Universidad de Buenos Aires
Facultad de Ingeniería

Teoría de Lenguajes (75.31)

Trabajo Práctico

Prolog

Integrantes

Axel Straminsky	XXXXX	axel_stram@hotmail.com
Demian Ferrerio	88443	epidemian@gmail.com
Martín Paulucci	XXXXX	martin.c.paulucci@gmail.com

1 de junio de 2011

Índice general

0.1. Hisotria

Prolog es un lenguaje de programación lógico ideado en los '70 por Colmerauer y Roussel.

La base conceptual del paradigma detrás del lenguaje fue desarrollada en forma independiente por Alfred Horn en los '50, al publicar *On sentences which are true of direct unions of algebras*, en la cual presenta un modelo lógico para el tratamiento de oraciones del lenguaje natural, donde se explican las luego denominadas "cláusulas de Horn".

Uno de los objetivo al crear este lenguaje fue el de poder crear programas que sirviesen como demostradores automáticos de teoremas.

Inicialmente era un lenguaje interpretado, hasta que en 1983 se desarrollo un compilador capaz de traducir Prolog a un conjunto de instrucciones de una máquina abstracta denominada Warren Abstract Machine (WAM), y desde entonces Prolog es semi-interpretado.

0.2. Paradigma

La programación lógica es un paradigma de programación dentro del paradigma de programación declarativa, y permite abstraerse del "cómo" y concentrarse en el "qué" a la hora de escribir programas. Particularmente Prolog implementa un tipo particular de lógica que es la lógica de primer orden.

Este paradigma tiene como característica principal la aplicación de las *reglas de la lógica* para inferir conclusiones a partir de datos. Conociendo la información y las condiciones del problema, la ejecución de un programa consiste en la búsqueda de un objetivo dentro de las declaraciones realizadas. Esta forma de tratamiento de la información permite pensar la existencia de *"programas inteligentes"* que puedan responder, no por tener en la base de datos todos los conocimientos, sino por poder inferirlos a través de la deducción.

Un programa lógico no tiene un algoritmo que indique los pasos que detallen la manera de llegar a un resultado, sino que está formado por expresiones que describen la solución (o más precisamente, la *declaran*). De esta manera, la clave para hacer un programa lógico es poder explicitar una declaración que describa correctamente la solución del problema.

0.3. Lógica de Primer Orden

La lógica de primer orden es un sistema de lógica formal usado en distintas disciplinas, como la matemática y la computación. La lógica de primer orden se distingue de la

lógica proposicional por su uso de *cuantificadores* ("para todo" o "existe").

Dentro de la lógica de primer orden, se pueden distinguir 2 partes clave: la sintaxis y la semántica. La sintaxis indica la colección de símbolos que son expresiones legales dentro de la lógica de primer orden, y la semántica determina el significado que poseen estos símbolos.

Existen 2 tipos de expresiones legales: los términos, que representan objetos, y las formulas, que representan proposiciones (expresiones que pueden ser verdaderas o falsas).

Los términos y las forumulas de la lógica de primer orden son cadenas de simbolos, que forman el alfabeto del lenguaje.

El conjunto de los términos se define inductivamente como:

- Variables: cualquier variable es un término.
- Funciones: cualquier expresión de la forma $f(t1, \cdots, tn)$ es un término.

El conjunto de las formulas se define inductivamente como:

- Simbolos de predicado: si P es un simbolo de predicado n-ario y $t1, \cdots, tn$ son terminos, entonces $P(t1, \cdots, tn)$ es una formula.
- Igualdad: si $t1$ y $t2$ son términos, entonces $t1 = t2$ es una formula.
- Negación: si P es una formula, entonces P es una formula.
- Conectivos Binarios: si P y Q son formulas, entonces $P \rightarrow Q$ es una formula.
- Cuantificadores: si P es una formula y x una variable, entonces $(paratodox)P$ y $(existealgunx)P$ son formulas.

0.4. Warren Abstract Machine (WAM)

La WAM es una máquina abstracta diseñada para ejecutar Prolog, consistente de una arquitectura de memoria y de un conjunto de instrucciones. Este diseño se convirtió en el standard de facto para la construcción de compiladores Prolog.

0.5. Aplicaciones del Paradigma Lógico

Uno de los principales campos de aplicación de este paradigma es en la Inteligencia Artificial, especialmente en todo lo relacionado con los Sistemas Expertos.

Un sistema experto es un programa que imita el comportamiento de un experto humano. Por lo tanto, contiene información (una base de conocimientos), y una herramienta para comprender las preguntas y encontrar las respuestas en la base de datos (un motor de inferencia).

También se aplica al Procesamiento Natural del Lenguaje, en donde se trata de dividir el lenguaje en partes y relaciones para tratar de comprender su significado. Otros casos en donde se utiliza este paradigma son:

- Bases de datos deductivas.
- Validación automática de programas.
- Programación Distribuida y Multiagente.
- Paralelización Automática de Programas.

0.6. Implementaciones del Paradigma Lógico

- SWI-Prolog: Soporta mulithreating.
- Mercury: Mezcla de mrogramación lógica y funcional.
- Fprolog: Añade lógica difusa.
- Prolog+: Añade clases y jerarquías de clases.
- LogTalk: añade *POO*.
- λprolog: soporta polimorfismo y programación de alto orden.

0.7. Programas lógicos

La estructura básica de un programa lógico son los términos. Un término puede ser una constante, una variable, o un término compuesto.

Las constantes denotan individuos particulares como enteros o átomos, mientras que las variables denotan a un único individuo no especificado.

Un término compuesto comprende un functor y una secuencia de uno o más términos llamados argumentos. Ejemplo:

```
padre(sergio, leandro).
```

Functor: padre
argumentos: sergio, leandro.

Los functores y las constantes empiezan con minúsculas, las variables con mayúsculas.

Los functores se caracterizan por su nombre (que es un átomo), y su aridad (número de argumentos). Las constantes, por ejemplo, son functores de aridad cero. Los functores establecen una relación entre sus argumentos, por lo cual se los llama también predicados.

Un programa lógico es un conjunto finito de clausulas. Una cláusula o regla es una sentencia lógica universalmente cuantificada, de la forma:

A ¡— $B_1, B_2, ..., B_n n >= 0(enPrologseusa : -envezde <----)$

Esta sentencia se lee declarativamente: "A se implica de conjunción de los B_i".$Asellamalacabezad$
$0, laclausulaseconocecomounhecho(factoclausulaunitaria)yseescribeA.$

Las clausulas de Prolog están basadas en las cláusulas de Horn, que son reglas del tipo Modus Ponens, es decir, si es verdad el antecedente, entonces es verdad el consecuente (en Prolog se escribe primero el consecuente y luego el/los antecedentes).

En lógica las cláusulas de Horn se escribirían así:

(mujer(A) $^p adre(B, A)) - -- > hija(A, B)$.

En Prolog no existe el concepto de asignación de variables, sino el de unificación. No hay un "estado" de las variables que se vaya modificando por sucesivas asignaciones, generalmente asociadas a posiciones de memoria, sino que las variables asumen valores al unificarse o "ligarse" con valores particulares temporalmente y se van sustituyendo durante la ejecución del programa. Cuando una variable no tiene ningún valor, se dice que está libre. A la unificación se le llama también Pattern Matching. Las variables en Prolog son similares a las variables matemáticas.

Un programa lógico contiene una base de conocimiento sobre la que se hacen consultas. La base de conocimiento está formada por hechos, que representan la información del sistema expresada como relaciones entre datos, y por reglas lógicas que permiten deducir consecuencias a partir de combinaciones entre los hechos y, en general, otras reglas. Se construye especificando la información del problema real en una base de conocimiento en un lenguaje formal y el problema se resuelve mediante un mecanismo de inferencia que actúa sobre ella. Así pues, una clave de la programación lógica es poder expresar apropiadamente todos los hechos y reglas necesarios que definen el dominio de un problema.

Internamente, existe un mecanismo, un "motor", que actúa como control de secuencia. Durante la ejecución de un programa va evaluando y combinando las reglas lógicas de la base de conocimiento para lograr los resultados esperados. La implementación del mecanismo de evaluación puede ser diferente en cada lenguaje del paradigma, pero en todos los casos debe garantizar que se agoten todas las combinaciones lógicas posibles para ofrecer el conjunto completo de respuestas alternativas posibles a cada consulta efectuada. El más difundido se denomina backtracking, que utiliza una estrategia de búsqueda primero en profundidad (Backtracking es un algoritmo para encontrar todas (o algunas) de las soluciones de un cierto problema computacional, que construye incrementalmente candidatos para la solución, mientras descartando los candidatos parciales que descubre no sirven para alcanzar la solución global).

0.8. Conclusión

Acá va la conclusión...
Así se hace una cita [1].

Bibliografía

[1] Leon Sterling & Ehud Shapiro, *The Art of Prolog: Advanced Programming Techniques*. The MIT Press, 2nd Edition, 1999.