



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Paradigmas de la programación

Alumno: Vazquez Guzman Jorge Antonio

Matrícula: 372504

Maestro: Carlos Gallego

Actividad No. : Practica 4

Tema - Unidad : ProLog

Fecha de Entrega : 30 de Mayo del 2024

Ensenada Baja California a 28 de Mayo del 2024



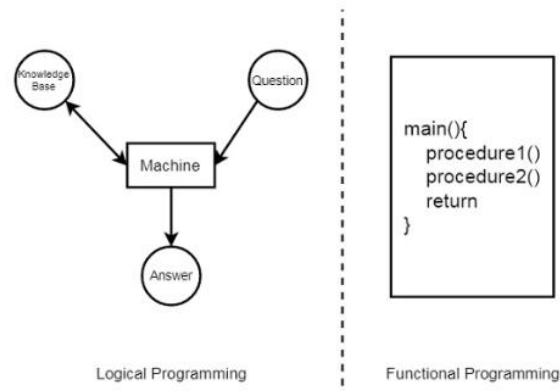
Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Desarrollo

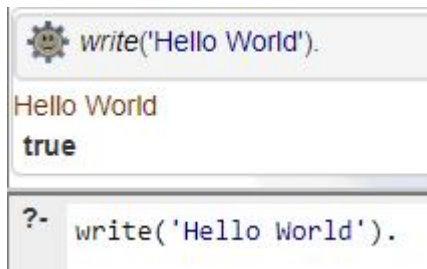
1. La diferencia entre programación lógica y programación tradicional, se pueden mostrar en la siguiente ilustración:

En la planificación funcional, necesitamos definir el proceso y determinar cómo funciona el proceso. Estos procedimientos funcionan paso a paso para resolver un problema específico basándose en un algoritmo.



2. En esta sección muestra como instalar prolog, esto no lo realice, opte por usar un compilador online.

3. Hello world, la manera de realizar consultas en prolog es con un “.” al final



4. Basics

Podemos definir la realidad como una relación clara entre las cosas y lo que esas cosas pueden hacer. Entonces la verdad es mentira. Se debe cumplir con ciertas condiciones:

Los nombres de propiedades/relaciones comienzan con letras minúsculas.

Los objetos aparecen como argumentos separados por comas entre paréntesis.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Como en el siguiente ejemplo:

```
cat(tom).  
loves_to_eat(kunal,pasta).  
of_color(hair,black).  
loves_to_play_games(nawaz).  
lazy(pratyusha).
```

Las consultas son algunas preguntas sobre las relaciones entre objetos y propiedades de los objetos. Entonces la pregunta puede ser cualquier cosa:

¿Tom es un gato? ¿A Kunal le encanta comer pasta? ¿Lili está feliz? ¿Ryan irá a jugar?

5. Relations

Hay varios tipos de relaciones, algunas de las cuales también pueden ser reglas. Una regla puede descubrir una relación incluso si la relación no está definida explícitamente como un hecho. En Prolog podemos rastrear la ejecución. Para rastrear la salida, debe ingresar al modo de rastreo escribiendo "trace".

6. Data objects

Los átomos son una variación de las constantes. Pueden ser cualquier nombre u objeto. Prolog también admite números reales, pero normalmente el caso de uso de números de punto flotante es mucho menor en los programas Prolog, porque Prolog es para cálculos simbólicos y no numéricos.

Las variables anónimas no tienen nombre. Las variables anónimas en el prólogo se escriben con un solo carácter de subrayado '_'. Y una cosa importante es que cada variable anónima individual se trata como diferente. No son iguales.

7. Operators

Los operadores de comparación se utilizan para comparar dos ecuaciones o estados.

Operador	Significado
$X > Y$	X es mayor que Y
$X < Y$	X es menor que Y
$X \geq Y$	X es mayor o igual que Y
$X \leq Y$	X es menor o igual que Y
$X = Y$	los valores de X e Y son iguales
$X \neq Y$	los valores de X e Y no son iguales



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Los operadores aritméticos se utilizan para realizar operaciones aritméticas.

Operador	Significado
+	Suma
-	Sustracción
*	Multiplicación
/	División
**	Fuerza
//	División entera
modificación	Módulo

8. Loop & Decision making

Las declaraciones de bucle se utilizan para ejecutar el bloque de código varias veces. En general, for, while y do- while son construcciones de bucle en lenguajes de programación (como Java, C, C++).

El bloque de código se ejecuta varias veces utilizando lógica de predicados recursiva. No hay bucles directos en algunos otros lenguajes, pero podemos simular bucles con algunas técnicas diferentes.

```
count_to_10(10) :- write(10),nl.  
count_to_10(X) :-  
    write(X),nl,  
    Y is X + 1,  
    count_to_10(Y).
```

Las declaraciones de decisión son declaraciones If-Then-Else. Entonces, cuando intentamos cumplir alguna condición y realizar alguna tarea, utilizamos las declaraciones de toma de decisiones.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

9. Conjunciones y disyunciones

Conjunciones:

La conjunción (lógica Y) se puede implementar utilizando el operador coma (,). Entonces, dos predicados separados por coma se unen con la declaración AND. Supongamos que tenemos un predicado, `parent(jhon, bob)`, que significa "Jhon es padre de Bob", y otro predicado, `male(jhon)`, que significa "Jhon es masculino". Entonces podemos hacer otro predicado que `padre(jhon,bob)`, que significa "Jhon es padre de Bob". Podemos definir padre predicado, cuando es padre Y es varón.

Disyunciones:

La disyunción (lógica OR) se puede implementar utilizando el operador de punto y coma (;). Entonces, dos predicados separados por punto y coma se unen con una declaración OR. Supongamos que tenemos un predicado, `padre(jhon, bob)`. Este dice que "Jhon es padre de Bob", y otro predicado, `madre(lili,bob)`, este dice que "lili es madre de bob". Si creamos otro predicado como `niño()`, esto será cierto cuando `padre(jhon, bob)` sea verdadero O `madre(lili,bob)` sea verdadero.

10. Lists

La lista es una estructura de datos simple que se usa ampliamente en programación no numérica. La lista consta de cualquier número de elementos, por ejemplo, rojo, verde, azul, blanco y oscuro. Se representará como `[rojo, verde, azul, blanco, oscuro]`. La lista de elementos irá entre corchetes. Una lista puede estar vacía o no vacía .

Operaciones	Definición
Comprobación de membresía	Durante esta operación, podemos verificar si un elemento determinado es miembro de una lista especificada o no.
Cálculo de longitud	Con esta operación podemos encontrar la longitud de una lista.
Concatenación	La concatenación es una operación que se utiliza para unir/agregar dos listas.
Eliminar objetos	Esta operación elimina el elemento especificado de una lista.
Agregar elementos	La operación de agregar agrega una lista a otra (como un elemento).
Insertar elementos	Esta operación inserta un elemento determinado en una lista.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

La concatenación de dos listas significa agregar los elementos de la segunda lista después de la primera. Agregar dos listas significa agregar dos listas o agregar una lista como un elemento.

Operaciones de reposicionamiento	Definición
Permutación	Esta operación cambiará las posiciones de los elementos de la lista y generará todos los resultados posibles.
Artículos inversos	Esta operación organiza los elementos de una lista en orden inverso.
Elementos de turno	Esta operación desplazará un elemento de una lista hacia la izquierda de forma rotatoria.
Encargar artículos	Esta operación verifica si la lista dada está ordenada o no.

Operaciones varias	Definición
Hallazgo de longitudes pares e impares	Verifica si la lista tiene un número par o impar de elementos.
Dividir	Divide una lista en dos listas, y estas listas tienen aproximadamente la misma longitud.
máx.	Recupera el elemento con valor máximo de la lista dada.
Suma	Devuelve la suma de elementos de la lista dada.
Combinar ordenar	Organiza los elementos de una lista determinada en orden (utilizando el algoritmo Merge Sort).

11. Recursion and structures

La recursividad es una técnica en la que un predicado se utiliza a sí mismo (puede ser con otros predicados) para encontrar el valor de verdad. Las estructuras son objetos de datos que contienen múltiples componentes. La coincidencia se utiliza para comprobar si dos términos dados son iguales (idénticos) o si las variables en ambos términos pueden tener los mismos objetos después de crear una instancia.

12. Backtracking

El término retroceso es bastante común en el diseño de algoritmos y en diferentes entornos de programación. En Prolog, hasta que llega al destino correcto, intenta retroceder. Cuando se encuentra el destino, se detiene.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

13. Different and Not

Aquí definiremos dos predicados: diferente y no. El predicado diferente comprobará si dos argumentos dados son iguales o no. Si son iguales, devolverá falso; de lo contrario, devolverá verdadero. El predicado not se usa para negar alguna afirmación, lo que significa que, cuando una afirmación es verdadera, entonces not(declaración) será falsa; de lo contrario, si la afirmación es falsa, entonces not(declaración) será verdadera.

14. Inputs and outputs

Para escribir la salida podemos usar el predicado write(). Este predicado toma el parámetro como entrada y escribe el contenido en la consola de forma predeterminada. write() también puede escribir en archivos. El predicado read() se utiliza para leer desde la consola. El usuario puede escribir algo en la consola, que puede tomarse como entrada y procesarlo. read() se usa generalmente para leer desde la consola, pero también se puede usar para leer desde archivos.

El tab() es un predicado adicional que se puede usar para poner algunos espacios en blanco mientras escribimos algo. Entonces toma un número como argumento e imprime esa cantidad de espacios en blanco.

Si queremos escribir en un archivo, excepto en la consola, podemos escribir el predicado tell(). Este predicado tell() toma el nombre del archivo como argumento. Si ese archivo no está presente, cree un archivo nuevo y escriba en él. Ese archivo se abrirá hasta que escribamos el comando dicho. Podemos abrir más de un archivo usando tell(). Cuando se llame, se cerrarán todos los archivos.

Cuando queremos leer desde un archivo, no desde el teclado, tenemos que cambiar el flujo de entrada actual. Entonces podemos usar el predicado see()



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

15. Built-In predicates

Predicado	Descripción
var(X)	tiene éxito si X es actualmente una variable sin instancias.
nueva(X)	tiene éxito si X no es una variable o ya se ha creado una instancia
átomo(X)	es cierto si X actualmente representa un átomo
número(X)	es cierto si X actualmente representa un número
entero(X)	es cierto si X actualmente representa un número entero
flotador(X)	es cierto si X actualmente representa un número real.
atómico(X)	es cierto si X actualmente representa un número o un átomo.
compuesto(X)	es cierto si X actualmente representa una estructura.
tierra(X)	tiene éxito si X no contiene ninguna variable no instanciada.

Cuando X no se inicializa, se mostrará verdadero; de lo contrario, se mostrará falso.

Predicados	Descripción
aleatorio(L,H,X).	Obtener valor aleatorio entre L y H
entre(L,H,X).	Obtener todos los valores entre L y H
éxito(X,Y).	Suma 1 y asignalo a X
abs(X).	Obtener el valor absoluto de X
máx(X,Y).	Obtener el valor más grande entre X e Y
mín(X,Y).	Obtener el valor más pequeño entre X e Y
redondo(X).	Redondear un valor cercano a X
truncar(X).	Convierta flotante a entero, elimine la parte fraccionaria
piso(X).	Redondear a la baja
techo(X).	Redondeo
raíz cuadrada (X).	Raíz cuadrada

16. Tree data structure

ht(Nodo,H). Esto encuentra la altura. También verifica si un nodo es hoja o no; de ser así, establece la altura H en 0; de lo contrario, encuentra de forma recursiva la altura de los hijos del nodo y les suma 1.

máx([X|R], M,A). Esto calcula el elemento máximo de la lista y un valor M. Entonces, si M es máximo, entonces devuelve M; de lo contrario, devuelve el elemento máximo de la lista que es mayor que M. Para resolver esto, si la lista dada está vacía, devuelva M como elemento máximo; de lo contrario, verifique si Head es mayor que M o no; de ser así, llame a max() usando la parte de la cola y el valor X; de lo contrario, llame a max() usando la cola y el valor M.

altura(N,H). Esto utiliza el predicado setof/3. Esto encontrará el conjunto de resultados utilizando el objetivo ht(N,Z) para la plantilla Z y lo almacenará en la variable de tipo lista llamada Set. Ahora encuentre el máximo de Set y el valor 0, almacene el resultado en H.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Conclusión

Prolog puede servir para algunas cosas específicas, sinceramente no encontré un motivo por el que usar prolog, además de ser un lenguaje desde mi punto de vista poco convencional. Si se desea encontrar más información sobre prolog se puede acceder al siguiente link:

<https://www.tutorialspoint.com/prolog/index.htm>

Y si se desea un compilador online es el siguiente:

<https://swish.swi-prolog.org/>