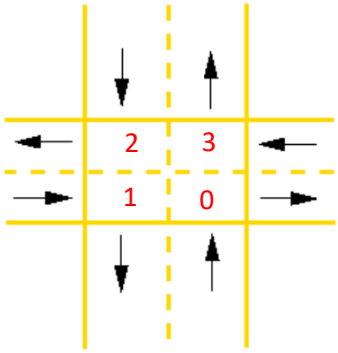
López Soto Miguel Ángel Sistemas Operativos

Intersección de caminos

Lenguaje de Programación: **Python 3.7**

Para la solución de este problema se consideró en primera instancia dividir la intersección en 4 cuadrantes:



Y dando la información sobre con cual semáforo se va a topar primero como un vector con la referencia del plano R2 Euclideo.

Para saber qué sección va a cruzar, propuse una fórmula que mapea las 4 posibles direcciones del vector con el número de sección.

Dado un vector u=(x,y) , Sección= (x-y(1-y)) % 4

El módulo sirve para mantener la cerradura en las secciones y asi mapear todos los vectores

(1,0) = 1

(-1,0) = -1 %4 = 3

(0,1) = 0

(0,-1) = 2

Asi en un arreglo de semáforos (mutex) podemos seleccionar el que vamos a bloquear para que pueda pasar el hilo y después de utilizarlo, lo liberamos.

Tomando en cuenta las refinaciones modelamos otras condiciones.

Para modelar el camino, se hace la siguiente operación:

(Posición actual – 1) % 4

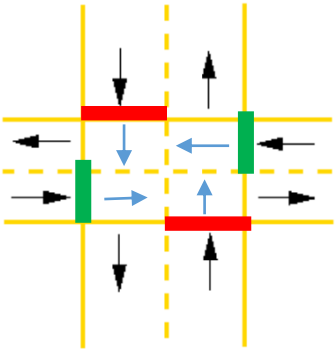
Asi podemos ir por ejemplo de 0 a 3 ((0-1)%4) y asi “ir derecho” o simplemente girar a la derecha.

Para modelar el giro a la derecha simplemente hacemos la operación anterior 1 veces. Y para hacer el giro a la izquierda hacemos la operación 3 veces.

Ahora, una opción fue poner una barrera en cada dirección para asi controlar el transito de los autos, el problema es que se necesitan n threads para que pueda avanzar y si hay threads que no llegan a ese número, nunca podrán avanzar por lo que sería una “barrera” pero controlada por tiempo asi como los semáforos en la vida real. Esto resuelve esa cuestión pero agrega que haya threads en espera aunque en los carriles perpendiculares no haya autos. Otra cuestión a resaltar es el giro a la izquierda ya que tiene que

bloquear una sección por donde esta pasando el carril contrario por lo que retrasa un poco el avance del carril contrario de donde nace el auto.

Ahora tocando el tema de los bloqueos mutuos hay una situación particular. Si después de que el semáforo cambio a rojo, aun hay autos en la intersección, si 2 autos se quedan en una sección y los autos a los que ya tienen “el verde” bloquean las secciones siguientes a los threads que se quedaron después del rojo y nadie puede avanzar.



Una solución a esto es añadir un tiempo de espera muy corto para que los threads que están después de que llego el rojo, logren terminar su recorrido. Esto puede resultar, pero no hay certeza de cuantos threads hay en la intersección. Otra opción es añadir un torniquete, aunque no estoy muy seguro.

Con esta solución aseguramos que si hay un auto siempre va a poder cruzar pero esto nos cuesta tiempo.

Aunque si los autos siguen derecho o van a la izquierda no hay tanto problema ya que avanzan muchos en una iteración.