Complementi ed Esercizi di Informatica Teorica

Vincenzo Bonifaci

14 aprile 2009

1 Problemi in NC

1.1 Le classi NC^k

Ricordiamo che con NC^k si indica la classe dei linguaggi decidibili da una famiglia (logspace-uniforme) di circuiti di profondità $O((\log n)^k)$ e dimensione polinomiale. I circuiti possono usare porte logiche di tipo AND, OR o NOT. Le porte AND e OR sono limitate ad avere *fan-in* 2 (due ingressi). La classe NC (l'acronimo sta per Nick's Class, dallo studioso Nick Pippenger) è l'unione delle classi NC^k per ogni $k \geq 0$.

A volte parleremo di una *funzione* (anziché un linguaggio) in NC^k. Con questo si deve intendere che esiste una famiglia di circuiti che calcola la funzione, e che questa famiglia gode delle stesse proprietà di cui sopra.

Nell'analizzare le famiglie di circuiti che daremo per i vari problemi, non dimostreremo esplicitamente che si tratta di famiglie logspace-uniformi; si può comunque verificare, almeno intuitivamente, che i circuiti costruiti hanno una struttura che varia in maniera omogenea al variare di n.

1.2 Il linguaggio PARITÀ

Sia

$$\text{PARITÀ} = \{x \in \{0,1\}^* \mid x \text{ ha un numero dispari di } 1\}$$

ovvero sia PARITÀ il linguaggio costituito dalle stringhe binarie aventi un numero dispari di 1. Ad esempio $1011 \in$ PARITÀ mentre $\varepsilon \notin$ PARITÀ. Il linguaggio PARITÀ è un linguaggio molto semplice (infatti è anche un linguaggio regolare) e può essere deciso da un circuito NC^1. Per vedere come, si consideri una stringa "composta" xy. Questa stringa ha un numero dispari di 1 se e solo se x ne ha un numero pari e y un numero dispari o viceversa. Quindi, indicando con $p(x)$ la funzione caratteristica dell'insieme PARITÀ, abbiamo $p(xy) = p(x) \oplus p(y)$ dove $\oplus$ indica la somma modulo due, o equivalentemente, l'operatore logico di OR esclusivo (XOR). Applicando ripetutamente questa proprietà ad una stringa di n bit $x = x_1 \dots x_n$, troviamo $p(x) = p(x_1) \oplus \dots \oplus p(x_n)$. Questa somma può essere calcolata velocemente in parallelo con un albero binario di XOR di profondità $O(\log n)$. L'operatore logico XOR non è disponibile direttamente nel nostro modello di circuiti ma possiamo realizzarlo facilmente notando che

$$x \oplus y = (\neg x \wedge y) \vee (x \wedge \neg y).$$

Abbiamo così dimostrato che PARITÀ $\in \mathsf{NC}^1$ e che quindi la parità di una stringa può essere calcolata in tempo parallelo $O(\log n)$.

Esercizio 1.1. Dimostrare che PARITÀ $\notin \mathsf{NC}^0$. Quali linguaggi appartengono alla classe NC^0?

Esercizio 1.2. Scrivere una espressione regolare che descriva il linguaggio PARITÀ.

1.3 Moltiplicazione di matrici booleane

Siano A e B due matrici $n \times n$ a elementi zero e uno. Vogliamo costruire un circuito NC^1 che calcoli il prodotto $C = AB$. Partendo dalla definizione di prodotto tra matrici abbiamo:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

che nel contesto booleano si può riscrivere come

$$C_{ij} = \bigvee_{k=1}^{n} A_{ik} \wedge B_{kj}.$$

Questa formula fornisce già un possibile circuito, ma l'idea non basta perché l'OR della formula ha un numero arbitrariamente alto (n) di ingressi mentre nel nostro modello le porte logiche possono avere al più due ingressi. Analogamente a quanto fatto nel caso precedente, comunque, possiamo calcolare questo OR a n ingressi con un albero binario di OR a due ingressi, di profondità $O(\log n)$. Abbiamo quindi che per calcolare C_{ij} (un elemento della matrice prodotto) sono sufficienti n porte AND e circa n porte OR per un totale di $O(n)$ porte e un tempo parallelo pari a $O(\log n)$ (profondità dell'albero). In questo modo *tutti* gli n^2 elementi della matrice C possono essere calcolati in parallelo nello stesso tempo utilizzando $O(n^3)$ porte logiche. Quindi anche la moltiplicazione di matrici booleane è un problema risolvibile con circuiti NC^1.

1.4 Raggiungibilità

Il problema della RAGGIUNGIBILITÀ è il seguente: dato un grafo orientato G, è possibile raggiungere il nodo n a partire dal nodo 1? L'algoritmo classico per risolvere questo problema visita il grafo a partire dal nodo 1 finché non si incontra il nodo n o finché non ci sono più nuovi nodi da visitare. Questo algoritmo però non può essere parallelizzato in modo efficiente, dato che un cammino dal nodo 1 al nodo n potrebbe arrivare ad avere una lunghezza pari a $n-1$, quando invece vogliamo determinare se questo cammino esiste in tempo $O((\log n)^k)$ per qualche k. Usiamo quindi un approccio totalmente diverso. Assumiamo di avere in ingresso la matrice di adiacenza A del grafo G' ottenuto da G aggiungendo tutti i loop da un nodo a se stesso. Quindi $A_{ij} = 1$ se e solo se esiste un arco da i a j in G oppure $i = j$. Consideriamo ora l'elemento (i, j) della matrice A^2:

$$A^2_{ij} = \bigvee_{k=1}^{n} A_{ik} \wedge A_{kj}.$$

Come abbiamo visto precedentemente, possiamo calcolare questa matrice con un circuito NC^1. Non è difficile rendersi conto dalla definizione che A^2_{ij} è pari a 1 *se e solo se esiste un cammino di lunghezza al più due dal nodo i al nodo j.* Analogamente A^l_{ij} vale 1 se e solo se esiste un cammino di lunghezza al più l dal nodo i al nodo j. Ipotizziamo ora per un momento che n sia una potenza di due. Possiamo ottenere la matrice $R = A^n$ elevando ripetutamente al quadrato la matrice A:

$$R = A^n = (((A^2)^2)\ldots)^2 \quad (\log n \text{ volte}).$$

Dato che ogni cammino in G ha lunghezza minore di n, l'elemento (i, j) di questa matrice vale 1 se e solo se esiste un cammino da i a j in G. Quindi dal nodo 1 si può raggiungere il nodo n se e solo se $R_{1n} = 1$.

Abbiamo così che il circuito che calcola la matrice R può essere ottenuto collegando in cascata $\log n$ circuiti che calcolano il quadrato di una matrice. Dato che ognuno di questi circuiti ha una profondità pari a $O(\log n)$ e dimensione $O(n^3)$, otteniamo un circuito con una profondità totale $O((\log n)^2)$ e una dimensione totale $O(n^3 \log n)$. RAGGIUNGIBILITÀ è quindi in NC^2.

Esercizio 1.3. Adattare l'algoritmo al caso in cui n non sia necessariamente una potenza di due.

Esercizio 1.4. Sia RAGG-4 il problema di stabilire, data la matrice di adiacenza di un grafo diretto G, se esiste un nodo di G dal quale ogni altro nodo è raggiungibile attraversando al più 4 archi. Dimostrare che il linguaggio L associato a RAGG-4 è in NC. A quale livello della gerarchia NC si trova L?