Alternanza, parallelismo e complessità

Pietro Battiston

25/09/2008

Si definisce SAT come l'insieme delle formule booleane soddisfacibili, ovvero di quelle formule booleane che risultano vere per almeno un'assegnazione di valori alle variabili. Non è noto se SAT $\in$ P (=PTIME), ovvero se esistano algoritmi per stabilire in tempo polinomiale (rispetto alla lunghezza dell'input) se una data formula booleana appartenga a SAT. È invece semplice verificare che SAT $\in$ NP (l'insieme dei problemi decisionali algoritmicamente risolubili in tempo polinomiale con una macchina non deterministica).

Uno dei problemi aperti più importanti della teoria della complessità computazionale (l'unico, tra i 7 problemi del Millennium Prize del Clay Institute, ad essere nato in campo informatico) è se P = NP.

A questo riguardo un risultato fondamentale è il seguente:

Teorema 1. *(Cook, 1971 [2]): SAT è NP-completo.*

dove per "completo" si intende rispetto a riduzioni polinomiali. Di conseguenza, se SAT $\in$ P, allora P = NP,

La congettura più accreditata è che P $\neq$ NP; di conseguenza la NP-completezza di un problema è considerata un'indicazione del fatto che probabilmente il problema è intrattabile in tempo polinomiale. A partire dal classico risultato di Cook molti altri problemi sono stati dimostrati NP-completi tramite una riduzione a SAT ([4]).

Valgono le inclusioni

$$\mathrm{P} \subset \mathrm{NP} \subset \mathrm{EXPTIME} \tag{$*$}$$

dove EXPTIME è l'insieme dei problemi decisionali algoritmicamente risolubili in tempo esponenziale. Segue dal teorema di gerarchia ([3]) che

$$\mathrm{P} \neq \mathrm{EXPTIME}\ ,$$

e quindi almeno una delle due inclusioni in (*) è stretta.

Se oltre ai connettivi booleani ammettiamo la possibilità di quantificare (universalmente ed esistenzialmente) su variabili proposizionali otteniamo l'insieme delle *formule booleane quantificate* e possiamo studiarne il sottoinsieme QSAT di quelle vere. Ad esempio la formula $\forall A(\neg A \rightarrow \exists B(A \rightarrow B))$ appartiene a

QSAT. Il problema di stabilire se una formula appartiene a QSAT è, almeno apparentemente, più complesso del problema analogo per SAT (che equivale al caso particolare di QSAT in cui tutti i quantificatori sono esistenziali e prenessi). Si pone quindi il problema, che affrontiamo in questa tesi, di studiare la complessità computazionale di QSAT.

La prima osservazione è che anche QSAT appartiene ad EXPTIME. Come vedremo però ci sono risultati che suggeriscono che QSAT $\notin$ NP (il risultato opposto confuterebbe alcune congetture generalmente ritenute vere). Per poter esporre questi risultati occorre introdurre delle classi di complessità che dipendono dallo spazio di memoria piuttosto che dal tempo di calcolo. Una delle più importanti di tali classi è PSPACE (l'insieme dei problemi decisionali risolubili in spazio polinomiale). Valgono le inclusioni

$$\mathrm{P} \subset \mathrm{NP} \subset \mathrm{PSPACE} \subset \mathrm{EXPTIME}$$

e si congettura che le inclusioni siano tutte strette. In questa tesi esporremo una dimostrazione del seguente:

Teorema 2. *(Stockmeyer, Meyer, 1973 [7]): QSAT è PSPACE-completo*

Ne segue che a meno che NP = PSPACE, QSAT $\notin$ NP, confermando l'intuizione che QSAT sia più complesso di SAT.

L'interesse della classe PSPACE è che essa può essere presa come possibile definizione della classe dei problemi decidibili in tempo polinomiale con un calcolatore capace di una forte forma di parallelismo. Vale infatti, a livello informale, il paradigma "Spazio deterministico = Tempo parallelo". Una delle forme in cui è possibile precisare questo paradigma è il seguente:

Teorema 3. *(Chandra, Kozen, Stockmeyer, 1981 [1]): PSPACE = APTIME*

dove ATIME è la classe dei problemi risolubili in tempo polinomiale con una macchina di Turing alternante (un modello di una certa forma di parallelismo). Le macchine di Turing alternanti sono una generalizzazione di quelle non-deterministiche: mentre le prime permettono di fare non-deterministicamente delle scelte prendendo l'OR dei risultati delle subcomputazioni, nelle macchine alternanti si ammette la possibilità di ramificazioni non-deterministiche sia di tipo OR che di tipo AND (ciò permette di simulare i quantificatori rispettivamente esistenziali e universali). L'utilità teorica delle macchine alternanti è che esse facilitano la progettazione di algoritmi per quei problemi nella cui definizione compaiono, implicitamente o esplicitamente, molti quantificatori alternati. Uno dei modi per dimostrare il teorema 3 è far vedere che QSAT è completo, oltre che per PSPACE, anche per APTIME, come vedremo in questa tesi.

QSAT risulta particolarmente interessante anche perché esso costituisce un paradigma per studiare la complessità di un'ampia classe di giochi ad informazione completa in cui due giocatori alternano le mosse. Ad esempio il problema di stabilire chi abbia una strategia vincente nel gioco del GO (generalizzato a scacchiere $n \times n$) si dimostra essere PSPACE-completo dimostrandone l'equivalenza a QSAT ([5, 6]).

Riferimenti bibliografici

[1] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 1981.

[2] Stephen A. Cook. The complexity of theorem-proving procedures. *Proceedings of the third annual ACM symposium on Theory of computing*, 1971.

[3] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 1965.

[4] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 1972.

[5] David Lichtenstein and Michael Sipser. Go is polynomial-space hard. *Journal of the ACM*, 1980.

[6] Christos H. Papadimitriou. *Computational Complexity*. 1994.

[7] L. Stockmeyer and A. Meyer. Word problems requiring exponential time. *Proceedings of the 5th ACM Symposium on the Theory of Computing*, 1973.