



Introducción a la unidad

Si hasta ahora pensabas que programar en ensamblador, C, Java, Ruby e inclusive en LISP, era todo lo que había en el mundo de los lenguajes de programación, pues no es así. Con los lenguajes funcionales se puede programar el movimiento de los discos de la torre de Hanoi sin una estructura *if* o *case*, o sin el *while* o el *for*, o sin declarar una variable ni mucho menos asignarle un valor. Todo se hace a través de funciones de programación matemáticamente declaradas. Pero también podemos prescindir de las funciones, o por lo menos, las podemos combinar con expresiones de “*cierto o falso*” para programar cualquier cosa, como una expresión de este tipo: “un caballo es un mamífero”, la cual es cierta. Parece algo inusual, pero hay un paradigma que en su definición pura no necesita de otra cosa más que de expresiones lógicas (como la de caballo) para realizar programas. Se llama paradigma lógico.

Objetivo particular de la unidad

Identificar los conceptos más importantes que dan soporte y fundamento al paradigma de programación lógico, y aplicarlos en la resolución de problemas algorítmicos propios de la automatización de la información y manipulación de datos.



Unidad V. Paradigma lógico



LO QUE SÉ

En no más de una cuartilla, contesta las siguientes preguntas de acuerdo a lo que sepas.

1. ¿Qué es la lógica matemática?
2. ¿Qué es una proposición lógica?
3. ¿Qué es un teorema?
4. ¿Qué es una tabla de verdad?
5. ¿Qué es un axioma?
6. ¿Qué es una tautología?
7. ¿Cuáles son las leyes de Morgan?

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**.

Temas de la unidad V

- 1 Definición
- 2 Hechos
- 3 Reglas
- 4 Cláusulas de Horn
- 5 Predicados
- 6 Introducción a los lenguajes lógicos
- 7 Campos de aplicación



Unidad V. Paradigma lógico



Resumen de la unidad

La programación lógica (no debe confundirse con la disciplina de la lógica computacional), se basa en el uso de la lógica como 'lenguaje de programación'.

En este tipo de programación se especifican:

- Ciertos **hechos y reglas de inferencia**.
- Un **objetivo** (meta) a probar.

La programación lógica es declarativa, es decir, se especifican hechos, reglas y objetivos, donde se pretende llegar al objetivo a partir de los hechos y las reglas.

En este sentido, un motor de inferencias trata de demostrar o comprobar que el objetivo es resultado lógico de los hechos y las reglas.

La programación lógica comprende dos paradigmas de programación:

- Programación declarativa: donde la base de desarrollo se centra en el concepto de predicado, o relación entre elementos
- Programación funcional: se basa en el concepto de función (que viene a ser una evolución de los predicados), de corte más matemático.



Unidad V. Paradigma lógico



Tema 1. Definición

Objetivo del tema

Identificar las bases de la programación lógica.

Desarrollo

La programación lógica aplica el *corpus*¹ de conocimiento sobre lógica para el diseño de lenguajes de programación y es una combinación de programación declarativa y programación funcional. Esto quiere decir que describen relaciones entre las variables en términos de funciones y reglas de inferencia (procedimiento que infiere hechos a partir de otros hechos conocidos), dejando al traductor la responsabilidad de encontrar el mejor algoritmo para encontrar el resultado buscado.

Los programas lógicos están contruidos únicamente por expresiones (enunciados) lógicas las cuales o son verdaderas o son falsas, y algoritmos específicos para implementar las reglas de inferencia. No hay propiamente expresiones interrogativas o imperativas como en los otros paradigmas.

¹ Conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación". Fuente: Diccionario de la Lengua Española en línea, Vigésima segunda edición. <http://www.rae.es>



Unidad V. Paradigma lógico



ACTIVIDAD 1

Lee el artículo Programación lógica de Gerardo Rossel (**ANEXO 1**), en un documento no mayor a una cuartilla, indicando los siguientes puntos:

1. Estructuras lógicas
2. Listas
3. Prolog y sus principios de programación

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad V. Paradigma lógico



Tema 2. Hechos

Objetivos del tema

Identificar a los hechos como aquellos elementos que establecen relaciones entre entidades.

Desarrollo

“La programación lógica trabaja con expresiones y sus relaciones que se especifican a través de reglas:

‘Para cumplirse el hecho A debe cumplirse B_1 y B_2 y B_3 y... B_n ’

Estas reglas se llaman cláusulas de Horn y constituyen un subconjunto de los predicados de primer orden de la lógica matemática. Un hecho P es un caso especial de regla para el cual no hay condiciones B_i que deban cumplirse, así se escribe como P .²

Hay dos partes diferenciadas en la programación lógica:

Partes de la programación lógica.

1. La base de hechos y las relaciones entre estos, que se conoce como base de conocimiento, cuya relación está a cargo del programador, y
2. El motor de inferencia, que es mecanismo de exploración de la base de hechos y relaciones para elaborar conclusiones.³

² Francisco Martínez, Francisco A. Martínez Gil, Gregorio Martín Quetglás: *Introducción a la programación estructurada*, Universidad de Valencia, 2003, p. 52. También disponible en formato digital, en: <http://books.google.es/books?id=-cVzTPOWf3kC>.

³ Francisco Martínez, Francisco A. Martínez Gil, Gregorio Martín Quetglás: *Introducción a la programación estructurada*, Universidad de Valencia, 2003, p. 52. También disponible en formato digital, en: <http://books.google.es/books?id=-cVzTPOWf3kC>.



Unidad V. Paradigma lógico



Veamos el siguiente ejemplo en Prolog (lenguaje lógico) de una base de conocimiento (los comentarios se ponen entre llaves):

```
Matrimonio (María, Juan). {Hecho}
Matrimonio (Lupita, José). ). {Hecho}
Matrimonio (Carmen, Paco) ). {Hecho}.
Matrimonio (Elena, Federico) ). {Hecho}.
Padres (Carmen, Lupita, José) ). {Hecho}.
Padres (Paco, María, Juan) ). {Hecho}.
Padres (Elena, María, Juan) ). {Hecho}.
Suegros (L) :- Matrimonio (L, X), Padres (X, H, I). {Regla}
```

Este programa funciona de la siguiente manera: el intérprete pregunta a Prolog y éste responde usando el motor de inferencia. El signo ?- indica que Prolog está listo para procesar una pregunta tras haber leído la base de conocimiento:

```
?- Suegros (Carmen).
    X = Paco
    H = María
    I = Juan
    no.
```

Prolog intenta encontrar en los hechos las variables que satisfagan la regla, esto es, que hagan la regla verdadera. Se van probando en el orden en que están escritos en la base de conocimiento. Explorará todas las combinaciones en busca de soluciones. La palabra *no* del final indica que no encontró más soluciones y despliega en pantalla todas las variables que verifican la regla.



Unidad V. Paradigma lógico



ACTIVIDAD 1

Lee Introducción al lenguaje de programación lógica Prolog de Kathryn M. Jones Pérez y Gustavo A. Yong Morales (**ANEXO 2**). Después genera un documento, no mayor a una cuartilla indicando en tus propias palabras los siguientes conceptos:

- Hechos
- Reglas
- Variables
- Consultas

Indica además, dos ejemplos para Hechos, Reglas y variables en Prolog.

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**



Unidad V. Paradigma lógico



Autoevaluación

De las siguientes frases identifica cuáles son reglas y cuáles son hechos. Relaciona las columnas arrastrando cada enunciado a su posición correcta. Al final obtendrás tu calificación de manera automática.

Reglas	Hechos

- Si un animal es ungulado y tiene rayas negras, entonces es una cebra. (regla)
- Si un animal rumia y es mamífero, entonces es ungulado.(regla)
- El animal es mamífero. (hecho)
- El animal tiene pezuñas. (hechos)
- Si un animal es mamífero y tiene pezuñas, entonces es ungulado. (regla)
- El animal tiene rayas negras. (hecho)

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad V. Paradigma lógico



Tema 3. Reglas de inferencia

Objetivos del tema

Identificar las reglas para la declaración de inferencia.

Desarrollo

Para explicar qué son las reglas de inferencia, debemos saber que los enunciados lógicos de un programa lógico están compuestos de los siguientes elementos o tokens:

Elementos o tokens de los enunciados lógicos.

- **Constantes:**

Son por lo general números o nombres. También se les llama átomos, ya que no se pueden dividir en partes más pequeñas. 1 es un ejemplo de constante.

- **Predicados:**

Son los nombres de funciones que son verdaderas o falsas. Pueden tomar varios argumentos. La función natural (2) es un ejemplo de predicado.

- **Funciones:**

Se distinguen las funciones que son verdaderas o falsas (predicados) del resto de las funciones. La función sucesor(x) es un ejemplo de función.

- **Variables:**

Representan cantidades todavía no especificadas. Ejemplo: x es una variable.

- **Conectores:**

Incluyen las operaciones *y*, *o* y *no*, además de las operaciones sobre datos booleanos. Otros conectores son la implicación \rightarrow y la equivalencia \leftrightarrow .

- **Cuantificadores:**

Son operaciones que introducen variables. Existen cuantificadores universales y existenciales.



Unidad V. Paradigma lógico



- **Símbolos de puntuación:**

Incluyen paréntesis izquierdo y derecho, la coma y el punto. Los paréntesis sirven para encerrar argumentos y para agrupar operaciones.

Las reglas de inferencia son las formas de derivar o de probar nuevos enunciados a partir de un conjunto dado de enunciados. Estas reglas nos permiten construir el conjunto de todos los enunciados que pueden derivarse o comprobarse a partir de un conjunto dado de enunciados. Se trata de enunciados que son verdaderos siempre que los enunciados originales sean verdaderos. Ejemplo:

Los siguientes enunciados son lógicos:

Un caballo es un mamífero.

Un ser humano es mamífero.

Los mamíferos tienen cuatro patas y ningún brazo, o dos pies y dos brazos.

Un caballo no tiene brazos.

Un ser humano tiene brazos.

Un ser humano no tiene patas.

Los enunciados derivados de los anteriores pueden ser:

patas(caballo,4).

brazos(caballo,0).

pies(caballo,0).

patas(serHumano,0).

brazos(serHumano,2).

pies(serHumano,2).

Los primeros enunciados lógicos se le llama **axiomas**, y los enunciados derivados se les llama teoremas.



ACTIVIDAD 1

Revisa el contenido de las siguientes páginas:

Capítulo 2: Inferencia Lógica, Falacias y Errores en el Razonamiento, (Primer curso de lógica matemática <http://camediga.blogspot.com/2008/04/primer-curso-de-lgica-matemtica.html>)

Reglas De Inferencia

(<http://www.mitecnologico.com/Main/ReglasDeInferencia>)

Presenta en un documento no mayor a dos cuartilla, los siguientes puntos:

- Tipos de reglas de Inferencia.
- Tabla de resultados de inferencias
 - $P \wedge Q \rightarrow R$
 - $(P \wedge Q \rightarrow R) \wedge (P \wedge Q)$
 - $[(P \wedge Q \rightarrow R) \wedge (P \wedge Q)] \rightarrow R$
- 4 ejemplos del uso de la tabla anterior.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



Unidad V. Paradigma lógico



Autoevaluación

Con base en las siguientes declaraciones indica si las expresiones son verdaderas o falsas. Al final obtendrás tu calificación de manera automática.

Declaraciones

- es Papa (juan, fidel).
- es Papa (juan, mario).
- esPapá (fidel, julio).
- esPapá (fidel, maría).
- esPapá (mario, marcos).
- esPapá (mario, pedro).
- esPapá (mario, luis).
- esMamá (lucy, luis).
- esMamá (lucy, marcos).
- esMamá (susana, julio).
- esMamá (susana, maria).

	Verdadera	Falsa
Expresiones		
1. Julio es hijo de Juan.	()	()
2. Susana es mamá de María.	()	()
3. Luis es hijo de Juan.	()	()
4. Mario es papa de Pedro.	()	()
5. Lucy es mamá de Luis.	()	()



Unidad V. Paradigma lógico



Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad V. Paradigma lógico



Tema 4. Cláusulas de Horn⁴

Objetivo del tema

Analizar las Cláusulas de Horn para la declaración de axiomas.

Desarrollo

Una cláusula de Horn es un enunciado

$$a_1 \text{ y } a_2 \text{ y } a_3 \dots \text{ y } a_n \rightarrow b$$

donde a los a_i sólo se les permite ser enunciados simples sin conectores. No existen conectores “o” ni cuantificadores. En el enunciado se dice que a_1 hasta a_n ; esto implica que b es verdadero si todos los a_i son verdaderos. A b se le llama cabeza de la cláusula, y a a_1, \dots, a_n el cuerpo de la cláusula. Se utilizan para expresar la mayoría, pero no la totalidad, de los enunciados lógicos. Ejemplo:

Veamos cómo queda la descripción lógica para el algoritmo del MCD entre dos números enteros positivos u y v :

El mcd de u y de 0 es u .

El mcd de u y de v , si v no es 0, es el mismo que el mcd de v y del residuo de dividir u entre v .

Pasando lo anterior a predicados nos queda:

Para toda u , $\text{mcd}(u,0,u)$.

Para toda u , para todas v , para todas w , no cero(v) y $\text{mcd}(v,u \text{ mod } v,w) \rightarrow \text{mcd}(u,v,w)$.

⁴ Ver, Kenneth C. Louden, *Programming Languages: Principles and Practice*, Second Edition, Brooks Cole, 2003.



Unidad V. Paradigma lógico



Para traducir estos enunciados a cláusulas Horn, sólo hay que eliminar los cuantificadores:

$\text{mcd}(u,0,u)$.

$\text{no cero}(v)$ y $\text{mcd}(v,u, \text{mod } v,w) \rightarrow \text{mcd}(u,v,w)$.

ACTIVIDAD 1

Analiza el contenido de los Capítulos 1, 2 y 3 del documento Lógica Proposicional para Informática

(http://www.fismat.iteso.mx/pag_def/estructura/matematicas/asignaturas/logica/apoyos/proposicional3.pdf)

Elabora un documento no mayor a tres cuartillas, indicando en sus propias palabras los siguientes conceptos:

- Tabla de conectivas: Negación, Conjunción, Disyunción, Condicional y Equivalencia. (Incluir ejemplos de la vida real, para cada uno de ellos).
- Significado de los siguientes símbolos con un ejemplo ilustrativo:
 - \neg
 - \square
 - \square
 - \rightarrow
 - \leftrightarrow
 - \equiv
- Expresa en tus propias palabras el concepto de Razonamiento.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma



Unidad V. Paradigma lógico



ACTIVIDAD 2

Analiza los Temas 4.4.1 y 4.4.3 del Capítulo 4 - Técnicas Semánticas de Estudio de Validez Proposicional.

http://www.fismat.iteso.mx/pag_def/estructura/maticas/asignaturas/logica/apoyos/proposicional3.pdf

Posteriormente elabora un documento, no mayor a una cuartilla, indicando las características de las cláusulas de Horn.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma



Unidad V. Paradigma lógico



Autoevaluación

De las siguientes expresiones indique cuales son clausulas de Horn y cuáles no. Escribe sobre la línea la expresión si es o no es según sea el caso. Al final obtendrás tu calificación de manera automática.

1. $\neg p \vee \neg q \vee r$ _____
2. $\neg p \vee \neg q \vee \neg r$ _____
3. $\neg p \vee q \vee r$ _____
4. r _____
5. $\neg p \vee \neg q$ _____

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad V. Paradigma lógico



Tema 5. Predicados⁵

Objetivo del tema

Identificar el uso de los predicados en el contexto de la programación lógica.

Desarrollo

Como ya se mencionó en el tema de reglas de inferencia, los predicados son los nombres de funciones que son verdaderas o falsas. Pueden tomar varios argumentos.

En el cálculo de predicados, los argumentos a los predicados y funciones deben ser combinaciones de variables, constantes y funciones (a éstos se les llama términos). Los términos no pueden tener predicados, cuantificadores o conectores. Las funciones de la programación funcional no son otra cosa más que la evolución de los predicados de la programación lógica.

ACTIVIDAD 1

Lee el tema 6.1 Predicados y Cálculo de Predicados (del capítulo 6) <http://www.fceia.unr.edu.ar/~iilcc/libro/PDF/capitulo06.pdf>, de los apuntes de la clase Introducción a la Informática, de Silvia Bianchi y Federico Guimpel. Posteriormente, elaborar un ensayo sobre los elementos del Cálculo de predicados. El documento no deberá ser mayor a una cuartilla.

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**

⁵ Ver, Kenneth C. Louden, *Programming Languages: Principles and Practice*, Second Edition, Brooks Cole, 2003.



Unidad V. Paradigma lógico



Autoevaluación

Relaciona las columnas arrastrando sobre las líneas el inciso correspondiente. Al final obtendrás tu calificación de manera automática.

a) Marco era un hombre.	__1. Pompeyano(Marco)
b) Marco era pompeyano.	__2. Para todo x y y tenemos que: (Persona(x) \square Gobernante(y) \square IntentaAsesinar(x,y) \rightarrow \neg Leal(x, y)
c) Todos los pompeyanos eran romanos.	__3. Para todo x Existe y tal que Leal(x, y)
d) César fue un gobernante.	__4. Gobernante(César)
e) Todos los romanos o eran leales a César o lo odiaban.	__5. Romano(x) \rightarrow (Leal(x, César) \square Odia(x, César))
f) Todo el mundo es leal a alguien.	__6.IntentaAsesinar(Marco, César)
g) La gente sólo intenta a asesinar los gobernantes a los que no es leal.	__7. Hombre(Marco)
h) Marco intentó asesinar a César.	__8. Pompeyano(x) \rightarrow Romano(x)

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad V. Paradigma lógico



Tema 6. Introducción a los lenguajes lógicos

Objetivo del tema

Reconocer al lenguaje PROLOG como el principal representante de la programación lógica.

Desarrollo

La mayoría de los lenguajes de programación lógica están basados en la lógica de primer orden o cálculo de predicados de primer orden. Esta lógica sirve para cuantificar variables individuales, por ejemplo, “*las plantas son seres vivos*”, en donde “*plantas*” está cuantificando universalmente a todas las plantas.

El lenguaje de programación lógica por excelencia es Prolog, cuyo nombre viene del francés *Programation et Logique*, y el cual es un lenguaje de programación interpretado y cuenta con muchas variantes. Desde el punto de vista del programador, la ventaja principal de Prolog es su facilidad para escribir programas claramente legibles de forma rápida y con pocos errores.



Unidad V. Paradigma lógico



ACTIVIDAD 1

Del libro Lenguaje de Programación PROLOG, leer el tema 1. Fundamentos de PROLOG <http://mural.uv.es/mijuanlo/PracticasPROLOG.pdf> y generar un documento de no más de dos cuartillas indicando la sintaxis y dos ejemplo de aplicación de los siguientes elementos:

- Hechos
- Preguntas
- Reglas

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

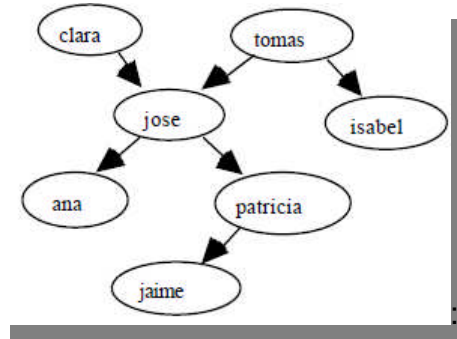


Unidad V. Paradigma lógico



Autoevaluación

Observa el diagrama e indica cuáles de las siguientes expresiones en PROLOG, corresponden a éste. Escribe sobre la línea frente a cada enunciado la palabra corresponde o no corresponde, según sea el caso. Al final obtendrás tu calificación de manera automática.



Expresiones

- 1) progenitor(clara,jose). _____
- 2) progenitor(jose,clara). _____
- 3) progenitor(jose, patricia). _____
- 4) progenitor(patricia, jose) . _____
- 5) progenitor(jaime,patricia). _____

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad V. Paradigma lógico



Tema 7. Campos de aplicación

Objetivo del tema

Identificar los campos de aplicación de la programación lógica.

Desarrollo

La programación lógica tiene su hábitat natural en el desarrollo de aplicaciones de inteligencia artificial, como los sistemas expertos, la demostración automática de teoremas, reconocimiento del lenguaje natural, y en menor medida en aplicaciones de propósito general.

Un campo muy socorrido para la programación lógica son los diccionarios de palabras de los celulares, los cuales incorporan un diccionario y un algoritmo que va restringiendo las posibles palabras que puedan ser resultado de la presión de una serie de teclas. Para ello se definen dos predicados que brindan la información necesaria del teléfono celular:

1. Predicado que define el mapeo entre las teclas de celular y los caracteres:

```
teclado([ (1, [1]), (2, [2,a,b,c]), (3, [3,d,e,f]),  
(4, [4,g,h,i]), (5, [5,j,k,l]), (6, [6,m,o,n]),  
(7, [7,p,q,r,s]), (8, [8,t,u,v]), (9, [9,w,x,y,z]),  
(0, [0]), (*, [-]), (#, [#])  
]
```

2. Predicado que define un diccionario de palabras:

```
diccionario([ [1,a], [l,a], [c,a,s,a], [a], [d,e], [r,e,j,a], [t,i,e,n,e],  
[c,a,s,a,m,i,e,n,t,o], [d,e,l], [a,n,t,e,s]  
])
```




ACTIVIDAD 1

Busca en la Web y descarga algunos de los siguientes ambientes de programación en Prolog⁶ Posteriormente implementa al menos 5 algoritmos de los que se listan a continuación, y envía los códigos fuentes para su revisión.

ADA PD Prolog es un intérprete antiguo (1986) y lento para MS-DOS de Automata Design Associates.

Arity/Prolog32 es un completo entorno de programación en Prolog para Windows de 32 bits que incluye un verdadero compilador, editor, depurador, intérprete y ayuda. Para descargarlo hay que rellenar un formulario.

B-Prolog es un completo sistema CLP (Constraint Logic Programming) que ejecuta programas Prolog y CLP(FD). Hay versiones para Windows y varios Unix, incluyendo Linux. Se distribuye también el código fuente.

Ciao es un entorno de programación multi-paradigma que ofrece un completo sistema Prolog de acuerdo con el estándar ISO, permitiendo tanto restricciones como extensiones al lenguaje. Se distribuye el código fuente y binarios para Windows. Aunque las páginas están en inglés, los autores pertenecen a un grupo de investigación de la Universidad Politécnica de Madrid.

CU-Prolog es un lenguaje CLP experimental adecuado para el procesamiento de lenguajes naturales. Hay versiones Unix, MS-DOS (djcup) y Macintosh (MacCup).

⁶ Lista extraída de <http://www.rodoval.com/paginalen.php?len=Prolog> (04/03/09).



Unidad V. Paradigma lógico



DGKS es un intérprete de Prolog escrito en Java. Se puede descargar y ejecutar localmente o directamente a través de la página web.

ESL Prolog-2 (PD Version) es una versión limitada pero de buen rendimiento de un intérprete Prolog para MS-DOS (de 1991).

GNU Prolog es un compilador conforme al estándar ISO con algunas extensiones. Incluye también un intérprete interactivo con depurador. Ha sido portado a muchas versiones de Unix (incluyendo Linux) y a Windows (con Cygwin o MSVC++). Se distribuyen ejecutables para Linux y Windows y el código fuente.

JIProlog es un intérprete de Prolog compatible con Java que permite añadir la potencia de Prolog a cualquier aplicación o applet Java.

jProlog es un intérprete Prolog escrito en Java.

Kernel Prolog es un intérprete Prolog escrito en Java con un sistema innovador de built-ins basado en la extensión Fluents.

K-Prolog es un compilador de Prolog para Windows, Linux y otros Unix.

LPA PROLOG Professional es un compilador antiguo de Prolog para MS-DOS de 16 bits, ofrecido ahora gratuitamente para uso personal.

NU-Prolog se distribuye sólo para la enseñanza y la investigación. Sólo código fuente.

Open Prolog es una implementación de Prolog para Macintosh (MacOS 7.5.5 y posteriores). Es postcardware, o sea, sus usuarios deben enviarle al autor



Unidad V. Paradigma lógico



(Mike Brady) una postal.

Qu-Prolog (de la Universidad de Queensland) es un Prolog extendido, diseñado principalmente como lenguaje de prototipos y como lenguaje táctico para la demostración interactiva de teoremas. Se distribuye gratuitamente sólo para uso no comercial. Sólo para Unix y Linux.

Reform Prolog es un proyecto abandonado sin terminar, pero se distribuye una versión beta (código fuente).

Strawberry Prolog es un compilador para Windows y Unix/Linux. Sólo la Light Edition es gratuita.

SWI-Prolog es un compilador dirigido principalmente a la investigación y la educación. Se distribuyen fuentes y binarios para Linux, Windows y MacOS X (Darwin).

Visual Prolog es un entorno de programación en Prolog para Windows sucesor de Turbo Prolog y PDC Prolog. La Personal Edition es gratuita, pero sólo debe usarse para su aprendizaje y no se debe distribuir los ejecutables generados, que mostrarán al principio un letrero.

W-Prolog es un intérprete de un lenguaje tipo Prolog implementado en Java. Es muy portable y puede ejecutarse como aplicación o como applet.

XSB es una extensión de Prolog para incluir una implementación eficiente de memorización y una implementación inicial de HiLog. Se puede descargar el código fuente y ejecutables para Windows.



Unidad V. Paradigma lógico



YAP es un compilador de Prolog de alto rendimiento, desarrollado en la Universidad de Oporto. Se distribuyen las fuentes y binarios para Linux, SPARC/Solaris y Windows.

Algoritmos

- Ordenamiento burbuja (bubble sort)
- Ordenamiento por inserción (insert sort)
- Ordenamiento de mezcla (merge sort)
- Ordenamiento rápido (quick sort)
- Ordenamiento ingenuo (naive sort)

Realiza tu actividad y guárdala en tu computadora, una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



Unidad V. Paradigma lógico



Cuestionario

Responde las siguientes preguntas.

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.

1. ¿De qué están constituidos los programas lógicos?
2. ¿Qué es una base de conocimiento?
3. ¿Qué es un motor de inferencia?
4. ¿Qué son las reglas de inferencia?
5. ¿Qué es una cláusula de Horn?
6. ¿Qué es un predicado?



Unidad V. Paradigma lógico



Examen de Autoevaluación

Elige la opción que conteste correctamente cada una de las siguientes oraciones. Al final obtendrás tu calificación de manera automática.

1. Paradigma que aplica el *corpus* de conocimiento sobre lógica para el diseño de lenguajes de programación y es una combinación de programación declarativa y programación funcional:

- a) Paradigma funcional
- b) Paradigma imperativo
- c) Paradigma lógico
- d) Paradigma orientado a objetos

2. Son formas de derivar o de probar nuevos enunciados a partir de un conjunto dado de enunciados:

- a) Expresiones lógicas
- b) Reglas de inferencia
- c) Predicados
- d) Hechos

3. Es el mecanismo de exploración de la base de hechos y relaciones para elaborar conclusiones:

- a) Motor de inferencia
- b) Cuantificadores
- c) Conectores
- d) Predicados



Unidad V. Paradigma lógico



4. La base de hechos y las relaciones entre éstos se conoce como:

- a) Reglas de inferencia
- b) Motor de inferencia
- c) Base de conocimiento
- d) Cuantificadores

5. Son por lo general números o nombres. También se les llama átomos, ya que no se pueden dividir en partes más pequeñas:

- a) Predicados
- b) Reglas de Horn
- c) Conectores
- d) Constantes
- e)

6. Son los nombres de funciones que son verdaderas o falsas. Pueden tomar varios argumentos:

- a) Predicados
- b) Conectores
- c) Cláusulas de Horn
- d) Variables

7. Son operaciones que introducen variables. Existen universales y existenciales:

- a) Predicados
- b) Cuantificadores
- c) Constantes
- d) Conectores



Unidad V. Paradigma lógico



8. Incluyen paréntesis izquierdo y derecho, la coma y el punto. Los paréntesis sirven para encerrar argumentos y para agrupar operaciones:

- a) Reglas de inferencia
- b) Constantes
- c) Símbolos de puntuación
- d) Funciones

9. Incluyen las operaciones *y*, *o* y *no*, además de las operaciones sobre datos booleanos. Otros son la implicación \rightarrow y la equivalencia \leftrightarrow :

- a) Conectores
- b) Símbolos de puntuación
- c) Cláusulas de Horn
- d) Reglas de inferencia

10. Representan cantidades todavía no especificadas:

- a) Constantes
- b) Variables
- c) Funciones
- d) Reglas de Horn



Unidad V. Paradigma lógico



Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción

LO QUE APRENDÍ

Elabora un mapa conceptual de la programación lógica, haciendo hincapié en los siguientes puntos:

- Definiciones
- Principios
- Aplicación
- Lenguajes
- Procesamiento lógico

Realiza tu actividad en un documento en Word, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza tu archivo donde lo guardaste, selecciónalo y presiona **Subir este archivo** para guardarlo en la plataforma.



Unidad V. Paradigma lógico



Glosario de la unidad

Constantes

Son por lo general números o nombres. También se les llama átomos, ya que no se pueden dividir en partes más pequeñas. 1 es un ejemplo de constante.

Predicados

Son los nombres de funciones que son verdaderas o falsas. Pueden tomar varios argumentos. La función natural(2), es un ejemplo de predicado.

Funciones

Se distinguen las funciones que son verdaderas o falsas (predicados) del resto de las funciones. La función sucesor(x) es un ejemplo de función.

Variables

Representan cantidades todavía no especificadas. Ejemplo: x es una variable.

Conectores

Incluyen las operaciones *y*, *o* y *no*, además de las operaciones sobre datos booleanos. Otros conectores son la implicación \rightarrow y la equivalencia \leftrightarrow .

Cuantificadores

Son operaciones que introducen variables. Existen cuantificadores universales y existenciales.

Símbolos de puntuación

Incluyen paréntesis izquierdo y derecho, la coma y el punto. Los paréntesis sirven para encerrar argumentos y para agrupar operaciones.

Argumento lógico

Colección finita de afirmaciones (proposiciones) dividida en premisas y conclusión. Las premisas y la conclusión debe ser susceptibles de recibir un valor de verdad. El argumento lógico puede ser correcto o incorrecto.



Unidad V. Paradigma lógico



Cláusula

Es una fórmula disyunción de cualquier número de fórmulas atómicas afirmadas o negadas. ($n=0$ cláusula vacía que no puede satisfacerse, $A \vee \neg A$ es la forma de cláusula que siempre es verdad)

Cláusula de Horn

Es aquella en la que aparece uno y sólo un átomo afirmado y cualquier número de átomos negados. Existen tres tipos de cláusulas:

- Un átomo simple (Hecho) : P
- Una implicación (Regla) cuyo antecedente consiste en una conjunción de literales positivos y cuyo consecuente es un solo literal positivo: $P \wedge Q \supset R$
- Un conjunto de literales negativos (Objetivo) escritos en forma de implicación, con un antecedente compuesto por la conjunción de literales positivos, y un consecuente vacío: $P \wedge Q$

Cláusula negativa

Es aquélla en la que aparecen sólo átomos negados.

Cláusula afirmativa

Es aquélla en la que aparecen sólo átomos afirmados.

Axioma

Es una "verdad evidente" que no requiere demostración, pues se justifica a sí misma, y sobre la cual se construye el resto de conocimientos por medio de la deducción. El axioma gira siempre sobre sí mismo, mientras los postulados y conclusiones posteriores se deducen de éste.



Unidad V. Paradigma lógico



Teorema

Es una afirmación que puede ser demostrada como verdadera dentro de un marco lógico.

Leyes de Morgan

Las leyes de De Morgan son una parte de la Lógica proposicional y analítica, y fue creada por Augustus De Morgan (Madura, 1806 - Londres, 1871). Las leyes de De Morgan declaran que la suma de n variables globalmente negadas (o invertidas) es igual al producto de las n variables negadas individualmente; y que inversamente, el producto de n variables globalmente negadas es igual a la suma de las n variables negadas individualmente.

Proposición

Es una declaración la cual puede ser verdadera o falsa, por ejemplo: $5 > 4$, $2+2=5$, "Pedro comió a las 3", "Me gusta la sopa".



Unidad V. Paradigma lógico



MESOGRAFÍA

Bibliografía básica


Bibliografía complementaria

Sitios electrónicos



Unidad V. Paradigma lógico



(ANEXO 1) Descarga el archivo de la plataforma **Programación lógica** de Gerardo Rossel  **Programación lógica**.



Unidad V. Paradigma lógico



(ANEXO 2) Descarga el archivo de la plataforma PDF  **Kathryn y Gustavo**