Lección 6: Propiedades de un programa concurrente

- Propiedades de un PC
- Propiedades de seguridad
- Prueba de propiedades de seguridad
- Propiedades de equidad

Propiedades de un PC

- **Propiedad** de un PC: atributo cierto para cualquier historia
- **Propiedad de seguridad**: garantiza que nada malo va a suceder
  - corrección parcial de un programa secuencial
  - exclusión mutua
  - ausencia de bloqueos (totales y/o parciales)
- **Propiedad de vivacidad**: garantiza que algo bueno sucederá, tarde o temprano
  - terminación de un programa secuencial
  - un mensaje llegará a su destino
  - un recurso solicitado será asignado
  - el comportamiento es "equitativo"

Prueba de propiedades de seguridad

- Algunas propiedades de seguridad se pueden expresar mediante la no verificación de un predicado "**B**" en ningún estado alcanzable
- **Prueba por el método de las aserciones críticas**:
  - sea "**B**" un predicado que identifica los estados "malos"
  - sea "**{Q} S {R}**" una verificación
  - **Si** *cada aserción crítica "**C**" cumple "**C** $\rightarrow \neg$**B**"*

    entonces *el programa cumple "$\neg$**B**"*

Prueba de propiedades de seguridad

- **Prueba por el método del invariante**:
  - sea "B" un predicado que identifica los estados "malos"
  - sea "`{Q} S {R}`" una verificación
  - sea "`I`" un invariante global
  - **Si** "$I \rightarrow \neg B$"

    entonces *el programa cumple* "$\neg B$"

Propiedades de equidad

- Supongamos como propiedad de vivacidad: todos los procesos activos terminan
  - ¿Posibles causas de que no se cumpla?
- Las propiedades de vivacidad vienen condicionadas por las **políticas de "scheduling"**
  - determinan, en cada instante, qué acciones elegibles han de ejecutarse a continuación
  - viene condicionada por la disponibilidad de recursos en el sistema
- La **equidad** ("fairness") es la garantía de que todo proceso tiene la posibilidad de evolucionar, independientemente de lo que hagan los restantes procesos
  - Se dice que el programa tiene un **comportamiento equitativo**

Propiedades de equidad

- La política de "scheduling" condiciona la equidad de un programa concurrente
- Diversos tipos de equidad, según la política de "scheduling" aplicada:
  - **Equidad incondicional**: se garantiza que ***toda acción atómica no condicionada elegible será tarde o temprano ejecutada***
  - **Equidad débil**: la incondicional y que ***toda acción atómica sincronizada elegible, cuya guarda se haga TRUE para siempre, será, tarde o temprano, ejecutada***
  - **Equidad fuerte**: la incondicional y que ***toda acción atómica sincronizada elegible cuya guarda se haga TRUE infinitas veces será, tarde o temprano, ejecutada***

Ejemplo

asumimos un único procesador

- Un "scheduler" que asigne el procesador al proceso "Bucle" provocaría un **comportamiento no equitativo**
- Cualquier **"scheduler" incondicionalmente equitativo** garantiza la equidad del programa (y, por lo tanto, terminación)

```
Vars
 esperar:Bool:=TRUE

Bucle::
Mq esperar
  seguir
FMq

Acaba::
  esperar:=FALSE
```

Otro ejemplo

asumimos un único procesador

- Un **"scheduler" incondicionalmente equitativo o débilmente equitativo NO** garantiza la equidad del programa (por lo tanto, no termina)
- Su comportamiento será **equitativo si y sólo si la política de "scheduling" es fuertemente equitativa**

```
Vars
 seguir: Bool:=TRUE
 intentar: Bool:=FALSE

Bucle::
 Mq seguir
   intentar:=TRUE
   intentar:=FALSE
 FMq

Acaba::
  <await intentar
     seguir:=FALSE
  >
```

Ejercicio

- Definir el predicado más débil posible, P, para que, asumiendo un "scheduler" débilmente equitativo el programa no se bloquee.

```
Vars x:Ent := ???

                --P
P1::                    P2::
<await x>=3             <await x>=2
   x := x-3                 x := x-1
>                       >
```