

EIF400 – Paradigmas de Programación

Proyecto de programación #3

Profesor: Mag. Georges Alfaro S.

PLANTEAMIENTO DEL PROBLEMA

Para este proyecto deberán implementar un conjunto de predicados para representar las conexiones una red simple de área local y poder resolver o calcular algunas características sobre la red, como accesibilidad o confiabilidad.

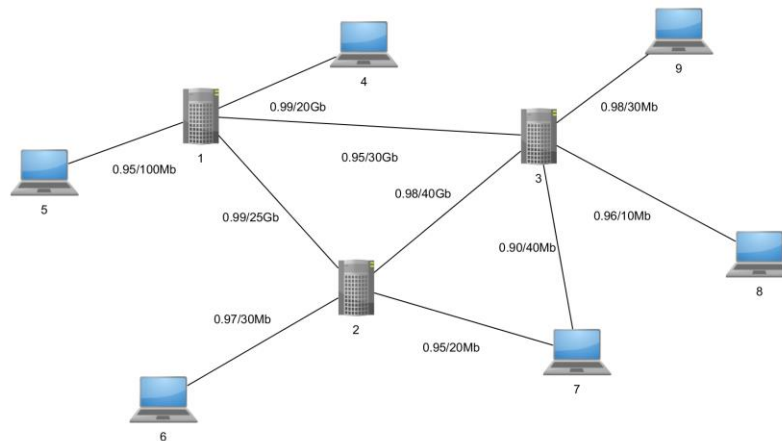
OBJETIVOS DEL PROYECTO

El proyecto tiene como objetivo estudiar algunas de las técnicas básicas de programación declarativa en un tipo de aplicación práctica.

DESCRIPCIÓN DEL PROYECTO

Descripción general.

La red se representará como un grafo no dirigido. Cada nodo representará un equipo conectado a la red, y los arcos describirán algunas características de las conexiones correspondientes.

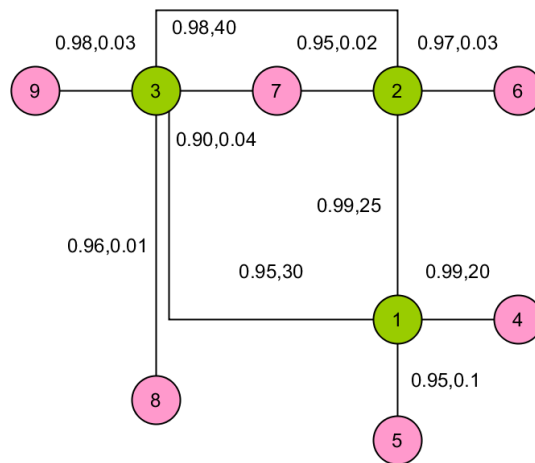


Debe escribir los predicados necesarios para representar **un grafo no dirigido** de cualquier orden. Cada vértice o nodo del grafo puede estar conectado con cualquier número de vértices. Cada arco del grafo tiene dos valores asociados, representando el nivel de confiabilidad de la conexión y la velocidad máxima de transmisión permitida, respectivamente.

El nivel de confiabilidad se registra como un valor de punto flotante entre 0 y 1. Un valor de 0.95, por ejemplo, indica que esa conexión es confiable en un 95% o, lo que es lo mismo, que

tiene una probabilidad de fallo del 5%.

La velocidad máxima de transmisión es un valor numérico positivo. Un valor de 100 puede representar una velocidad de transmisión de 100 Mb/s (o 100 Gb/s). Observe que este valor es un escalar, es decir, no tiene ninguna unidad de medida asociada. Si una conexión tiene una velocidad de 20 Mb/s y otra 3 Gb/s, podrían representarse con los valores 20.0 y 3000.0 o de la misma manera 0.02 y 3.0.



Cuando se intenta comunicar dos nodos a través de una conexión cualquiera, es posible que dicha conexión falle, según la probabilidad asociada a cada una.

La forma específica de los predicados queda a criterio de los estudiantes. Pueden existir ciclos en el grafo, pero ningún nodo estará conectado consigo mismo. También podrían existir conexiones redundantes entre dos nodos, excepto cuando uno de los dispositivos sea un cliente. Un cliente puede tener varias conexiones, pero no directamente con otros clientes.

Los predicados a completar son los siguientes:

| | |
|-------------------------------------|---|
| conexion(Grafo,A,B). | El predicado tiene éxito si los nodos A y B se encuentran conectados en grafo (que representa la red). La red tiene nodos que son servidores, enrutadores o clientes. Una ruta puede existir entre dos dispositivos cualesquiera, pero no puede pasar “a través” de un cliente. Los nodos intermedios de la ruta deben ser servidores o enrutadores (señalados en color verde en el ejemplo del grafo anterior). En todos los casos, Grafo es una estructura que representa el grafo completo de la red. |
| conexion(Grafo,A,B,Ruta). | Igual que en el caso anterior, el predicado tiene éxito si los nodos A y B se encuentran conectados. El parámetro Ruta contendrá la ruta de conexión (lista de nodos o arcos). El predicado puede encontrar cualquier ruta, y mostrará todas las rutas posibles cuando se soliciten soluciones alternativas. Hay que considerar la posibilidad de que existan ciclos en la ruta, y éstos deberán evitarse. Una ruta que conecte los nodos A y B no tendrá entonces ningún nodo o arco repetido. |
| velocidad_maxima(Grafo,A,B,Ruta,V). | El predicado tiene éxito si V es la mayor velocidad de transmisión posible entre los nodos A y B a través de la ruta Ruta . La velocidad máxima de transmisión es el valor mínimo de la velocidad entre dos nodos de la ruta. |
| velocidad_maxima(Grafo,A,B,V). | El predicado tiene éxito si V es la velocidad máxima posible entre A y B, considerando todas las rutas posibles. |
| confiabilidad(Grafo,A,B,Ruta,P). | El predicado tiene éxito si P es la confiabilidad de la conexión entre los nodos A y B a través de la ruta Ruta. El valor de P es el producto de los valores de confiabilidad entre cada uno de los nodos de la ruta. |

CONSIDERACIONES DE IMPLEMENTACIÓN

Escriba los predicados necesarios utilizando el lenguaje de programación Prolog. El proyecto se revisará utilizando la versión de SWI-Prolog utilizada en clase (versión 7.6.4).

Incluya en el código los comentarios necesarios para explicar la implementación de cada predicado definido, e incluya las pruebas que considere necesarias para verificar su correcto funcionamiento.

Es posible utilizar predicados propios del lenguaje que no se hayan visto en clase, pero no se podrán usar bibliotecas o predicados que implementen de alguna manera los solicitados.

Deberán utilizar recursión de cola siempre que sea posible.

Evite hacer comparaciones de variables. Por ejemplo, si tiene un predicado similar al siguiente:

```
p(A,B, ...) :- ..., A = B, ..., .
```

deberá utilizar en su lugar:

```
p(A,A, ...) :- ..., ..., .
```

Tampoco deberán aparecer variables únicas (*singleton*) en ningún predicado.

Puede utilizar el predicado `write()` para mostrar resultados intermedios, pero no sustituye directamente el cálculo de valores.

Por ejemplo, si tiene un predicado que suma dos valores numéricos, podría hacerlo de la siguiente manera:

```
suma(X,Y,Z) :- Z is X + Y.
```

Pero sería **incorrecto** si lo resuelve así:

```
suma(X,Y) :- Z is X + Y, write(Z).
```

ENTREGA Y EVALUACIÓN

El proyecto debe entregarse **por medio del aula virtual, en el espacio asignado para ello**. La entrega se hará en la semana 17 del curso. **(jueves 15 de noviembre de 2018)**. **No se aceptará ningún proyecto después de esa fecha**, ni se admitirá la entrega del proyecto por correo electrónico. El proyecto se puede realizar en grupos de **cuatro personas, como máximo**.

En caso de que las funciones no trabajen adecuadamente, efectúe un análisis de los resultados obtenidos, indicando las razones por las cuales el programa no se ejecuta correctamente, y cuáles son las posibles correcciones que se podrían hacer.

La definición de cada predicado tiene el mismo valor.

Observaciones generales:

- Se debe indicar en **cada documento entregado** el nombre completo y cédula de cada participante del grupo, indicando el nombre del curso, ciclo lectivo y descripción del trabajo que se entrega. Esto incluye comentarios en cada archivo fuente entregado.
- Los trabajos no se copiarán de ninguna llave USB u otro dispositivo en el momento, sino que se deben entregar en el formato adecuado.
- Cualquier trabajo práctico que no sea de elaboración original de los estudiantes (plagio) se calificará con nota 0 (cero) y se procederá como lo indiquen los reglamentos vigentes de la universidad.