Programación Lógica: Prolog

Docente : Oscar Alonso Ramírez
email: oalonso@uv.mx

Paradigmas de Programación Lógico

Recordando:

- Basado en lógica formal de primer orden
- Se considera un paradigma declarativo
- Un programa es un conjunto de sentencias en forma lógica
- Se expresan hechos y reglas acerca del dominio de un problema

Paradigmas de Programación Lógico

Recordando:

- Los hechos representan relaciones lógicas verdaderas. Por ejemplo: persona(carlos).
- La resolución de los cómputos la realiza un motor de inferencia, el cual no realiza compilación sino interpretación
- El programador sólo se preocupa por definir el dominio de su problema y de realizar consultas al motor de inferencia

Recordando

- Facilita la resolución de problemas relacionados con búsqueda y planificación, así como sistemas expertos
- Poco eficientes debido al costo que implican las resoluciones lógicas

Prolog

- Paradigma lógico
- Declarativo
- Resolución de teoremas mediante aplicación de reglas lógicas
- Motor de inferencia
- Interpretado
- Tipificación dinámica
- Evaluación ansiosa

Introducción

- Prolog es creado por Alain Colmerauer y Robert Kowalski en 1972

Alain Colmerauer:

- Estudió en Grenoble Institute of Technology
- Profesor en University of Aix-Marseille
- Creó su compañia Prologia

Prolog

Características:

- No se expresa cómo hacer las cosas sino qué se debe hacer
- Se trabaja a nivel de lógica de primer orden en lugar de lógica proposicional
- Los programas no se ejecutan, se le pregunta al interprete algo

Prolog

Características:

- A las variables no se les asigna un valor, éstas unifican un valor
- Muchas de las clausulas de control tradicionales no están (o al menos no son necesarias) esto incluye aunque no se limita a: if, while, for, or
- La negación tiene un significado especial (negación de mundo cerrado)
- Las estructuras de datos son diferentes (no hay arreglos por ejemplo)

Prolog

Prolog cuenta con:

- Recursividad
- Listas (no en el sentido C o Lisp)
- Predicados
- MGU (unificador más general)
- Motor de inferencia

Prolog

- Prolog es Turing completo, por lo tanto permite hacer lo mismo que con los demás lenguajes
- Sin embargo provee mayor expresividad
- La ganancia en expresividad repercute en la eficiencia

Prolog

Desventajas:

- La base de usuarios de Prolog es reducida (pocas bibliotecas third-party)
- A pesar de pretender ser un lenguaje intuitivo puede no ser amigable para algunas personas
- Puede ser difícil expresar un problema en forma de predicados
- Es difícil (o no muy limpio) realizar tareas extra-lógicas (accesos a bases de datos, manejo de archivos, hilos, sockets, etc.).

Prolog

Cuando utilizar Prolog

- Problemas que pueden expresarse de forma natural en términos de lógica de primer orden
- Problemas de planificación y búsqueda con restricciones
- Sistemas expertos
- Manejo de ontologías (web semántica)

Cuando No utilizar Prolog

- Operaciones matemáticas complejas
- GUIs
- Sistemas administrativos
- Reconocimiento de patrones

Prolog

- El modelo de ejecución en Prolog es muy diferente al de otros lenguajes
- En Prolog no se sigue el modelo de entrada y salida tradicional, la idea es definir una base de conocimientos
- Dada una base de conocimientos y una pregunta (query) el interprete computa cosas automáticamente utilizando su motor de inferencia

Lógica de primer orden

- También llamada lógica de predicados o cálculo de predicados
- Históricamente desarrollada para tratamientos matemáticos
- Tiene el poder expresivo suficiente para definir prácticamente a todas las matemáticas

Lógica de primer orden

- Para describir el mundo usualmente utilizamos oraciones declarativas:
  (i) *Toda madre ama sus hijos*
  (ii) *Marge es madre de Bart*
- Mediante un razonamiento podemos extraer conclusiones
  (iii) *Marge ama Bart*
- Este ejemplo define un *universo* de personas y algunas *relaciones* entre dichos individuos

Lógica de primer orden

- El ejemplo anterior refleja la idea principal de programación lógica
- La sintáxis de estas sentencias debe ser definida precisamente
- Las reglas deben ser formalizadas cuidadosamente
- Un sistema formal necesitará de un alfabeto

Lógica de primer orden

- El alfabeto del lenguaje de lógica de predicados consiste de:
  - variables
  - constantes
  - predicados
  - funciones
  - conectores lógicos
  - cuantificadores
  - símbolos auxiliares

Constantes

- Expresión lingüística que refiere a una entidad

Variables

- Expresión lingüística que NO refiere a una entidad, su referencia no está determinada.

Predicados

- Representan relaciones entre los individuos de un mundo
- Ejemplo: La tierra es más grande que la luna
  - masGrande(tierra, luna)
- Ejemplo: Pedro es el padre de Jośe
  - padre(pedro, jose). Otra opción: hijo(jose, pedro)
- Tambien pueden representar propiedades
- Ejemplo: Pedro es hombre
  - hombre(pedro)
- Notar que la semántica de la relación es opaca (la interpretación es abierta)

Conectores lógicos

- $\vee$ simboliza disyunción
- $\wedge$ simboliza conjunción
- $\rightarrow$ simboliza implicación
- $\leftrightarrow$ simboliza co-implicación
- $\neg$ simboliza negación

Cuantificadores

- Dos cuantificadores lógicos:
- Para todo $\forall$: indica que todos los elementos de un conjunto dado cumplen con cierta propiedad
  - $\forall X$ que sea un hombre, X es inteligente
- Existe $\exists$: indica que al menos un elemento de un conjunto dado cumple con cierta propiedad
  - $\exists X$ tal que X es menor a 0

Poniendo todo junto

- Todas las madres aman a sus hijos
- $\forall X(\forall Y((madre(X) \wedge hijo\_de(Y, X) \rightarrow ama(X, Y))))$
- Lo anterior se puede leer como: Para todo X y Y, si X es madre y Y es hijo de X, entonces X ama a Y
- Otro ejemplo: Marge tiene un hijo
- $\exists X hijo\_de(X, marge)$

Ejercicios

Formaliza las siguientes oraciones

- No todos los animales son inteligentes
- Todos los gatos tienen cola
- Ningún gato negro da mala suerte