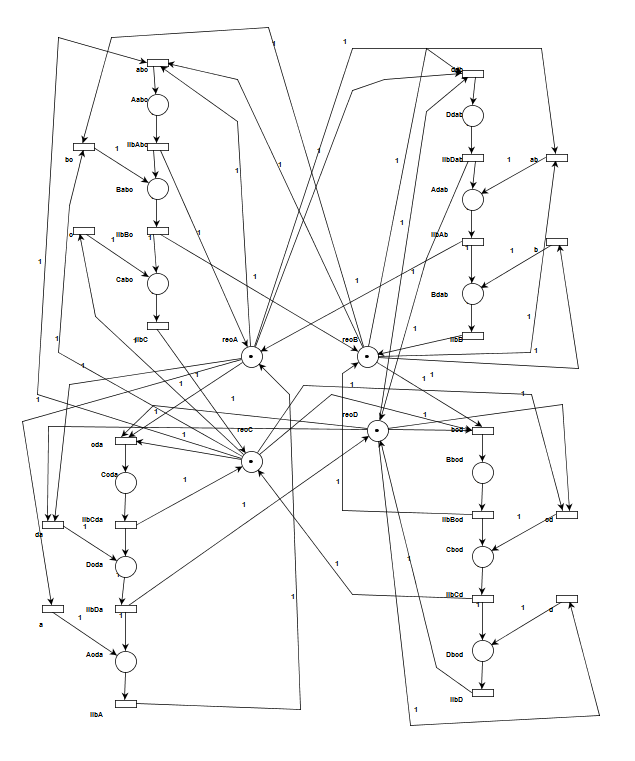
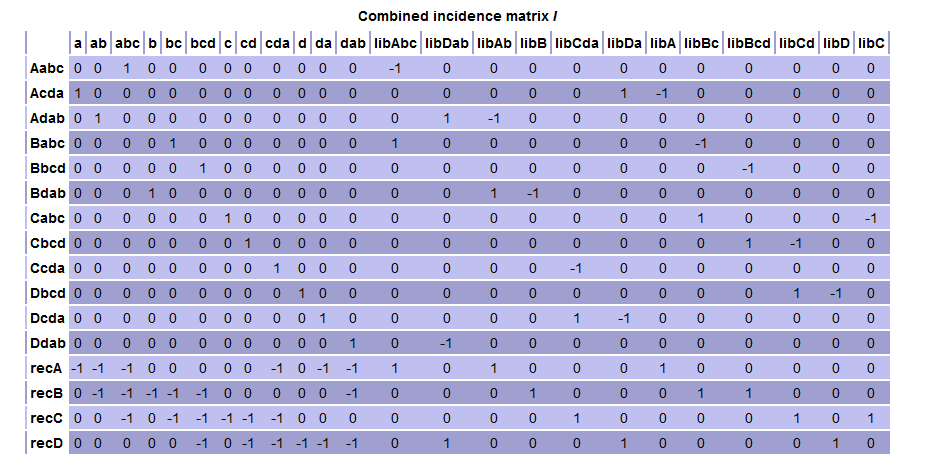
Analisis de red de Petri



Esta es la red de Petri desarrollada para resolver la situación del cruce de la calle para los autos. Se pueden apreciar las plazas con los recursos al centro (recursos de esquina A, B, C y D).



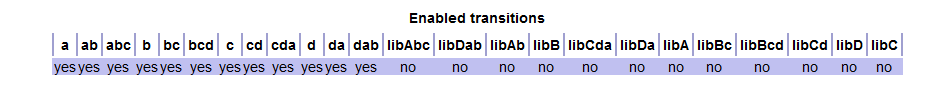
Esta es la matriz de incidencia combinada de la red. Las plazas se encuentran representadas por las filas. Las que se llaman recA, recB, etc. Representan, si tienen token, que el recurso esta disponible. El resto de las plazas representan en mayúscula que recurso esta utilizando, y en minúscula que secuencia/s. Por ejemplo: si hay un token en Aabc implica que se esta usando el recurso A para la secuencia abc. Si hay un token en D bcd implica que se esta usando el recurso d en la secuencia bcd, cd o d solamente.

Para los disparos, utilizamos la siguiente nomenclatura:

Por un lado los que tienen nombres en minúsculas representan que recursos se solicitan: bcd usa y necesita los recursos b, c y d (es un giro a la izquierda).

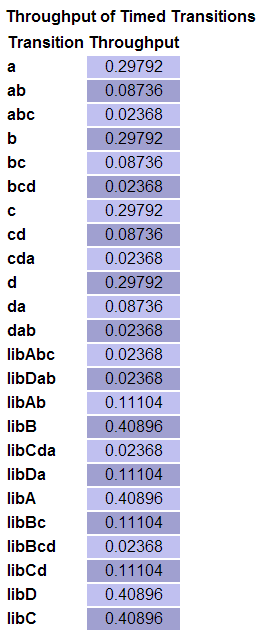
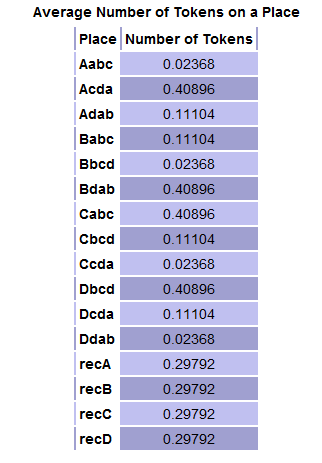
Por otro lado los que empiezan con lib representa un disparo donde se libera el recurso indicado en mayúscula, pero todavía se retienen los indicados en minúsculas: libDa de ejecutarse esta transición implica que se libera el recurso D pero todavía se retiene el A

A continuación vemos que transiciones están habilitadas al inicio:



Podemos ver que se permite el movimiento de todos los posibles caminos al inicio, al estar disponibles los cuatro recursos.

A continuación hacemos un análisis de las probabilidades de encontrar un token en una plaza o de que se ejecute una transición (asignando mismas probabilidades a todas las transiciones)



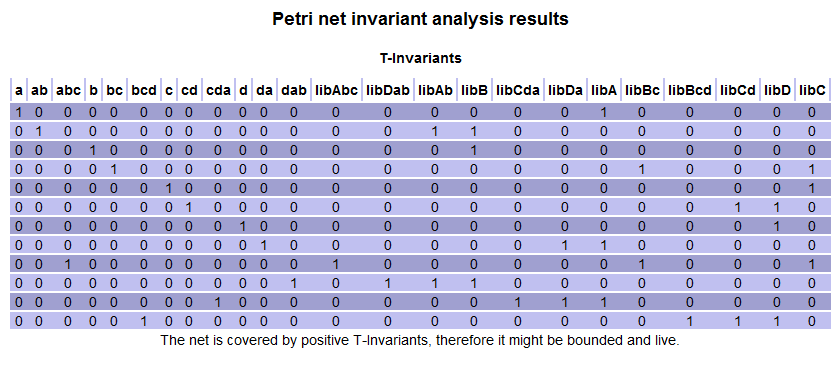
Estas dos tablas nos permiten analizar varias cosas:

Por un lado, al no haber plazas que no tengan probabilidad de ser ocupadas,ni transiciones que no tengan probabilidad de ser ejecutadas, nos muestra que no hay interbloqueo, y que todas las plazas pueden llegar a ser tomadas de alguna forma.

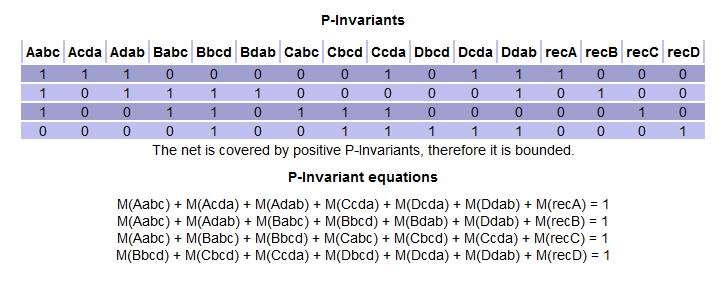
Ademas podemos ver tanto en las plazas como en las transiciones, que todos los autos que quieran tomar la misma cantidad de recursos, tienen la misma probabilidad de que puedan tomarlos: por ejemplo, la probabilidad de que ocurra la transición a, b, c o d es la misma (es decir que los que quieran tomar un solo recurso tienen la misma probabilidad. Lo mismo sucede por ejemplo con ab, bc, cd y da. Esto nos muestra que la red es simetrica en cuanto a cada recurso.

Tambien vemos que hay mucha mas probabilidad de que se pueda ejecutar una transición que pida un recurso que una que pida dos o que pida tres. Esto tiene mucho sentido, ya que es mas fácil que se encuentre un solo recurso disponible a que haya mas. Esto lo vamos a tener que tener en cuenta a la hora de asignar prioridades en nuestro monitor

Ahora haremos un análisis de los invariantes de nuestra red:



Esta tabla nos muestra cada conjunto de transiciones que al ejecutarse nos lleven a un mismo estado de marcación. Podemos ver que son 12 los t-invariantes lo que nos muestra los 12 posibles caminos que se pueden tomar en una esquina, es decir, el conjunto de todos los movimientos posibles de un auto en una esquina.



Esto nos muestra que de los miembros de cada ecuación de los p-invariantes solo uno puede tener el token en determinado momento, con esto podemos determinar que no se multiplican los tokens (garantizamos que el recurso sea tomado una sola vez), ni se destruyen (no desaparece el recurso); naturalmente tenemos una ecuación por cada posible recurso (A, B, C o D)

Políticas

En la planificación del monitor, a través de la red de Petri, hemos decido implementar el uso de una política para solventar una posible inanición y manejo de acumulación en las colas de acción.

**Manejo de acumulación en colas**

Por lo observado de la probabilidad de disparar determinadas transiciones, tenemos que la probabilidad disminuye a medida que necesitamos más recursos simultáneos para esa transición. Esto quiere decir que al ser más probable la que necesiten menos recursos, se van a disparar con mayor frecuencia produciendo una acumulación menor en estas (se considera que la distribución de entrada en las colas de acción es aleatoria con distribución probabilística normal).

Solución:

Es por esto que decidimos que en frente al caso, en donde se encuentren recursos, para poder disparar acciones de diferentes cantidad de tokens (teniendo en cuenta solo las acciones mutuamente excluyentes entre ellas), se le dará prioridad a la acción que necesite más recursos. Estableciendo así, tres niveles de prioridades, uno cada tipo de acción: (acción de 1 token, 2 tokens, y 3 tokens).

|  |  |  |
| --- | --- | --- |
| Cantidad de tokens necesarios por acción | Acciones | Prioridad |
| 1 token | A, b, c, d | 1 |
| 2 tokens | Ab, bc, cd, da | 2 |
| 3 tokens | Abc, bcd, cda, dab | 3 |

**A mayor número de prioridad la prioridad es mayor.**

Para un mejor entendimiento se da el siguiente ejemplo:

Dado un momento determinado, un auto ingresa al monitor y devuelve el token A y el token B siendo que no hay ningún otro recurso disponible. En este caso se podrían disparar las acciones:

* A
* B
* AB

Frente a esta situación la acción con prioridad a ejecutarse será la acción AB, ya que posee una prioridad de 2, frente a la prioridad 1 de las otras dos acciones.

**Inanición:**

La inanición se puede dar frente a la situación en la cual lleguen continuamente autos que deseen utilizar acciones de menor cantidad de recursos y nunca se pueda juntar la combinación de recursos necesarios para una acción que necesitan mayor cantidad de tokens, ya que cuando está disponible uno, otro de los que necesita está siendo ocupado. La política de prioridad presentada anteriormente no resuelve este problema ya que le uso de esa prioridad se da en el caso que se encuentren la posibilidad de ejecutar en ese momento acciones mutuamente excluyentes entre sí.

Ejemplo:

Llegan continuamente autos para realizar la acción a, y acción b. Debido a esto se produce inanición para efectuar las acciones ab, abc ya que nunca consigo la combinación de estos recursos.

Solución:

Definimos que un hilo cae en una posible inanición cuando DESDE su ingreso al monitor, ya han entrado K cantidad de hilos luego de él, y todavía no ha conseguido los recursos que necesita para ejecutar la acción deseada.

Una vez que un hilo entra en la categoría de posible inanición, se producen las siguientes modificaciones al flujo normal del monitor:

Cuando un hilo ingresa con la intención de solicitar recursos, verifica si el hilo más viejo en estado de posible inanición requiere alguno de los recursos que esté está solicitando, si es así, no retira ningún recurso y se ubica en la cola de acción correspondiente.

Cuando un hilo ingresa para devolver recursos, verifica cuales son todas las acciones posibles con los recursos disponibles previamente, mas lo que él devuelve, si hay inactivos en ellas, se despierta al mas antiguo, si no hay, continua con el flujo normal.