Lenguajes de Programación Lógica

Integrantes:

- -Jorge Denegri Castillo
- -Cristian Mellado Baca
- -Jhon Berly Taype Alccaccahua

16.1 Introducción

¿Qué es la programación lógica?

- Es un paradigma de programación diferente.
- No le dices a la computadora cómo hacer algo paso a paso.
- Le das hechos y reglas, y haces preguntas.
- El programa usa lógica simbólica para deducir respuestas.
- Es una forma de programación declarativa.

Comparación con otros paradigmas

Paradigma	¿Cómo funciona?
Imperativo (C, Java)	Paso a paso, instrucciones exactas
Funcional (Haskell)	Basado en funciones matemáticas
Lógico (Prolog)	Basado en hechos y reglas + inferencia lógica

16.2 Introducción al Cálculo de Predicados

¿Que es el cálculo de predicados?

Sistema lógico para representar conocimiento

- Expresar proposiciones (V o F)
- Representar relaciones
- Inferir nuevos hechos

padre (manuel, piero)
padre (piero, carlos)

Manuel es el padre de Piero Piero es el padre de Carlos El sistema puede deducir que:

abuelo (manuel, carlos)

Manuel es el abuelo de Carlos

Proposiciones, términos y conectores logicos

Proposiciones Atómicas

mayor_que (5, 3) tiene (gato, bigotes)

Terminos simples:

Constantes: 5, perro, jose... **Variables**: X, Y, Persona...

Términos compuestos:

ciudad (arequipa)
vive_en (carlos, ciudad (arequipa))

Proposiciones Compuestas

CONECTORES	EJEMPLOS
¬ negación	¬es_hombre(maria)
∧ conjunción	es_madre(maria, ana) ∧ es_mujer(maria)
V disyunción	Ilueve V nieva
→ implicación	es_madre(X, Y) \rightarrow es_padre_o_madre(X, Y)
↔ equivalencia	es_soltero(X) ↔ ¬casado(X)

Cuantificadores

Son símbolos que indican cuántos elementos del dominio hacen verdadera una proposición lógica

C. Universal ∀ X. P(X) "para todo valor de X, la proposición P(X) es verdadera"

C. Existencial 3 X. P(X) "existe al menos un valor de X tal que la proposición P(X) es verdadera"

Cuantificador	Símbolo	Ejemplo	Significado
Universal	∀	$\forall X. gato(X) \rightarrow mamifero(X)$	Todos los gatos son mamíferos
Existencial	3	∃X. estudia(X) ∧ vive_en (X, Arequipa)	Existe alguien que estudia y vive en Arequipa

Forma Clausular

Forma estándar y simplificada de escribir proposiciones lógicas

B1 V B2 V ... V Bn
$$\leftarrow$$
 A1 \land A2 \land ... \land Am

Si todas las A son verdaderos, entonces al menos una de las B es verdadera

Ejemplo	Significado
gusta (bob, trucha) ← gusta (bob, pescado) ∧ pescado (trucha)	Si a Bob le gusta el pescado y la trucha es un pescado, entonces a Bob le gusta la trucha
padre(luis, alan) \forall padre(luis, nicole) \leftarrow padre(alan, bob) \land madre(nicole, bob) \land abuelo(luis, bob)	Si Luis es abuelo de Bob, y Bob es hijo de Alicia y Nicole, entonces Luis es padre de Alicia o de Nicole

16.3 Cálculo de predicados y demostración de teoremas

Resolución

- -Los matemáticos usan axiomas para derivar teoremas, En programación lógica, las proposiciones en forma clausal permiten automatizar este proceso mediante un mecanismo llamado resolución.
- -Resolución es una regla de inferencia que permite derivar nuevas proposiciones a partir de un conjunto de proposiciones dadas, todas en forma clausal.

Mecánica de la resolución:

Definición:

La resolución es una regla de inferencia que combina proposiciones en forma clausal para derivar nuevas proposiciones.

Mecánica:

- Dadas dos proposiciones:
 - $\blacksquare P_1 \leftarrow P_2$
 - $Q_1 \leftarrow Q_2$
 - Si P_1 = Q_2 , se infiere: $Q_1 \leftarrow P_2$.
- o Ejemplo
 - mayor(ana, juan) ← madre(ana, juan)
 - sabio(ana, juan) ← mayor(ana, juan)
 - Resolución infiere: sabio(ana, juan) ← madre(ana, juan)

Unificación y Backtracking en la Resolución

Unificación:

- o Proceso de asignar valores a variables para que las proposiciones coincidan.
- \circ Ejemplo: Para unificar hombre(X) con hombre(juan), se asigna X = juan.
- Las asignaciones son temporales (*instanciaciones*).

• Retroceso (Backtracking):

- Si una unificación falla, el sistema retrocede para probar otras asignaciones.
- Ejemplo: Si mayor(ana, juan) no se puede probar con una instanciación, se intenta con otra (por ejemplo, cambiando X a otro valor).

Prueba por Contradicción

- Para probar un teorema, se niega (se asume falso) y se busca una inconsistencia con las proposiciones dadas.
- Método común en matemáticas, adaptado para sistemas lógicos automáticos.

Proceso:

- Se parte de hipótesis (proposiciones dadas) y la negación del teorema (meta).
- La resolución busca una contradicción, probando que el teorema es verdadero.
- Ejemplo: Probar sabio(ana, juan):
 - Base:
 - $mayor(ana, juan) \leftarrow madre(ana, juan).$
 - sabio(ana, juan) ← mayor(ana, juan).
 - madre(ana, juan).
 - Negamos: ¬sabio(ana, juan).
 - Resolución encuentra una contradicción, confirmando que sabio(ana, juan) es verdadero.

Simplificación con Cláusulas de Horn

- Cláusulas de Horn tienen una sola proposición en el lado izquierdo (con cabeza) o ninguna (sin cabeza).
- Ejemplo con cabeza: gusta(josé, atún) ← gusta(josé, pescado) Λ pescado(atún).
- Ejemplo sin cabeza: padre(josé, juan).

• Ventaja:

- Simplifican la resolución, haciéndola más eficiente para sistemas automáticos.
- La mayoría de las proposiciones pueden convertirse en cláusulas de Horn.

16.4 Una Visión General de la Programación Lógica

¿Qué es un Lenguaje Declarativo?

- No se dan instrucciones paso a paso.
- Se escriben proposiciones lógicas (no órdenes).
- Se basa en lógica simbólica.

Semántica Declarativa vs Semántica Imperativa

Característica	Declarativa	Imperativa
¿Qué indica?	Qué se quiere	Cómo se hace
¿Depende del contexto?	No	Sí
¿Requiere seguimiento previo?	No	Sí (tipos, valores, alcance)

Ejemplo de programación lógica

```
sort(old\_list, new\_list) \subset permute(old\_list, new\_list) \land sorted(new\_list)

sorted(list) \subset \forall i (1 \le i < n) \rightarrow list(i) \le list(i + 1)
```

Desafíos de la Programación Lógica

- La eficiencia es un gran problema.
- La resolución puede ser muy lenta.
- Aún falta evolucionar el paradigma lógico para problemas grandes.

Gracias