Esercizi di teoria per SPA

Tiziano Villa

Anno Accademico 2008-9

Nome e Cognome:
Matricola:
Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	X	
problema 2	X	
problema 3	X	
problema 4	X	
problema 5	X	
problema 6	X	
problema 7	X	
problema 8	X	
problema 9	X	
problema 10	X	
problema 11	X	
totale	X	

1. Data la definizione di cofattore generalizzato come funzione incompletamente specificata

$$co(f,g) = (fg, \bar{g}, \bar{f}g),$$

si dimostri che che il cofattore di Shannon f_x e' una copertura di $co(f,x)$, cioe'

$$fx \subseteq f_x \subseteq fx + \bar{x};$$

inoltre si dimostri che e' l'unica copertura di $co(f,x)$ indipendente da x.

Si dimostrino le seguenti proprieta' del cofattore di Shannon:

(a) $xf_x + \overline{x}f_{\overline{x}} = f$

(b) $(f_x)_y = f_{xy}$

(c) $(fg)_x = f_x g_x$

(d) $(\overline{f})_x = \overline{(f_x)}$

Si dimostrino le seguenti proprieta' del cofattore generalizzato:

(a) $f = g\ co(f,g) + \overline{g}\ co(f,\overline{g})$

(b) $co(co(f,g),h) = co(f,gh)$

(c) $co(fg,h) = co(f,h)co(g,h)$

(d) $co(\overline{f},g) = \overline{co(f,g)}$

2. (a) Si dimostri che $\overline{f} = x\overline{(f_x)} + \overline{x}\overline{(f_{\overline{x}})}$.

   (b) Si applichi il paradigma di ricorsione monotona per complementare la funzione:

$$f = xy'z' + wxy' + wy'z + wx'z + wyz + xyz + w'xy'z$$

   usando il processo di fusione indicato dal punto precedente.

   (c) Si complementi f usando la legge di De Morgan.

   (d) Si calcoli

$$U \sharp \mathcal{F},$$

   dove U e' il cubo universale e $\mathcal{F}$ e' la copertura data sopra

$$\mathcal{F} = \{xy'z', wxy', wy'z, wx'z, wyz, xyz, w'xy'z\}.$$

   L'operazione $\sharp$ tra due cubi α e β e' definita come segue:

$$\alpha \sharp \beta = \begin{cases} a_1.b_1' & a_2 & \dots & a_n \\ a_1 & a_2.b_2' & \dots & a_n \\ \dots & & & \\ a_1 & a_2 & \dots & a_n.b_n' \end{cases}$$

   Che cosa si ottiene ?

   (e) Si calcoli

$$U \sharp (U \sharp \mathcal{F}).$$

   Che cosa si ottiene ?

(f) Si applichi il paradigma di ricorsione monotona per complementare la funzione:

$$f = ab'c' + ab'd + b'cd + a'cd + bcd + abc + ab'cd'$$

(g) E' noto che l'operazione di fusione delle chiamate ricorsive si esegue con la seguente procedura MERGE_WITH_CONTAINMENT, dove data una funzione g s'indicano rispettivamente con $H_0 = \{\cup_{i=1}^{l} c_i\}$ e $H_1 = \{\cup_{j=1}^{k} d_j\}$ le coperture di $g_{x'}$ e g_x:

MERGE_WITH_CONTAINMENT(H_0, H_1) {
$H_2 = \{c_j \mid \exists j\ s.t. c_j \subseteq d_i\} \cup \{d_j \mid \exists\ i s.t. d_j \subseteq c_i\}$
$H_0 = H_0 \setminus H_2$
$H_1 = H_1 \setminus H_2$
return($x'H_0 + xH_1 + H_2$)
}

Un'alternativa alla procedura MERGE_WITH_CONTAINMENT e' la seguente procedura MERGE_WITH_COVER:

MERGE_WITH_COVER(H_0, H_1) {
$H_2 = \{c_j \mid c_j \subseteq \{\cup d_i\}\} \cup \{d_j \mid d_j \subseteq \{\cup c_i\}\}$
$H_0 = H_0 \setminus H_2$
$H_1 = H_1 \setminus H_2$
return($x'H_0 + xH_1 + H_2$)
}

Si ripeta il calcolo del complemento di

$$f = ab'c' + ab'd + b'cd + a'cd + bcd + abc + ab'cd'$$

usando MERGE_WITH_COVER invece che MERGE_WITH_CONTAINMENT e si confrontino i risultati.

Che cosa si puo' dire delle proprieta' delle coperture del complemento ottenute usando l'una o l'altra procedura di fusione ?

Traccia. Che cosa si puo' dire circa l'irridondanza o la primalita' delle coperture ottenute ?

3. (a) Si dimostri che p e' un primo di f se e solo se e' un cubo massimale (un cubo massimale non e' contenuto in nessun altro cubo) dell'insieme che contiene i seguenti cubi:

   i. $p = qx$, q e' un primo di f_x, $q \not\subseteq f_{\bar{x}}$

   ii. $p = r\bar{x}$, r e' un primo di $f_{\bar{x}}$, $r \not\subseteq f_x$

   iii. $p = qr$, q e' un primo di f_x, r e' un primo di $f_{\bar{x}}$

   (b) Si dimostri il teorema precedente anche nel caso in cui la terza clausola iii. sia sostituita da

   iii. $p = \text{CONSENSO}(qx, r\bar{x})$, q e' un primo di f_x, r e' un primo di $f_{\bar{x}}$.

   L'operazione CONSENSO tra due cubi α e β e' definita come segue:

$$\text{CONSENSO}(\alpha, \beta) = \begin{cases} a_1 + b_1 & a_2.b_2 & \dots & a_n.b_n \\ a_1.b_1 & a_2 + b_2 & \dots & a_n.b_n \\ \dots \\ a_1.b_1 & a_2.b_2 & \dots & a_n + b_n \end{cases}$$

   (c) Si consideri il risultato del calcolo $U \sharp (U \sharp \mathcal{F})$, eseguito per l'esercizio 2. (e). Data una qualsiasi copertura $\mathcal{F}$ di una funzione f, si puo' dimostrare un'affermazione generale sul risultato del calcolo $U \sharp (U \sharp \mathcal{F})$?

4. Date le funzioni

$$f_1 = \overline{a}\overline{b}c + \overline{a}b\overline{c} + \overline{a}\overline{b}\overline{c}$$
$$f_2 = \overline{a}bc + \overline{a}\overline{b}c + abc + \overline{a}b\overline{c}$$

   (a) Si generino tutti i primi di f_1 usando la procedura monotona ricorsiva di generazione dei primi;
   (b) Si generino tutti i primi di f_2 usando la procedura monotona ricorsiva di generazione dei primi;
   (c) Si generi con l’algoritmo di tautologia la tavola di Quine-McCluskey ridotta di f_1;
   (d) Si generi con l’algoritmo di tautologia la tavola di Quine-McCluskey ridotta di f_2;

   Nota: in alternativa alla procedura monotona si possono calcolare i primi con l’algoritmo di espansione rispetto al complemento della funzione.

5. (a) Si generino tutti i primi della funzione multi-uscita (f_1, f_2) (dove f_1 e f_2 sono date nel problema precedente), usando l'algoritmo di espansione rispetto al complemento della funzione.

   (b) Si generi la tavola di Quine-McCluskey ridotta della funzione multi-uscita (f_1, f_2) del problema precedente:

$$f_1 = \overline{a}\overline{b}c + \overline{a}b\overline{c} + \overline{a}\overline{b}\overline{c}$$
$$f_2 = \overline{a}bc + \overline{a}\overline{b}c + abc + \overline{a}b\overline{c}$$

6. (a) Si considerino le seguenti due affermazioni:

Teorema Data una funzione $ff = (f, d, r)$, sia $\mathcal{F} = \mathcal{G} \cup \{\alpha\}$, dove $\mathcal{F}$ e' una copertura di ff ed α e' un implicante primo disgiunto da $\mathcal{G}$.

Allora α e' un primo essenziale se e solo se

$$\text{CONSENSO}(\mathcal{G}, \{\alpha\})$$

non copre α.

L'operazione CONSENSO tra due cubi α e β e' definita come segue:

$$\text{CONSENSO}(\alpha, \beta) = \begin{cases} a_1 + b_1 & a_2.b_2 & \ldots & a_n.b_n \\ a_1.b_1 & a_2 + b_2 & \ldots & a_n.b_n \\ \ldots \\ a_1.b_1 & a_2.b_2 & \ldots & a_n + b_n \end{cases}$$

Nota:

- e' vuoto quando distanza ≥ 2;
- contiene un solo cubo quando distanza = 1;
- contiene n cubi quando distanza = 0.

Corollario Data una funzione $ff = (f, d, r)$, sia $\mathcal{F}$ una copertura di f, $\mathcal{D}$ una copertura di d e α un implicante primo di ff.

Allora α e' un primo essenziale se e solo se $\mathcal{H} \cup \mathcal{D}$ non copre α, dove

$$\mathcal{H} = \text{CONSENSO}((\mathcal{F} \cup \mathcal{D}) \sharp \alpha, \alpha),$$

e l'operazione $\sharp$ tra due cubi α e β e' definita come segue:

$$\alpha \sharp \beta = \begin{cases} a_1.b_1' & a_2 & \ldots & a_n \\ a_1 & a_2.b_2' & \ldots & a_n \\ \ldots \\ a_1 & a_2 & \ldots & a_n.b_n' \end{cases}$$

Si noti che invece di $\sharp$ si puo' usare l'operazione $\tilde{\sharp}$ definita come:

$$\alpha \tilde{\sharp} \beta = \begin{cases} a_1.b_1' & a_2 & \ldots & a_n \\ a_1.b_1 & a_2.b_2' & \ldots & a_n \\ \ldots \\ a_1.b_1 & a_2.b_2 & \ldots & a_n.b_n' \end{cases}$$

Sfruttando il precedente corollario e' possibile ridurre la prova di essenzialita' a una verifica di contenimento.

Si argomenti sulla correttezza del teorema e corollario enunciati.

(b) Applicando il metodo precedente, calcolare quali implicanti della seguente copertura sono essenziali:

```
.mv 3 0 5 5 5
11111 00001 11110
01100 00011 01010
01010 00100 11111
00110 01001 11010
00001 11111 10110
.e
```

7. Usando il paradigma di ricorsione monotona si calcoli il complemento $\overline{f}$ della seguente funzione f:

```
.mv 3 0 5 5 5
11111 00001 11110
01100 00011 01010
01010 00100 11111
00110 01001 11010
00001 11111 10110
.e
```

La copertura precedente di f e' debolmente monotona ?

La copertura ottenuta di $\overline{f}$ e' debolmente monotona ?

8. (a) Si dimostri che se una funzione f e' fortemente monotona, allora il suo complemento $\overline{f}$ e' una funzione fortemente monotona.

   (b) La copertura del complemento ottenuta con il procedimento di complementazione monotona e' minima ? Dimostrarlo o proporre un controesempio.

9. Sia dato un circuito che puo' avere uno dei seguenti otto guasti:

$$\{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8\}.$$

Si sono calcolati sei vettori di collaudo:

$$\{v_2, v_2, v_3, v_4, v_5, v_6\},$$

ciascuno dei quali puo' rilevare alcuni dei guasti, secondo la seguente lista:

(a) v_1 rileva la presenza dei guasti $\{g_1, g_3, g_5\}$;

(b) v_2 rileva la presenza dei guasti $\{g_2, g_4, g_6\}$;

(c) v_3 rileva la presenza dei guasti $\{g_1, g_2, g_4, g_7\}$;

(d) v_4 rileva la presenza dei guasti $\{g_3, g_5, g_8\}$;

(e) v_5 rileva la presenza dei guasti $\{g_2, g_6, g_8\}$;

(f) v_6 rileva la presenza dei guasti $\{g_3, g_4, g_7\}$.

Si trovi un sottoinsieme minimo dei vettori di guasto che rilevi tutti gli otto possibili guasti, se presenti. Si supponga che il circuito abbia al piu' un guasto e che tutti i vettori di collaudo abbiano lo stesso costo.

Traccia. Lo si modelli come un problema di copertura monotona.

10. (a) Si ottenga il diagramma di decisione binaria ridotto e ordinato secondo l'ordine $x_1 < x_2 < x_3 < x_4$ (senza lati complementati) della funzione

$$f = x_1 x_2 + \overline{x}_2 x_3 + \overline{x}_1 \overline{x}_3.$$

Si applichi la procedura *apply_ite*. Si mostrino i passi del procedimento seguito.

Traccia di soluzione.

Si considerino come disponibili i diagrammi delle costanti 0 e 1 e delle variabili x_1, x_2 e x_3. Si costruiscano prima i diagrammi dei singoli prodotti $x_1 x_2$, $x_2 x_3$ e $\overline{x}_1 \overline{x}_3$ usando *apply_ite* per calcolare i prodotti. Poi si costruisca il diagramma della somma dei prodotti usando *apply_ite* per calcolare le somme.

(b) Si ottenga il diagramma di decisione binaria ridotto e ordinato della precedente funzione usando anche i lati complementati. Per farlo si ripeta la costruzione precedente usando anche il caso terminale

$$ITE(F, 0, 1) = F'$$

per scoprire quando usare lati complementati.

Si mostrino i passi del procedimento seguito.

Si adotti la convenzione che le negazioni compaiono solo sui lati *else*, per cui i lati *else* si etichettano con un cerchietto vuoto se sono regolari o con un cerchietto pieno se sono complementati (regola per verificare il diagramma della funzione: poiche' i lati *then* non sono mai complementati, il valore di una funzione per il mintermine $1, 1, \ldots, 1$ sara' 0 se e solo se il lato uscente dal nodo associato alla funzione e' complementato).

Addendum sull'operazione ITE

I casi terminali sono

$$
\begin{aligned}
ITE(F,1,0) &= F \\
ITE(1,F,G) &= F \\
ITE(0,F,G) &= G \\
ITE(F,G,G) &= G \\
ITE(F,0,1) &= \overline{G}
\end{aligned}
$$

Ci sono molte triple che danno il medesimo risultato, ad es., $ITE(\overline{x}_1 + x_2x_4, (x_1+x_2)x_4, 0)$ e $ITE(x_2x_4(x_1+\overline{x}_3), 1, \overline{x}_1x_2x_3x_4)$ producono il medesimo risultato x_2x_4.

Sarebbe auspicabile, ma praticamente difficile, identificare tutte le triple che producono il medesimo risultato, perche' potremmo associarle allo stesso elemento della "computed table". Ci sono dei casi speciali in cui si possono identificare facilmente triple equivalenti, ad es.,

$$ITE(F,G,F) = ITE(F,G,0) = ITE(G,F,G) = ITE(G,F,0) = F.G$$

Per questi casi si trasforma una tripla in una forma convenzionale prima di cercarla nella "computed table". In tal modo tutte le triple che sono associate alla stessa forma hanno il medesimo elemento nella tavola.

I seguenti casi permettono di stabilire equivalenze tra triple:

(a) Due argomenti dell'ITE sono uguali, es. $ITE(F,F,G)$

(b) Due argomenti dell'ITE sono l'uno il complemento dell'altro, es. $ITE(F,G,\overline{F})$

(c) Almeno un argomento e' costante, es. $ITE(F,G,0)$

Per arrivare a delle triple convenzionali, si applica una prima serie di trasformazioni che rimpiazza il massimo numero di argomenti con una costante:

$$
\begin{aligned}
ITE(F,F,G) &\Rightarrow ITE(F,1,G) \\
ITE(F,G,F) &\Rightarrow ITE(F,G,0) \\
ITE(F,G,\overline{F}) &\Rightarrow ITE(F,G,1) \\
ITE(F,\overline{F},G) &\Rightarrow ITE(F,0,G)
\end{aligned}
$$

Una seconda serie di trasformazioni usa le seguente uguaglianze

$$
\begin{aligned}
ITE(F,1,G) &= ITE(G,1,F) \\
ITE(F,G,0) &= ITE(G,F,0) \\
ITE(F,G,1) &= ITE(\overline{G},\overline{F},1) \\
ITE(F,0,G) &= ITE(\overline{G},0,\overline{F}) \\
ITE(F,G,\overline{G}) &= ITE(G,F,\overline{F})
\end{aligned}
$$

per scegliere la tripla il cui primo argomento ha la variabile piu' in alto nel suo grafo con indice minimo nell'ordinamento.

Un'altra serie di trasformazioni usa le seguenti uguaglianze

$$ITE(F,G,H) = ITE(\overline{F},H,G) = \overline{ITE(F,\overline{G},\overline{H})} = \overline{ITE(\overline{F},\overline{H},\overline{G})}$$

In questo caso si sceglie la tripla ai cui due primi argomenti si punta con lati non complementati. Per es., se F non e' complementato e G e' complementato, si sostituisce $ITE(F,G,H)$ con $ITE(F,\overline{G},\overline{H})$ e se ne prende il complemento (da $ITE(F,G,H) = \overline{ITE(F,\overline{G},\overline{H})}$. Similmente, se F e' complementato e G non e' complementato, si si sostituisce $ITE(F,G,H)$ con $ITE(\overline{F},H,G)$ (da $ITE(F,G,H) = ITE(\overline{F},H,G)$).

Tra le conseguenze di queste trasformazioni c'e' la possibilita' di applicare la legge di De Morgan: $F.G = \overline{(\overline{F}+\overline{G})}$; infatti

$$F.G = ITE(F,G,0) = \overline{ITE(\overline{F},1,\overline{G})} = \overline{(\overline{F}+\overline{G})}$$

(si ha $ITE(F,G,0) = \overline{ITE(\overline{F},1,\overline{G})}$ da $ITE(F,G,H) = \overline{ITE(\overline{F},\overline{H},\overline{G})}$).

11. Il grafo delle precedenze (V, E) ha come nodi le operazioni $v_i,\ i = 0, 1, \ldots, n$ (v_0 e v_n operazioni nulle) e come lati (v_j, v_i) le relazioni di precedenza (cioe' $(v_j, v_i) \in E$ se v_j deve precedere v_i).

    Inoltre siano $d_i,\ i = 0, 1, \ldots, n$ i tempi di esecuzione delle operazioni $v_i,\ i = 0, 1, \ldots, n$ ($d_0 = d_n = 0$), e $\mathcal{T} : V \rightarrow \{1, 2, \ldots, n_{res}\}$ l'unico tipo di risorsa che realizza una data operazione.

    Si consideri il grafo delle precedenze in Fig. 1.

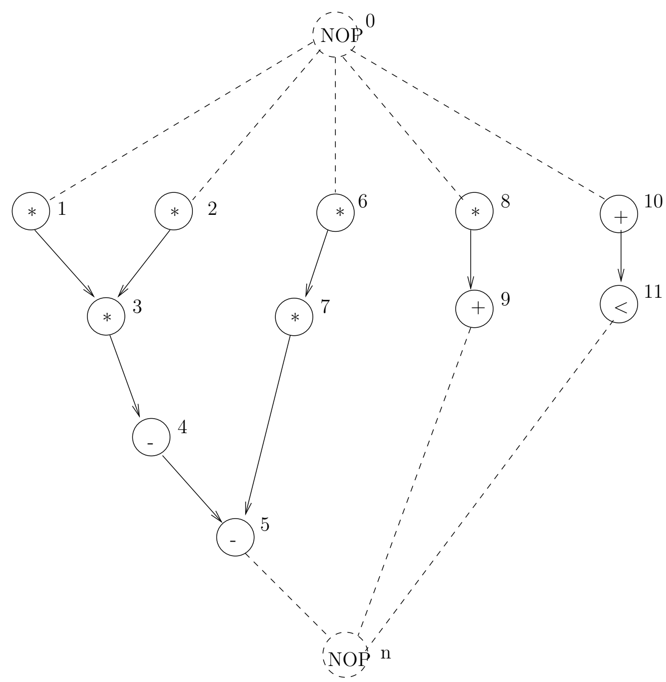

Figure 1: Grafo delle precedenze (da un algoritmo per integrare numericamente con Eulero in avanti l'equazione differenziale $y'' + 3xy' + 3y = 0$).

   (a) (Schedulazione senza vincoli) Si applichi al grafo delle precedenze in Fig. 1 l'algoritmo di schedulazione ASAP e se ne mostrino i passi.

   (b) (Schedulazione con vincolo sulla latenza) Si applichi al grafo delle precedenze in Fig. 1 l'algoritmo di schedulazione ALAP con latenza $\overline{\lambda} = 4$ e se ne mostrino i passi.

   (c) (Schedulazione euristica per latenza minima con l'algoritmo a lista)

      i. Si applichi l'algoritmo a lista al grafo delle precedenze in Fig. 1 con $a_1 = 2$ moltiplicatori e $a_2 = 2$ unita' aritmetico-logica. La funzione

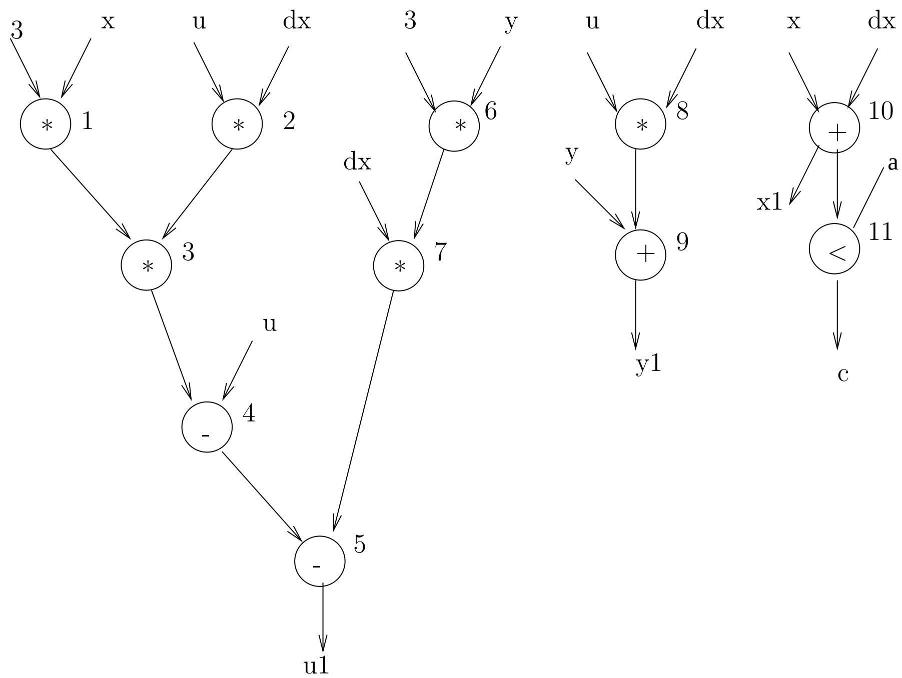

Figure 2: Grafo del flusso dei dati (da un algoritmo per integrare numericamente con Eulero in avanti l'equazione differenziale $y'' + 3xy' + 3y = 0$).

di priorita' e' data dalla seguente etichettatura: $v_1 \rightarrow 4$, $v_2 \rightarrow 4$, $v_3 \rightarrow 3$, $v_4 \rightarrow 2$, $v_5 \rightarrow 1$, $v_6 \rightarrow 3$, $v_7 \rightarrow 2$, $v_8 \rightarrow 2$, $v_9 \rightarrow 1$, $v_{10} \rightarrow 2$, $v_{11} \rightarrow 1$.

Inoltre i tempi di esecuzione delle operazioni siano $d_1 = 2$ per il moltiplicatore e $d_2 = 1$ per l'unita' aritmetico-logica.

Si usi come funzione di priorita' il peso del percorso piu' lungo al pozzo.

ii. Si applichi l'algoritmo a lista al grafo delle precedenze in Fig. 1 con $a_1 = 3$ moltiplicatori e $a_2 = 1$ unita' aritmetico-logica. Inoltre i tempi di esecuzione delle operazioni siano $d_1 = 2$ per il moltiplicatore e $d_2 = 1$ per l'unita' aritmetico-logica.

Si usi come funzione di priorita' il peso del percorso piu' lungo al pozzo.

(d) (Schedulazione euristica per risorse minime con l'algoritmo a lista)

Si applichi l'algoritmo a lista al grafo delle precedenze in Fig. 1 nell'ipotesi che tutte le operazioni richiedono un tempo unitario e che si richieda una latenza di 4 cicli.

(e) (Schedulazione per latenza minima con vincoli sulle risorse)

Si consideri la formulazione classica con ILP della schedulazione con vincoli sulle risorse, rispetto alle variabili $X = \{x_{il}, i = 0, 1, \ldots, n; l = 1, 2, \ldots, \overline{\lambda} + 1\}$, con $x_{il} = 1$ solo quando l'operazione v_i inizia al passo l della schedulazione, e $\overline{\lambda}$ maggiorante sulla latenza:

Si noti che $x_{il} = 0$ per $l < t_i^S$ o $l > t_i^L$ per ogni operazione $v_i, i = 0, 1, \ldots, n$, dove t_i^S sono i minoranti calcolati dall'algoritmo ASAP ($x_{il} = 1 \rightarrow l \geq t_i^S$) e t_i^L sono i maggioranti calcolati dall'algoritmo ALAP ($x_{il} = 1 \rightarrow l \leq t_i^L$), da cui

$$\Sigma_l \; x_{il} = \Sigma_{l=t_i^S}^{t_i^L} \; x_{il}, \; i = 0, 1, \ldots, n$$

(il tempo d'inizio d'ogni operazione e' unico, percio' l'inizio t_i di ogni operazione v_i puo' essere scritto come $t_i = \Sigma_l \; l.x_{il}$, $t_i^S \leq t_i \leq t_i^L$).

I vincoli sono:

$$(*) \; \Sigma_l \; x_{il} = 1, i = 0, 1, \ldots, n$$

(il tempo d'inizio d'ogni operazione e' unico)

$$(**) \; \Sigma_l \; l.x_{il} \geq \Sigma_l \; l.x_{jl} + d_j, \; i, j = 0, 1, \ldots, n, \; (v_j, v_i) \in E$$

(le relazioni di precedenza devono essere soddisfatte)

$$(***) \; \Sigma_{i:\mathcal{T}(v_i)=k} \; \Sigma_{m=l-d_i+1}^{l} x_{im} \leq a_k, \; k = 1, 2, \ldots, n_{res}, \; l = 1, 2, \ldots, \overline{\lambda}+1$$

(i vincoli sulle risorse devono essere soddisfatti ad ogni passo della schedulazione)

Ogni insieme di tempi iniziali che soddisfa i vincoli precedenti ci da' una soluzione ammissibile.

Si consideri la schedulazione con vincoli sulle risorse del grafo in Fig. 1. Si assuma che ci siano due tipi di risorse: moltiplicatore e unita' aritmetico-logica (che esegue somme/sottrazioni e confronti). Entrambe le risorse eseguano in un ciclo. Si assuma anche che si possano usare al piu' due moltiplicatori e due unita' aritmetico-logiche: $a_1 = 2$, $a_2 = 2$.

i. Usando l'algoritmo euristico di schedulazione a lista ("list scheduling") si trovi un maggiorante sulla latenza.

ii. Con gli algoritmi ASAP and ALAP si trovino dei vincoli sui tempi d'inizio, t_i^S and t_i^L.

iii. Si scrivano in dettaglio tutte le equazioni precedenti per l'unicita' dei tempi di partenza, le relazioni di precedenza e le quantita' di risorse disponibili.

iv. Si risolvano i vincoli precedenti (*), (**), (***) per ottenere una soluzione ammissibile.

v. Si risolvano i vincoli precedenti (*), (**), (***) con la funzione di minimo

$$min\ \mathbf{c}^T \mathbf{t}$$

e $\mathbf{c} = [0, 0, \ldots, 1]^T$ equivalente a

$$min\ t_n = \Sigma_l\ l.x_{nl}$$

che corrisponde a minimizzare la latenza della schedulazione (poiche' $\overline{\lambda} = t_n - t_0$ una soluzione ottima implica $t_0 = 1$).

vi. Si risolvano i vincoli precedenti (*), (**), (***) con la funzione obiettivo

$$min\ \mathbf{c}^T \mathbf{t}$$

e $\mathbf{c} = [1, 1, \ldots, 1]^T$ equivalente a

$$min\ \Sigma_i t_i = \Sigma_i \Sigma_l\ l.x_{il}$$

che corrisponde a minimizzare il tempo iniziale per tutte le operazioni.

(f) (Schedulazione per risorse minime con vincoli sulla latenza)
L'obiettivo dell'ottimizzazione e' una somma pesata delle risorse usate rappresentata dal vettore $\mathbf{a}$, esprimibile come

$$min\ \mathbf{c}^T \mathbf{a}$$

dove $\mathbf{c}$ e' il vettore dei costi delle singole risorse. Nelle equazioni (***) gli $a_k,\ k = 1, 2, \ldots, n_{res}$ sono adesso incognite ausiliarie.
C'e' inoltre un vincolo sulla latenza

$$t_n = \Sigma_l\ l.x_{nl} \leq \overline{\lambda} + 1$$

Per i dati del problema precedente, assumendo inoltre $\mathbf{c} = [5, 1]^T$ (il moltiplicatore costa 5 unita' di area e l'unita' aritmetico-logica ne costa una) e un maggiorante sulla latenza di $\overline{\lambda} = 4$, si riscrivano i vincoli del tipo (***) (quelli dei tipi (*) e (**) sono uguali al caso precedente) e si minimizzi la funzione di costo

$$\mathbf{c}^T\mathbf{a} = 5.a_1 + 1.a_2$$