PRÁCTICA 2
REPRESENTACIÓN DEL CONOCIMIENTO E INFERENCIA
INTELIGENCIA ARTIFICIAL
TERCER CURSO ITIG-ITIS
CURSO ACADÉMICO 1997/98

Alvaro Barreiro

November 19, 1997

1 Un sistema de representación del conocimiento e inferencia

En [1] puede encontrarse la documentación y código fuente de un sistema de representación del conocimiento e inferencia. El lenguaje de representación del conocimiento es 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 supone que todas la variables están cuantificadas universalmente. Por simplicidad este lenguaje no permite símbolos de funciones ni igualdad. Obsérvese también que se necesita una distinción sintáctica entre variables y constantes.

A continuación describimos las líneas maestras para la construcción de un sistema de este tipo.[1]

1. Una tarea preliminar es la especificación en lenguaje BNF de la sintaxis del lenguaje.

2. Se implementan 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).

3. Se implementan 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:

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

4. También es necesaria una rutina de unificación. En [2] 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:

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

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

[1] esta descripción no pretende sustituir a la documentación del programa ni evitar el estudio del código

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 [2] se supone 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 supone 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

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

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

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

5. El paso siguiente es el diseño e implementación de la base de conocimiento con funciones adecuadas de almacenamiento y recuperación.

6. A continuación se implementa un mecanismo de inferencia forward-chaining (encadenamiento hacia delante). De nuevo en [2] 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

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

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

7. Para poder responder a las preguntas de un usuario el sistema debe tener un mecanismo de backward-chaining (encadenamiento hacia atrás). De nuevo en [2] 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.

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

   - Crear una base de conocimiento
   - Elimina uan base de conocimiento
   - Almacena una sentencia p en la base de conocimiento; si es el caso disparar el proceso de inferencia

- Informa si una sentencia p existe en la base de conocimiento. Para ello deben obtenerse todas las sentencias de la base de conocimiento que pueden unificarse con p
- Borrar una sentencia p de la base de conocimiento
- Encontrar todas las posibles respuestas (si existe alguna) a la prueba de una sentencia p en la base de conocimiento
- Listar la base de conocimiento

2 Contenidos de la práctica

En esta práctica se realizarán cuatro tareas:

- **tarea 1**. 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.
  - Describir conocimiento general del dominio y escribirlo como axiomas o sentencias lógicas acerca de los términos del dominio.
  - Describir ocurrenciar específicas del dominio escribiendo sentencias atómicas sobre los predicados que ya forman parte del dominio
  - Formular preguntas al sistema y obtener las respuestas del mismo.
- **tarea 2**. Estudio en detalle de la implementanción de la solución propuesta.
- **tarea 3**. Estudio de las limitaciones del sistema y posibles soluciones. Estudio de posibles ampliaciones al sistema.
- **tarea 4 (opcional)**. Implementación de alguna de las mejoras o ampliaciones de la tarea 3.

3 Defensa de la práctica

La práctica se defenderá en el laboratorio de prácticas asignado a la asignatura en el día señalado por el profesor de la misma. La defensa consistirá en la entrega de una memoria, demostración de la práctica y contestación a las preguntas del profesor. La memoria incluirá obligatoriamente los resultados de las tareas 1 y 3 y si es el caso la descripción de las implementaciones propias de la tarea 4.

References

[1] `http://www.dc.fi.udc.es/ai/~barreiro/iadocen/ialab.htm#anteriores`

[2] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, 1995.