Valutazione Efficiente di Aggregati Ricorsivi in Programmazione Logica

Mario Alviano

Department of Mathematics, University of Calabria, 87030 Rende (CS), Italy
alviano@mat.unical.it

Answer set programming (ASP), anche nota come programmazione logica disgiuntiva (DLP) con semantica dei modelli stabili [1,2], è un potente formalismo per la rappresentazione della conoscenza e il ragionamento di senso comune. ASP è particolarmente adatta per rappresentare conoscenza incompleta e ragionamento non monotono ed è utilizzata in intelligenza artificiale (AI) anche per applicazioni di diagnostica e planning. Un punto di forza di ASP è il suo elevato potere espressivo. Infatti, essa consente di esprimere tutti i problemi appartenenti alla classe di complessità Σ_2^P ($\mathrm{NP}^{\mathrm{NP}}$). Tale elevata espressività è rilevante anche per AI, in quanto importanti problemi di questa disciplina sono completi per il secondo livello della gerarchia polinomiale, quali ad esempio diagnostica e planning in presenza di conoscenza incompleta. Di fatto, questi problemi non sono risolvibili con riduzioni (polinomiali) a SAT, mentre sono direttamente rappresentabili in ASP [3]. Sebbene abbia un elevato potere espressivo, neanche ASP consente di rappresentare e ragionare naturalmente su proprietà che interessano insiemi di dati nel loro complesso, aspetto rilevante in diversi domini applicativi. Per colmare questa lacuna, sono state proposte diverse estensioni sintattiche di ASP, la più importante delle quali è l'introduzione delle funzioni di aggregazione. Tra queste, assumono particolare rilevanza le funzioni di aggregazione ricorsive [4]. Più precisamente, un'aggregazione è ricorsiva se i dati aggregati dipendono dalla valutazione dell'aggregato stesso.

Esempio 1. Si consideri un insieme di compagnie, ognuna delle quali può possedere una percentuale di azioni delle altre. Una compagnia ha il controllo di un'altra compagnia se ne controlla, direttamente e/o indirettamente, più del 50% delle azioni. Nell'esempio in Figura 1, a controlla b e, grazie al 20% di c posseduto da b, esercita un controllo anche su c. Si noti che per trattare questo problema occorre eseguire la somma delle azioni controllate, che dipendono a loro volta dalla valutazione di questa somma; in altre parole, è necessaria un'aggregazione ricorsiva. La codifica di questo problema in ASP è riportata in Figura 2.

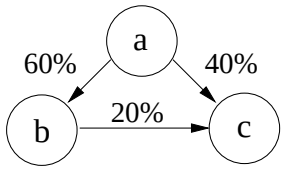

Figura 1. Un'istanza di Company Controls

```
company(a).       owns(a,b,60).
company(b).       owns(a,c,40).
company(c).       owns(b,c,20).

cv(X,X,Y,S)    :- owns(X,Y,S).
cv(X,Z,Y,S)    :- controls(X,Z), owns(Z,Y,S).
controls(X,Y) :- company(X), company(Y),
                 #sum{S,Z : cv(X,Z,Y,S)} > 50.
```

Figura 2. Codifica di Company Controls

Durante la tesi sono state analizzate le proprietà del ricco frammento di ASP con aggregati che consente la disgiunzione, la negazione non-monotona e gli aggregati ricorsivi monotoni e antimonotoni; il linguaggio che ne risulta è la *Disjunctive Logic Programming with Monotone and Antimonotone Aggregates* ($\mathrm{DLP}^{\mathcal{A}}_{m,a}$) [5].

I principali contributi del lavoro di tesi possono essere riassunti come segue.

- Analisi delle proprietà della $\mathrm{DLP}^{\mathcal{A}}_{m,a}$.
  - Definizione di una nuova e intuitiva nozione di insieme infondato [6] per $\mathrm{DLP}^{\mathcal{A}}_{m,a}$.
  - Dimostrazione formale di alcune proprietà che consentono di ottimizzare la computazione dei modelli stabili di un sottoinsieme della programmazione logica con aggregati.
  - Prova che gli insiemi infondati possono essere utilizzati con profitto nella computazione $\mathrm{DLP}^{\mathcal{A}}_{m,a}$ per il controllo di stabilità e per il taglio dello spazio di ricerca.
- Progettazione di algoritmi per la valutazione efficiente di $\mathrm{DLP}^{\mathcal{A}}_{m,a}$.
  - Dimostrazione dell'insufficienza della tecnica semi-naive [7] per l'istanziazione degli aggregati ricorsivi e presentazione di una nuova tecnica che corregge questa mancanza.
  - Definizione di alcuni operatori per il taglio dello spazio di ricerca, fra i quali estensione di Fitting [8] e well-founded [6] al linguaggio $\mathrm{DLP}^{\mathcal{A}}_{m,a}$.
  - Presentazione di algoritmi per la computazione modulare dell'operatore well-founded.
- Implementazione e sperimentazione di un prototipo per $\mathrm{DLP}^{\mathcal{A}}_{m,a}$.
  - Realizzazione di un sistema prototipale per $\mathrm{DLP}^{\mathcal{A}}_{m,a}$ ottenuto implementando i precedenti risultati in DLV, il quale supportava solo aggregazioni non ricorsive.
  - Valutazione sperimentale delle caratteristiche dei programmi con aggregati ricorsivi e del prototipo realizzato.

Il prototipo realizzato costituisce la prima implementazione di aggregati ricorsivi in ASP disgiuntiva. Dunque, i risultati ottenuti e descritti nella tesi possono essere concretamente utilizzati per risolvere problemi di natura applicativa nel campo dell'intelligenza artificiale. In particolare, nella tesi vengono mostrate delle codifiche con aggregati ricorsivi per alcuni di questi problemi, fra cui *Knapsack*, *Party Invitation*, *Seating*, *Car Sequencing*, *Employee Raise* e *Social Golfer*.

Riferimenti bibliografici

1. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. NGC **9** (1991) 365–385
2. Leone, N., Rullo, P., Scarcello, F.: Disjunctive Stable Models: Unfounded Sets, Fixpoint Semantics and Computation. Information and Computation **135**(2) (1997) 69–112
3. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. ACM TOCL **7**(3) (2006) 499–562
4. Faber, W., Leone, N., Pfeifer, G.: Recursive aggregates in disjunctive logic programs: Semantics and complexity. In: JELIA 2004. LNCS 3229, (2004) 200–212
5. Alviano, M., Faber, W., Leone, N.: Using Unfounded Sets for Computing Answer Sets of Programs with Recursive Aggregates. CILC (2007)
6. Van Gelder, A., Ross, K.A., Schlipf, J.S.: The Well-Founded Semantics for General Logic Programs. JACM **38**(3) (1991) 620–650
7. Terracina, G., Leone, N., Lio, V., Panetta, C.: Experimenting with Recursive Queries in Database and Logic Programming Systems. Theory and Practice of Logic Programming (2007)
8. Fitting, M.: A Kripke-Kleene semantics for logic programs. JLP **2**(4) (1985) 295–312