APPUNTI SULLA CONCORRENZA E SUI DIAGRAMMI DI PRECEDENZA

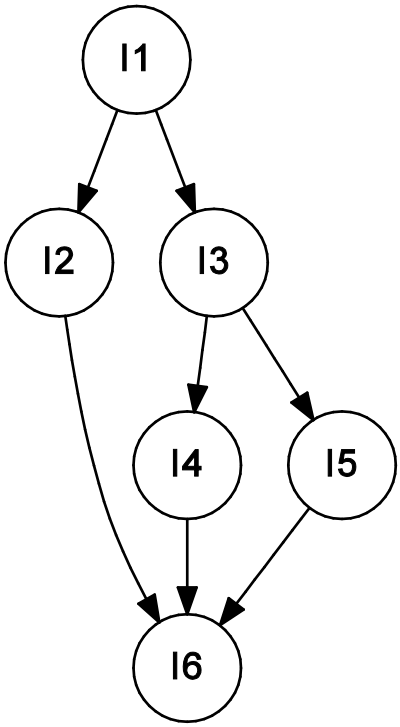
Una possibile sequenza di istruzioni (o esecuzioni/azioni elementari)

```
I1      a <- 10;
I2      c <- a + 4;
I3      b <- a - 2;
I4      d <- b + a;
I5      e <- b + 1;
I6      f <- d + e + c;
```

Segnaliamo le dipendenze (o precedenze); per es. I2 non può cominciare prima che I1 sia finito, ecc.

```
I1
I2 (dopo I1); I3 (dopo I1)
I4 (dopo I3); I5 (dopo I3)
I6 (dopo I2, I4 e I5)
```

La notazione grafica classica è quella dei **diagrammi di precedenza**, che hanno il seguente aspetto:



Per la cronaca, il diagramma a lato è stato disegnato con Graphviz (http://www.graphviz.org); ecco una delle possibili definizioni:

```
digraph ddpdemo {
        graph [rankdir=TB];
        node [shape=circle,
           fontname=Arial];
        I1 -> {I2 I3};
        I3 -> {I4 I5};
        {I2 I4 I5} -> I6;
}
```

L'interpretazione è la seguente: I2 non può cominciare prima che I1 sia finito, ecc.
Evidenziamo le coppie concorrenti (cioè che possono essere in esecuzione nello stesso istante, contemporaneamente)

`I2 con I3; I4 con I5` (queste coppie si “vedono” immediatamente)
ma anche:
`I2 con I4, I2 con I5` (meno evidenti)

© CESARE BOTTARO, 2009-2010, NON RIPRODUCIBILE SENZA PERMESSO DELL'AUTORE

Condizioni di Bernstein

È necessario garantire che il risultato delle elaborazioni non dipenda dal “percorso” seguito; a questo scopo si devono verificare, per tutte le coppie di attività potenzialmente concorrenti, le tre condizioni seguenti:

1) $D_i \bigcap R_j = \varnothing$
2) $D_j \bigcap R_i = \varnothing$
3) $R_i \bigcap R_j = \varnothing$

Le condizioni si applicano sugli insiemi degli ingressi e delle uscite (detti rispettivamente **dominio**, indicato con D, e **codominio**/range, indicato con R) delle elaborazioni; in pratica si verificano sulle variabili.
Si potrebbero riassumere dicendo che gli ingressi di ciascuna delle azioni parallelizzabili devono essere disgiunti dalle uscite di ciascun'altra azione, e che le uscite delle azioni parallelizzabili devono essere sempre disgiunti. Infine si noti che è sufficiente che una delle condizioni sia falsa per negare la potenziale parallelizzabilità delle azioni.
Verifichiamo le condizioni nel caso delle nostre 4 coppie di azioni.

Cond.	I2 e I3	I4 e I5	I2 e I4	I2 e I5
1	$\{a\} \bigcap \{b\} = \varnothing$	$\{a,b\} \bigcap \{e\} = \varnothing$	$\{a\} \bigcap \{d\} = \varnothing$	$\{a\} \bigcap \{e\} = \varnothing$
2	$\{a\} \bigcap \{c\} = \varnothing$	$\{b\} \bigcap \{d\} = \varnothing$	$\{a,b\} \bigcap \{c\} = \varnothing$	$\{b\} \bigcap \{c\} = \varnothing$
3	$\{c\} \bigcap \{b\} = \varnothing$	$\{d\} \bigcap \{e\} = \varnothing$	$\{c\} \bigcap \{d\} = \varnothing$	$\{c\} \bigcap \{e\} = \varnothing$

In tutti questi casi le 3 condizioni sono verificate; di fatto ciò non sorprende, perché le azioni erano state disposte in una sequenza corretta già in fase di determinazione delle precedenze.

Semplificazione dei diagrammi di precedenza

Quando in un diagramma di precedenza compaiono percorsi che conducono ad un'azione successiva secondo una via “più breve”, vanno eliminati. Infatti questi percorsi rappresentano attese inutili, perché sono già garantite dal passaggio attraverso altre azioni. Esempio nelle figure accanto.

Diagramma originale

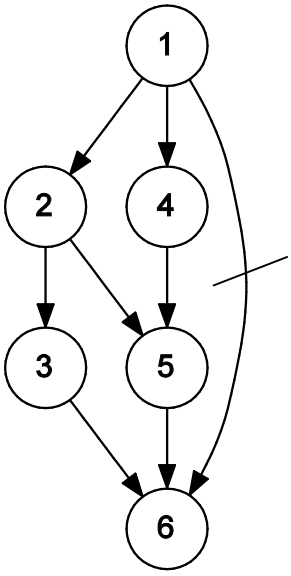
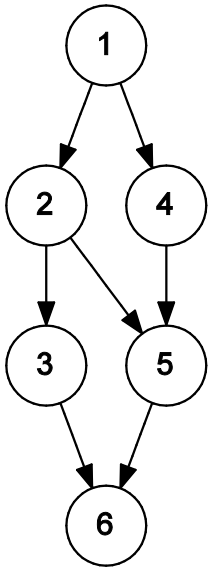


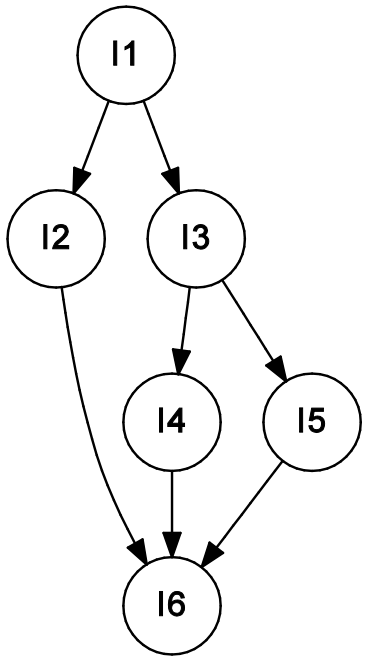
Diagramma semplificato



Traduzione dei diagrammi di precedenza con parbegin/parallel/cobegin

I diagrammi si possono tradurre con parbegin {} o con costrutti fatti da coppie fork-join, per le quali occorre usare contatori appositi.
Mentre le fork-join possono tradurre tutti i diagrammi di precedenza, le parbegin possono tradurre solo diagrammi **strutturati**, cioè in cui le strutture parallele hanno un solo ingresso e una sola uscita.

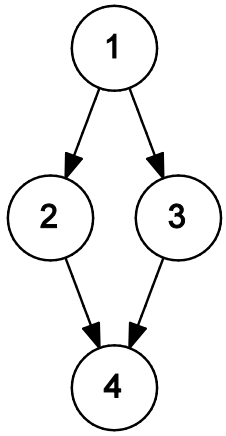
Traduzione del primo diagramma con parbegin




```
I1
parbegin {
     I2
     {
          I3
          parbegin {
               I4
               I5
          }
     }
}
I6
```

Da notare che è bene rispettare rigorosamente l'indentazione, se non si vuole perdere il controllo della struttura.
Si noti che c'è una sequenza interna, dove I3 precede la struttura parallela formata da I4 e I5. Quindi occorre un livello apposito di parentesi graffe.

Vediamo il caso più semplice possibile (una sola biforcazione) in entrambe le notazioni



Uso di **parbegin**

```
1;
parbegin {
     2;
     3;
}
4;
```

Uso di **fork-join**

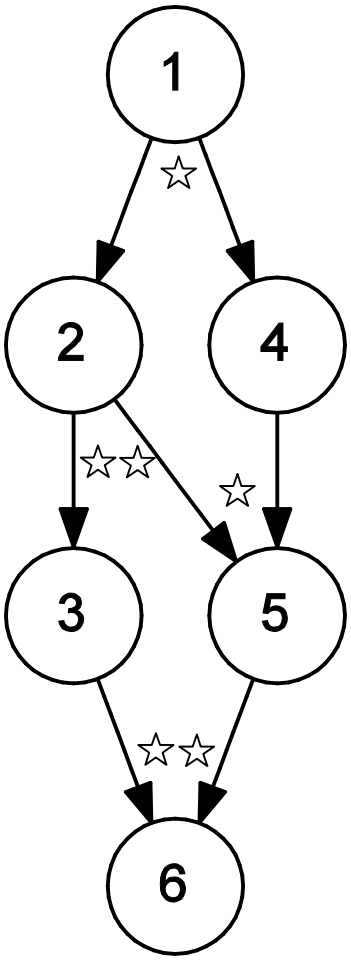
```
     1;
     c <- 2;
     fork l3;
     2;
     goto l4;
     // ---
l3:  3;
l4:  join c;
     4;
```

NOTA: traduzione solo didattica, **non** va fatta in questo caso

© CESARE BOTTARO, 2009-2010, NON RIPRODUCIBILE SENZA PERMESSO DELL'AUTORE

Traduzione dei diagrammi di precedenza con fork-join

Come detto sopra, tutti i diagrammi si possono tradurre in codice con i costrutti fork-join, anche i diagrammi non strutturati. I diagrammi si riconoscono perché contengono sezioni con due ingressi o con due uscite. Vediamo la traduzione in un caso “classico” di doppia biforcazione.




```
      1;
      c1 <- 2; // biforco
      fork l4; // rinvio 4
      2;
      c2 <- 2; // biforco
      fork l5; // rinvio 5
      3;
      goto l6;
      // ---
l4:   4;
l5:   join c1; // attendo 2 volte
      5;
l6:   join c2; // attendo 2 volte
      6;
```

Le “stelline” nel diagramma servono solo ad identificare la posizione di biforcazione e ricongiungimento relative ai contatori utilizzati, che abbiamo chiamato c1 (una stellina) e c2 (due stelline).
Il significato dell’istruzione fork è biforcare l’esecuzione, cioè procedere contemporaneamente all’istruzione successiva e a quella indicata in etichetta.
L’istruzione join decrementa il contatore associato e permette di procedere solo se assume il valore zero, cioè se tutti i rami concorrenti sono stati eseguiti. Va associata a un’etichetta che precede l’azione su cui l’esecuzione si ricongiunge.

Un’ultima annotazione: se un diagramma è strutturato, **deve** essere tradotto con parbegin; tradurlo con fork-join senza necessità è un errore, perché complica inutilmente la scrittura e l’interpretazione del codice.

© CESARE BOTTARO, 2009-2010, NON RIPRODUCIBILE SENZA PERMESSO DELL'AUTORE