PRÁCTICA 2. CURSO ACADÉMICO 1995/96 REPRESENTACIÓN DEL CONOCIMIENTO Y RAZONAMIENTO

Alvaro Barreiro

Enero 1996

INTELIGENCIA ARTIFICIAL E INGENIERIA DEL CONOCIMIENTO
4 CURSO LICENCIATURA EN INFORMÁTICA

1 OBJETIVO

Construir desde sus fundamentos un sistema de representación del conocimiento y razonamiento basado en lógica de predicados. El estudiante debe percatarse de la ventaja de disponer de un sistema propio que puede crecer añadiéndole prestaciones propias de los sistemas inteligentes.

El lenguaje que se implementará será una versión restringida de la lógica de predicados, conteniendo sólo sentencias atómicas y cláusulas de Horn de la forma $A1 \wedge A2 \wedge ... \wedge An \Rightarrow C1$ donde cada uno de los antecedentes y el consecuente son sentencias atómicas. Se asume que todas la variables están cuantificadas universalmente. Por simplicidad este lenguaje no permite símbolos de funciones ni igualdad. Obsérvese también que se necesitará una distinción sintáctica entre variables y constantes.

Por tanto una tarea preliminar será la especificación en lenguaje BNF de la sintaxis del lenguaje.

2 MÉTODO

Se proponen los siguientes pasos para la realización del sistema

1. Implementar los tipos de datos y funciones de acceso para las expresiones lógicas del lenguaje. Una implementación adecuada puede ser la clásica de OPERADOR/ARGUMENTOS tanto para las sentencias atómicas (Predicado/Términos) como para las cláusulas Horn (Identificador/Consecuente y Antecedentes).

2. Implementar los tipos de datos y funciones de acceso para substituciones. Una substitución θ es una lista que vincula variables con individuales particulares. Por ejemplo $\theta = \{x/Pepe, y/Pepa\}$ Debe implementarse también una rutina de substitución que devuelve el resultado de aplicar una substitución θ a una sentencia α:

   $SUBST(\{x/Pepe, y/Pepa\}, Conoce(x, y)) = Conoce(Pepe, Pepa)$

3. Implementar una rutina de unificación. En el material suplementario se propone el seudocódigo de una rutina de unificación. Dadas dos expresiones p y q la rutina de unificación debe devolver una substitución θ que haga que p y q parezcan la misma expresión:

   $UNIFY(p, q) = \theta$ donde $SUBST(\theta, p) = SUBST(\theta, q)$

   Por ejemplo:

   $UNIFY(Conoce(Pepe, x), Conoce(Pepe, Pepa)) = \{x/Pepa\}$

   $UNIFY(Conoce(Pepe, x), Conoce(y, Pepa)) = \{x/Pepa, y/Pepe\}$

   Si no se puede encontrar tal sustitución la rutina debe devolver un fallo. Obsérvese por tanto que será necesario distinguir entre una substitución vacía y el resultado fallido de una unificación.

   En la rutina de unificación que se propone en el material suplementario se asume una implementación de las expresiones lógicas de tipo OPERADOR/ARGUMENTOS. De ahí se deduce el comportamiento de las funciones COMPOUND?, LIST?, OP?, ARGS?, FIRST?, REST?.

   Se asume también que se renombran las variables del segundo argumento en caso de conflicto de nombres.

   En principio

   $UNIFY(Conoce(Pepe, x), Conoce(x, Pepa))$ devuelve fallo pero es irrelevante que en la Base de Conocimiento tuviésemos $Conoce(x, Pepa)$ ó $Conoce(z, Pepa)$, con lo que renombrando las variables del segundo argumento

   $UNIFY(Conoce(Pepe, x), Conoce(z, Pepa)) = \{x/Pepa, z/Pepe\}$

   Obsérvese también que la parte occur-check del seudocódigo propuesto es necesaria porque no se puede unificar una expresión que contiene una variable dada con esa misma variable. Si queremos unificar $P(x)$ y $P(F(x))$ y sustituimos $x/F(x)$ nos quedará $P(F(x))$ y $P(F(F(x)))$, y realizando de nuevo la misma sustitución nunca llegaríamos a eliminar x.

   Téngase en cuenta que la rutina de unificación nos permitirá inferir con Modus Ponens Generalizado:

   Para las sentencias atómicas $p_i, p_i\prime, q$; si hay una substitución θ tal que $UNIFY(p_i\prime, pi) = \theta \forall i$, entonces

$$\frac{p_1', p_2', \ldots, p_n', (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{SUBST(\theta, q)}$$

Así por ejemplo si en nuestra base de conocimiento tuviésemos

$Conoce(Pepe, x) \Rightarrow Odia(Pepe, x)$

$Conoce(Pepe, Pepa)$

y ya que $UNIFY(Conoce(Pepe, x), Conoce(Pepe, Pepa)) = \{x/Pepa\}$ podemos inferir

$SUBST(\{x/Pepa\}, Odia(Pepe, x)) = Odia(Pepe, Pepa)$

4. Diseñar e implementar la base de conocimiento con funciones adecuadas de almacenamiento y recuperación.

5. Implementar un mecanismo de inferencia forward-chaining (encadenamiento hacia delante). En el material suplementario se propone el seudocódigo de un algoritmo de encadenamiento hacia delante.

   Se supone que al añadir una nueva sentencia atómica a la base de conocimiento se dispara el proceso de inferencia y se añaden a la base de conocimiento todas las sentencias que pueden inferirse. Si las premisas se pueden igualar de varias formas se inferirá cada conclusión correspondiente. Por otra parte si la sentencia ya estaba en la base de conocimiento el algoritmo no hace nada. Para esto último el algoritmo utiliza el concepto de renaming: una sentencia es idéntica a otra excepto en el nombre de las variables. Por ejemplo $Conoce(x, Pepe)$ y $Conoce(y, Pepe)$ son idénticas excepto en el nombre de las variables ya que sólo se diferencias en la elección de x ó y.

   El seudocódigo propuesto utiliza también la idea de composición de sustituciones: $COMPOSE(\theta_1, \theta_2)$ es la sustitución tal que

   $SUBST(COMPOSE(\theta_1, \theta_2), p) = SUBST(\theta_2, SUBST(\theta_1, p))$

   Por ejemplo: si

   $\theta_1 = \{y/Pepa\}, \theta_2 = \{x/Pepe\}, p = Conoce(x, y)$

   $SUBST(COMPOSE(\theta_1, \theta_2), p) = SUBST(\{x/Pepe\}, Conoce(x, Pepa)) = Conoce(Pepe, Pepa)$

6. Implementar un mecanismo de backward-chaining (encadenamiento hacia atrás). En el material suplementario se propone el seudocódigo de un algoritmo de encadenamiento hacia atrás.

   El algoritmo propuesto está diseñado para responder a preguntas propuestas a la base de conocimiento. Primero se chequea si la respuesta puede encontrarse directamente en la base de conocimiento. Luego se buscan todas las implicaciones cuyo consecuente se puede unificar con la pregunta y se validan las premisas de esas implicaciones también con encadenamiento hacia atrás. Si la premisa es una conjunción de elementos el algoritmo los procesa uno a uno construyendo el unificador global para toda la premisa.

En el algoritmo propuesto debe prestarse especial atención a la forma en la que se responde a la pregunta.

En cualquier caso los algoritmos del material suplementario [1] son propuestas que el estudiante puede modificar o cambiar por otros siempre que consiga los objetivos propuestos.

7. Una vez completados los pasos anteriores se estará en disposición de proporcionar un interfaz del sistema que incluya los siguientes comandos:

   - `create BCnombre`. Crea una base de conocimiento vacía con el nombre indicado
   - `empty BCnombre`. Elimina la base de conocimiento con el nombre indicado
   - `save BCnombre p`. Almacena la sentencia p en la base de conocimiento; si es el caso dispara el proceso de inferencia
   - `in BCnombre p`. Informa si la sentencia p existe en la base de conocimiento. para ello devuelve todas las sentencias de la base de conocimiento que pueden unificarse con p
   - `drop BCnombre p`. Borra la sentencia p de la base de conocimiento
   - `proof BCnombre p`. Encuentra todas las posibles respuestas (si existe alguna) a la prueba de p en la base de conocimiento
   - `list BCnombre`. Lista la base de conocimiento

3 MEMORIA

La realización de la práctica exige la redacción de una memoria organizada necesariamente con los siguientes apartados

- Índice
- Resumen de no mas de 150 palabras con los objetivos fundamentales de la práctica.
- Introducción. Breve introducción de no mas de 2 páginas en la que se debe clasificar el sistema construído en el marco de los sistemas de representación del conocimiento y razonamiento
- Material y métodos. Breve descripción del material usado (hardware y software) y consideraciones sobre portabilidad. Descripción del lenguaje de representación y de su implementación. Descripción de los procedimientos de inferencia y prueba y su implementación.

[1] Russell S., Norvig P. Artificial Intelligence: A Modern Approach, Prentice-Hall, Englewood Cliffs, NJ, 1995.

- Resultados. Demostrar el funcionamiento del sistema sobre un dominio y problemas concretos. Sígase la siguiente metodología:
  - Una vez elegido y comprendido el dominio, escribir un vocabulario de predicados y constantes de ese dominio (y funciones si se incluyesen en el lenguaje de representación). De esta manera se produce una ontología del dominio que determina que tipo de cosas existen en el dominio.
  - Describir conocimiento general del dominio y escribirlo como axiomas o sentencias lógicas acerca de los términos del dominio.
  - Describir instancias específicas del dominio escribiendo escribiendo sentencias atómicas sobre los conceptos que ya forman parte del dominio
  - Formular preguntas al sistema y obtener las respuestas del mismo.
- Discusión. En base a los estudios teóricos y resultados del apartado anterior discútase
  - Limitaciones del sistema construido
  - Ampliaciones al sistema construido
- Conclusiones. Un glosarios de no mas de 10 conclusiones bien diferenciadas en no mas de 1 página.
- Bibliografía.
- Apéndices. Todo el código de la práctica. Cualquier otra cosa que se considere de interés.