



Universidad Autónoma de Baja California

Facultad de ingeniería, Arquitectura y diseño

Ingeniería en Software y Tecnologías Emergentes

Paradigmas de la Programación

Docente: Gallegos Mariscal José Carlos

Alumno: Diarte Salas Gilberto

Matricula: 360954

Grupo: 941

Práctica: Paradigma Lógico, Prolog.

Fecha de Entrega: 31/mayo/2024



Introducción a Prolog

Prolog es un lenguaje de programación lógica que se destaca por su capacidad para trabajar con hechos, reglas y consultas, facilitando la resolución de problemas que involucran relaciones complejas y razonamiento lógico. Creado en la década de 1970, Prolog es especialmente popular en áreas como la inteligencia artificial, la lingüística computacional y la programación de sistemas expertos. A diferencia de los lenguajes de programación imperativos, Prolog se basa en la declaración de hechos y reglas que describen relaciones entre datos, y permite a los programadores realizar consultas para inferir información basada en esos hechos y reglas. Este enfoque declarativo permite la expresión de lógica compleja de manera concisa y natural.

Desarrollo

food(burger).

food(sandwich).

food(pizza).

lunch(sandwich).

dinner(pizza).

meal(X) :- food(X).

Consulta 1: ?- food(pizza).

Resultado: true

Explicación: food(pizza) está explícitamente definido en la base de datos. Prolog encuentra el hecho y devuelve true.

Consulta 2: ?- meal(X), lunch(X).

Resultado: X = sandwich

Explicación:

- meal(X) es cierto si food(X) es cierto (debido a la regla meal(X) :- food(X)).
- Busca en la base de datos y encuentra que food(sandwich) es cierto.
- Luego verifica lunch(sandwich), que también es cierto en la base de datos.
- Así que X unifica con sandwich.



Consulta 3: ?- dinner(sandwich).

Resultado: false

Explicación: En la base de datos, solo dinner(pizza) está definido. No hay un hecho dinner(sandwich), por lo tanto, la consulta devuelve false.

studies(charlie, csc135).

studies(olivia, csc135).

studies(jack, csc131).

studies(arthur, csc134).

teaches(kirke, csc135).

teaches(collins, csc131).

teaches(collins, csc171).

teaches(juniper, csc134).

professor(X, Y) :- teaches(X, C), studies(Y, C).

Consulta 1: ?- studies(charlie, What).

Resultado: What = csc135

Explicación:

- La base de datos contiene el hecho studies(charlie, csc135).
- Prolog unifica What con csc135 y devuelve el resultado.

Consulta 2: ?- professor(kirke, C).

Resultado: C = charlie ; C = olivia

Explicación:

- Prolog busca hechos donde kirke enseña un curso.
- Encuentra teaches(kirke, csc135).
- Luego busca estudiantes que estudien csc135. Encuentra studies(charlie, csc135) y studies(olivia, csc135).
- La regla professor(X, Y) se satisface con X = kirke y Y = charlie, así como con X = kirke y Y = olivia.
- Prolog devuelve C = charlie y luego C = olivia al buscar todas las posibles soluciones.



Conclusión

Prolog ofrece una manera poderosa y única de abordar problemas mediante la programación lógica. Su uso de hechos y reglas permite a los programadores construir sistemas que pueden inferir nuevas informaciones y tomar decisiones basadas en el conocimiento existente. Aunque puede tener una curva de aprendizaje pronunciada para aquellos acostumbrados a lenguajes imperativos, su capacidad para manejar problemas de lógica y relaciones complejas lo hace invaluable en campos como la inteligencia artificial. Prolog sigue siendo una herramienta robusta para la construcción de sistemas expertos y el procesamiento de lenguajes naturales, demostrando la durabilidad y relevancia de la programación lógica en el desarrollo de software avanzado.