

02.20.01.01.04.03. Conclusiones.

Las ventajas que aporta la representación del conocimiento en reglas son las siguientes:

- *Facilidad de representación.* Las reglas son muy intuitivas, es fácil expresar en reglas el conocimiento de un amplio número de dominios.
- *Modularidad.* Cada regla es una pieza elemental de conocimiento independiente. A diferencia de un programa convencional, no hay orden prefijado en las reglas ni se hace referencia unas a otras. Esto facilita la comprensión del conocimiento.
- *Flexibilidad.* Cada regla puede ser añadida o eliminada en la base sin que ello tenga una repercusión importante en el comportamiento del sistema. En principio, cuantas más reglas tenga el sistema mejor calidad de respuestas tendrá el sistema. La falta de una determinada regla solo repercutirá en que en determinados casos muy específicos el sistema no sabrá dar soluciones. Esto es una diferencia importante con el estilo de programación habitual ya que la falta de una sentencia en un programa, o sólo el cambio de orden, puede hacer que el sistema completo no funcione.
- *Eficiencia.* En comparación con otras formas de representación del conocimiento la inferencia en reglas es más eficiente dado que está marcada la dirección de inferencia.

Sin embargo, como desventajas tiene:

- Dirección de inferencia prefijada. Los atributos que son premisa no pueden ser considerados como objetivos en otro razonamiento, cosa que sí es posible en representación en marcos y en restricciones. Esto resta generalidad en el uso del conocimiento para diferentes propósitos.
- Hay dominios de conocimiento que formulados en reglas dan lugar a bases de conocimiento muy grandes debido a que el conocimiento se expresa mejor mediante restricciones. Un ejemplo de este tipo de dominios son los relativos al comportamiento de sistemas físicos, que se expresan mejor mediante restricciones que representan ecuaciones que relacionan magnitudes físicas.

Las cualidades de la representación en reglas han hecho que sea una de las formas de representación más extendidas. Existe un gran número de herramientas informáticas para construcción de sistemas expertos que tienen como representación este método de forma única o combinado con algún otra representación. Por ejemplo la herramienta MI utiliza reglas con variables con encadenamiento hacia atrás y ART/IM maneja reglas de producción con encadenamiento hacia adelante. OPS-5 es un lenguaje muy extendido de reglas de producción con encadenamiento hacia adelante que maneja un método de inferencia muy eficiente (algoritmo RETE).

02.20.01.01.05. Restricciones.

En los últimos tiempos ha surgido en diferentes campos de la inteligencia artificial la necesidad de una nueva modelización y representación del conocimiento basada en restricciones. Así, si bien las reglas expresan influencias del tipo causa-efecto entre conceptos, una restricción es una ecuación que permite expresar múltiples relaciones causa-efecto, estableciendo puntos de sincronismo en el comportamiento de los objetos. Las restricciones por tanto incorporan la idea de ecuación cualitativa a las descripciones simbólicas propias de la inteligencia artificial, y por lo dicho, cubren esa laguna que dejan las reglas para expresar un tipo de conocimiento menos empírico y más general acerca del dominio.

El paradigma basado en restricciones surge como resultado de trabajos paralelos en diferentes campos de la inteligencia artificial, alguno de los cuales se estudiarán posteriormente, y son:

- El desarrollo de arquitecturas para razonamiento sobre sistemas físicos basado en técnicas de simulación cualitativa ha puesto de manifiesto las carencias del clásico modelo de razonamiento superficial basado en reglas deductivas. En estos casos la representación en restricciones proporciona modelos concisos para razonamiento cualitativo profundo que permite obtener conclusiones significativas evitando el coste computacional de la simulación numérica. Además, en muchos casos de sistemas físicos reales, los modelos construidos con reglas presentan problemas por la presencia de ciclos en las relaciones (realimentaciones e influencias bidireccionales entre las variables del sistema), problema resuelto por la naturaleza intrínsecamente bidireccional de las restricciones.
- Por otro lado, en el campo de los lenguajes de programación lógica, se presenta la posibilidad de aumentar la potencia expresiva de los mismos (sin ninguna pérdida en cuanto a sus propiedades semánticas) gracias a la aportación del paradigma basado en restricciones que puede reemplazar de forma ventajosa a la clásica unificación. En esta línea han surgido algunas realizaciones de lo que se han llamado CLP (Constraint Logic Programming) como el PROLOG-III debido a uno de los impulsores de la programación lógica (A. Colmenauer) que utiliza restricciones aritméticas y booleanas.

02.20.01.01.05.01. Tipos de restricciones.

Se pueden distinguir dos grandes líneas en la implementación de sistemas de satisfacción de restricciones de tipo numérico, apropiadas para aplicaciones de corte convencional y restricciones de tipo cualitativo, muy adecuadas para desarrollo de aplicaciones basadas en el manejo de información simbólica como suele ser el caso en inteligencia artificial.

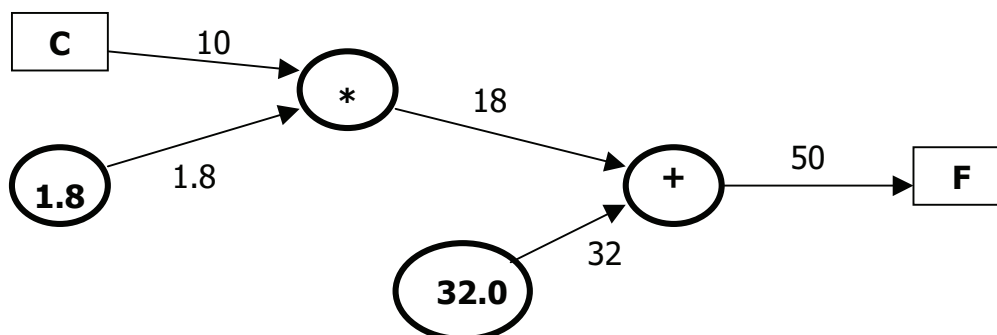
Las diferencias entre uno y otro tipo de sistema son sustanciales, tanto por el tipo de aplicaciones a las que van dirigidas como por sus características internas. Así por ejemplo en un sistema de restricciones cualitativas los dominios son dis-

cretos, pudiendo plantearse buscar enumeraciones de todas sus soluciones, tarea inviable en el caso de restricciones numéricas por la dimensión de la recta real. Por tanto es difícil obtener una visión unificada de uno y otro tipo de sistemas. Por el enfoque dado al presente libro, se hará una localización hacia las restricciones de tipo cualitativo.

02.20.01.01.05.02. Concepto de restricción.

Una restricción expresa una relación de equilibrio entre los atributos de uno o más objetos, y constituye la pieza básica de información para la comprensión y modelización del conocimiento del mundo real bajo este enfoque.

En la siguiente figura se muestra un ejemplo de restricción numérica que expresa la relación existente entre dos variables que miden la temperatura en grados Celsius y grados Fahrenheit.



Los sistemas de satisfacción de restricciones (SSR), son diseños para resolver problemas expresados en términos de restricciones. La modelización basada en restricciones proporciona un nuevo concepto de programación declarativa utilizable en una gran variedad de problemas del mundo real.

Básicamente un SSR acepta una descripción en forma de restricciones junto con los datos o conocimiento actual del estado del dominio para aplicar su estrategia general de resolución y generar los estados solución. Haciendo una analogía con otro modelo de programación declarativa como son los lenguajes de programación lógica (PROLOG), hay que distinguir entre el conocimiento que el usuario introduce en el sistema en forma de restricciones (reglas de un programa PROLOG) y la estrategia general para buscar soluciones consistentes (algoritmo de resolución/unificación en PROLOG).

El conocimiento expresado en forma de restricciones cualitativas puede ser utilizado para razonar sobre el dominio de discurso a dos niveles:

- En primer lugar se pueden eliminar aquellos valores de los dominios de las variables que no sean consistentes con las restricciones y el conocimiento actual del estado del dominio, o conocimiento a priori. Esto permite deducir el rango de valores en que se mueven los parámetros cualitativos, y por tanto obtiene como resultado una familia de posibles estados el dominio definida de forma implícita o intensiva. Este nivel de inferencia se denominará Filtrado.
- Por otro lado, es posible generar todas las Interpretaciones de las variables que intervienen en el sistema de restricciones. Una interpretación es una asignación de un único valor a cada variable de forma que todas las restricciones se verifican. En este caso lo que obtenemos pues es una enumeración de los estados consistentes del dominio a la luz del conocimiento contenido en las restricciones y del estado de las variables conocidas. Es decir en contraposición al nivel anterior, se obtienen los estados de forma explícita o extensiva. A este nivel se le denominará Generación de Interpretaciones.

No hay que olvidar que las restricciones constituyen una herramienta básica de representación del conocimiento, y como tal se utilizarán en arquitecturas modulares de Sistemas Basados en el Conocimiento (SBC). Es la estrategia de solución de problemas del SBC quien decide cuándo y cómo utilizar el conocimiento de un paquete de restricciones. De tal forma que, en función del tipo de problema a resolver y la estrategia utilizada, en ocasiones bastará con utilizar el nivel de filtrado y en otras se necesitará el nivel de la generación de interpretaciones.

Así pues todo sistema de restricciones debe constar como mínimo de un lenguaje de especificación de restricciones y un procedimiento capaz de interpretar esta especificación y de evaluar sus respuestas.

El caballo de batalla de las implementaciones de sistemas de restricciones radica precisamente en el procedimiento de evaluación de respuestas (ya sea a nivel de filtrado o de generación de interpretaciones), ya que normalmente están basados en algoritmos cuya curva de consumo de tiempo y de recursos de máquina crece de manera fuertemente exponencial con la dimensión del problema. Y esto es así por la gran combinatoria que intrínsecamente conlleva la solución de este tipo de problemas.

02.20.01.01.05.03. Aplicaciones de las restricciones.

Los sistemas de restricciones pueden ser utilizados en una gran variedad de aplicaciones informáticas de corte convencional como son la construcción de entornos de gráficos (por ejemplo una circunferencia es representada como la familia de puntos que verifican la restricción de equidistar de uno dado); programas de resolución y simulación de circuitos

eléctricos (donde las leyes de Kirchoff son expresadas como restricciones que afectan a la intensidad circulante y caída de tensión en los diferentes caminos); en bases de datos relacionales como una facilidad para la realización del subsistema de mantenimiento de la integridad (verificación automática de restricciones de integridad, chequeo de redundancias, etc.); etc. Este tipo de problemas se suelen abordar con restricciones de tipo numérico.

Igualmente las restricciones se han utilizado en muchos problemas relacionados con la inteligencia artificial. Así por ejemplo Waltz propuso una forma de interpretar dibujos lineales planos utilizando las técnicas de propagación de restricciones simbólicas. El problema en este caso es saber cuando una línea determina un borde entre dos objetos o entre dos planos del mismo objeto, de tal forma que las líneas se pueden etiquetar de tres formas líneas cóncavas, convexas y bordes. Cuando varias líneas coinciden en un vértice existen restricciones en el etiquetado de las líneas que convergen ya que físicamente, no todas las posibilidades pueden ocurrir. Estas restricciones se pueden propagar sucesivamente entre líneas adyacentes hasta obtener una comprensión completa de la escena lineal.

02.20.01.02. Representación paramétrica.

Como alternativa a la representación del conocimiento simbólica que utiliza símbolos para representar ideas y relaciones, está la representación en forma paramétrica que utiliza parámetros numéricos.

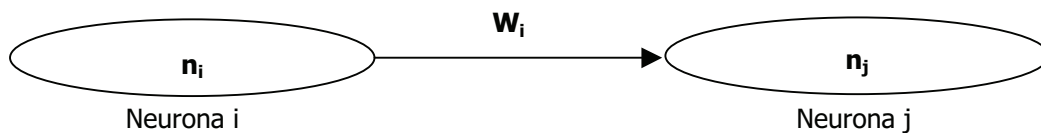
Esta representación es muy adecuada en los módulos de percepción de sistemas automáticos puesto que modelizan bien el reconocimiento de patrones como es el caso de reconocimiento de caracteres o fonemas. Sin embargo, debido a la representación numérica, no hay posibilidad de proporcionar explicaciones que justifiquen sus deducciones, al menos no de forma significativa para las personas que utilicen el sistema.

Algunos ejemplos de modelos de representación del conocimiento en forma paramétrica son el perceptrón y las redes neuronales.

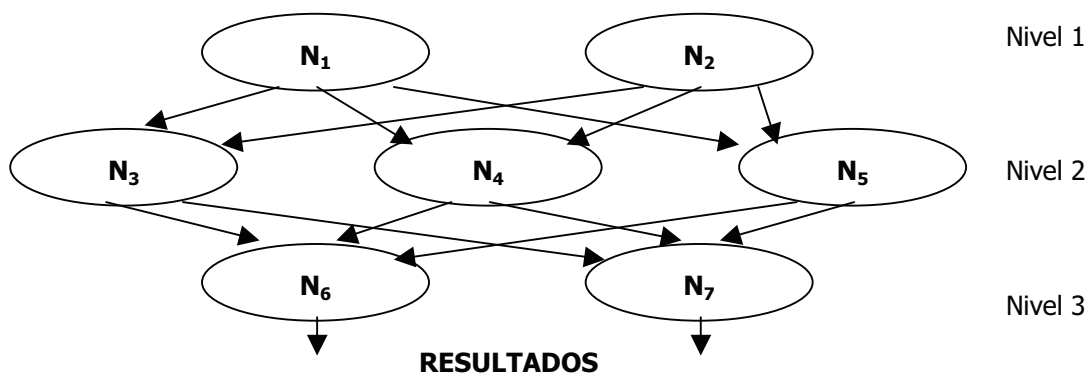
02.20.01.02.01. Redes neuronales.

La idea de la representación del conocimiento mediante redes neuronales responde a la posición filosófica estructuralista según la cual se puede simular un comportamiento inteligente si se utilizan estructuras de razonamiento semejantes a las que tienen los seres inteligentes, es decir, las neuronas y su estructura, que componen el cerebro.

La unidad básica de representación en redes neuronales es la neurona que se representa gráficamente con una circunferencia. Cada neurona dispone de una medida numérica llamada umbral de la neurona. Las neuronas pueden conectarse entre sí mediante arcos. Un arco une dos neuronas en un sentido y tiene asociado un número denominado peso que indica la fuerza de la asociación entre dos neuronas.



Las neuronas se agrupan en capas o niveles, distinguiéndose el nivel más bajo (el nivel 1) de las neuronas que se conectan con los datos, y el nivel más alto que corresponde a la conexión con los resultados. Entre dos niveles consecutivos, cada neurona del nivel inferior está conectada a todas las neuronas del nivel superior.



(ejemplo de red neuronal con 3 niveles)

La red neuronal, para que sea operativa, desde el punto de vista del razonamiento debe sufrir primero un proceso de entrenamiento que tiene como objetivo calcular los parámetros numéricos de la red. Por ejemplo, si se pretende construir una red para reconocimiento de caracteres, se debe entrenar primero para que reconozca cada uno de los caracteres. Una vez pasado el proceso de aprendizaje la red neuronal habrá reforzado algunas conexiones entre neuronas y