INSTITUTO TECNOLÓGICO DE IZTAPALAPA

Propuesta para el desarrollo del proyecto del titulado:

Simulación de Maquina de Turing en Petri Net

Presenta:

Ávila Gómez Giovanni Arturo

García Pacheco Axel David

Rojas Pérez José Ramón

No. De Control:

181080041

181080026

181080011

Asesor Interno:

Abiel Tomas Parra Hernández

CIUDAD DE MÉXICO **11 / 2020**

CRONOGRAMA DE PROYECTO

ACTIVIDAD \ FECHA	SEMANA 10	SEMANA 11	SEMANA12	SEMANA13	SEMANA14	SEMANA 15
	30 de Noviembre 2020	7 de Diciembre 2020	14 de Diciembre 2020	21 de Diciembre 2020	11 de Enero 2021	18 de Enero 2021
a) Portada (como se muestra en el archivo "portada para anteproyecto de residencia profesional")						
b) Resumen general de su proyecto final (párrafo de 5 a 10 líneas)						
c) Cronograma preliminar de actividades (como se muestra en el archivo "cronograma 2019")						
Análisis de riesgo (en una tabla con dos columnas: lista de riesgos y soluciones propuestas)						
"Justificación", en una página deben describir el problema de "Halting (Parada)"						
Proponer una hipótesis						
Metodología (tradicional o ágil)						
Requerimientos Funcionales y No Funcionales						
Diseño y desarrollo de su proyecto						
Diseño y desarrollo de su proyecto						
Diseño y desarrollo de su proyecto						
				GIOVANNI		
				AXEL		
				RAMON		

RESUMEN GENERAL DE PROYECTO

En este documento se presentarán las múltiples capacidades de desarrollo de nuestro sistema de modelado (Petri net), el cual nos permitirá ejecutar simulaciones de la estructura de un modelo (Maquina de Turing) y recopilar información sobre el rendimiento; así como también nos servirá como un modelo formal estudiar sistemas concurrentes y distribuidos como también su ampliación al estudio de las propiedades específicas del sistema, como su rendimiento, fiabilidad, etc.

ANÁLISIS DE RIESGO DEL PROYECTO

RIESGO	SOLUCIÓN
Intensidad en el ritmo de trabajo	Las cosas deben ser echas en tiempo y forma para evitar este riesgo
la carga de trabajo	Repartirnos el trabajo y ayudarnos mutuamente
incrementar el estrés	Tomar el fin de semana ratos libres para evitar el estrés
No entender el tema al 100%	Poner atención al profesor para que no existan dudas acerca del tema
Que no se guarde el proyecto	Guardar constantemente el proyecto para y respaldarlo para que no haya algún conflicto
Que el equipo no esté conforme con el trabajo de otros integrantes	Comunicarse entre todos para que no haya inconformidades

¿QUE ES LA RED PETRI?

Una herramienta matemática para el estudio de las comunicaciones con los autómatas. Algunas de las aplicaciones más importantes de las Redes de Petri, creada por Carl Petri en 1962, esta ha sido en el modelado y análisis de los protocolos de comunicación, en el modelado y análisis de los sistemas de manufactura. En esta área, se han utilizado para representar líneas de producción, líneas de ensamble automatizadas, sistema de producción automotriz, sistemas de manufactura flexible, sistemas just-in-time, etc.

Las Redes de Petri son grafos bipartidos que consisten de tres tipos de objetos:

1. Lugares.
2. Transiciones.
3. Arcos orientados

Un lugar se puede unir mediante un arco con una transición, y una transición se puede unir con un lugar también mediante un arco, pero nunca se podrán unir mediante un arco dos lugares o dos transiciones.

Regla de evolución. Un lugar P es un lugar de entrada de una transición T si existe un arco orientado que conecta este lugar a la transición. Un lugar P es un lugar de salida de una transición T si existe un arco orientado que conecta esta transición al lugar.

Cada lugar puede tener una marca (token) que indica cuando la condición asociada con este lugar es falsa o verdadera. En cualquier instante de tiempo, la distribución de lugares, llamado marcado de Petri define el estado del modelo. El marcado de una red de Petri es denotado por un vector M(p) de mx1 que representa el número de marcas en los lugares correspondientes. En el ejemplo M=[1,0,0,0,0]´

Una transición está activada si están marcados todos sus lugares de entrada. Una transición activada puede ser disparada si se verifica el evento asociado a la transición. El disparo de una transición supondrá quitar una marca de cada uno de sus lugares de entrada y añadir una marca a todos sus lugares de salida.

1(UNIVERSIDAD DE CANTABRIA CURSO 2012/13) consulta(10/12/20)

PROBLEMA DE LA PARADA

También conocido como problema de la detención y consiste en lo siguiente: dada una Máquina de Turing {\displaystyle My una palabra {\displaystyle w}, determinar si {\displaystyle M} terminará en un número finito de pasos cuando es ejecutada usando {\displaystyle w} como dato de entrada. Alan Turing, en su famoso artículo "On Computable Numbers, with an Application to the Entscheidungsproblem" (1936), demostró que el problema de la parada de la Máquina de Turing es indecidible (no computable o no recursivo), en el sentido de que ninguna máquina de Turing lo puede resolver.

2(Luis Enrique 2011) consulta 09/12/20

3(Francisco Salto 12/2009) consulta 09/12/20

Relevancia

Al ejecutar un conjunto de programas, este puede terminar después de un número finito de pasos o puede no terminar nunca. En la práctica, este último caso se manifiesta como programas que se quedan "trabados" o que entran a un bucle infinito. Por esta razón sería de gran utilidad resolver la siguiente pregunta en la práctica:

· Existe un programa P, tal que, dado un programa cualquiera q y unos datos de entrada x, muestre como salida 1 si q con entrada x termina en un número finito de pasos o muestre como salida 0 si q con x entra a un bucle infinito.

Conocer si existe el programa P es, en términos resumidos, el problema de la parada.

Sin embargo, hay que hacer notar que la sabiduría popular acerca de este problema hace pensar que nunca es posible demostrar que un programa termina. Esto es falso.

Lo que se afirma es que no existe una manera automática computable de saber si todos los programas del mundo terminan. No se niega que exista la prueba para programas concretos. De hecho, la construcción de pruebas para programas concretos es un paso obligatorio para demostrar su correctitud.

IRRESOLUBILIDAD DEL PROBLEMA

La irresolubilidad del problema se puede mostrar de varias formas, pero en esencia todas equivalen a un argumento diagonal de Cantor.

4 (Wikipedia consultado 13/12/20)

HIPÓTESIS

Primero, se analiza el problema de la parada por medio de las técnicas del análisis teórico en el que se demuestra rigurosamente el problema sin solución por cualquier computación habida y por haber; también, que tales problemas sean formulables aritméticamente.

En este problema de la parada de Turing, para el que se demostró que no existe un procedimiento algorítmico de decisión.

Una de las razones por la que es importante conocer que el problema de la parada es que este no tiene solución, ya que nos permite decidir si otros problemas son resueltos o no. Gracias a la Red Petri se logra deducir la existencia de tiempos óptimos para los problemas de la parada, es decir, el enfoque adoptado para el análisis no estándar.

¿POR QUÉ USAMOS LA METODOLOGÍA ÁGIL?

Usamos la metodología ágil por lo compleja que es y por la formulación que se requiere para el desarrollo del proyecto, ya que se necesita de un equipo eficiente para llevarlo a cabo, además de rapidez y flexibilidad por parte de todo.

También, porque esta metodología está enfocada a mejorar el resultado a futuro del proyecto, por ello la implementaremos en nuestro proyecto.

METODOLOGÍA AGIL

Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

También es un enfoque en tiempo real a la gestión de proyectos que ayuda a gestionar el proceso de programación que se va desarrollando, de forma progresiva a medida que va evolucionando. Con la metodología ágil creas, incorporas retroalimentación, pruebas y gestionas tus proyectos, todo ello de manera simultánea.

7(blog wam Encarna Abellan consulta 19/12/20)

Ventajas

A continuación, enumeramos algunas de las ventajas que nos brinda la gestión ágil de proyectos:

Mejora de la calidad del producto: Estas metodologías fomentan el enfoque proactivo de los miembros del equipo en la búsqueda de la excelencia del producto.

Mayor satisfacción del cliente: El cliente está más satisfecho al verse involucrado y comprometido a lo largo de todo el proceso de desarrollo

Mayor motivación de los trabajadores: Los equipos de trabajo autogestionados, facilitan el desarrollo de la capacidad creativa y de innovación entre sus miembros.

Trabajo colaborativo: La división del trabajo por distintos equipos y roles junto al desarrollo de reuniones frecuentes, permite una mejor organización del trabajo.

Uso de métricas más relevantes: Las métricas utilizadas para estimar parámetros como tiempo, coste, rendimiento, etc. son normalmente más reales en proyectos ágiles que en los tradicionales**.**

6(iebs consulta el 19/12/20)

Los valores de la metodología ágil

La metodología ágil como la conocemos en la actualidad nació en respuesta a los enfoques en cascada de la gestión de proyectos, en que estos se organizan como series de secuencias lineales, un grupo de desarrolladores de software redactó el Manifiesto para el Desarrollo Ágil de Proyectos. De acuerdo con lo que establecieron, los equipos de desarrollo ágil de software deberían valorar:

Las personas y las interacciones antes que los procesos y las herramientas
El software en funcionamiento antes que la documentación exhaustiva
La colaboración con el cliente antes que la negociación contractual
La respuesta ante el cambio antes que el apego a un plan.

5(Red Hat consulta 19/12/20)

LISTA DE REQUERIMIENTOS

Requerimientos funcionales:

Describen una interacción entre el sistema y su ambiente. Cómo debe comportarse el sistema ante determinado estímulo.
Describen lo que el sistema debe hacer, o incluso cómo NO debe comportarse.
Describen con detalle la funcionalidad del mismo.
Son independientes de la implementación de la solución.
Se pueden expresar de distintas formas.

Requerimientos no funcionales:

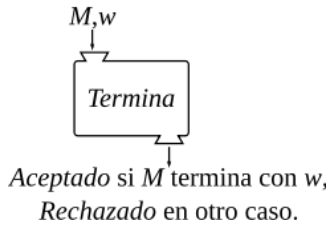
Describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema.
Requerimientos del producto: Especifican el comportamiento del producto (usabilidad, eficiencia, rendimiento, espacio, fiabilidad, portabilidad).
Requerimientos organizacionales: Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador (entrega, implementación, estándares).
Requerimientos externos: Interoperabilidad, legales, privacidad, seguridad, éticos.

DISEÑO

Existe un programa *P*, tal que, dado un programa **cualquiera** *q* y unos datos de entrada *x*, muestre como salida 1 si *q* con entrada *x* termina en un número finito de pasos o muestre como salida 0 si *q* con *x* entra a un bucle infinito.

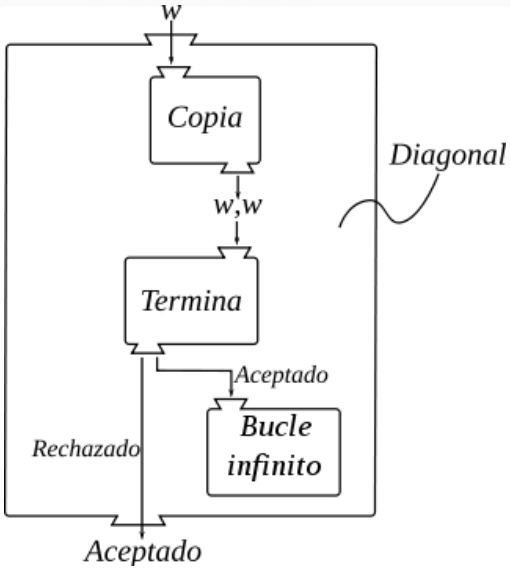
DESARROLLO

Supongamos que el problema de la parada tiene solución. Es decir, supongamos que existe una máquina de Turing que es capaz de determinar si otra máquina de Turing parará (terminará) con una entrada determinada. Llamemos *Termina* a esta máquina. Esta máquina recibiría como entrada la cadena M,w, donde M es la codificación de una máquina de Turing y w es la codificación de la cadena que se le alimenta a M. La máquina termina terminará en un estado de aceptación si M para ante la entrada w, y en otro caso terminará en un estado de rechazo, pero nunca entrará en un ciclo infinito.



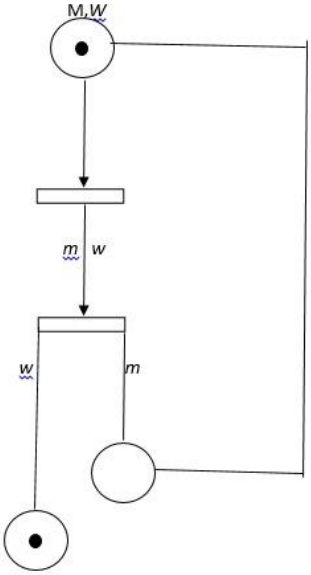
tomaremos la codificación de la máquina *Diagonal*, y la aplicaremos como entrada a la propia máquina *Diagonal*. Como w es la codificación de *Diagonal*, de lo anterior se sigue que *Diagonal* para ante sí misma como entrada si y solo si *Diagonal* no para ante sí misma como entrada. Esta conclusión es paradójica, pero todas las máquinas que hemos usado en la demostración, exceptuando *Termina*, son construibles; por lo tanto la clave de la demostración se encuentra, por reducción al absurdo, en la supuesta existencia de la máquina *Termina*. Al ser imposible la existencia de tal máquina *Termina*, que resolvía el problema de la parada, el problema de la parada no puede ser solucionado por ninguna máquina de Turing.

Diagonal para ante la entrada w → la máquina codificada en w no para ante la entrada w .



Las redes de Petri pueden describir fácilmente sistemas en los que se producen relaciones de paralelismo, exclusión mutua y sincronización entre procesos en este caso se comprueba que tanto las máquinas de Turing como la red Petri no resuelven el problema de la parada.

Se realizo la representación de la Red Petri implementando el problema de halting de la siguiente manera:



BIBLIOGRAFIAS

https://www.ctr.unican.es/asignaturas/MC_ProCon/Doc/PETRI_1.pdf
http://teodelacomp.blogspot.com/2011/05/52-problemas-de-halting-por-jose-luis.html
http://glossarium.bitrum.unileon.es/Home/teorema-de-turing
https://es.wikipedia.org/wiki/Problema_de_la_parada#Relevancia_en_la_pr%C3%A1ctica
https://www.redhat.com/es/devops/what-is-agile-methodology
https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/
https://www.wearemarketing.com/es/blog/que-es-la-metodologia-agile-y-que-beneficios-tiene-para-tu-empresa.html#:~:text=La%20metodolog%C3%ADa%20Agile%20es%20una,est%C3%A1%20enfocada%20a%20mejorar%20resultados.