

Simulación

Proyecto: Modelado y Simulación de Sistemas Dinámicos con el Formalismo DEVS

Año: 2017

Descripción de Proyecto

Considere dos jugadores $player_{pc}$ y $player_{human}$ que ingresan objetos por el inicio y final de una recta de longitud l respectivamente. El objeto que ingresa ya sea por la derecha o izquierda, trae como información su peso, y el mismo se desplaza intentando llegar al otro extremo.

Puede suceder que en un determinado momento 2 objetos viajen sobre la recta en sentido opuesto, con lo cual se producirá una colisión. El objeto que continua sobre la recta (objeto *vencedor*) es aquel cuya distancia recorrida multiplicada por su peso es mayor. El objeto *vencedor* continua pero con un peso disminuido según se detalla en la sección de consideraciones del proyecto. El objeto derrotado desaparece de la recta y en caso de empate desaparecen los dos.

Los objetos viajan a una velocidad constante V_c .

El sistema debe sacar una salida X cada vez que ocurre una colisión, donde que incluye la información sobre el objeto vencedor (objeto vencedor, peso con que continua, punto de colisión).

Al llegar un objeto al extremo izquierdo de la recta el sistema debe sacar una salida indicando $Llego_{der}$ (si llegó el objeto que ingresó por la derecha), o $Llego_{izq}$ en caso contrario.

Pueden haber sobre la recta varios objetos viajando en el mismo sentido.

El jugador $player_{pc}$ genera automáticamente el ingreso de los objetos con un interarribo exponencial de 10 unidades de tiempo y los pesos de cada objeto siguen una distribución uniforme entre 5 y 10.

El jugador $player_{human}$ prueba con distintas estrategias con el fin de que la cantidad de objetos que llegan al otro extremo sea superior a los que llegan a su extremo (derecha). En cada estrategia, previo a las corridas, genera los datos de los objetos que van a ingresar por la derecha.

Estrategias de $player_{human}$

1. Genera los interarribos con distribución exponencial con media 10. Genera los pesos con distribución exponencial con media 7,5. Ordena de menor peso a mayor peso los objetos y los envía a la cinta de acuerdo a los interarribos generados.
2. Idem al caso anterior, pero ahora los ordena de mayor a menor.
3. Idem al caso anterior, pero el ordenamiento por peso es primero el de mayor peso, luego el de menor peso, luego el siguiente de mayor peso, luego el siguiente de menor peso y así sucesivamente.
4. Solo genera los pesos con distribución exponencial con media 7,5. Cada vez que ocurre una colisión y “pierde su objeto” pone sobre la cinta el objeto de menor peso pero que supere al objeto del jugador $player_{pc}$ vencedor. En caso de empate ingresa cualquier objeto restante. Al inicio ingresa a la cinta aleatoriamente cualquiera de los objetos.

5. Solo genera los pesos con distribución exponencial con media 7,5. Cada vez que ocurre una colisión y “pierde su objeto” pone sobre la cinta el objeto de mayor de los que les quedan. En caso de empate ingresa cualquier objeto restante. Al inicio ingresa a la cinta aleatoriamente cualquiera de los objetos.
6. Analice alguna estrategia mejor que las anteriores y presente los resultados.

Objetivos

Se busca analizar la distintas estrategias a fin de evaluar la mejor de ellas para el jugador $player_{human}$.

- Especifique el comportamiento del juego bajo las distintas estrategias utilizando el formalismo DEVS.
- Implementar la especificación del juego en PowerDevs. Diseñe los módulos lo más parametrizado posible, de tal manera que sea sencillo modificar el juego, por ejemplo, utilizar otras estrategias.
- Utilice GNUPLOT para crear los gráficos.

Consideraciones:

- Graficar las colisiones (elegir el tipo de gráfico mas representativo) y las secuencias de llegadas de ambos lados.
- Generar para cada corrida 100 objetos de cada lado.
- La velocidad de los objetos es $V_c = 1$.
- Al inicio se comienza con un objeto de cada lado.
- La longitud de la cinta es: $l = 25$
- En caso de colisión el objeto vencedor continúa con un peso disminuido dado por la siguiente formula:

$$ObjVenc_{peso_nuevo} = ObjVenc_{peso_actual} \cdot \frac{ObjPerd_{peso_actual} \cdot ObjPerd_{dist_actual}}{ObjVenc_{peso_actual} \cdot ObjVenc_{dist_actual}} \quad (1)$$