \ldots \vee \neg x_n)$, definendo a^{π} e a^{ν} come i vettori di incidenza rispettivamente degli insiemi $A_{\pi} = \{x_i, \ldots, x_j\}$ dei letterali diretti e $A_{\nu} = \{x_k, \ldots, x_n\}$ dei letterali negati di C_j, diviene la disequazione lineare

$$\sum_{i=1}^{n} a_i^{\pi} x_i + \sum_{i=1}^{n} a_i^{\nu} \overline{x}_i \geq 1 \tag{8}$$

La modellazione del problema di imputazione è possibile in quanto vengono individuati legami tra le variabili y_i e le x_i. La procedura, applicata a problemi reali (forniti dall'Istat), risulta molto valida.

BIBLIOGRAFIA

[1] BRUNI R. and SASSANO A., *Finding Minimal Unsatisfiable Subformulae in Satisfiability Instances*, Proc. of 6th Internat. Conf. on Principles and Practice of Constraint Programming, LNCS, Springer 1894 (2000), 500-505.

[2] FELLEGI P. and HOLT D., *A Systematic Approach to Automatic Edit and Imputation*, Journal of the American Statistical Association, **71** (1976), 17-35.

[3] LOVELAND D. W., *Automated Theorem Proving: a Logical Basis*, North Holland (1978).

Dipartimento di Informatica e Sistemistica, Università di Roma «La Sapienza»
e-mail: bruni@dis.uniroma1.it
Dottorato in Ricerca Operativa (sede amministrativa: Roma I) - Ciclo XIII
Direttore di ricerca: Prof. Antonio Sassano, Università di Roma «La Sapienza»