Simulación (y 2)

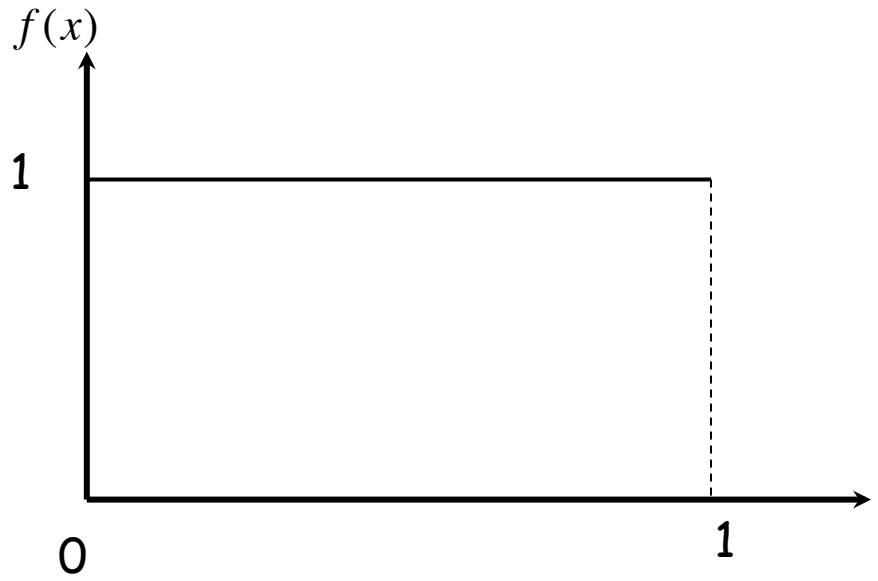
Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Grado en Ingeniería en Tecnologías de Telecomunicación, 4º

Generación de números aleatorios

Generación de números aleatorios

- Primero intentamos generar números aleatorios de una distribución uniforme
- Independientes
- Empleamos generadores de números pseudo-aleatorios (...)



Números pseudo-aleatorios

- Parece que fueran aleatorios
- Pero conocida la *semilla* se pueden predecir
- Incluso tienen un periodo
- Ejemplo: Linear Congruential Method

$$X_{i+1} = \left(aX_i + c\right) \bmod m$$

- ¿Y para una distribución no uniforme?

Implementación y uso

- Librerías habituales traen funciones como rand() o random() para generar números pseudo-aleatorios
- Suelen generar un número entero en un rango [0,MAX]
- Uniformes i.i.d.
- Son fáciles de transformar, por ejemplo si queremos un número entre 0 y 1 simplemente hacemos: rand()/MAX (en flotante)
- Si queremos un número real entre 1 y 100:

  99x(rand()/MAX) + 1
- Si queremos un dado de 6 caras (un número entero entre 1 y 6) haríamos:

  int(6x(rand()/(MAX+1))) + 1

Semilla

- El generador de números pseudo-aleatorios parte de una "semilla" para generar la secuencia (lo que sería el X_0)
- El siguiente número que obtenemos con la función es el siguiente de la secuencia
- Pero si ejecutamos el programa una segunda vez, parte de la misma semilla y repite la secuencia de valores
- Algunos lenguajes/librerías cambian ellos solos la semilla
- Poder repetir la secuencia es útil para depurar

$$X_{i+1} = \left(aX_i + c\right) \bmod m$$

Semilla

- Podemos cambiar la semilla
- Funciones como srand() o srandom (depende del lenguaje/ librería)
- Solo deberíamos usar esa función una vez en el programa, antes de empezar a generar números aleatorios
- Esa semilla tiene que ser un número diferente cada vez
- Se suele tomar algo derivado de la hora del día (en segundos, milisegundos, etc)
- Algo como: srand(time(NULL))
- Es un error habitual cambiar la semilla cada vez que vamos a generar un número aleatorio, NO debemos hacer eso

Semilla: Errores típicos

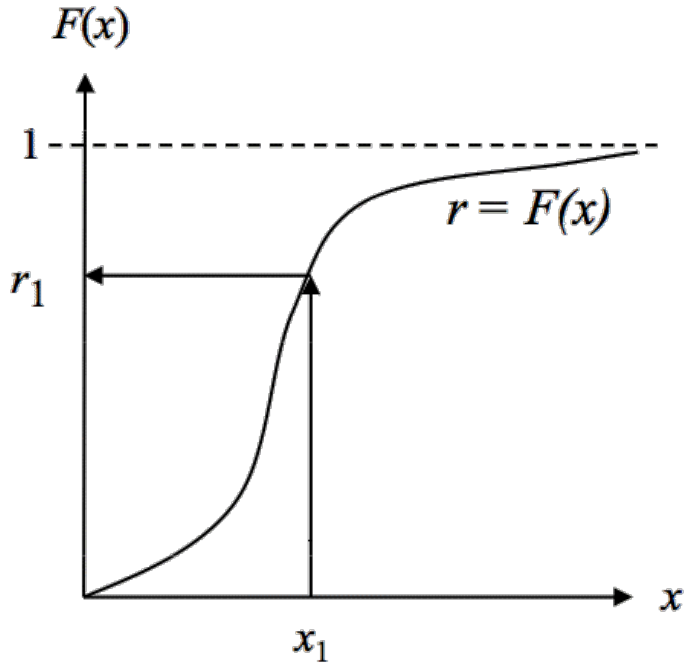
- Es un error habitual cambiar la semilla cada vez que vamos a generar un número aleatorio, NO debemos hacer eso
- La secuencia son números aproximadamente uniformes i.i.d pero no dice nada de su relación con los números de otra secuencia
- Si vamos a generar números para dos tareas (tiempos entre llegadas y tiempos de servicio) no debemos cambiar la semilla para generar de cada uno
- Podemos usar la misma secuencia de números para los dos, el primer nº para uno, el segundo para el otro, etc
- Algunas librerías permiten tener instancias independientes de la secuencia generadora de números

Semilla: Errores típicos

- Cuidado con los problemas colaterales y sutiles
- Ejemplo:
  - Generamos en un programa números aleatorios uniformes entre 1 y 100 con semilla K
  - Generamos en otro programa números aleatorios uniformes entre 80 y 10000 con la misma semilla K
  - Salen números diferentes pero *están correlados*
  - Podemos emplearlos en un sistema pero cuidado si luego pensamos que son independientes, pues no lo son
  - Por ejemplo si unos son tiempos entre llegadas y los otros son tiempos de servicio

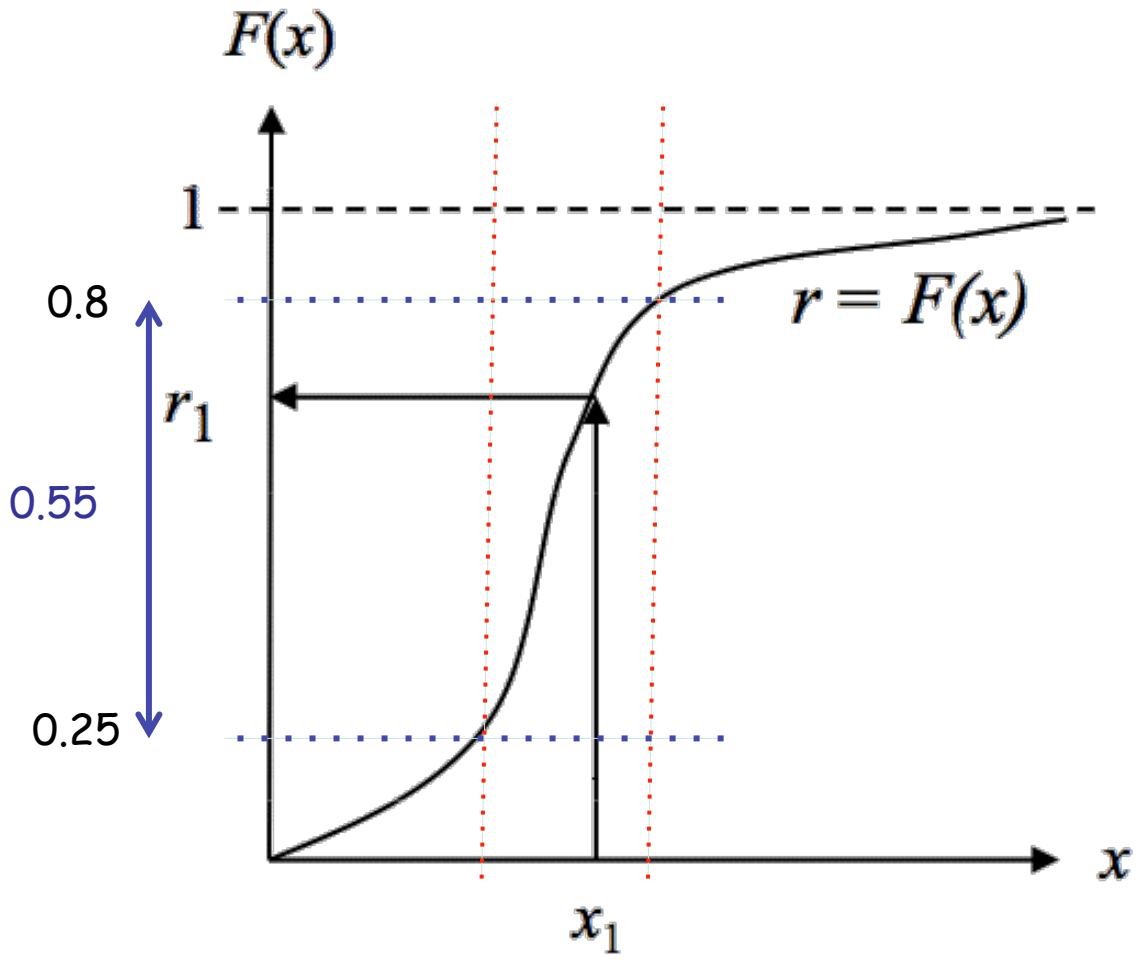
Inverse-transform Technique

- X variable aleatoria deseada
- F(x) = P[X<x]
- U variable aleatoria uniforme en [0,1]
- Se genera una muestra r_1 de U
- Se obtiene mediante la inversa de F(x) : $x_1 = F^{-1}(r_1)$
- x_1 es una muestra de la variable aleatoria X
- Esto es más sencillo si F(x) tiene una inversa “simple”



Ejemplo visual

- En ese rango la v.a. X tiene el 55% de sus valores
- Lo que estamos haciendo es que con un 55% de probabilidad (la uniforme) elegimos un valor de ese rango



Ejemplo: Distribución exponencial

$$f(x) = \lambda e^{-\lambda x}$$

$$F(x) = 1 - e^{-\lambda x}$$

$$R = 1 - e^{-\lambda X}$$

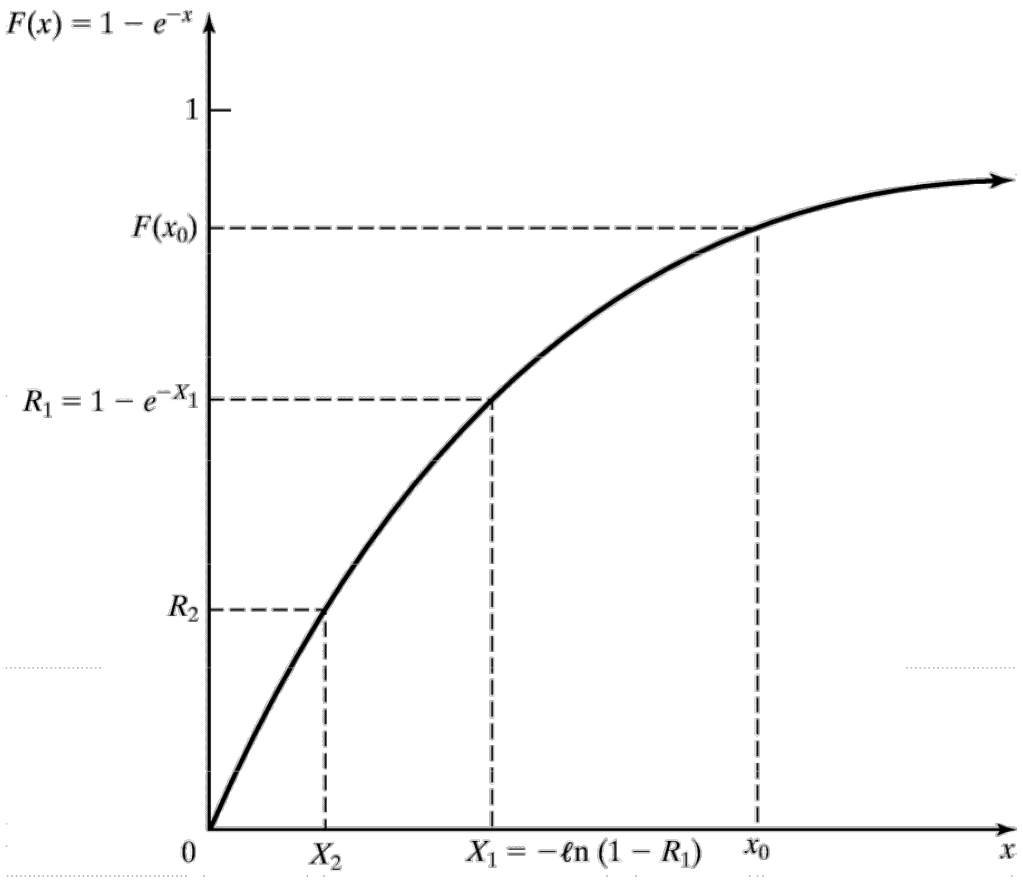
$$1 - R = e^{-\lambda X}$$

$$\ln(1-R) = -\lambda X$$

$$X = -\frac{\ln(1-R)}{\lambda} = F^{-1}(R)$$

ó

$$X = -\frac{\ln(R)}{\lambda} = F^{-1}(R)$$



(Tanto R como 1-R son variables aleatorias uniformes)

Ejemplo: Distribución exponencial

$$f(x) = \lambda e^{-\lambda x}$$

$$F(x) = 1 - e^{-\lambda x}$$

-log(rand()/MAX)/λ

$$R = 1 - e^{-\lambda X}$$

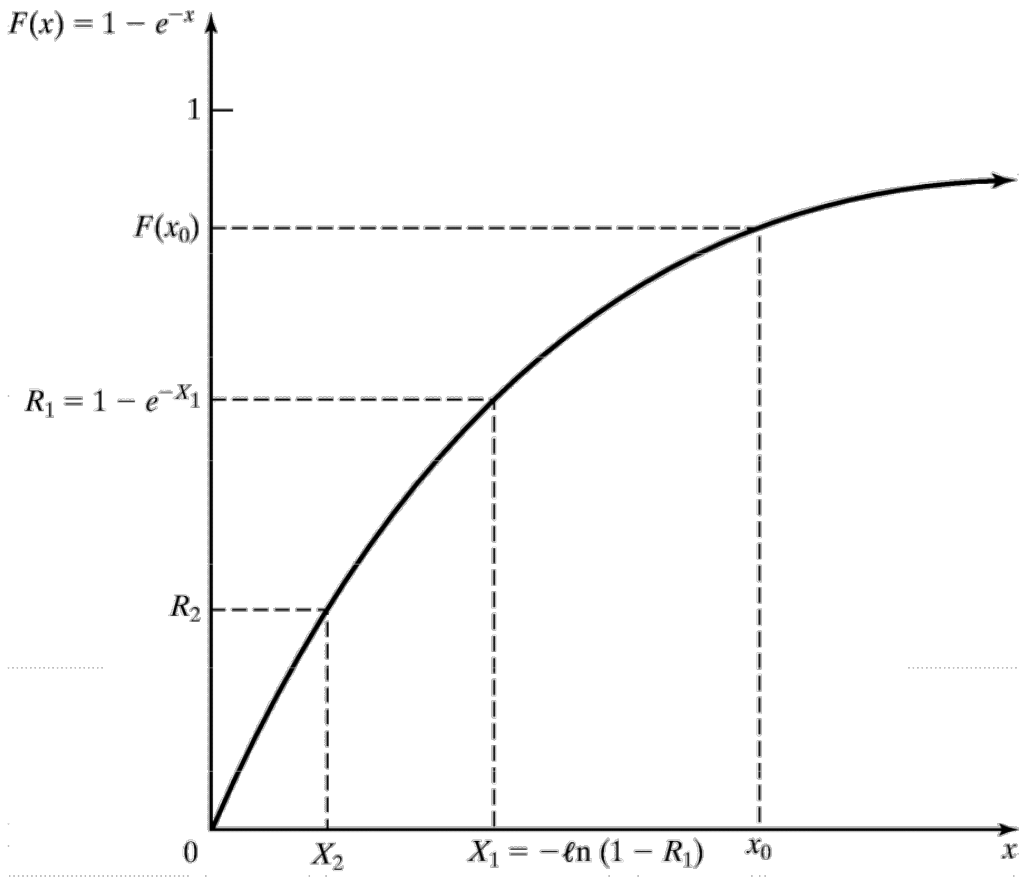
$$1 - R = e^{-\lambda X}$$

$$\ln(1 - R) = -\lambda X$$

$$X = -\frac{\ln(1 - R)}{\lambda} = F^{-1}(R)$$

ó

$$X = -\frac{\ln(R)}{\lambda} = F^{-1}(R)$$



(Tanto R como 1-R son variables aleatorias uniformes)

Inverse-transform Technique

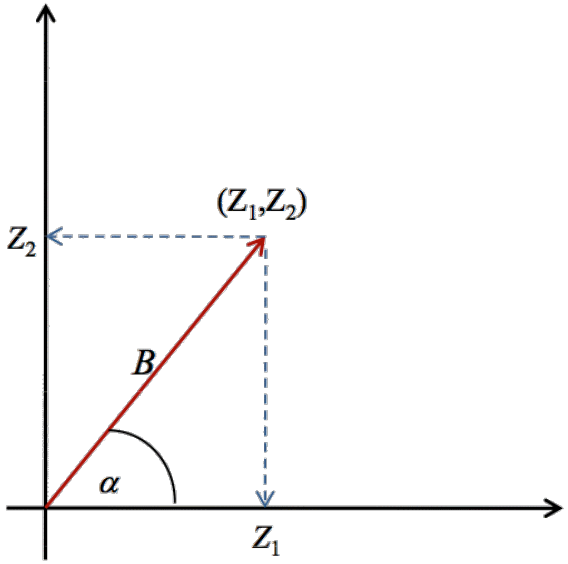
- Distribuciones “sencillas”: Triangular, Weibull, Pareto
- F(x) podría venir de una aproximación con datos experimentales
- Entonces se podría emplear interpolación para mejorar el resultado
- Para una variable aleatoria discreta es suficiente con una tabla
- “Difíciles”: Gamma, Normal, Beta
- Se tienen que emplear aproximaciones a F(x) o a $F^{-1}(x)$

Técnicas a partir de propiedades

Ejemplo: Gaussian distribution

- Z_1 y Z_2 variables aleatorias N(0,1)
- Supongamos que son coordenadas rectangulares de un punto (Z_1,Z_2)
- En coordenadas polares: $\begin{cases} Z_1 = B\cos(\alpha) \\ Z_2 = B\sin(\alpha) \end{cases}$
- El radio B es una variable aleatoria exponencial
- El ángulo α es una variable aleatoria uniforme
- Son independientes
- Así que se pueden obtener dos muestras de N(0,1) con dos muestras de una variable aleatoria uniforme

$$Z_1 = \sqrt{-2\ln(R_1)}\cos(2\pi R_2)$$
$$Z_2 = \sqrt{-2\ln(R_1)}\sin(2\pi R_2)$$



- Y para Y = N(μ,σ) :

$$Y = \mu + \sigma Z_i$$

Simulación de eventos discretos

Simulación de eventos discretos

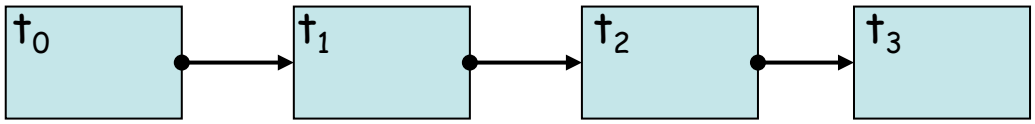
- Modelado con el tiempo de un sistema en el que todos los cambios de estado se producen en un conjunto discreto de puntos en el tiempo
- Empleo de métodos numéricos
  - En vez de métodos analíticos
  - El modelo se “corre” en vez de se “resuelve”
- Se lleva a cabo produciendo una secuencia de *snapshots* del sistema con el tiempo
- El *snapshot* en un instante *t* incluye
  - El estado del sistema en el instante *t*
  - Una lista de las actividades en progreso y cuándo terminarán
  - El estado de todas las entidades
  - Los valores de todos los contadores estadísticos

Future (pending) Events List (FEL)

- Es el mecanismo para hacer avanzar la simulación
- La FEL contiene los eventos planificados para este instante o posteriores aún sin procesar
- Cada evento contiene el instante de tiempo en que sucede
- Ordenados por instante de tiempo de menor a mayor
- Garantiza que los eventos tienen lugar en orden cronológico

Gestión de actividades con una FEL

- La duración se conoce al comenzar (determinista o aleatoria)
- En algunos entornos existe la posibilidad de cancelar
- Al comenzar la actividad se introduce un evento de finalización de la actividad en la FEL
- Ejemplo: Nueva llegada, actividad el tiempo hasta la siguiente (...)



CLOCK = $t < t_0$

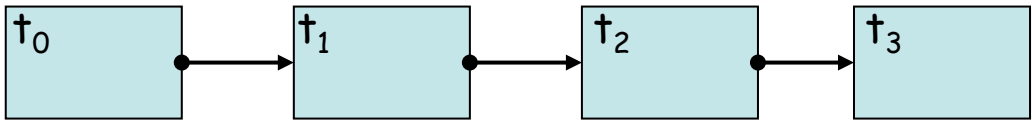
$$t_0 \leq t_1 \leq t_2 \leq t_3$$

Información fundamental a definir

- ¿Efectos de cada tipo de evento?
  - Cambios de estado
  - Cambios de atributos de entidades
- ¿Cómo se definen las actividades?
  - Deterministas, probabilísticas, ecuaciones
  - Qué tipo de evento marca su pricipio/final
  - Su comienzo es condicional al estado
- ¿Cómo comienza la simulación?
  - Primeros eventos
- ¿Cuándo finaliza la simulación?

Avance de la simulación

- *Snapshots* del sistema con el tiempo
- *Snapshot* incluye el estado del sistema y la FEL
- Esa FEL contiene las actividades en progreso y cuándo finalizan
- CLOCK = t = instante actual en la simulación
- Evento en t_0 = Evento inminente
- Se actualiza CLOCK = t_0
- Se retira el evento inminente de la FEL
- Se "ejecuta" el evento
- Eso crea un nuevo *snapshot* del sistema



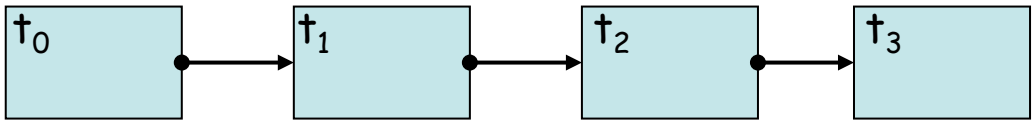
$CLOCK = t < t_0$

$t_0 \leq t_1 \leq t_2 \leq t_3$

Event-scheduling/Time-advance

Mientras queden eventos en la FEL

- Retirar el primero (evento inminente)
- Avanzar la variable de CLOCK hasta el instante del evento
- Procesar el evento: puede modificar el estado del sistema e introducir otros eventos futuros en la FEL manteniéndola ordenada
- Actualizar los contadores y estadísticos



¿ Fin de la simulación ?

- Cuando no queden eventos en la FEL
- En la inicialización introducir un evento futuro de finalización
  - Limita el tiempo simulado
  - No limita el tiempo real
- Detenerla al alcanzar una duración (tiempo real)
- Detenerla al alcanzar unas medidas una cierta precisión.

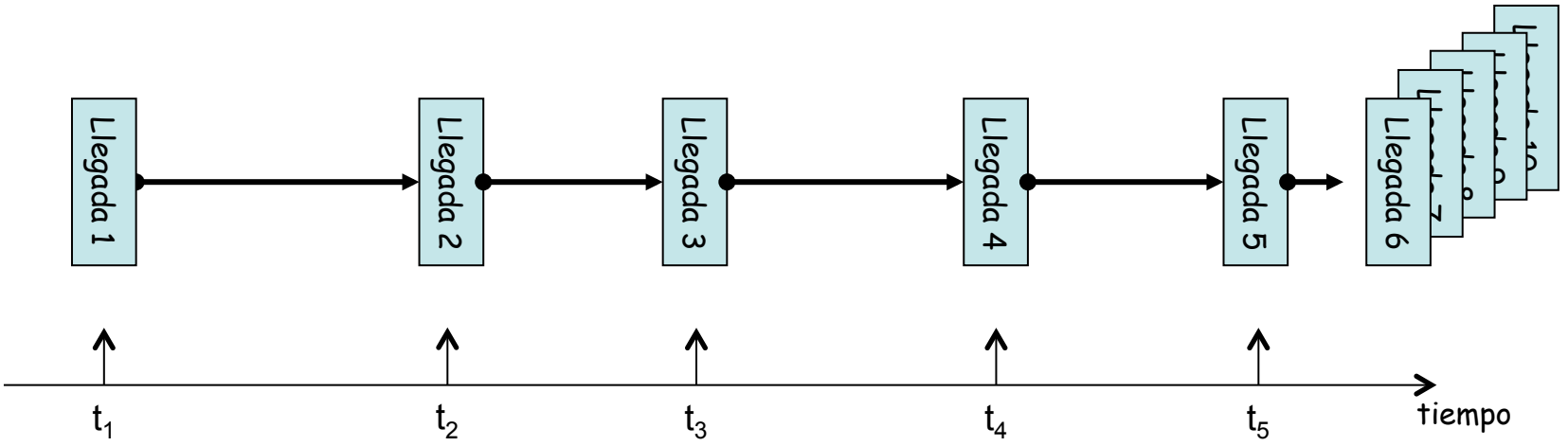
Gestión de la FEL

- Su longitud cambia durante toda la simulación
- Su gestión eficiente es vital
- Operaciones más frecuentes:
  - Retirar el primero
  - Insertar manteniendo el orden
- Puede soportar el eliminar un evento en concreto

Ejemplo de simulación con FEL

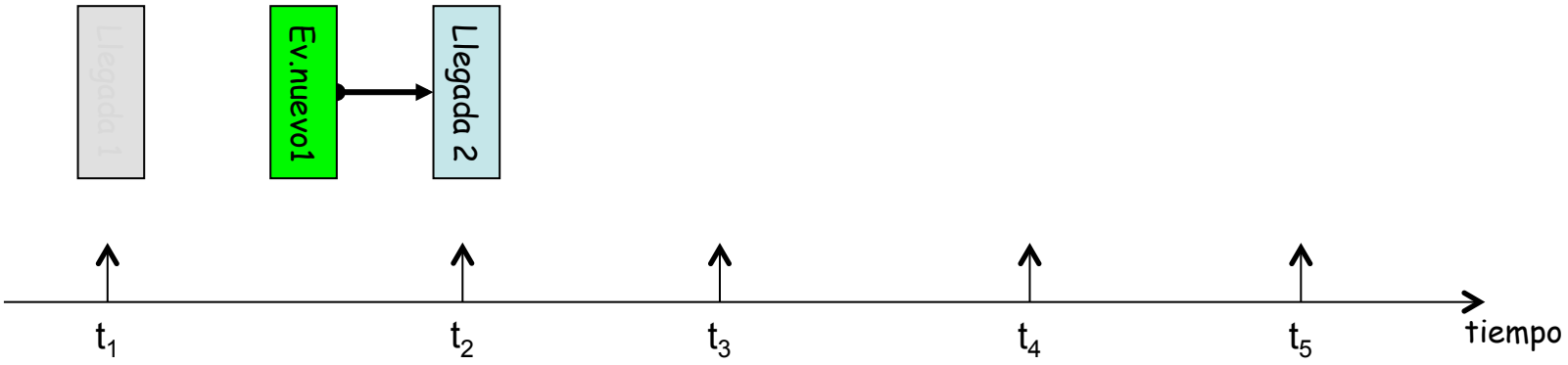
Ejemplo

- Iniciamos la FEL con todos los eventos de llegadas que se vayan a producir
- Esto es muy ineficiente
- Es común que trabajemos con millones, decenas de millones o centenares de millones de llegadas
- Un gran gasto de memoria
- ¿Mejor aproximación al problema? (...)



Ejemplo: Avance con llegadas

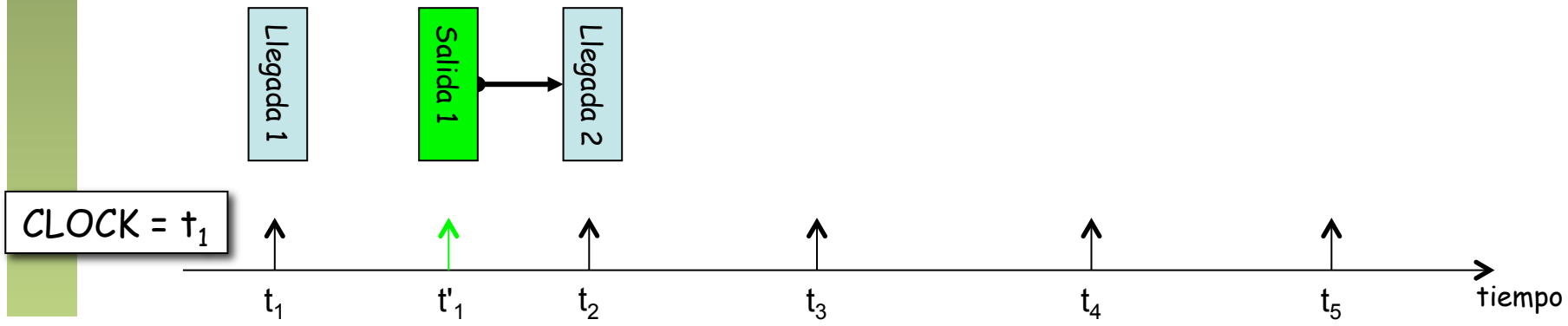
- Evento inicial: una llegada
- Se procesa
  - Se elimina de la lista
  - Se introducen eventos consecuencia de ella
  - Se introduce un nuevo evento que es la siguiente llegada donde le corresponda (…)



Ejemplo: Cajero automático

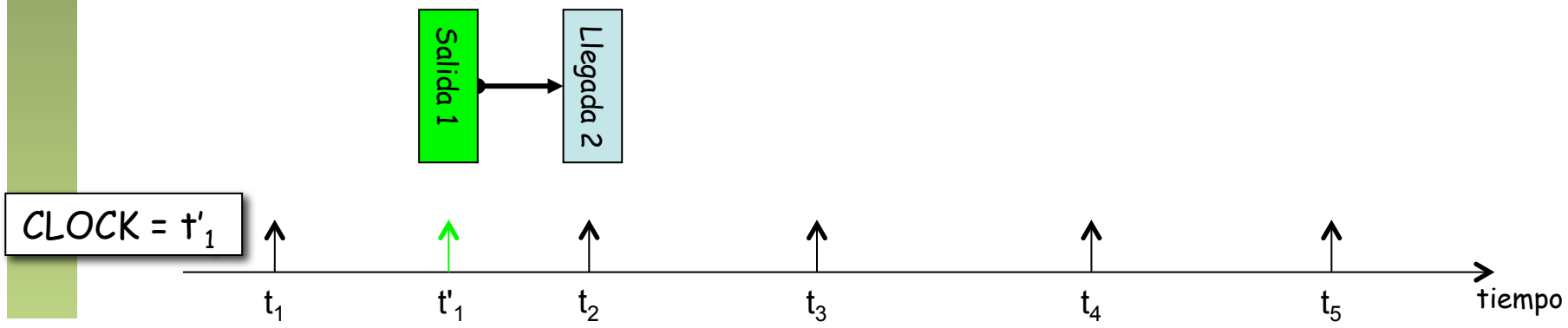
Ejemplo: Cajero automático

- Evento único en la FEL: llegada del primer cliente (t_1)
- Se procesa el evento inminente:
  - Se avanza el reloj al instante de tiempo de este evento
  - El evento es el de esa primera llegada
  - Se elimina (…)
  - Introduce eventos consecuencia de ella: evento de cuándo termina de usar el cajero (t'_1)
  - Se introduce un nuevo evento que es la siguiente llegada (t_2) (…)
  - Se actualizan estadísticas (1 cliente en el sistema)



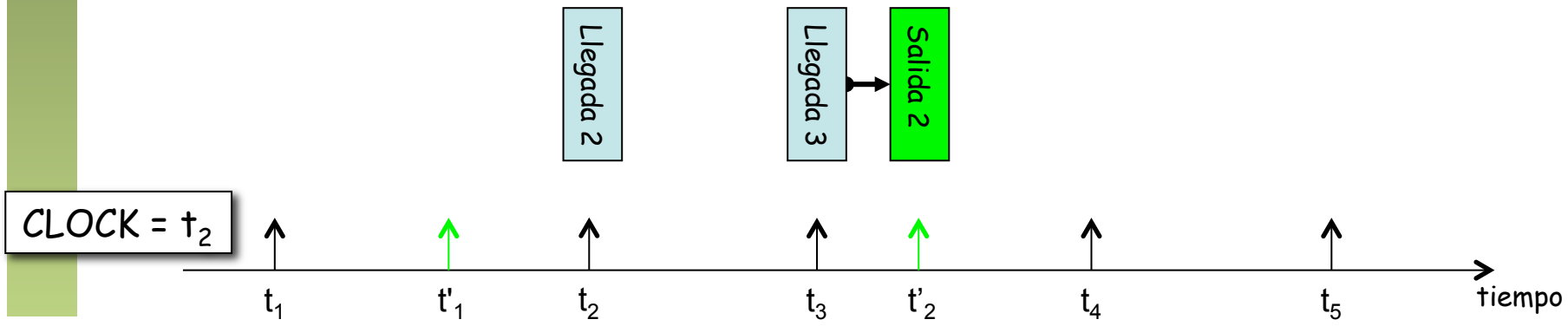
Ejemplo: Cajero automático

- Se procesa el evento inminente:
  - Se avanza el reloj al instante de tiempo de este evento
  - El evento es la salida del primer cliente
  - Se elimina (…)
  - Se actualizan estadísticas (0 clientes en el sistema)
  - En este caso suponemos que la salida no genera más eventos



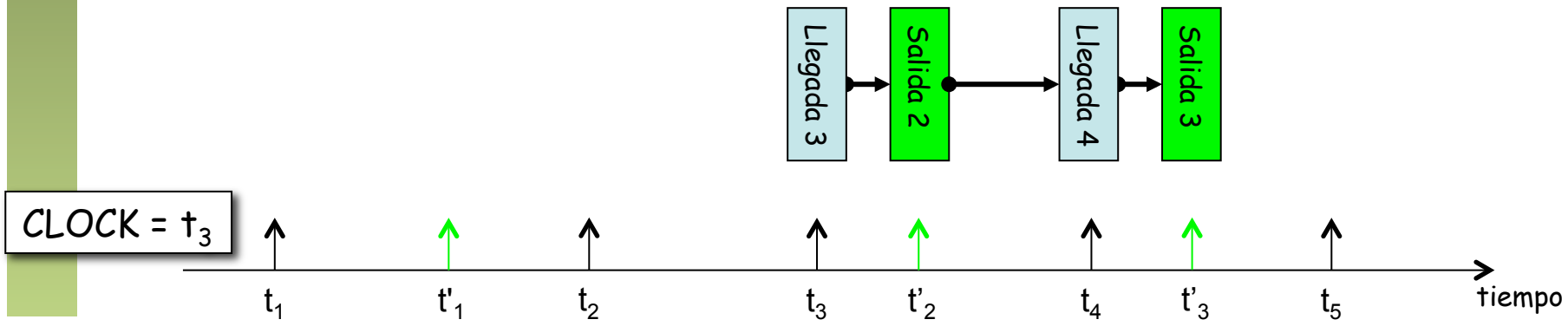
Ejemplo: Cajero automático

- Se procesa el evento inminente:
  - Se avanza el reloj al instante de tiempo de este evento
  - El evento es el de la segunda llegada
  - Se elimina (…)
  - Introduce eventos consecuencia de ella: evento de cuándo termina de usar el cajero (t'_2)
  - Se introduce un nuevo evento que es la siguiente llegada (t_3)
  - Supongamos que la siguiente llegada se producirá antes de que salga este (…)
  - Se actualizan estadísticas (1 cliente en el sistema)



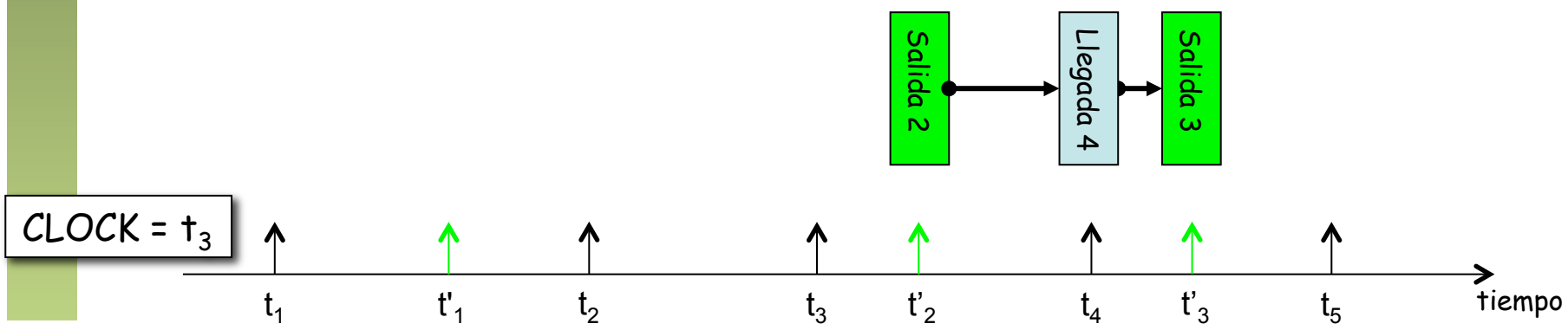
Ejemplo: Cajero automático

- Se procesa el evento inminente:
  - Se avanza el reloj al instante de tiempo de este evento
  - El evento es el de la tercera llegada
  - Se elimina (…)
  - Introduce eventos consecuencia de ella: evento de cuándo termina de usar el cajero (t'_3)
  - Se introduce un nuevo evento que es la siguiente llegada (t_4)
  - Supongamos que la siguiente llegada se producirá entre la segunda y tercera salida (…)
  - Se actualizan estadísticas (2 clientes en el sistema)



Ejemplo: Cajero automático

- Se procesa el evento inminente:
  - Se avanza el reloj al instante de tiempo de este evento
  - El evento es la salida del segundo cliente
  - Se elimina (…)
  - Se actualizan estadísticas (1 clientes en el sistema)
  - Etc.



Simulation Tools

- Librerías de utilidades
- Simuladores programables
- Simuladores controlables (gráfico, script)
- Simuladores de redes (ns2, OMNeT++, SSFNet, Parsec, Qualnet, OPNET, JiST/SWANS …)