Análisis de terminación de reglas, un enfoque con red de Petri

Joselito Medina Marín y Xiaoou Li

Sección de Computación
Departamento de Ingeniería Eléctrica
CINVESTAV-IPN
A.P. 14-740, Av.IPN 2508, México D.F., 07360, México
jmedina@computacion.cs.cinvestav.mx, lixo@cs.cinvestav.mx

Resumen El análisis de reglas en bases de datos activas es muy importante, ya que pueden detectarse problemas como el de No-terminación. Este problema se origina porque la relación que guardan las reglas, la acción de una regla es el evento que activa a otra, genera cadenas entre ellas y si una es disparada, entonces se produce un disparo consecutivo de reglas hasta volver con la primera regla disparada, provocando un disparo inifinito de reglas. El problema de No-terminación debilita al sistema manejador de la base de datos, por muy robusto que éste sea. En este artículo presentamos el análisis de la no-terminación en el disparo de reglas, utilizando los fundamentos teóricos de la Red de Petri Coloreada Condicional. Se propone como método de análisis la aplicación de la matriz de incidencia en la red de Petri, para detectar la no terminación del disparo de reglas en bases de datos activas.

1. Introducción

Las bases de datos activas (BDA), a diferencia de las bases de datos (BD) tradicionales, tienen la capacidad de reaccionar ante la ocurrencia de eventos interés y, si se presentan las condiciones apropiadas, se ejecutan una o varias acciones en la BD [1] [2]. Esta capacidad que tienen las BDA se logra mediante la definición de una base de reglas *evento-condición-acción* (reglas ECA ó reglas activas), cuya forma general es:

on evento
if condición
then acción

En las cuales se especifica el evento que se debe detectar, la condición que se evaluará y la acción que será ejecutada en la BD. El desarrollo de una base reglas ECA es una tarea sencilla cuando se define un reducido número de reglas sin embargo, si se desarrolla una base de reglas con un número considerable elementos, es posible perder el control que se tenga sobre las relaciones existentes entre las reglas, ocasionando dificultades para detectar la presencia de ciclos infinitos durante el procesamiento de reglas y, en consecuencia, el problema

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 56-67, 2003. © Centro de Investigación en Computación, IPN, México.

no-terminación. La no terminación del disparo de reglas conduce al sistema manejador de BD a un estado inconsistente, provocando caídas al sistema y pérdida de información, entre otras cosas.

Un modelo estructural para modelar las reglas ECA fue propuesto por los mismos autores de este artículo, denominado Red de Petri Coloreada Condicional (CCPN) [3],[4]. Una CCPN es una extensión de una red de Petri (PN) tradicional, con capacidades para modelar reglas ECA. En este artículo se analiza la información almacenada en la matriz de incidencia de la CCPN, la cual nos ayuda a determinar si el disparo de reglas activas termina o no. Además, en la CCPN se realiza el análisis de terminación de una base de reglas activas. Para llevar a cabo este análisis, se propone un algoritmo para encontrar todas las rutas cíclicas de la CCPN a partir de la matriz de incidencia respectiva. A continuación se muestra la definición de la CCPN, presentada en [5].

1. Red de Petri coloreada condicional

na PN es un tipo particular de gráfica dirigida bipartita, compuesta por dos tipos de objetos. Estos objetos son **lugares y transiciones**, los arcos que los unen están dirigidos de lugares a transiciones o de transiciones a lugares. Gráficamente, los lugares se representan por círculos y las transiciones por barras rectángulos. En su forma más simple, una PN se representa por una transición unida con sus lugares de entrada y lugares de salida. Esta red básica se utliza para representar varios aspectos de un sistema que se pretende modelar. Con la finalidad de estudiar el comportamiento dinámico de los sistemas modelados, cada lugar puede tener cero o más tokens. Los tokens se representan gráficamente por pequeños círculos rellenos. La presencia o ausencia de un token dentro de un lugar indica si una condición asociada con este lugar es falsa verdadera, ó también nos indica si un dispositivo que se está modelando se encuentra disponible o no. Una extensión de PN para modelar reglas ECA y a vez llevar a cabo la simulación del comportamiento de las mismas es la Red Petri Coloreada Condicional (CCPN) [3] [4]. La extensión CCPN se propuso para ofrecer un modelo estructural de representación de reglas activas. El modelo original de PN no cubre el manejo de información (colores) en los tokens. Para modelación de sistemas donde es necesario manejar datos se crearon las redes Petri coloreadas (CPN). La CCPN, además de manejar información de la BD tal como una CPN modela un sistema con manejo de datos), también evalúa parte condicional de la regla. La condición de cada regla se almacena en las transiciones, donde se lleva a cabo su evaluación, auxiliándose de la información que contienen los tokens.

En la figura 1 se muestra un ejemplo con tres reglas activas. Las dos primeras reglas tienen el mismo evento que las activa (insert_VENTAS), por lo tanto, se crea un lugar p_0 para este evento, una transición copy t_0 para crear dos salidas p_0 y evaluar, por separado, a las reglas. El evento de la regla 3 es la acción la regla 1, por lo que el lugar de salida p_4 pasa a ser el lugar de entrada de la tercera regla, concatenndo así la primer y tercer regla. Cuando ocurre el evento insert_VENTAS, la información del registro agregado se almacenará en un token que se desplazará de p_0 a t_0 y, a su vez, de t_0 se desplazará un token hacia y otro hacia p_2 (p_1 y p_2 representan copias del lugar p_0). Con la información de los tokens se evalúan las condiciones almacenadas en t_1 y t_2, y si éstas cumplen, entonces se generan tokens con información sobre la acción de la regla. Los tokens que lleguen a p_4 serán enviados a t_3 para evaluar la condición que tiene almacenada, y si el resultado es positivo, entonces se envía a p_5 el token referente a la acción de la regla. Antes de dar una definición formal de la CCPN

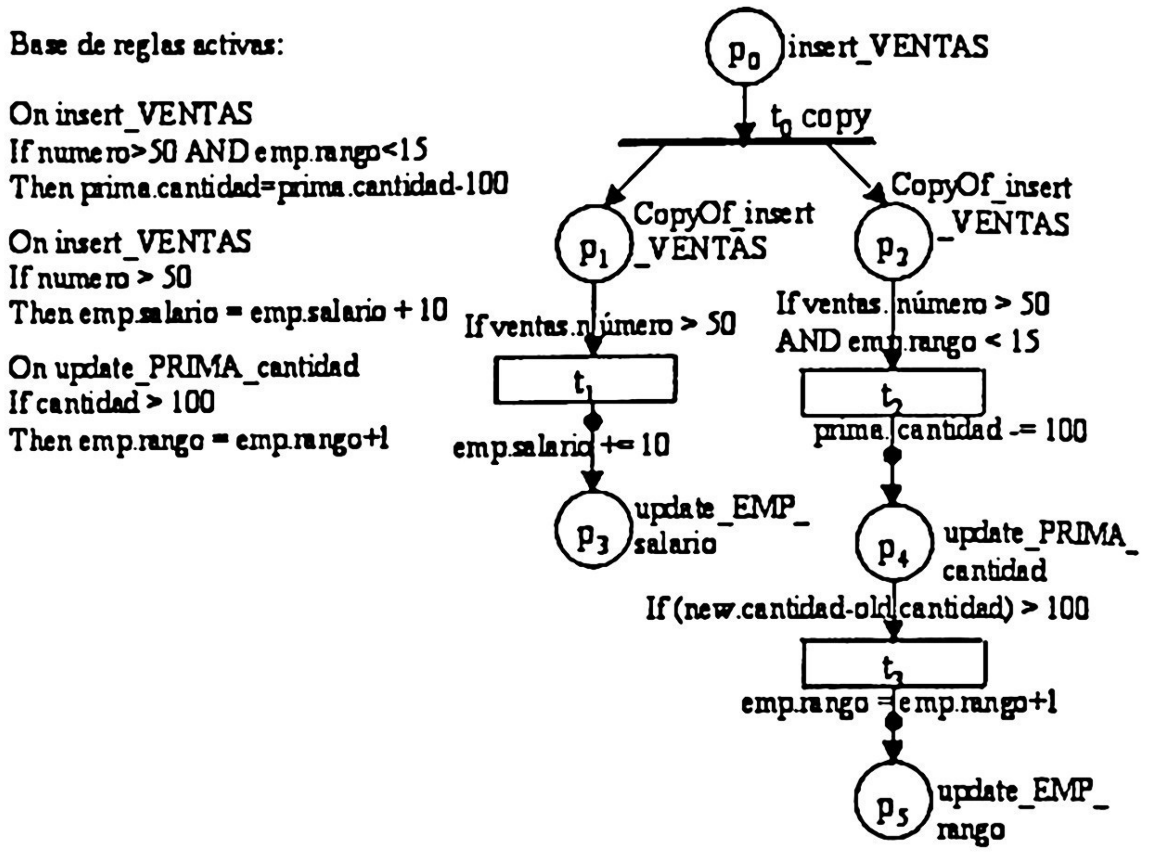

Figura 1. Ejemplo de una red de Petri Coloreada Condicional.

definiremos algunos elementos necesarios en la definición.

Definición 1. *Se le llama "color" a la estructura que tiene el token acorde lugar donde se coloca.*

Definición 2. *La CCPN tiene cuatro tipos de lugares:*
$P_{insert\_tabla} = \{p_{it} \mid p_{it}$ *representa un lugar de tipo "INSERT"*$\}$,
$P_{update\_tabla\_columna} = \{p_{utc} \mid p_{utc}$ *representa un lugar de tipo "UPDATE"*$\}$,
$P_{delete\_tabla} = \{p_{dt} \mid p_{dt}$ *representa un lugar de tipo "DELETE"*$\}$,
$P_{copy} = \{p_c \mid p_c$ *representa un lugar de tipo "COPY"*$\}$

Definición 3. *La CCPN tiene dos tipos de transición:*
$T_{regla} = \{t_r \mid t_r$ *representa la transición que almacena la parte condicional de regla ECA*$\}$ *y*
$T_{copy} = \{t_c \mid t_c$ *representa la transición que genera duplicados del token proveniente de su lugar de entrada*$\}$

Las transiciones $t \in T_{regla}$ se representan por un rectángulo. Mientras que las transiciones $t \in T_{copy}$ se representan por una línea gruesa.

Definición 4. *La CCPN, al igual que las PN, tiene arcos de entrada* $A_{in} = \{(p,t) \mid p \in P,\ t \in T,\ (p,t) \in \{PxT\}\ \}$ *y de salida* $A_{out} = \{(t,p) \mid t \in T,\ p \in P,\ (t,p) \in \{TxP\}\ \}$

Formalmente una CCPN se define de la siguiente manera:

Definición 5. *Una PN Coloreada Condicional (CCPN) es una 9-tupla:*

$$CCPN = \{\Sigma, P, T, A, N, C, Cond, Action, I\}$$

donde:

Σ *Conjunto finito de tipos de dato, los cuales son soportados por los lugares, llamado conjunto de colores.*

$P = \{p_1, p_2, \ldots, p_m\}$ *es un conjunto finito de lugares. Este conjunto está compuesto por* $P = P_{insert\_tabla} \cup P_{update\_tabla\_columna} \cup P_{delete\_tabla} \cup P_{copy}$

$T = \{t_1, t_2, \ldots, t_n\}$ *es un conjunto finito de transiciones. Este conjunto está compuesto por dos subconjuntos, es decir,* $T = T_{regla} \cup T_{copy}$

A : *conjunto finito de arcos dirigidos de* P *a* T *o de* T *a* P. A *está formado por los conjuntos* A_{in} *y* A_{out}, $A = A_{in} \cup A_{out}$

$N : A \rightarrow \{PxT\} \cup \{TxP\}$ *es una función nodo que asocia a cada arco un par ordenado* $(p,t) \in A_{in}$ *ó* $(t,p) \in A_{out}$, *donde el primer elemento del par ordenado es el nodo origen y el segundo es el destino.*

$C : P \rightarrow \Sigma$ *es una función color, que relaciona a cada lugar* $p \in P$ *un tipo de dato* $Type(p) \in \Sigma$.

$Cond : T_{regla} \rightarrow Bool$, *es una función condición que evalúa la parte condicional de la regla ECA almacenada en la transición* $t \in T_{regla}$.

Acción : $T_{regla} \rightarrow C(p)_{MS}$ *es una función acción que asocia cada transición* $t \in T_{regla}$ *con un multi-conjunto de un tipo de dato* $C(p)$, donde p es el lugar de salida de t.

$I : P \rightarrow C(p)_{MS}$ *es una función de inicialización que relaciona cada lugar* $p \in P$ *con un multi-conjunto de un tipo de dato* $C(p)$.

Regla de disparo de transiciones Habiendo dado una definición formal de la CCPN, ahora describiremos su comportamiento. Uno de los elementos más importantes que rigen el comportamiento de la PN son los tokens. Los tokens de CCPN no son tokens simples, sino que tienen la capacidad adicional de almacenar información de un cierto tipo de dato. Su definición es la siguiente:

Definición 6. *Un elemento token es un par* (p,c) *donde* $p \in P$ *y* $c \in C(p)$. *El conjunto de todos los elementos token está denotado por* TE. *Una marca* M *es un multi-conjunto sobre* TE. *La marca inicial* M_0 *es obtenida al evaluar las expresiones de inicialización:*

$$\forall (p,c) \in TE : M_0(p,c) = I(p).$$

El conjunto de todas las marcas es expresado por M. Otro elemento mental que rige el comportamiento de la CCPN es la transición. Una transici $t \in T_{copy}$ se dispara en una marca M_x si y sólo si:

$$\forall p \in^{\cdot} t : M_x(p) > 0$$

Una transición $t \in T_{regla}$ se dispara en una marca M_y si y sólo si se satisfac siguiente:

$$\forall p \in^{\cdot} t : M_y(p) > 0, Type(Cond(t)) = verdadero$$

La expresión $^{\cdot}t$ representa al conjunto de los lugares de entrada $p \in P$ de transición $t \in T$ y el conjunto de los lugares de salida $p \in P$ de una transic $t \in T$ se representa por la expresión $t^{\cdot}$.

1.2. Trabajos relacionados

Se han desarrollado diferentes enfoques para realizar el análisis de bases glas ECA. En [6] se presenta un análisis de bases de reglas activas basado algoritmo de "propagación", el cual utiliza un álgebra relacional extendida determinar cuando la acción de una regla afecta a la condición de otra y así tectar la formación de cadenas entre las relaciones de reglas. En la metodolog que se está proponiendo, la CCPN muestra gráficamente, y a simple vista, relación existente entre la acción de una regla y el evento de otra, además, bido a que la parte condicional de cada regla se almacena en transiciones transición por cada regla), es fácil analizar si la acción de una regla afecta condición de la otra regla con quien se encuentra conectada. Fraternali, boschi y Tanca en [7], construyen una "Hipergráfica de disparo". En esta gráfi los nodos son restricciones y los arcos son reglas, que viajan desde una restricci hacia un conjunto de restricciones. Los ciclos en la gráfica se interpretan secuencias infinitas de reglas. Las reglas que son parte de los ciclos se elimin de la gráfica, pero no se eliminan en el sistema, solamente se marcan para monitoreo especial en tiempo de ejecución y prevenir ciclos infinitos. En trabajo de investigación, los ciclos presentes en la gráfica de la CCPN consideran ciclos infinitos en el disparo de reglas, porque la acción de una del ciclo puede provocar que la condición de la regla que le precede sea falsa.

El artículo [8] describe la implementación del método de la Gráfica de paro Refinado (RTG) para el análisis de terminación de reglas activas. El mé do RTG estudia el contenido de ciclos de reglas en una gráfica de disparo. análisis demuestra que una regla no puede disparar a otra, el arco que conect estas dos reglas en la gráfica de disparo se elimina, hasta obtener una gráfica disparo refinada. Al igual que en nuestra investigación, se realiza un análisis haustivo para determinar si una regla dispara a otra; pero a diferencia de [8], nuestro caso solamente analizamos aquellas reglas que se encuentran dentro rutas cíclicas. En [9] y [11] se utilizan métodos de análisis de PN para detectar presencia de ciclos en el disparo de reglas. En [9] utilizan el árbol de alcanzab idad ("reachability tree") y en [11] aplican el árbol de cobertura ("coverabili tree"). En estos métodos concluyen que con la presencia de ciclos en la PN, probable que ocurra el problema de no terminación del disparo de reglas. En el presente documento, también se propone la aplicación de un método de análisis de PN para detectar el problema de no terminación: la matriz de incidencia, donde, además de encontrar los ciclos en la CCPN, se estudia la posibilidad del disparo infinito de las reglas que forman parte de los ciclos.

Otro trabajo interesante es un método, basado en redes de Petri, para analizar las reglas de bases de datos activas y determinar si las reglas (1) presentan ciclos, (2) son inconsistentes y (3) entran en contradicciones. Los autores proponen una Red de Petri con Restricciones (Constraint Petri Net, CPN), la cual es una sistema de red Predicado/Transición y almacena conjuntos de restricciones en cada transición [10]. Sin embargo, este método no realiza una búsqueda exhaustiva para detectar las contradicciones de las reglas. Finalmente, en [12], Detlef Zimmer, Rainer Unland y Axel Meckenstock, proponen a las redes de Petri como la base para un análisis de reglas en tiempo de compilación. Argumentan que el análisis basado en redes de Petri es un enfoque que puede utilizarse para una especificación detallada del comportamiento de las reglas. Precisamente, en la CCPN se ha observado que, muchas de las características de las reglas activas, pueden ser representadas.

2. Matriz de incidencia de la CCPN

El comportamiento dinámico de muchos sistemas estudiados en ingeniería pueden describirse mediante ecuaciones diferenciales o ecuaciones algebraicas. El comportamiento dinámico de las PN también se describen por medio de ecuaciones. Uno de los métodos de análisis del comportamiento de un sistema modelado con redes de Petri es la *matriz de incidencia* [13], la cual almacena información relevante sobre la relación existente entre los lugares (representados por las columnas de la matriz, y las transiciones (representados por los renglones). La definición formal de la matriz de incidencia es la siguiente.

Definición 7. *Para una red de Petri N, con n transiciones y m lugares, la matriz de incidencia $A = [a_{ij}]$ es una matriz de números enteros de $n \times m$. El valor para cada elemento de la matriz está dado por:*

$$a_{ij} = a_{ij}^{+} - a_{ij}^{-}$$

donde $a_{ij}^{+} = w(i,j)$ es el peso del arco que conecta una transición $t_i \in T$ con su lugar de salida $p_j \in P$ y $a_{ij}^{-} = w(j,i)$ es el peso del arco que conecta una transición $t_i \in T$ con su lugar de entrada $p_j \in P$.

En la matriz de incidencia se almacena suficiente información sobre la estructura completa de una PN, y es posible reconstruir la PN original a partir de los valores almacenados en la matriz de incidencia. Para el caso particular de este enfoque, la matriz de incidencia almacena solamente valores -1, 0 y 1, porque los pesos de los arcos en la CCPN siempre es 1. El valor a_{ij} de A, que representa la relación entre una transición $t_i \in T$ y su lugar de entrada $p_j \in P$ es -1, porque

$a_{ij}^+ = 0$ y $a_{ij}^- = 1$, por lo tanto $a_{ij} = 0 - 1 = -1$. De manera similar, el valor de A que representa la relación entre una transición $t_i \in T$ y su lugar de $p_j \in P$ es 1, porque $a_{ij}^+ = 1$ y $a_{ij}^- = 0$, por lo tanto $a_{ij} = 1 - 0 = 1$. El de a_{ij} es 0 cuando t_i y p_j no tienen relación alguna, es decir, que p_j no lugar de entrada ni de salida para t_i. Lo anterior puede resumirse de la siguiente manera:

$$a_{ij} = \begin{cases} -1 & \text{El lugar } p_j \in P \text{ es un lugar de entrada a la transición } t_i \in T \\ 0 & \text{No existe un arco que conecta al lugar } p_j \in P \text{ con la transición } \\ & t_i \in T \text{ y viceversa} \\ 1 & \text{El lugar } p_j \in P \text{ es un lugar de salida de la transición } t_i \in T \end{cases}$$

En cada una de las columnas de A aparecerá, a lo más, un valor $a_{ij} =$ Si el lugar $p_j \in P$ es un lugar de entrada, entonces en su respectiva columna de A aparecerá un valor de -1. En cambio, si el lugar p_j solamente es lugar salida, entonces no existirán valores de -1 en la columna j de la matriz en la matriz no existen columnas donde solo aparezca un valor de -1 y el 0's, entonces todos los lugares son lugares de entrada y salida a la vez.

En base a las propiedades particulares de la matriz de incidencia para CCPN, se presentan las siguientes definiciones.

Definición 8. *Un lugar es un nodo inicial NI si la columna presenta solamente valores 0's y un valor de -1.*

Definición 9. *Un nodo terminal NT es un lugar que representa la acción, no el evento, de una o varias reglas. En la matriz de incidencia, este lugar caracteriza por almacenar solamente valores 0's y 1's en la columna correspondiente. La ausencia de valores -1's es debido a que el lugar no representa lugar de entrada para alguna transición.*

3. Análisis de terminación

Un tópico importante en diseño de bases de reglas activas es el problema No-terminación. El problema de no-terminación se presenta cuando la acción de una regla r_0 provoca el disparo de una regla r_1 y, la regla r_1, provoca disparo de la regla r_2 y así sucesívamente, hasta llegar a una regla r_n, donde esta última provoca el disparo de la regla r_0, produciendo el disparo infinito reglas. Es necesario, determinar la Terminación o No-terminación del disparo reglas en una BDA para evitar estados incosistentes en la BD. A continuación presentamos algunas definiciones necesarias en la demostración del teorema Terminación, basado en CCPN:

Definición 10. *Una ruta R es una secuencia de pares ordenados (i, j), obtenida a partir de la matriz de incidencia de la CCPN. La secuencia de pares ordenados (i, j) es el desplazamiento sobre la matriz de incidencia que describe la conexión que existe entre los lugares y las transiciones. Las parejas ordenadas de una ruta están formadas de la siguiente manera:*

$$(a, b), (a, c), (d, c), \cdots, (w, x), (y, x), (y, z)$$

Los valores de i y j son los índices en la matriz de incidencia. El primer par ordenado está compuesto por la transición t_a y su lugar de entrada p_b, continuando con la ruta, el segundo par ordenado tiene la misma transición t_a y su lugar de salida p_c, el tercer par ordenado (d, c) tiene el mismo lugar p_c pero con la transición t_d, de la cual es su lugar de entrada. El órden de los valores de cada par ordenado debe ser -1 y 1, de forma alternada, como se muestra en la siguiente tabla:

(a,b),	(a,c),	(d,c),	···,	(w,x),	(y,x),	(y,z)
-1	1	-1	···	-1	1	-1

Para formar una ruta, primero se debe localizar, en las columnas, el lugar que represente a un nodo inicial; si no existe ningún nodo inicial (existen ciclos) tomamos el lugar de la primer columna. El lugar de la primer columna siempre almacenará un valo -1, porque, de acuerdo al algoritmo para formar una CCPN [5], el lugar p_0 es el evento de la primera regla. En la columna seleccionada localizamos el valor -1, siendo la coordenada de -1 el primer elemento de R. Posteriormente, en el renglón donde se localizó el valor -1 se busca el valor 1, y la coordenada donde se ubique será el segundo elemento de R. Sucesivamente se busca el siguiente -1 en las columnas y el siguiente 1 en los renglones, hasta que no se encuentre uno de éstos valores. Si alguna de las coordenadas ya existe dentro de la secuencia, entonces se trata de una ruta cíclica. A continuación se presenta un algoritmo recursivo para encontrar las parejas ordenadas que forman una ruta:

```
siguienteNodo(int valor, int indice)
  if(valor == -1)
    boolean flag = false;
    for(i=0; i<n; i++)
      if(A[i][indice] == valor)
        flag = true;
        if(noExisteNodo(i,indice))
          agregaNodo(i,indice);
          siguienteNodo(1,i);  // busca el valor 1 en el rengl\'{o}n
          eliminaUltimoNodo();
        else  // Se encontro una ruta c\'{i}clica
          agregaNodo(i,indice);
          imprimeRuta();
          eliminaUltimoNodo();
    if(!flag)  // Se encontro una ruta ac\'{i}clica
      imprimeRuta();
  else
    for(j=0; j<m; j++)
      if(A[indice][j] == valor)
        agregaNodo(indice,j);
        siguienteNodo(-1,j);  // busca el valor -1 en la columna j
        eliminaUltimoNodo();
```

Definición 11. *Una ruta cíclica RC es una ruta R donde el último par ordenado* (y,z) *ya se encuentra listado en R, es decir,* $y = q_k$ *y* $z = r_k$, $k = 1,2,...,|R|-1$.

Definición 12. *Una ruta acíclica RA es aquella donde la última pareja ordenada* (i,j) *de la ruta es diferente de sus antecesoras y le corresponde una valor de en su matriz de incidencia respectiva, es decir, que RA tiene un nodo terminal*

Dadas estas definiciones, presentamos el siguiente teorema de terminación:

Teorema 1. *Si todas las rutas R de una CCPN son acíclicas, entonces el disparo de las reglas termina.*

Demostración. Dado que las rutas R de una CCPN son acíclicas, por definición de R no existen nodos en las rutas que se repitan, todos los nodos, en cada ruta, son diferentes, por lo que cada ruta tiene un nodo inicial y un nodo terminal. Suponiendo que la pareja ordenada (i_t, j_t) es un nodo terminal de la ruta R, el lugar j_t es un lugar que representa la acción de una o mas reglas, mas no el evento activador de otras, terminando en este punto con la ejecución consecutiva de reglas, por lo tanto el disparo de reglas ECA termina.

A partir del teorema presentamos el siguiente corolario:

Corolario 1. *Si existen rutas cíclicas RC en la CCPN, y existe al menos una transición* $t \in T_{rule}$, *donde* $Cond(t) = false$ *dentro de cada RC, entonces el disparo de las reglas ECA termina.*

Demostración. Dado que en la CCPN existen rutas cíclicas RC entonces existen parejas ordenadas (i,j) que se repiten según la definición de RC. Se presenta un disparo consecutivo de reglas cuando cada acción de reglas, dentro de la ruta cíclica, provoca que la condición de la siguiente regla sea verdadera, sin embargo, de acuerdo a los argumentos del corolario, existe al menos una transición, tal que $Type(Cond(t)) = falso$, truncando de esta manera la activación de una siguiente regla, convirtiendo así la última pareja ordenada (i,j), con valor de -1, en un nodo terminal, y con la existencia de un nodo terminal en la ruta R entonces el disparo de las reglas ECA termina.

4. Caso de estudio

Para mostrar la viabilidad que ofrece nuestro enfoque en la detección de ciclos infinitos, tomamos la base de reglas que se presenta en [10], la cual se describe a continuación:

Regla 1: Cuando la prima de un empleado se modifica, si la cantidad es mayor que \$100.00, entonces el rango del empleado se incrementa en uno.

Regla 2: Cuando el rango de un empleado se modifica (supongamos que se incrementa), entonces la prima del empleado se incrementa diez veces el nivel del nuevo rango.

Regla 3: Cuando se obtienen las ventas del mes y el número de éstas es superior a 50, entonces se incrementan \$10.00 al salario del empleado.

Regla 4: Cuando se obtienen las ventas del mes y el número de éstas es superior a 100, entonces el rango del empleado se incrementa en un nivel.

Regla 5: Cuando el nivel del rango de un empleado se modifica y el rango alcanzó el nivel 15, entonces el salario del empleado se incrementa en un 10 %.

Regla 6: Cuando se obtienen las ventas del mes y el número de éstas es superior a 50 y el rango de empleado que las obtuvo es menor de del nivel 15, entonces el monto de su prima se decrementa \$100.00.

Las tablas que forman parte de la BD son:

```
EMPLEADO(emp_id, nombre, rango, salario)
PRIMA(emp_id, cantidad)
VENTAS(emp_id, mes, numero)
```

La CCPN generada a partir de éstas cuatro reglas se muestra en la figura 2.

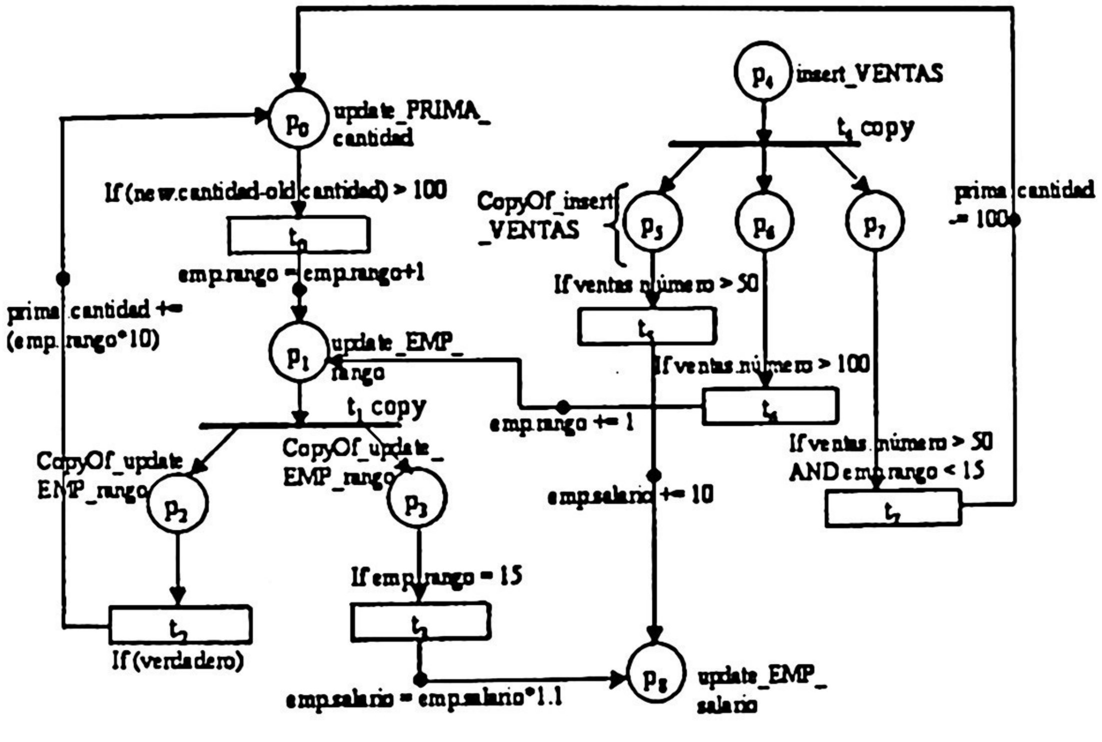

Figura 2. CCPN obtenida a partir de la base de reglas.

La matriz de incidencia A correspondiente a la CCPN es la siguiente

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

Las rutas R que se obtienen de A son las siguientes:

$R[0]$: (4,4), (4,5), (5,5), (5,8)
$R[1]$: (4,4), (4,6), (6,6), (6,1), (1,1), (1,2), (2,2), (2,0), (0,0), (0,1), (1,1)
$R[2]$: (4,4), (4,6), (6,6), (6,1), (1,1), (1,3), (3,3), (3,8)
$R[3]$: (4,4), (4,7), (7,7), (7,0), (0,0), (0,1), (1,1), (1,2), (2,2), (2,0), (0,0)
$R[4]$: (4,4), (4,7), (7,7), (7,0), (0,0), (0,1), (1,1), (1,3), (3,3), (3,8)

Se puede observar que son cinco rutas diferentes que se pueden generar el disparo de las reglas, de las cuales, dos de ellas contienen la misma secuencia cíclica de nodos ($R[1]$ y $R[3]$). Dado que existen rutas cíclicas, es posible que el disparo de reglas sea infinito, sin embargo, de acuerdo a la descripción del Corolario 1, debe existir al menos una transición, dentro de la ruta cíclica, que condición sea falsa. Dentro de la ruta se encuentran las transiciones $t_0, t_2 \in T_{rule}$, pero la condición que se evalúa en t_2 siempre es verdadera. La condición de verifica si el campo cantidad de la tabla PRIMA se incrementó en más de \$100, el lugar p_0, que representa al evento de actualizar el campo cantidad en la tabla PRIMA, recibe información de t_2 y de t_7. La transición t_7 no provoca el disparo de t_2 porque la actualización del campo cantidad es un decremento. La transición t_2 actualiza el valor de prima.cantidad incrementándole en diez veces el valor del rango, por lo que si el valor del rango es igual a un número mayor que 10 entonces el valor prima.cantidad se incrementará en más de \$100, evaluando en verdadero la condición de t_0, incrementando a la vez el valor de rango y produciendo una vez más un incremento mayor de \$100 para prima.cantidad, generando de esta manera en un disparo infinito de reglas. Se puede concluir de que si el valor de emp.rango es menor de 10, entonces el disparo de las reglas termina.

5. Conclusiones y trabajo futuro

En este artículo se presentó un enfoque para atacar el problema de No-terminación en BDA, diferente a los desarrollados en la actualidad. En este enfoque se utiliza la CCPN para analizar la base de reglas ECA, con el apoyo de la matriz de incidencia, herramienta de análisis en la teoría de PN. La matriz de incidencia de una CCPN presenta propiedades importantes en la detección de ciclos. Al igual que en muchos trabajos relacionados, se parte de que la ausencia de ciclos garantiza la terminación del disparo de las reglas ECA. En nuestro caso, además de probar la terminación del disparo de reglas debido a la ausencia de ciclos, probamos que, aún con la presencia de ciclos, el disparo de reglas termina, si existe al menos una regla cuya condición sea falsa y, de esta manera, romper el ciclo infinito en que pueda caer el disparo de las reglas.

En el caso de estudio presentado, la estructura de la CCPN ofrece un número menor de lugares y transiciones que los trabajos relacionados donde se aplican las redes de Petri. En otros enfoques solamente verifican la no existencia de ciclos en la gráfica de disparo de las reglas, para determinar que el disparo de reglas termina (como es el caso de donde se tomó la base de reglas), mientras que en este trabajo se verifica que las condiciones presentes en el ciclo sean verdaderas.

para asegurar que el disparo de las reglas no va a terminar. Por el momento, la CCPN solo detecta eventos simples. Se tiene propuesto, como trabajo a seguir, incorporar al modelo de la CCPN la detección de eventos compuestos, y a la vez, considerarlos en el análisis de No-terminación.

Referencias

1. Paton, N.W., Diaz O.: Active Database Systems. In: ACM Computing Surveys, Vol. 31, No. 1,1999, pp. 64-103
2. Chakravarthy, S.: Early Active Database Efforts: A Capsule Summary. In: IEEE Transactions on Knowledge and Data Engineering, Vol.7, No.6,1995, pp.1008-1010
3. Li, X., Medina Marín, J., and Chapa, S. V.: A Structural Model of ECA Rules in Active Database. In: Mexican International Conference on Artificial Intelligence (MICAI'02), Mérida, Yucatan, México, April 22-26, 2002
4. Medina Marín, J., y Li, X.: Red de Petri Coloreada Condicional y su Aplicación en Sistemas de Bases de Datos Activas. En: Octava Conferencia de Ingeniería Eléctrica 2002 (CIE'02), México, D.F., México, Septiembre 4, 5 y 6, 2002.
5. Medina Marín, J.: Red de Petri Coloreada Condicional (CCPN) y su Aplicación en Bases de Datos Activas. En: Tesis de Maestría de la Sección de Computación, Depto. de Ingeniería Eléctrica, CINVESTAV-IPN, México. Septiembre de 2002.
6. Baralis, E. and Widom, J.: An Algebraic Approach to Static Analysis of Active Database Rules. In: ACM Transactions on Database Systems, Vol. 25, No. 3, September 2000, pp. 269-332.
7. Fraternali, P., Paraboschi, S., and Tanca, L.: Automatic Rule Generation for Constraint Enforcement in Active Databases. In: U.W. Lipeck and B. Thalheim (Eds.), Modelling Database Dynamics. London: Springer-Verlag, 1993. Selected Papers from the Fourth International Workshop Foundations of Models and Languages for Data and Objects, pp. 153-173.
8. Urban, S.D., Tschudi, M.K., Dietrich, S.w. and Karadimce, A.P.: Active Rule Termination Analysis: An Implementation and Evaluation of the Refined Triggering Graph Method. In: Journal of Intelligent Information Systems, No. 12, 1999, pp. 27-60.
9. Kokkinaki, A.I.: On using Multiple Abstraction Models to Analyze Active Databases Behavior. In: Biennial World Conference on Integrated Design and Process Technology, Berlin, Germany, June, 1998.
10. Guisheng, Y., Qun, L., Jianpei, Z., Jie, L., Daxin, L.: Petri Based Analysis Method For Active Database Rules. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 2, 1996, pp. 858-863
1. Zimmer, D.: Rule Termination Analysis based on a Rule Meta Model. In: Cadlab Report 2, Cadlab, Bahnhofstr. 32, 33102 Paderborn, Germany, April 1995
12. Zimmer, D., Unland, R., Meckenstock, A.: Using petri nets for rule termination analysis. In: Proceedings of the Workshop on Databases: Active and Real-Time 1996 (CIKM'96), Rockville, Maryland, 1996.
13. Murata, T.: Petri Nets: Properties, analysis, and applications". In: Proceedings of the IEEE, 77(4):541-580, 1989.