Caroselli Stefano

ALGORITMI & STRUTTURE DATI

TESINA Nº 2

Studio dei circuiti in una griglia.

1. STUDIO DELLA GRIGLIA

È data una griglia n×m, in cui sono presenti coppie di vertici, disposte in maniera arbitraria; si vuole generare un algoritmo che studi se è possibile o meno collegare tra loro i due vertici di ogni coppia in modo tale che siano verificate le seguenti condizioni:

1. ogni collegamento è una retta spezzata, composta al più da due segmenti;
2. tali segmenti devono essere paralleli ai lati della griglia;
3. i collegamenti non si possono intersecare.

Osserviamo inizialmente che il numero massimo di collegamenti che possiamo fare tra due vertici è due: dipende se ci vogliamo muovere prima in orizzontale, o in verticale; una possibile soluzione del problema è data quindi da un vettore **v** ad f componenti booleane (f è il numero dei fili, ossia dei collegamenti tra i vertici) dove **v**[i]=true se partiamo dal primo vertice movendoci in verticale, **v**[i]=false in caso contrario.

Ecco i due possibili casi che possiamo avere, con le rispettive assegnazioni booleane:

Abbiamo indicato con un tratteggio verde il percorso "vero" e con un tratteggio rosso il percorso "falso".

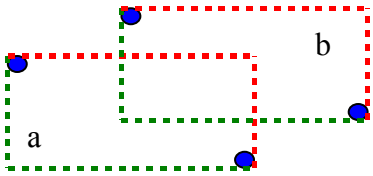
Osserviamo che questo problema è sempre risolvibile, , in quanto un algoritmo che sfrutta la tecnica del *back-tracking* andrebbe per tentativi fino a trovare una possibile soluzione; se dopo aver esaminato tutti i possibili casi non ne avesse trovata nessuna, concluderebbe che il problema non ha soluzione; ma questo procedimento implicherebbe, nel peggiore dei casi, un esame di tutte le possibili combinazioni vero/falso di tutti i fili, e questo ha un costo di f^2 ! Dobbiamo trovare un algoritmo migliore.

Iniziamo con lo studio delle possibili situazioni in cui si possono trovare due coppie di vertici. Per comodità di studio possiamo identificare ogni coppia con il rettangolo generato dall'unione dei due suoi cammini; tale rettangolo (filo) sarà "vero" se dovremo prendere il cammino vero, sarà "falso" in caso contrario. Se, per semplicità, scartiamo i casi in cui i rettangoli sono tangenti, abbiamo che i casi principali che si possono verificare sono i seguenti:

- I fili sono indipendenti : i due rettangoli non si intersecano;
- I fili sono incompatibili : i due rettangoli hanno 4 punti d'intersezione;
- I fili sono compatibili se verificano determinate *clausole* : i due rettangoli hanno esattamente 2 punti d'intersezione.

Uno studio più approfondito mostra che il terzo caso ha 5 sottocasi, che si distinguono per le clausole da verificare. Se consideriamo l'equazione booleana associata ad ogni caso, e la poniamo in forma *normale congiuntiva*, osserviamo che le clausole sono date dalle parentesi con all'interno operatori *or* ($\vee$).

Esempio: consideriamo i fili a : [(1,3) ; (4,1)] e b : [(2,4) ; (5,2)]



Abbiamo una soluzione booleana data da : (a-true $\vee$ b-false) che possiamo scrivere formalmente : (a $\vee$ not-b).

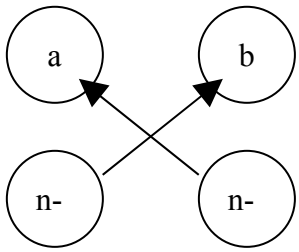
La soluzione finale del problema sarà data quindi da un'assegnazione che verifichi contemporaneamente tutte le clausole, e che non esista se esiste almeno una coppia di fili incompatibili, o se non si riuscisse a trovare alcuna combinazione.
Ad esempio una equazione booleane formata da tutte le clausole può essere:

[1] : $(a \vee b) \wedge (\text{not-}c \vee d) \wedge (e \vee f) \wedge (\text{not-}a \vee e) = \text{true}$

dove ogni parentesi determina per l'appunto una clausola.

Studiando le singole clausole osserviamo che ogni combinazione di valori è possibile, eccetto il caso (false $\vee$ false) : in quel caso la clausola avrebbe risultato false, e quindi l'intera equazione non avrebbe soluzione.

2. STUDIO DEL GRAFO ASSOCIATO



Data una clausola (a $\vee$ b) possiamo associarvi 4 vertici e 2 spigoli di un grafo orientato G :

La condizione che non possono essere a=false e b=false contemporaneamente, implica una condizione sulla assegnazione booleana sui vertici : se x e y sono due vertici, ed esiste uno spigolo da x ad y, allora : [*] non può essere x=true e y=false allo stesso tempo.
Data una equazione booleana del tipo di [1], allora possiamo associarvi un grafo orientato G, con 2f vertici, ed un numero di spigoli dipendente dalle clausole; risolvere l'equazione equivale a determinare una buona assegnazione di valori booleani ai vertici, in modo da non violare la condizione [*].
Possiamo a questo punto enunciare alcuni risultati che ci aiuteranno nel costruire l'algoritmo:

Teorema

L'equazione non è soddisfabile $\Leftrightarrow \exists$ almeno una c.f.c. di G contenente un vertice x e il suo complementare not-x.

Dim.

Supponiamo per assurdo che esista una c.f.c. contenente x e not-x; sia ad esempio un assegnazione in cui x=true, quindi not-x=false; consideriamo un cammino da x a not-x : poiché x=true, non ci resta che assegnare true al successivo di x, altrimenti avremmo uno spigolo da un vertice true ad un false; di conseguenza, percorrendo il cammino dovremmo sempre dare un'assegnazione true ai vertici, fino ad arrivare a not-x, che però è contrassegnato false; abbiamo comunque trovato uno spigolo che non verifica la condizione [*]. Analogamente se avessimo supposto x=false, ci sarebbe stato un tale spigolo percorrendo il cammino da not-x a x. Q.E.D.

Da questo teorema ricaviamo anche un'altro importante risultato:

Proposizione

I vertici di una stessa c.f.c. hanno tutti la medesima assegnazione booleana.

Quindi possiamo passare dallo studio del grafo G allo studio del grafo G_C delle c.f.c. di G costruito nel seguente modo:

- Ad ogni vertice di G_C corrisponde una c.f.c. di G

- $\exists$ uno spigolo da X a Y in G_C $\Leftrightarrow$ $\exists$ uno spigolo da un vertice $x \in X$ ad un vertice $y \in Y$ in G

Possiamo subito vedere che se l'equazione è soddisfabile, allora:

Proposizione
Se $\exists$ una componente X con vertici {a, b, c, etc…} allora $\exists$ anche una componente Y con vertici {not-a, not-b, not-c, etc…} che per comodità chiameremo not-X.

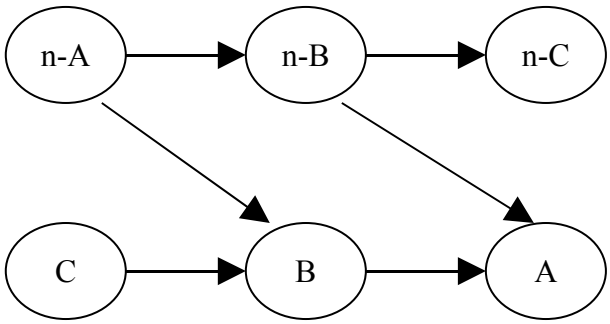
Un'altra importante proprietà di G_C è la seguente:

Proposizione
G_C è un grafo aciclico.

Di conseguenza è possibile dare un ordinamento ai vertici di G_C per mezzo di un *sort topologico* in modo da poter dare una corretta assegnazione di valori booleani; infatti si può osservare che:
- Se un vertice X di G_C si trova in una metà del vettore ordinato **w**, allora not-X si trova nella metà rimanente.
- Condizione necessaria affinché l'assegnazione sia corretta è che l'ultimo vertice in **w** sia posto true.

Possiamo quindi cominciare assegnando l'ultimo vertice true, e il suo opposto a false, e tornare indietro fino a metà del vettore, in quanto per quanto visto, la prima metà sarà stata posta false.

Esempio :
Consideriamo il seguente grafo G_C : effettuando un t-sort abbiamo un possibile ordinamento :
not-A, C, B, not-B, A, not-C.
Ne segue che una possibile assegnazione è: not-C=true, A=true, not-B=true, ossia:
A=true, B=false, C=false.



3. CONCLUSIONE

In questo modo siamo riusciti ad assegnare in modo corretto valori true/false a tutti i vertici di G_C e quindi a tutti i vertici di G; quindi ogni filo ha un corretto valore booleano, e abbiamo risolto il problema. Inoltre il costo di questo algoritmo – se tralasciamo I costi dovuti alla creazione delle clausole e del grafo – è il costo del t-sort, che si effettua tramite una V.I.P.,e questa ha un costo lineare in f (numero fili).

4. OSSERVAZIONI

Nel nostro algoritmo possiamo evitare di seguire l'intero ragionamento per risolvere il problema, in quanto molti passi intermedi possono essere evitati; ad esempio non è necessario creare il grafo delle c.f.c., in secondo luogo ordinarlo, e infine assegnare i valori ai vertici, ma è possibile, durante la V.I.P. assegnare valori ad un rappresentante della c.f.c. (ad esempio quello più in alto nello sp-tree) senza dover ordinare in un vettore le componenti: infatti la V.I.P. termina di esaminare i rappresentanti, cominciando da quelli che andrebbero ordinati in fondo, e quindi andrebbero posti a true; si può quindi dare tale assegnazione, e contemporaneamente si può quindi anche porre a false i loro opposti, e assegnare tali valori anche ad ogni altro vertice della c.f.c.
Ovviamente per fare questi passaggi saranno necessarie varie strutture d'appoggio, ma il costo totale sarà minore.