

# Seminario 1 PCTR

## 1. Presentación a nivel conceptual de los problemas del productor/consumidor, filósofos comensales y lectores/escritores.

Los problemas que pueden tener siempre vendrán dados por la naturaleza del enunciado del problema. De estos destacan:

- Buffer limitado (los productores no pueden producir mas por buffer lleno o los consumidores no pueden consumir más buffer vacío).
- 1 productor muchos consumidores, 1 consumidor muchos productores.
- Problemas de los 5 palillos y los 5 filósofos.

## 2. Construir un programa concurrente que modele el grafo de sincronización de la Figura 1, utilizando cobegin/coend.

```
Begin
  S1;
  Cobegin
    S3;
    Begin
      S2;
      S4;
      Cobegin
        S5;
        S6;
      Coend
    End
  Coend
  S7;
End;
```

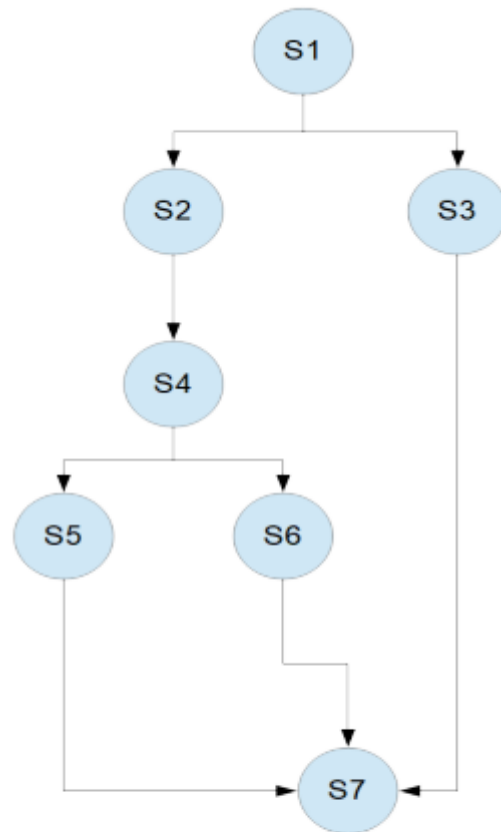


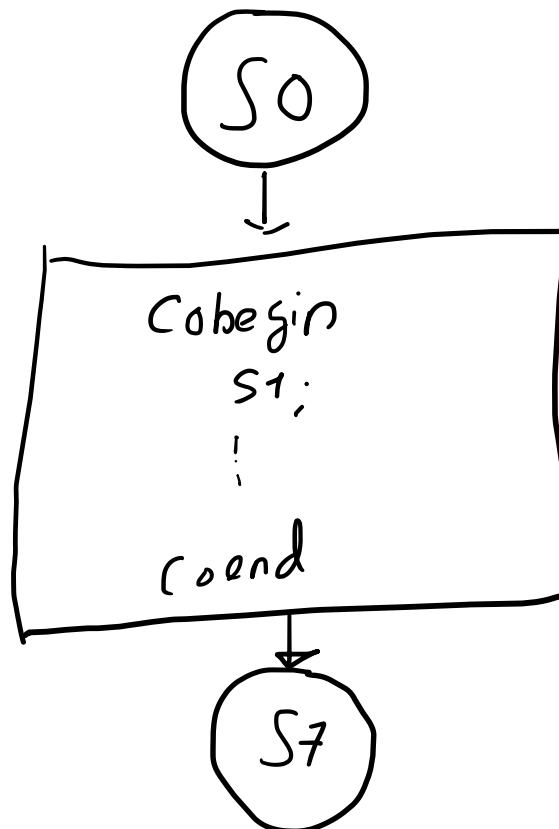
Figura 1: Grado de Precedencia

3. Dado el siguiente segmento de código concurrente, obtener su grafo de precedencia.

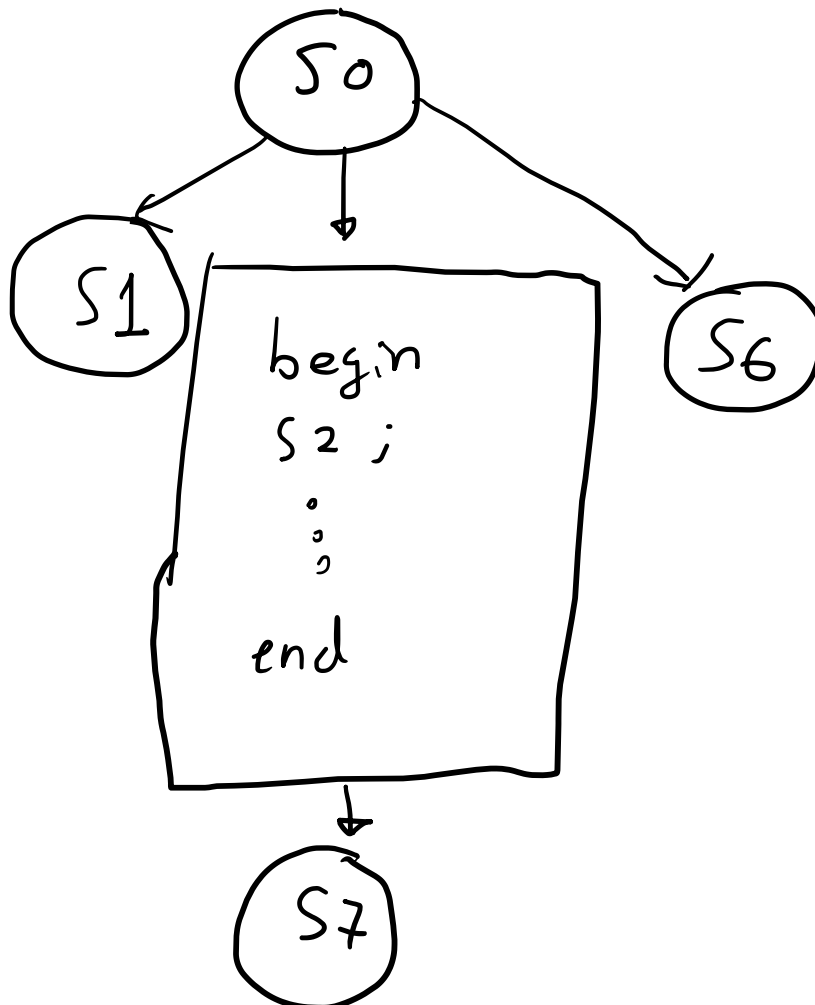
```
s0;  
cobegin  
  s1;  
begin  
  s2;  
cobegin  
  s3; s4;  
coend;  
s5;  
end;  
s6;  
coend;  
s7;
```

Voy a realizar el proceso a seguir:

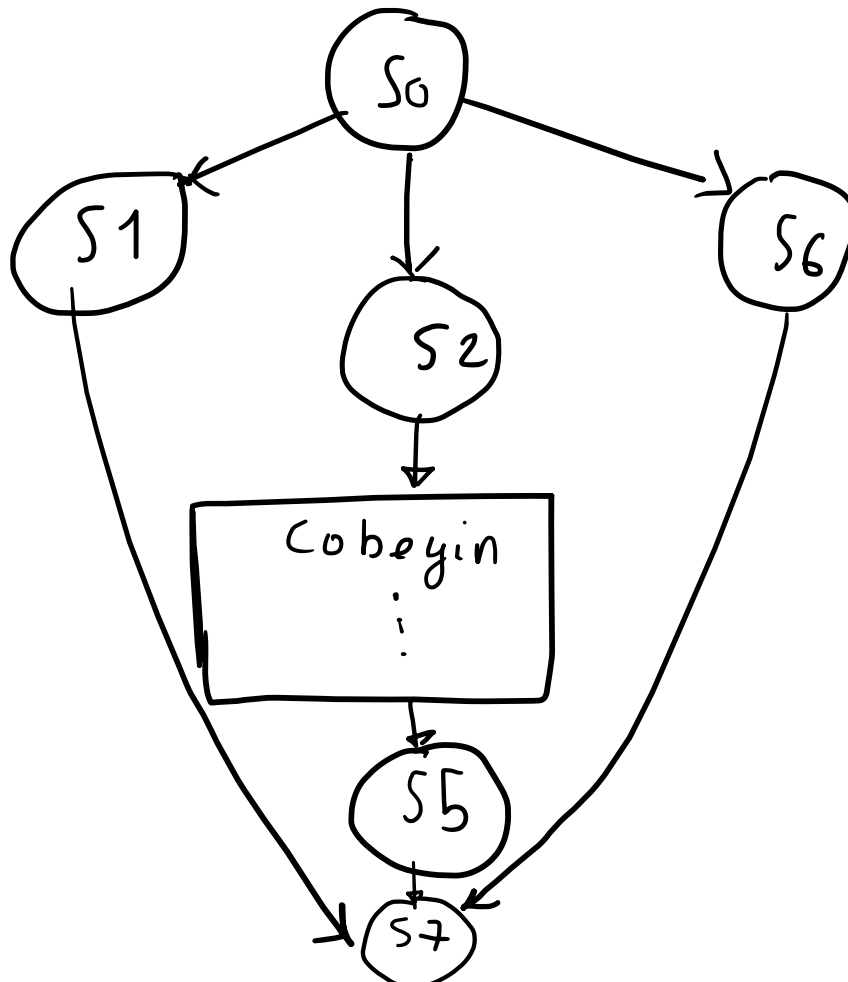
1-



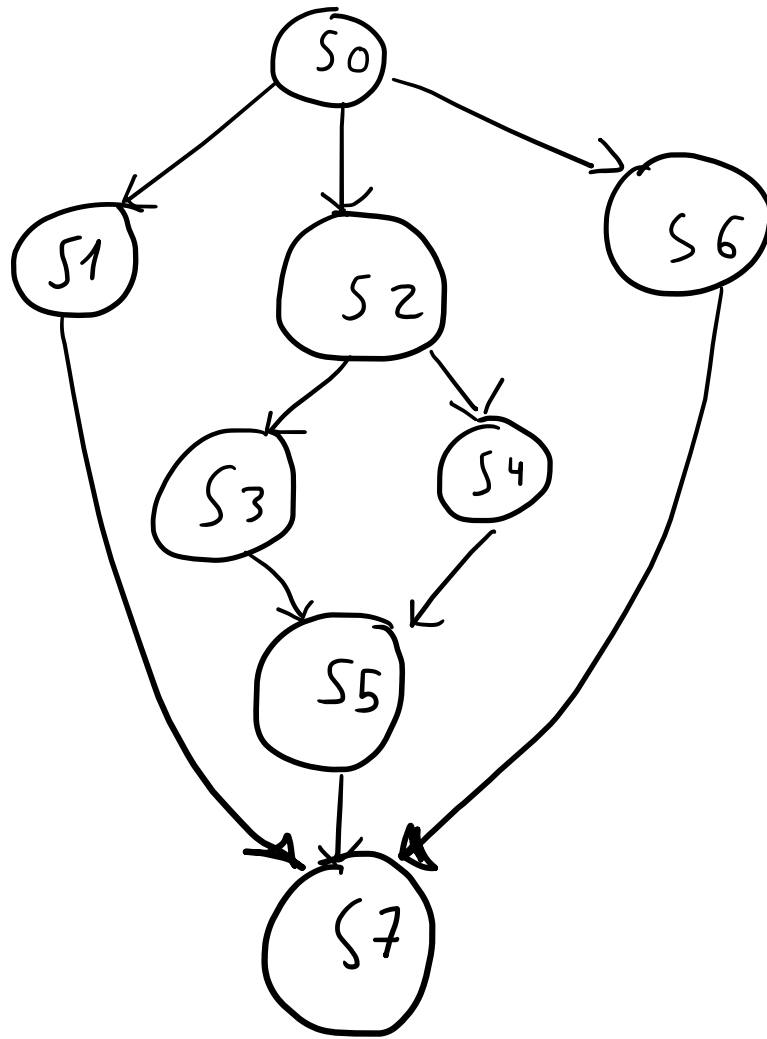
2-



3-



9-



4. Construir un programa concurrente que utilice la concurrencia para copiar un fichero f en un fichero g, utilizando el par cobegin/coend.

```
Var f,g: File of T;
Var r,s: T;
Begin
  abrirLectura(f);
  abrirEscritura(g);
  leer(f,r);
  while not finFichero(f) do
    begin
      s := r
      cobegin
        escribir(g,s)
        leer(f,r)
      conend
    end
    escribir(g,r) //ponte un ejemplo y veras porque hay que ponerlo
  end
```

5. Utilizar las ecuaciones de Bernstein para determinar cuáles de las siguientes instrucciones admiten ejecución concurrente:

```
S1: cuad:=x*x;
S2: m1:=a*cuad;
S3: m2:=b*x;
S4: z:=m1+m2;
S5: y:=z+c;
```

-----	S1	S2	S3	S4	S5
S1	-----	NO	SI	SI	SI
S2	-----	-----	SI	NO	SI
S3	-----	-----	-----	NO	SI
S4	-----	-----	-----	-----	NO
S5	-----	-----	-----	-----	-----