]

FONDO MONETARIO INTERNACIONAL SOFISTICADO

PROGRAMACIÓN CONCURRENTE

Redes de Petri

TRABAJO PRÁCTICO N°2

Autores:

* Daniele, Francisco || francisco.daniele@mi.unc.edu.ar
* Endrek, Marco || mendrek@mi.unc.edu.ar
* Martinez, Brenda Sofía || sofia.martinez@mi.unc.edu.ar
* Palacios Lucero, Matías Iván || ivan.palacios@mi.unc.edu.ar

[**Enunciado**](#_heading=h.kbk4p6iwe8wo) **2**

[**Trabajo Realizado**](#_heading=h.qu4pkgj8aqd3) **3**

[**Decisiones**](#_heading=h.1g506yd5mm8r) **6**

[**Resultados**](#_heading=h.bi5se72ptjqw) **7**

[**Sequence Diagram**](#_heading=h.cslv0g9r4zyz) **8**

[**Class Diagram**](#_heading=h.o916zrbgp4yv) **9**

# 

# Enunciado

## 

# Trabajo Realizado

# La primera parte de este trabajo consiste en analizar la siguiente Red de Petri propuesta, justificando sus propiedades, determinando sus invariantes y realizando una breve descripción de lo que representa. Además debemos, si es que lo tiene, eliminar el deadlock tratando de obtener el mayor paralelismo posible en la red.

# 

# 

# 

# 

# 

# 

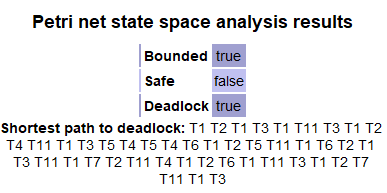
# Analizandola con el software PIPE llegamos a lo siguiente:

# **Propiedades**

# La RdP no está libre de deadlock. Al tener deadlock, la RdP no es viva.

# La RdP está limitada ya que está cubierta por invariantes de plaza positivos, nunca va a existir un lugar que posea más de 3 tokens.

# La RdP no es segura, nos damos cuenta viendo la marca inicial m0 dónde ya se observan lugares con más de 1 token.



# 

# 



# **Invariantes**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **Clasificación**

# 

# 

# 

# 

# **Conclusiones**

# Según el análisis realizado anteriormente podemos realizar la siguiente descripción sobre la RdP:

* Las plazas {P3, P4, P7, P10, P14} representan recursos compartidos en el sistema. Serían por ejemplo, máquinas de una fábrica que se pueden utilizar para un proceso a la vez. Estas limitan la cantidad de hilos que pueden estar realizando las tareas que representan las plazas que están encerradas entre las transiciones que quitan y ponen tokens a {P3, P4, P7, P10, P14}.
* Las plazas {P1, P12} son plazas idle, que corresponden a buffers del sistema. Cuando un token (hilo) está allí, se encuentra esperando poder disparar la primera transición de un invariante para comenzar a realizar tareas.
* El resto de las plazas son plazas donde se realizan actividades relacionadas con el proceso.
* Analizando los invariantes de plaza (fig. 3):
  + El primero determina que entre la plaza idle (1) y {P2, P5, P6, P9, P13} que son las que representan la realización de procesos que harán los hilos de P1 siempre habrá 3 hilos.
  + El segundo nos dice que, cómo P10 representa un único recurso compartido, las tareas que representan P9 y P11 se realizan en exclusión mutua.
  + El tercero nos dice que como hay 3 tokens en la plaza idle P12, entre esta y {P8, P11, P15} donde los hilos realizan tareas siempre va a haber esos 3 tokens.
  + El cuarto y el último son similares, nos indican que se poseen 2 tokens de recurso compartido por lo que las únicas opciones son: o qué ningún recurso esté siendo utilizado, que uno sólo sea utilizado por una plaza de alguno de los dos caminos, o que los 2 recursos sean utilizados por la plaza de un camino o uno usado por un camino y el otro por el restante.
  + El quinto determina que sólo habrá un hilo a la vez “decidiendo” qué conjunto de tareas realizará. Si las del invariante T1T2T11 o del invariante T1T3T4T5T6.
  + El sexto determina que al P4 representar un recurso, sólo podrá haber 1 hilo realizando la tarea de P5 a la vez.
* Observamos que tiene 3 invariantes de transición (secuencias de disparo que te llevan al estado inicial) que representan una secuencia de tareas que realizaría el sistema que modela la Red.
  + Invariante T1T2T11: este no conlleva problemas ya que no comparte recursos con otro invariante del sistema.
  + Invariantes T1T3T4T5T6 y T7T8T9T10: estos invariantes comparten recursos y realizan una tarea en exclusión mutua. Aquí observamos que mientras haya tokens en uno sólo de los caminos, no hay problema. Pero cuándo en las plazas 6 (o 15) hay 2 tokens y en la 11 (o 9) hay uno realizando la tarea, se interbloquean estos 2 invariantes y no se pueden disparar más. Y a partir de ahí, dependiendo del interleaving, puede quedarse disparando infinitamente el invariante T1T2T11 o, si todos los tokens de P1 pasaron a estar en las plazas 6, 9 o 13 se llega al deadlock.

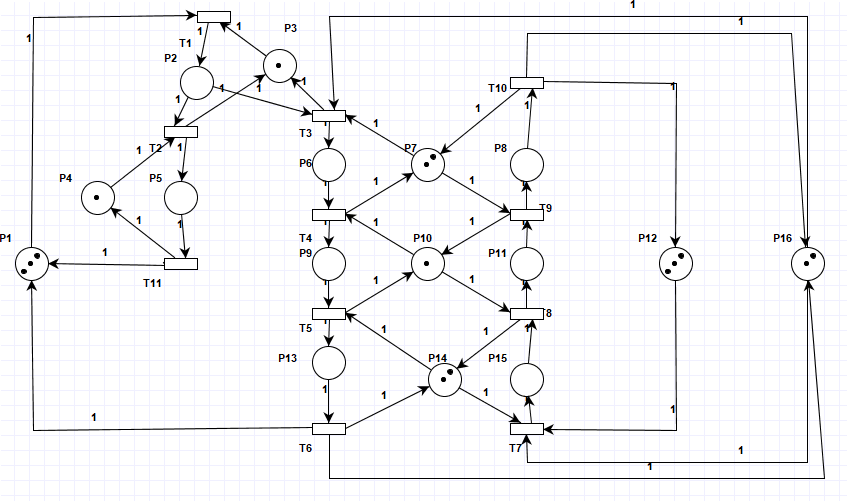
**Eliminación del deadlock**

Para poder seguir con el trabajo había que averiguar el origen del deadlock y encontrar una manera eficiente de eliminarlo. Para esto fuimos probando diferentes cosas, siempre agregando plazas de control en distintos puntos para mantener lo más simple y entendible para nuestro criterio (evitamos uso de brazos inhibidores para no agregar elementos y complejidad a la red), empezando muy restrictivamente hasta poder llegar a una RdP con el mayor paralelismo que pudimos.

Mención especial a la primera “solución” que propusimos por no haber entendido correctamente la consigna: directamente eliminamos dos recursos compartidos lo cual no cumplía con los requerimientos del trabajo.

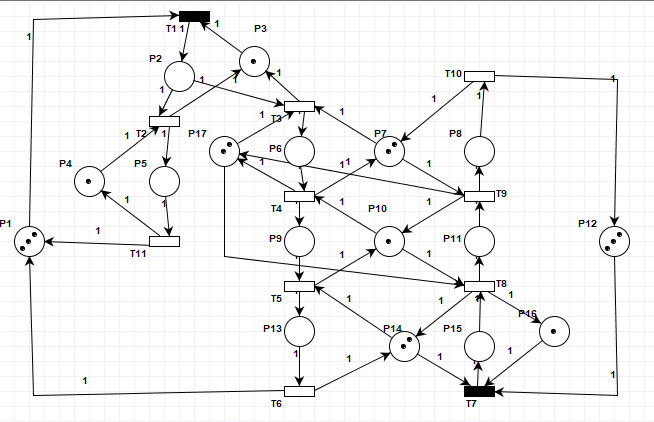
A continuación describimos cómo se fue llegando a la red final:

* Lo primero que hicimos fue agregar P16 con 2 tokens limitando así que entre las plazas que encierran los invariantes conflictivos solo pueda haber 2 hilos trabajando. Esto, si bien soluciona el deadlock, no es eficiente ya que secuencializa mucho el trabajo.





Luego de seguir probando, llegamos a las siguientes 2 redes similares, las cuales no limitan la cantidad de hilos trabajando en {P6, P9, P13, P8, P11, P15} (están sólo limitados por los recursos iniciales, tal cómo la red inicial de la consigna). Tienen una ligera diferencia, ya que una limita a los lugares de “entrada” P6 y P15 a un hilo a la vez y la otra sólo hace esto con P15.



# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **Elección de la red**

# Luego realizamos un análisis mediante PIPE comparando estas dos redes para determinar cuál permitía más paralelismo y en base a esto tomar la decisión de con qué red continuar el trabajo:

|  | **Red fig. 7** | | **Red fig. 6** | |
| --- | --- | --- | --- | --- |
| **Marcados** | 54 | | 58 | |
| **Promedio de tokens por plaza de interés según los diferentes marcados** | | | | |
| **P1** | 0,41 | | 0,26 | |
| **P12** | 1,29 | | 1,52 | |
| **P2** | 0,8423 | | 0,6381 | |
| **P5** | 0,4902 | | 0,4109 | |
| **P6** | 0,7394 | | 1,0142 | |
| **P8** | 0,3064 | | 0,2074 | |
| **P9** | 0,2918 | | 0,3905 | |
| **P11** | 0,4101 | | 0,2776 | |
| **P13** | 0,2265 | | 0,2845 | |
| **P15** | 0,9959 | | 0,9973 | |
| **Suma plazas IDLE** | | 1,7 |  | 1,78 |
| **Suma plazas de tareas** | | 4,3026 |  | 4,2205 |

# A partir de la tabla anterior concluimos que la Red de Petri de la figura 7 es ligeramente “mejor” ya que aporta mayor paralelismo que la otra. Esto se observa al realizar el análisis del promedio de tokens por plaza por marcado, observamos que la suma de tokens promedio en las plazas idle (dónde no “trabajan”) es menor en dicha red, lo que implica (y se observa en la tabla) que la de las plazas no marcadas que no son recursos, y por lo tanto representan tareas, es mayor.

# En base a este análisis, decidimos elegir la Red de Petri de la figura 7 para realizar la siguiente parte del trabajo.

A continuación, a partir del análisis de dicha red mediante PIPE, observamos lo siguiente:

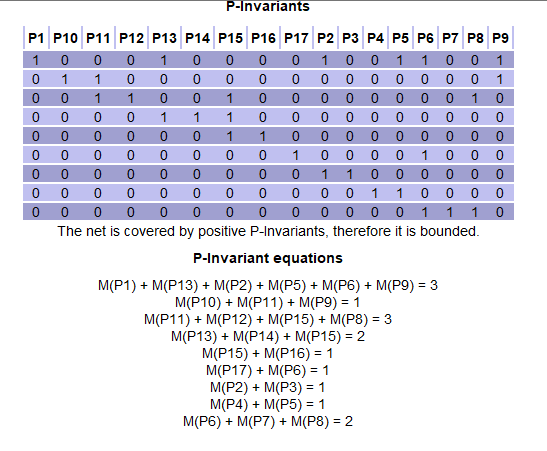
**Propiedades**

# La RdP ahora está libre de deadlock y por lo tanto es viva.

# La RdP sigue siendo limitada ya que está cubierta por invariantes de plaza positivos, nunca va a existir un lugar que posea más de 3 tokens, por esto también se mantiene no segura, con lugares con más de 1 token.

**Invariantes**

Como se pide en la consigna, esta nueva red mantiene sus invariantes de transición.



# **Clasificación**

La clasificación de esta RdP sigue siendo la misma que la anterior, observada en fig. 4.

# **Conclusiones**

# Como hicimos con la red original, a partir del análisis realizado por software de la nueva RdP, procedemos a una breve descripción de la misma:

* Las plazas {P3, P4, P7, P10, P14} siguen representando recursos compartidos en el sistema y limitan la cantidad de hilos que pueden estar realizando las tareas que representan las plazas que utilizarían los tokens de estas plazas.
* Las plazas {P1, P12} son plazas idle, que corresponden a buffers del sistema.
* Las plazas {P16, P17} son plazas de control, utilizadas para eliminar el deadlock original, las cuáles limitan a 1 los tokens en las plazas {P6, P15}.
* El resto de las plazas son plazas donde se realizan actividades relacionadas con el proceso.
* Analizando los invariantes de plaza (fig. 8) vemos que se mantienen los mismos 7 invariantes de plaza de la red original y además aparecen 2 nuevos:
  + M(P16) + M(P15) = 1. Determina que en P15 sólo podrá haber un hilo realizando la tarea.
  + M(P17) + M(P6) = 1. En P6 sólo podrá haber un hilo realizando la tarea.
* Se mantienen los mismos 3 invariantes de transición, pero esta vez entre ellos no hay problemas de concurrencia que originen un deadlock.

# 

# **Criterio de hilos**

En la figura 9 se determinan los hilos necesarios para la ejecución del sistema con el mayor paralelismo posible. Las flechas indican hilos. A continuación se indica el por qué de la decisión:

* Hilo amarillo: P3 me limita el disparo de T1 de a un hilo por vez, por lo que sólo necesitaría uno que se encargue del disparo de esa transición. Cómo en P2 hay un conflicto, debemos dejar la tarea de ejecutar las siguientes transiciones a otros hilos.
* Hilo naranja: P4 limita a 1 la cantidad de hilos que pueden disparar T2 y T11, por lo que habrá uno solo encargado de realizar la tarea de P5 y disparar dichas transiciones.
* Hilos rojos: la red puede tener hasta 3 hilos realizando simultáneamente tareas en P6, P9 y P13 por lo que encargamos a 3 hilos distintos la tarea de disparar las transiciones T3, T4, T5 y T6.
* Hilos azules: es el mismo caso que el anterior, puede haber hasta 3 hilos realizando tareas en P8, P11 y P15 por lo que encargamos a 3 hilos distintos el disparo de T7, T8, T9 y T10.

# 

# Políticas

# 

# Resultados

# 

# Sequence Diagram

# Class Diagram