

# VERIFICACIÓN DE LA CONSISTENCIA DE UNA BASE DE CONOCIMIENTOS BASADA EN REGLAS

Ing. CARLOS PALACIO

*Investigador del Laboratorio  
de Informática*

## 1. INTRODUCCIÓN

Este trabajo tiene como finalidad abarcar un problema, que actualmente es objeto de estudio e investigación y es el de la coherencia de una Base de Conocimientos Basada en Reglas de Producción. Es sabido que un conjunto de reglas representan el conocimiento de un cierto dominio y como tal su implantación correcta constituye una parte muy importante en el desarrollo de un sistema experto.

Se describirá una herramienta para la verificación de la consistencia de un conjunto de reglas. Se verá su implementación para un caso en particular, pero es importante destacar que los principios de esta herramienta pueden ser adaptados a distintos conjuntos de reglas que representan el conocimiento de un dominio específico.

## 2. EL PORQUE DE UNA EFICAZ VERIFICACIÓN DE LA BASE DE CONOCIMIENTOS

El Ingeniero del Conocimiento debe asegurarse que el sistema dará a sus usuarios informes precisos y soluciones correctas a sus problemas. El proceso de verificación de un sistema para comprobar si este es completo y confiable tiene dos componentes diferentes:

- á. verificar que la Base de Conocimientos contenga toda la información necesaria (integridad).
- b. verificar que los programas puedan interpretar y aplicar esta información correctamente.

La depuración de la Base de Conocimientos, o sea el proceso de verificación de si ésta es correcta y completa, es un componente del **gran problema de adquisición del conocimiento** debido a que antes de que el conocimiento pueda ser introducido en un sistema de computación, debe sufrir una serie de transformaciones:

1. En primer término, el ser humano adquiere experiencia en algún dominio específico a través del estudio, de la investigación y de la experiencia.
2. Luego, el experto intenta formalizar esta experiencia y expresarla en

la representación interna de un sistema experto; por ejemplo: reglas de producción, marcos o redes semánticas.

3. Finalmente, el conocimiento es añadido a la Base de Conocimientos del sistema en una forma comprensible para la máquina como son las expresiones LISP o PROLOG.

Los problemas que pueden aparecer son debido a que:

1. El conocimiento del experto es incompleto, inconsistente o incluso parcialmente erróneo.

2. El conocimiento exacto y completo no se ha adecuado correctamente a la representación elegida.

3. El experto que no conoce los computadores trabaja con un ingeniero del conocimiento que no está familiarizado con el dominio del problema. En este caso, las faltas de entendimiento que se producen, generalmente no son conocidas hasta que ocurren problemas en el funcionamiento del sistema y que afectan a las prestaciones del mismo.

4. Se cometen errores de sintaxis cuando una Base de Conocimientos es introducida en el sistema.

Por lo tanto, este proceso involucra la prueba y refinamiento del sistema de manera que permita descubrir y corregir la variedad de errores que pueden ocurrir durante el proceso de transferencia de experiencia desde un experto humano a un sistema de computación.

### 3. CARACTERÍSTICAS DE UNA VERIFICACIÓN ASISTIDA

Como se ha apuntado, la Base de Conocimientos de un sistema experto es construida, generalmente, por la colaboración entre expertos en el problema e ingenieros del conocimiento. Los expertos en la materia formulan sus conocimientos y los ingenieros del conocimiento codifican éste para que pueda ser utilizado por el sistema. Esta tarea dificultosa y consumidora de tiempo puede ser facilitada por un programa que:

1. Busque inconsistencias y huecos en la Base de Conocimientos.
2. Ayude y facilite la comunicación entre el experto y el ingeniero.
3. Prevea una representación clara y entendible del conocimiento tal como el sistema lo utilizará.

Por los trabajos más importantes que se han realizado hasta el momento, se distinguen dos modos de realizar la verificación de reglas:

1. **Interactivo;** que en base a un intercambio de información entre el experto y el programa se determinan las inconsistencias en la Base de Conocimientos.
2. **Sistemático;** que en base a una verificación que realiza un programa verificador entrega un listado con las inconsistencias encontradas en la Base de Conocimientos.

Una asistencia automatizada para los ingenieros del conocimiento puede

ayudar a identificar rápidamente problemas en la Base de Conocimientos y permitir a los expertos el descubrir huecos en su conocimiento o errores en su razonamiento.

#### 4. COMPROBACIÓN DE LAS REGLAS DE PRODUCCIÓN

Las herramientas de prueba de una base se ven claramente incrementadas si se cuenta con un programa que pruebe su integridad y consistencia en forma sistemática.

La búsqueda de un prototipo de programa, que realice esta forma de verificación sistemática en una Base de Conocimientos Basada en Reglas de Producción, constituye el objetivo de los siguientes temas.

##### 4.1. MATRIZ DE CONSISTENCIAS

La especificación de la Base de Conocimientos puede ser formulada de manera que consista en un conjunto de variables, cada una de las cuales debe ser asignada a un dominio en particular, y un conjunto de predicados que los valores de las variables deben satisfacer.

$$\begin{array}{l}
 X_1 \quad X_2 \quad \dots \quad X_n \quad P_1 \quad X_1 \quad \cap \quad P_2 \quad X_2 \quad \dots \cap \quad P_n \quad X_n \\
 P_{12} \quad X_1, X_2 \quad \cap \quad P_{13} \quad X_1, X_3 \quad \dots \cap \quad P_{n-1n} \quad (X_{n-1}, X_n) \\
 \vdots \\
 P_{n-2n-1n} \quad X_{n-2}, X_{n-1}, X_n \quad \dots \cap \quad P_{n-k..n} \quad (X_n - k, \dots, X_n)
 \end{array}$$

El conjunto de variables consta de una lista finita de  $n$  variables  $V = (v_1, \dots, v_n)$ . La variable  $v_i$ , tiene asociada un dominio finito  $D_i = (v_{i1}, \dots, v_{im_i})$ , del cual puede tomar cualquiera de los  $M_j$  valores. Un conjunto de relaciones  $R = [R_1, \dots, R_n]$ , especifica cuáles valores son compatibles mutuamente para el conjunto de  $n$  variables. En particular una relación  $R_j$ :

$$R_j \leq D_{i1}^j * D_{i2}^j * \dots * D_{ir_j}^j$$

especifica qué pares de valores son consistentes (llamados por su argumento  $r_j$ ), para las variables  $i^j$ . Estas relaciones limitadoras pueden en la práctica ser dadas como tablas de la verdad, y más concretamente, representadas como una matriz  $[r_{ki}^j]$ , tal que el elemento  $[r_{ki}^j] = 1$ , si y sólo si, el valor  $k$  de la variable  $i$  es consistente con el valor  $l$  de la variable  $j$ . De lo contrario  $[r_{ki}^j] = 0$ . Esto sería esencialmente una representación de todas las relaciones entre pares de variables.

Esta matriz se puede implementar de manera tal que dé las relaciones

consistentes de todas las variables que están incluidas en la Base de Conocimientos. Lo ideal sería que las relaciones fueran todas aquellas pertenecientes al dominio completo, pero evidentemente, en algunos casos el dominio de un sistema experto basado en reglas es bastante extenso e incierto, en cuyo caso las relaciones se harían en base a un dominio acotado. Pero no por ser **acotado** significa que sea poco efectivo, sino que puede ser suficientemente potente como para cubrir gran parte de las expectativas esperadas. En efecto, se puede cubrir aquellas relaciones consistentes que ayuden a determinar incoherencias en la Base de Conocimientos. La habilidad reposa en saber determinar ese dominio acotado como para poder formar la matriz con las relaciones consistentes de todas las variables que están incluidas en la Base de Conocimientos.

### 4.2. IMPLEMENTACION DE LA MATRIZ DE CONSISTENCIAS

La matriz, que se denominará matriz-consistencia, contendrá todas las variables de la Base de Conocimientos y los valores del dominio acotado. A continuación, en la figura 1, se muestra un formato simplificado de la matriz de consistencias.

Cada elemento de la matriz de consistencias estará identificado por cuatro subíndices [I, J, K, L]; los dos primeros identificarán las dos variables y los dos restantes identificarán los valores asignados a esas variables.

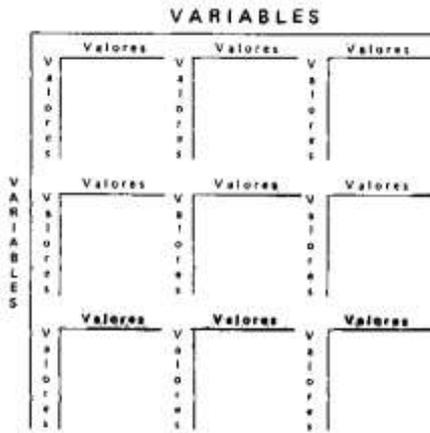


Figura 1. Representación esquemática de la Matriz.

Evidentemente, para completar la implementación se debe contar con dos vectores suplementarios:

1. vector de variables
2. vector de valores

### 4.3. VECTOR DE VARIABLES

La Base de Conocimientos contiene un conjunto de variables  $V = (v_1, v_2, \dots, v_n)$ , y estará expresada como un vector que se denominará vector-variable. El contenido de cada elemento será una variable del conjunto  $V$ . De esta manera, el vector de variables tendrá tantos elementos como variables tenga  $V$ :

vector-variable [1] =  $v_1$

vector-variable [2] =  $v_2$

vector-variable [n] =  $v_n$ .

### 4.4. VECTOR DE VALORES

El conjunto de valores que ayudará a especificar las relaciones consistentes estará dado por un vector denominado vector-valor. Cada componente de dicho vector tendrá la siguiente estructura:

((<VALOR>) (<CONTEXTO>))

Cada elemento contendrá una lista compuesta de dos partes; la primera indica el valor propiamente dicho, y la segunda el contexto de "aplicabilidad" de ese valor.

Con la ayuda del vector de variables y del vector de valores se puede implementar la matriz de consistencias de forma definitiva. De los cuatro subíndices que identifican un elemento de la matriz de consistencias, los dos primeros corresponden a los subíndices de las variables en el vector de variables y los segundos corresponden a los subíndices de los valores en el vector de valores.

De esta manera, el elemento matriz-consistencia [I, J, K, L] contendrá un 1 o un 0 si la variable vector-variable [I] con el valor vector-valor [K], es consistente o no con la variable vector-variable [J] con el valor vector-valor [L].

Debido a la simetría de la matriz, se puede tener en cuenta las siguientes consideraciones:

- a. Limitar la atención a las relaciones consistentes que cumplan con la condición  $I < J$ .
- b. Determinar los valores que puede tomar una variable, independientemente del resto, en la relación  $I = J$ .

#### 4.5. SINTAXIS DE LAS REGLAS DE PRODUCCIÓN

Las reglas, en formato LISP, es una lista compuesta, a la vez, de tres listas que se denominarán: contexto, antecedente y consecuente (figura 2).

```
Contexto      : ( (CONT1) (CONT2) ... (CONTn) )
Antecedente  : ( (ANTE1) (ANTE2) ... (ANTEn) )
Consecuente  : ( (CONS1) (CONS2) ... (COIMSn) )
```

**Figura 2.** Sintaxis de las Reglas de Producción.

La primera, define el contexto de aplicabilidad de la regla y está en relación con el contexto descrito en los valores del vector de valores. Como se verá más adelante, un valor será aplicado a una variable de la regla si el contexto definido en el valor está incluido en el contexto de la regla.

El antecedente es siempre una conjunción de cláusulas cada una de las cuales conforma una lista dentro de la lista "Antecedente".

El consecuente indica una o más conclusiones, las cuales son obtenidas si las cláusulas del antecedente son satisfechas.

Para que el usuario pueda visualizar las reglas de una manera más simple, las representaciones externa e interna son distintas, tal como muestra la figura 3.

Para poder distinguir los predicados, funciones, variables y constantes, se debe aplicar una nomenclatura determinada. En este caso, se adopta que la primera letra para el caso de las variables debe estar comprendida entre la 'U' y la 'Z', para las constantes, entre la 'A' y la 'E', para las funciones, entre la 'P' y la 'U', y para los predicados, entre la 'P' y la 'T'.

##### Representación Interna

```
Contexto      : ( ((CONT1) (CONT2)
```

##### Representación Externa

```
Contexto      :( ((CONT1) (CONT2) ... (CONTn))
Antecedente   : ((PRED1 VAR1 CONST1) (PRED2 VAR2) ...
                (PREDpVARkCONSTm))
Consecuente   : ((PREDi VARi CONSTi) ... (PREDj VARj)) )
```

### Representación Externa

Contexto : 1- CONT1  
2- CONT2  
.

Antecedente : n - CONTn  
n + 1 - PRED1 (VAR1 CONST1)  
n + 2 - PRED2 (VAR2)  
.

Consecuente : n + p - PREDp (VARk CONSTm)  
n + p + 1 - PREDi (VARi CONSTi)  
.

n + p + j - PREDj (PREDj (VARj))

**Figura 3.** Representación de las reglas.

#### 4.6. HEURÍSTICAS PARA LA DETECCIÓN DE INCONSISTENCIAS

Cuando el conocimiento está representado en forma de reglas de producción las inconsistencias aparecen como:

- **CONFLICTOS:** Dos reglas llegan en la misma situación pero con resultados distintos.
- **REDUNDANCIA:** Dos reglas suceden en la misma situación y tienen los mismos resultados.
- **SUBORDINACIÓN:** Dos reglas tienen los mismos resultados pero una contiene restricciones adicionales para las situaciones en las cuales ella pueda tener lugar. Siempre que la regla más restrictiva sea satisfecha, la menos también se activará, resultando por tanto una redundancia.
- **INCOMPLETITUD:** Ocurre cuando una regla ante una determinada situación no produce resultado alguno.

Estas condiciones pueden ser detectadas con la sintaxis que se describió, ya que permite examinar una o dos reglas y determinar: si existen situaciones en las cuales ambas puedan ocurrir, y si el resultado de aplicar las dos reglas es el mismo, si tienen conflictos o si no están relacionadas.

Con estas definiciones de inconsistencias, se puede formalizar las heurísticas que ayuden a descubrir los fallos en la Base de Conocimientos. Como primer paso está el de tomar dos reglas y aplicar las heurísticas de la figura 4.

<b>HEURÍSTICAS</b>	
<b>Antecedente</b>	<b>Consecuente</b>
Cons.1 =NULO Cons.2 = NULO Ante.1 Incluido Ante.2 y Cons.1 = Cons.2 Ante.2 Incluido Ante.1 y Cons.1 = Cons.2 Ante.1 = Ante.2 y Cons.1 = Cons.2 Ante.1 = Ante.2 y Cons.1 <> Cons.2	Incompletitud.1 Incompletitud.2 Subordinación.1 Subordinación.2 Redundancia. 1.2 Conflicto. 1.2

**Figura 4.** Heurísticas para la detección de inconsistencias.

#### 4.7. DESCRIPCIÓN DEL PROCEDIMIENTO

La matriz de consistencias, junto con el vector de valores y el vector de variables, permiten determinar la combinación de valores de los argumentos del antecedente y del consecuente de cada una de las reglas. El contexto de aplicabilidad de un valor debe estar incluido en el contexto de la regla para que este pueda ser asignado a la variable correspondiente. Las variables utilizadas en un conjunto de reglas deben estar destinadas a tomar ciertos valores dentro del dominio. Esta consideración se verá claramente en el ejemplo de implementación.

El procedimiento de verificación, básicamente, realiza los siguientes pasos:

1. Para cada regla del conjunto hacer:
  - 1.1. Extraer sus variables
  - 1.2. Conformar una tabla con todos los valores consistentes según la matriz-consistencia.
  - 1.3. Obtener una tabla con todos los casos posibles de la regla, basados en los resultados obtenidos en el paso 1.2.
2. Comprobar, por cada par de reglas, los casos obtenidos en el paso 1.3. e introducir las inconsistencias en una tabla, denominada Tabla.inconsistencia. La comprobación se realiza según las heurísticas definidas anteriormente.
3. Imprimir Tabla.inconsistencia.

#### 4.8. GENERALIDADES DEL PROGRAMA DE VERIFICACIÓN

El programa se puede considerar dividido en dos partes:

1. Una de ellas se encarga de suministrar las facilidades para la comunicación

con el usuario. Estas facilidades se expresan, principalmente, en forma de menú y tienen como principal objetivo la de ayudar al usuario a acceder a las fuentes de información, esto es: a la matriz de consistencia, vector de valores, vector de variables y conjunto de reglas, otorgándole la posibilidad de introducir, modificar, eliminar y consultar sobre cualquiera de ellas. Algunas de las características se detallan a continuación:

- 1.1. Obtención de forma sistemática del vector de variables. Es decir, que para el usuario el proceso de generación y actualización de dicho vector es transparente.
- 1.2. Actualización de la matriz de consistencias por cambios realizados tanto en el vector de valores como en el conjunto de reglas. Así, por ejemplo, si se da el caso de la modificación de una regla que acarrea la eliminación y/o el ingreso de variables, el sistema actualizará la matriz de consistencias manteniendo las relaciones anteriores ajenas a los cambios efectuados.
- 1.3. Acceso a la matriz de consistencias de forma gráfica y sencilla.

La segunda parte es el proceso de verificación propiamente dicho, es decir, la ejecución del procedimiento que, en función de las fuentes de información y las heurísticas, especifica las inconsistencias existentes. La figura 5 muestra los módulos principales con que cuenta el programa.

## 5. UN EJEMPLO DE IMPLEMENTACION

### 5.1. REGLAS DE PRODUCCIÓN PARA CIRCUITOS LÓGICOS

Recientes avances en metodología de diseño en tecnología de fabricación, han transformado al "hardware" digital en una complejidad sin precedentes. La desventaja de esa complejidad es que aumenta sustancialmente la dificultad para razonar sobre el diseño de un determinado circuito. Un sistema basado en reglas de producción es una forma de representación muy potente que puede ser usado para asistir a los diseñadores a través de simulación, diagnóstico y generación de "test" para sus diseños.

En función de los cuatro componentes lógicos AND, OR, XOR, y NOT y de los distintos tipos de conexiones básicas que se pueden realizar, se obtienen las siguientes reglas de producción:

- R1 – CONTEXTO : CONJUNCION  
 REGLA :  $PAND(X), PI(1,X,1), PI(2,X,1) \rightarrow PO(1,X,1)$
- R2 - CONTEXTO : CONJUNCION  
 REGLA :  $PAND(X), PI(V,X,0) \rightarrow PO(1,X,1)$
- R3 - CONTEXTO : DISYUNCION  
 REGLA :  $POR(X), PI(V,X,1) \rightarrow PO(1,X,1)$

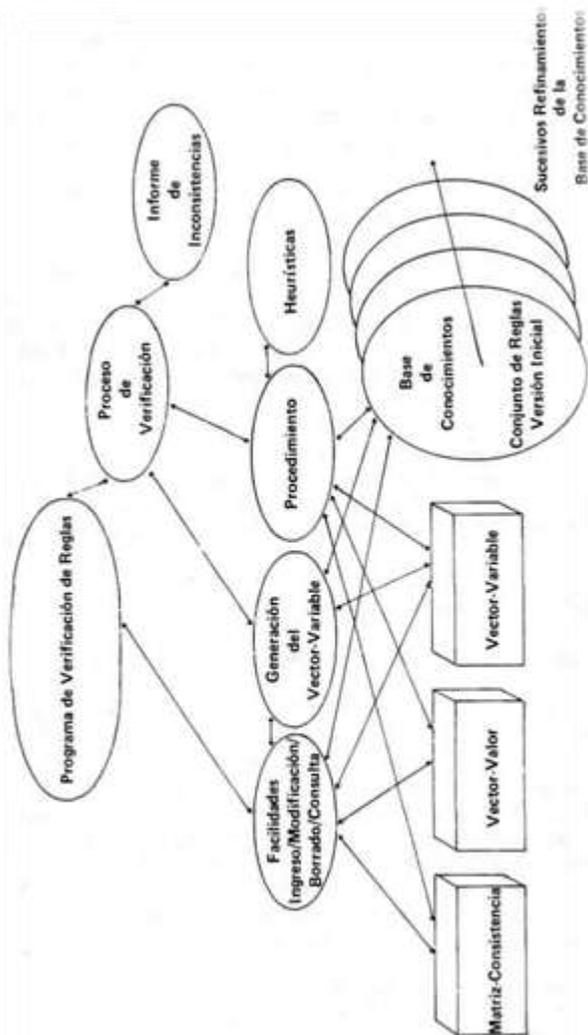


Figura 5. Módulos del Programa.

- R4- CONTEXTO : DISYUNCIÓN  
 REGLA : POR (X), PI (1, X, 0), PI (2, X, 0)  $\rightarrow$  PO (1, X, 0)
- R5- CONTEXTO : DISYUNCIÓN EXCLUSIVA  
 REGLA : PXOR (X), PI (1, X, U), PI (2, X, Z), PDIS (U, Z)  $\rightarrow$  PO (1, X, 1)
- R6- CONTEXTO : DISYUNCIÓN EXCLUSIVA  
 REGLA : PXOR (X), PI (1, X, Z), PI (2, X, Z)  $\rightarrow$  PO (1, X, 0)
- R7- CONTEXTO : NEGACIÓN  
 REGLA : PNOT(X), PI (1, X, 1)  $\rightarrow$  PO (1, X, 0)
- R8- CONTEXTO : NEGACIÓN  
 REGLA : PNOT(X), PI (1, X, 0)  $\rightarrow$  PO (1, X, 1)
- R9- CONTEXTO : CONJUNCIÓN  
 DISYUNCIÓN  
 DISYUNCIÓN EXCLUSIVA  
 NEGACIÓN  
 REGLA : PCON (PIN (W, X), PIN (V, Y)), PI (W, X, Z)  $\rightarrow$  PI (V, Y, Z)
- R10 CONTEXTO : CONJUNCIÓN  
 DISYUNCIÓN  
 DISYUNCIÓN EXCLUSIVA  
 NEGACIÓN  
 REGLA : PCON (POUT(W, X), PIN (V, Y)), PO(W, X, Z)  $\rightarrow$  PI (V, Y, Z)
- R11- CONTEXTO : CONJUNCIÓN  
 DISYUNCIÓN  
 DISYUNCIÓN EXCLUSIVA  
 NEGACIÓN  
 REGLA : PCON (POUT (W, X), POUT (V, Y)), PO (W, X, Z)  $\rightarrow$  PO(V, Y, Z)

## 5.2. COMPOSICIÓN DE LAS FUENTES DE INFORMACIÓN

### 5.2.1. VECTOR DE VARIABLES

En base al conjunto de reglas descrito el sistema obtendrá el siguiente vector de variables:

vector-variable [0] = X  
 vector-variable [1] = V  
 vector-variable [2] = U  
 vector-variable [3] = Z  
 vector-variable [4] = W  
 vector-variable [5] = Y

Como se ha dicho anteriormente, las variables están destinadas a tomar determinados valores del dominio. En este caso, las variables tienen las siguientes asignaciones:

## 5.22. VECTOR DE VALORES

El vector de valores debe incluir valores que abarquen de la mejor manera el dominio del conocimiento. En este caso sencillo, existirán valores que representarán a los distintos componentes, a las entradas/salidas, y a los números de entradas/salidas físicas de los componentes.

Se adoptará el valor VY1 como representativo de un componente conjuntivo, VX1 de uno disyuntivo, VO1 de uno disyuntivo exclusivo y VN1 de uno negativo. Los valores 0 y 1 representan los niveles de entrada/salida, y, V1 y V2 las entradas/salidas físicas de los componentes.

Por lo tanto, el vector de valores quedará implementado de la siguiente manera:

Indice	Valor	Contexto
0	VY1	CONJUNCIÓN
1	VX1	DISYUNCIÓN
2	VO1	DISYUNCIÓN EXCLUSIVA
3	VN1	NEGACIÓN
4	0	CONJUNCIÓN DISYUNCIÓN DISYUNCIÓN EXCLUSIVA NEGACIÓN
5	1	CONJUNCIÓN DISYUNCIÓN DISYUNCIÓN EXCLUSIVA NEGACIÓN
6	V1	CONJUNCIÓN DISYUNCIÓN DISYUNCIÓN EXCLUSIVA NEGACIÓN
7	V2	CONJUNCIÓN DISYUNCIÓN DISYUNCIÓN EXCLUSIVA NEGACIÓN

### 5.2.3. MATRIZ DE CONSISTENCIA

Las dimensiones de la matriz de consistencias, que se formará en base a los dos vectores, será de  $6 \times 6 \times 8 \times 8$ , debido a las 6 variables y 8 valores.

El siguiente paso es identificar las relaciones consistentes con 1, ya que todos los elementos de la matriz tienen asignados, por defecto, el valor 0.

Para una mejor interpretación se considerará la matriz de consistencias como una matriz de submatrices.

La primera submatriz relaciona a la variable X con sí misma, y como se había señalado, servirá para determinar los valores que puede tomar la variable independientemente del resto. En este caso, como X está destinada a tomar los valores de los componentes, esto es: VY1, VX1, VO1 y VN1, se incluirá un 1 en los elementos que relacionan dichos componentes, y se lee de la siguiente manera:

- La variable X con el valor VY1 es consistente con la variable X con el valor VY1.
- La variable X con el valor VX1 es consistente con la variable X con el valor VX1.
- La variable X con el valor VO1 es consistente con la variable X con el valor VO1.
- La variable X con el valor VN1 es consistente con la variable X con el valor VN1.

La segunda submatriz relaciona la variable X y V. Esta última tiene asignados los valores para las entradas/salidas físicas de los componentes, por lo que hará consistencias en aquellas relaciones donde se les asigne a X un valor de un componente y a V el valor V1 o V2. Es decir, que se leería de la siguiente manera:

- La variable X con el valor VY1 es consistente con la variable V con el valor V1.
- La variable X con el valor VX1 es consistente con la variable X con el valor V1.

El modo de operar en cada una de las restantes relaciones y submatrices es similar a las analizadas.

### 5.2.4. PROCESO DE VERIFICACIÓN

Con la implementación del vector de valores, la matriz de consistencias y las reglas de producción, es factible de realizar el proceso de verificación.

La opción **PROCESO** del menú principal accede al procedimiento de verificación.

Con el conjunto de reglas que se ha definido el proceso no hubiera detectado ninguna inconsistencia. Pero si se incluyen las siguientes reglas:

- R12- CONTEXTO : CONJUNCIÓN  
 REGLA :  $PAND(X), PI(1, X, 1), PI(2, X, 1) \rightarrow PO(1, X, 0)$
- R13- CONTEXTO : DISYUNCIÓN  
 REGLA :  $PI(V, Y, 1), POR(Y) \rightarrow PO(1, Y, 1)$
- R14- CONTEXTO : NEGACIÓN  
 REGLA :  $PNOT(X), PI(2, X, 0) \rightarrow$
- R15- CONTEXTO : DISYUNCIÓN EXCLUSIVA  
 REGLA :  $PXOR(X), PI(1, X, Z) \rightarrow PO(1, X, 0)$

se obtendría las inconsistencias mostradas en las figuras 7a y 7b. En ellas se interpreta que la regla R6 está subordinada por la regla R15 para los casos de componentes disyuntivos y con cualquiera de las entradas (1 ó 0). La regla R13 y R3 son redundantes, ya que para el valor VX1 ambas tienen los mismos antecedentes y las mismas acciones, tanto para la entrada en V1 como en V2. La regla R14 no posee consecuente, de tal manera que representa una incompletitud en el conjunto de reglas. Las reglas R12 y R1 están en conflicto debido a que las cláusulas en los antecedentes de ambas son iguales y producen acciones distintas. En este caso la regla R12 es errónea.

R6	-SUBORDINACION-	R15	-SUBORDINACION-
Antecedente:	1 — PXOR (V01) 2 — PI (1, V01, 1) 3 — PI (2, V01, 1)	Antecedente:	1 — PXOR (V01) 2 — PI (1, V01, 0)
Consecuente:	4 — PO (1, V01, 0)	Consecuente:	3 — PO (1, V01, 0)
R15	-SUBORDINACION-	R13	-REDUNDANCIA-
Antecedente:	1 — PXOR (V01) 2 — PI (1, V01, 1)	Antecedente:	1 — PI (V2, VX1, 1) 2 — POR (VX1)
Consecuente:	3 — PO (1, V01, 0)	Consecuente:	3 — PO (1, VX1, 1)
R6	-SUBORDINACION-	R3	-REDUNDANCIA-
Antecedente:	1 — PXOR (V01) 2 — PI (1, V01, 0) 3 — PI (2, V01, 0)	Antecedente:	1 — POR (VX1) 2 — PI (V2, VX1, 1)
Consecuente:	4 — PO (1, V01, 0)	Consecuente:	3 — PO (1, VX1, 1)
Presione una tecla . . .		Presione una tecla . . .	

Figura 7a.

R12-CONFLICTO-	R13 -REDUNDANCIA-
Antecedente: 1 — PAND (VY1)	Antecedente: 1 — PI (V1, VX1, 1)
2 — PI (1,VY1,1)	2 — POR (VX1)
3 — PI (2, VY1,1)	Consecuente: 3 — PO (1, VX1, 1)
Consecuente: 4 — PO (1, VY1, 0)	
R1 -CONFLICTO-	R3 -REDUNDANCIA-
Antecedente: 1 — PAND (VY1)	Antecedente: 1 — POR (VX1)
2 — PI (1,VY1, 1)	2 — PI (V1,VX1, 1)
3 — PI (2,VY1,1)	Consecuente: 3 — PO (1, VX1, 1)
Consecuente: 4 — PO (1, VY1, 1)	
	R14 -INCOMPLETITUD-
	Antecedente: 1 — PNOT (VN1)
	2 — PI (2,VN1,0)
	Consecuente:
Presione una tecla para retornar:	Presione una tecla . . .

**Figura 7 b.**

## 6. CONCLUSIONES

- El presente trabajo ha desarrollado un método que cumple con la utilización de una herramienta que verifica la consistencia y coherencia de una Base de Conocimientos Basada en Reglas. Dicho método responde a las exigencias originadas por la necesidad de generalización de una herramienta cuya aplicabilidad sea independiente del dominio del conocimiento que se trate.
- Como futura línea de investigación queda abierto el camino del desarrollo, a partir del método propuesto, de un procedimiento más eficiente que el desarrollado, suponiendo que la Base de Conocimientos contenga "lagunas". Para la solución de este problema sería necesario incluir las metas a alcanzar por el conjunto de reglas, y las heurísticas necesarias.