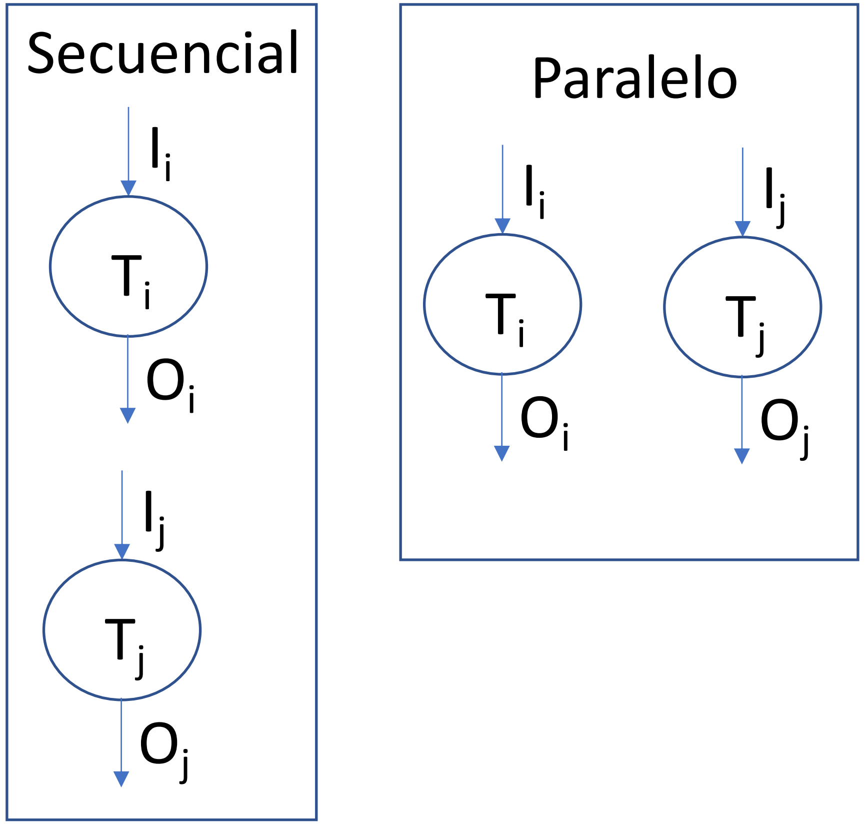
Transparencias adicionales T2

Condiciones de Bernstein



1. Si $O_i \cap I_j \neq \emptyset$, entonces puede suceder que una variable de entrada es usada por T_j cuando no ha sido modificada por T_i (**dependencia de flujo**)
2. Si $O_j \cap I_i \neq \emptyset$, entonces puede suceder que una variable de salida de T_j sea modificada, no habiéndose sido leída por T_i (**anti-dependencia**)
3. Si $O_i \cap O_j \neq \emptyset$, entonces puede suceder las dos tareas estén intentando a la vez modificar una variable de salida (**dependencia de salida**)

Condiciones de Bernstein para la independencia de tareas:

$$O_i \cap I_j = \emptyset \quad O_j \cap I_i = \emptyset \quad O_i \cap O_j = \emptyset$$

Para que $\mathbf{T_i}$ y $\mathbf{T_j}$ (en secuencial $\mathbf{T_i}$ precede a $\mathbf{T_j}$) puedan ejecutarse en paralelo es necesario que independientemente de la ejecución en paralelo de ambos el resultado final sea el mismo que la ejecución secuencial

Dependencia entre iteraciones en un bucle (Tr T2-15)

```
for (i=1; i<n; i++) {
    b[i] = b[i] + a[i-1];
    a[i] = a[i] + c[i];
}
```

Hay dependencia entre iteraciones:
Hay una dependencia de flujo en la variable a[1], ya que a[1] es una variable de salida para la iteración i=1 y de entrada para la iteración i=2. Si por ejemplo, la iteración i=1 la realizara el hilo H0 y la iteración i=2 la realizara el hilo H1, podría ocurrir que se realizase antes la iteración i=2 que la de i=1, por lo que el valor b[2] se calcularía mal.

Problema: Cuando las variables a[i] (salida) y a[i-1] (entrada) se usan en la misma iteración, existirá una dependencia entre iteraciones

i=1
b[1] = b[1] + a[0]
a[1] = a[1] + c[1]

i=2
b[2] = b[2] + a[1]
a[2] = a[2] + c[2]

i=3
b[3] = b[3] + a[2]
a[3] = a[3] + c[3]

……

i=n-3
b[n-3] = b[n-3] + a[n-4]
a[n-3] = a[n-3] + c[n-3]

i=n-2
b[n-2] = b[n-2] + a[n-3]
a[n-2] = a[n-2] + c[n-2]

i=n-1
b[n-1] = b[n-1] + a[n-2]
a[n-1] = a[n-1] + c[n-1]

Solución

i=1

$b[1] = b[1] + a[0]$

$a[1] = a[1] + c[1]$

i=2

$b[2] = b[2] + a[1]$

$a[2] = a[2] + c[2]$

i=3

$b[3] = b[3] + a[2]$

$a[3] = a[3] + c[3]$

……

i=n-3

$b[n-3] = b[n-3] + a[n-4]$

$a[n-3] = a[n-3] + c[n-3]$

i=n-2

$b[n-2] = b[n-2] + a[n-3]$

$a[n-2] = a[n-2] + c[n-2]$

i=n-1

$b[n-1] = b[n-1] + a[n-2]$

$a[n-1] = a[n-1] + c[n-1]$

$b[1] = b[1] + a[0]$

i=1

$a[1] = a[1] + c[1]$

$b[2] = b[2] + a[1]$

i=2

$a[2] = a[2] + c[2]$

$b[3] = b[3] + a[2]$

$a[3] = a[3] + c[3]$

……

$b[n-3] = b[n-3] + a[n-4]$

i=n-2

$a[n-3] = a[n-3] + c[n-3]$

$b[n-2] = b[n-2] + a[n-3]$

i=n-1

$a[n-2] = a[n-2] + c[n-2]$

$b[n-1] = b[n-1] + a[n-2]$

$a[n-1] = a[n-1] + c[n-1]$

```
b[1] = b[1] + a[0];
for (i=1; i<n-1; i++) {
    a[i] = a[i] + c[i];
    b[i+1] = b[i+1] + a[i];
}
a[n-1] = a[n-1] + c[n-1];
```

Cuestión 1-1 (boletín)

Según las condiciones de Bernstein, indica el tipo de dependencias de datos existente entre las distintas iteraciones en los casos que se presentan a continuación. Justifica si se puede eliminar o no esa dependencia de datos, eliminándola en caso de que sea posible

a)

```
for (i=1;i<N;i++)
   x[i-1] = x[i-1] + x[i];
```

$i=1 \quad x[0]=x[0]+\boxed{x[1]}$

$i=2 \quad \boxed{x[1]}=x[1]+\boxed{x[2]}$

$i=3 \quad \boxed{x[2]}=x[2]+x[3]$

....

$i=N-3 \quad x[N-4]= x[N-4]+\boxed{x[N-3]}$

$i=N-2 \quad \boxed{x[N-3]}= x[N-3]+\boxed{x[N-2]}$

$i=N-1 \quad \boxed{x[N-2]}= x[N-2]+x[N-1]$

- Hay antidependencias en las iteraciones del bucle **for**
- No se pueden eliminar

Cuestión 1-1 (boletín)

b)

```
for (i=0;i<N;i++) {
    a[i] = a[i] + y[i];
    x =2.0+ a[i];
}
```

i=0	a[0]=a[0]+y[0]
	x=2.0+a[0]
i=1	a[1]=a[1]+y[1]
	x=2.0+a[1]
i=2	a[2]=a[2]+y[2]
	x=2.0+a[2]
….	…..
i=N-1	a[N-3]=a[N-3]+y[N-3]
	x=2.0+a[N-3]
i=N-2	a[N-2]=a[N-2]+y[N-2]
	x=2.0+a[N-2]
i=N-1	a[N-1]=a[N-1]+y[N-1]
	x=2.0+a[N-1]

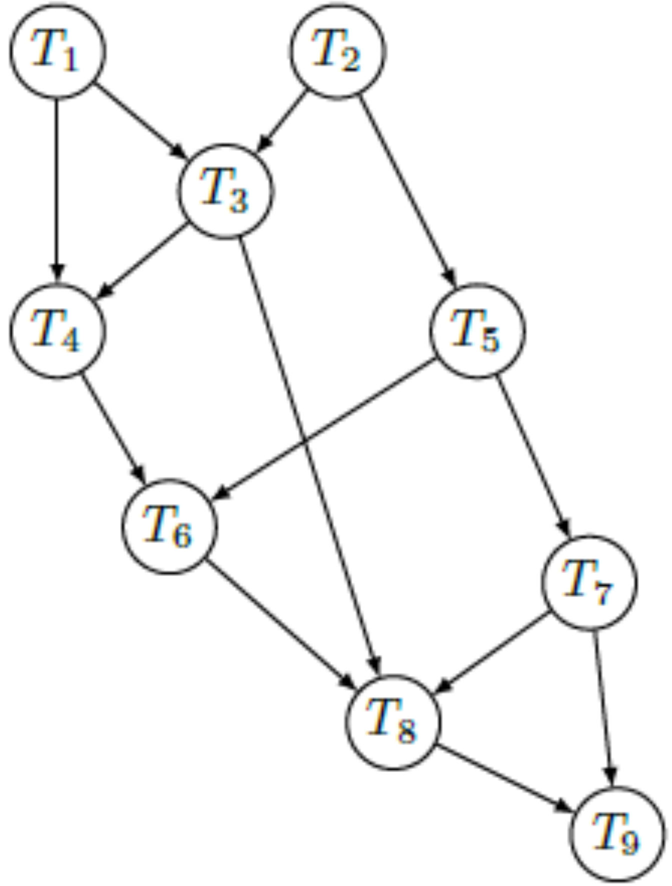
Hay dependencias de salida en las iteraciones del bucle **for** pues la variable x es de salida en diferentes iteraciones (dependencia de salida)

Solución

```
for (i=0;i<N;i++) {
    a[i] = a[i] + y[i];
}
x =2.0+a[N-1];
```

Cuestión 1-2 (boletín)

a) Dado el siguiente grafo de dependencias, en donde el coste de las tareas es igual a 1, determinar los caminos críticos, la longitud del camino crítico, el grado máximo de concurrencia y el grado medio de concurrencia



Solución:

Caminos críticos: T1−T3−T4−T6−T8−T9
T2−T3−T4−T6−T8−T9

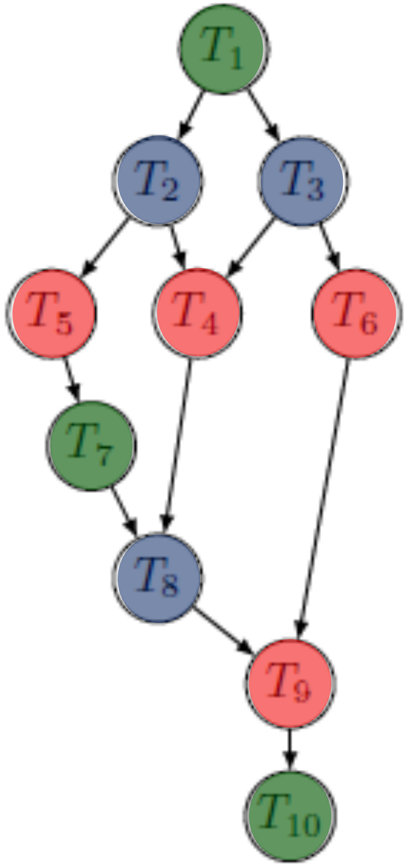
Longitud del camino crítico: L = 6

El grado máximo de concurrencia: 2

El grado medio de concurrencia: $M = \frac{9}{6} = 1.5$

Cuestión 1-2 (boletín)

b) Repite el apartado anterior para el siguiente grafo. Nota: en este caso el coste de cada tarea viene dado en flops (para un tamaño de problema n) según la tabla mostrada



T_1, T_7, T_{10}	$\frac{1}{3}n^3$
T_2, T_3, T_8	n^3
T_4, T_5, T_6, T_9	$2n^3$

Solución:

Camino crítico: T1–T2–T5–T7–T8–T9–T10

Longitud del camino crítico:

$$L = \frac{1}{3}n^3 + n^3 + 2n^3 + \frac{1}{3}n^3 + n^3 + 2n^3 + \frac{1}{3}n^3 = 7n^3$$

Grado máximo de concurrencia: 3

Grado medio de concurrencia:

$$M = \frac{3 \cdot \frac{1}{3}n^3 + 3 \cdot n^3 + 4 \cdot 2n^3}{7n^3} = \frac{12n^3}{7n^3} = \frac{12}{7}$$

Fórmulas para el cálculo de costes

$$\boxed{\sum_{i=0}^{n-1} 1 = n} \quad \boxed{\sum_{i=0}^{n-1} k = kn} \quad \boxed{\sum_{i=a}^{b} 1 = b - a + 1} \quad \boxed{\sum_{i=a}^{b} k = k(b - a + 1)} \quad \boxed{\sum_{i=0}^{n-1} i \simeq \frac{n^2}{2}} \quad \boxed{\sum_{i=0}^{n-1} i^2 \simeq \frac{n^3}{3}}$$

```
for( i=0; i<n; i++)
   for( j=0; j<n; j++)
      x = x + a[i][j];
```

$$\sum_{i=0}^{n-1} \boxed{\sum_{j=0}^{n-1} 1} = \sum_{i=0}^{n-1} n = n^2 \text{ flops}$$

```
for( i=0; i<n; i++)
   for( j=i; j<n; j++)
      x = x + 3*a[i][j];
```

$$\sum_{i=0}^{n-1} \boxed{\sum_{j=i}^{n-1} 2} = \sum_{i=0}^{n-1} 2(n-1-i+1) = \sum_{i=0}^{n-1} 2(n-i) = 2\sum_{i=0}^{n-1} n - 2\sum_{i=0}^{n-1} i \simeq 2n^2 - 2\frac{n^2}{2} = n^2 \text{ flops}$$

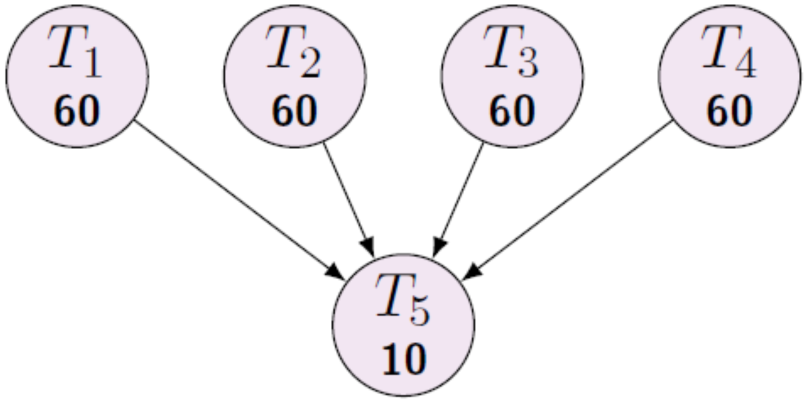
```
for( i=0; i<n; i++)
   for( j=i; j<n; j++)
         for( k=i; k<n; k++)
            x = x + a[i][k];
```

$$\sum_{i=0}^{n-1} \sum_{j=i}^{n-1} \boxed{\sum_{k=i}^{n-1} 1} = \sum_{i=0}^{n-1} \boxed{\sum_{j=i}^{n-1} (n-i)} = \sum_{i=0}^{n-1} (n-i)^2 = \sum_{i=0}^{n-1} n^2 - \sum_{i=0}^{n-1} 2ni + \sum_{i=0}^{n-1} i^2 \simeq \frac{n^3}{3} \text{ flops}$$

$$\boxed{n^3} = \sum_{i=0}^{n-1} n^2, \qquad \boxed{2n\sum_{i=0}^{n-1} i \simeq n^3} = \sum_{i=0}^{n-1} 2ni, \qquad \boxed{\frac{n^3}{3}} \simeq \sum_{i=0}^{n-1} i^2$$

Ejemplo

Determinar el speed-up y la eficiencia para la ejecución paralela de las tareas T_1, T_2, T_3 y T_4 de la figura, considerando un número de hilos igual a 4 (p=4)



$$t_1 = 60 + 60 + 60 + 60 + 10 = 250$$

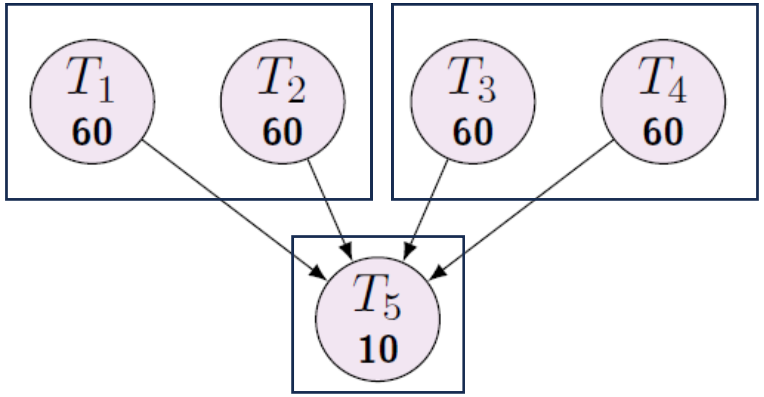
$$t_4 = 60 + 10 = 70$$

$$S_4 = \frac{t_1}{t_4} = \frac{250}{70} \approx 3.57$$

$$E_4 = \frac{S_4}{4} \approx 0.89$$

Ejemplo

Determinar el speed-up y la eficiencia para la ejecución paralela de las tareas T_1, T_2, T_3 y T_4 de la figura, considerando un número de hilos igual a 2 (p=2)



$$t_1 = 60 + 60 + 60 + 60 + 10 = 250$$

$$t_2 = 60 + 60 + 10 = 130$$

$$S_2 = \frac{t_1}{t_2} = \frac{250}{130} \approx 1.92$$

$$E_2 = \frac{S_2}{2} \approx 0.96$$