UNIVERSIDAD NACIONAL DE LUJÁN
Centro Regional CHIVILCOY
Licenciatura en Sistemas de Información
Asignatura: MATEMÁTICA DISCRETA II Cuatrimestre – Año 2020

Unidad 1: LÓGICA PROPOSICIONAL – LÓGICA DE PREDICADOS

TEORÍA

CONCEPTUALIZACION DE PARADIGMA LOGICO: HISTORIA

La visión Procedural afirma que el conocimiento humano está caracterizado básicamente por "know how"; esto significa, ejecutar alguna secuencia de operaciones en la resolución de problemas. Los Lenguajes de Programación Convencional deben verse como procedurales en tal sentido. En los niveles más bajos no existen diferencias claras entre datos y programas, en forma independiente de algunos contextos específicos de ejecución.
El programador está en control sobre qué operaciones aplicar y cómo interactúan con otras componentes interiores y exteriores al procedimiento: estas cuestiones constituyen la potencia de la representación.

El punto de vista Declarativo permite concebir al Intérprete como el único Proceso Activo en el Sistema, tal que las componentes restantes se traducen en entidades pasivas. Las Representaciones Declarativas están basadas usualmente en un Modelo Axiomático de la Lógica Matemática e intentan plasmar claras distinciones entre Axiomas, Reglas de Deducción e Inferencia Lógica, estableciendo como característica central en el marco del Paradigma, a la Demostración. Las Demostraciones se deben definir como la Reducción de *fbf* (Fórmulas Bien Formadas) a Axiomas, empleando únicamente secuencias válidas de transformación; en consecuencia, los Axiomas son las últimas primitivas para definir "un mundo" en cada sistema.

La Lógica arrastra una Historia de gran valor en la Ciencia y en la Cultura. En el Año 1987, MALPAS sintetizó la siguiente visión sobre el desarrollo del Silogismo Aristotélico; así, las leyes básicas sobre las que se funda esta Lógica Filosófica la constituyen los siguientes Principios:

1- Identidad (las cosas son siempre iguales a si mismas);

2- Contradicción (nada puede ser simultáneamente verdadero y falso);

3- Principio del Tercero Excluido (cualquier cosa tiene que ser verdadera o falsa).

La Inferencia Lógica se encarga del proceso de obtener conclusiones desde las premisas, donde cada paso en un argumento válido puede ser definido, constituyendo un silogismo.
Los Argumentos son <u>Consistentes</u> si es imposible derivar una contradicción desde un conjunto de axiomas. Un Conjunto de Axiomas se considera <u>Completo</u> si algún argumento verdadero también puede demostrarse.
En el Siglo XIX, DE MORGAN y BOOLE efectuaron severas críticas a la Lógica Aristotélica, debido a que fue formulada en términos de Lenguaje Natural con su inherente imprecisión; en consecuencia, se avinieron a diseñar una notación más formal y rigurosa. La Historia Moderna de la Lógica Simbólica se inicia con Frege's *Begriffsschrift*, con la estructura de un Cálculo Simbólico para la Representación y Manipulación de Formas Lógicas, sobre las cuales se conceptualizó posteriormente el fundamento del Cálculo de Predicados. La Lógica de Predicados presenta mayor potencia expresiva que la Lógica Proposicional; pues, en tanto la Lógica Proposicional no permite variables de algunas clases, la Lógica de Predicados permite: Fórmulas Atómicas, Composición de Predicados Simbólicos y algunos Argumentos (Términos); cada uno de los predicados puede definir relaciones arbitrarias entre argumentos, representándolos a través de constantes, variables o funciones.

CONCEPTUALIZACION DE PARADIGMA LOGICO: HISTORIA

Una *fbf* (Fórmula Bien Formada) resulta desde la combinación de fórmulas atómicas con conectivos lógicos (no, y, o, implicación, equivalencia, existe y para todo). Las Variables deben ser cuantificadas (mediante los cuantificadores existencial o universal). La Lógica de Predicados de Primer Orden solamente permite variables en las cuales se representan objetos, en tanto que la Lógica de Predicados de Segundo Orden también permite a los predicados propios servir como valor de las variables.

La Lógica de Predicados ofrece un conjunto de técnicas para demostrar si una Fórmula Bien Formada puede ser derivada como una consecuencia de un conjunto de axiomas y parece un paso natural para computarizar cada proceso; sobre estos conceptos, Hilbert estructuró su idea de elaborar un Programa con el propósito de basar la totalidad del conocimiento matemático sobre fundamentos puramente lógicos y por ende, proveer demostraciones mecánicas para sus teoremas, sin embargo, tanto GÖDEL como TURING probaron la Incompletitud y la Indecidibilidad del postulado.

La Técnica de Inferencia RESOLUCION (introducida por ROBINSON en el Año 1965 y refinada en el Año 1971 por KOWALSKI y KUEHNER) constituye el algoritmo básico del Lenguaje PROLOG; tal que corresponde a una Estrategia Inferencial empleada en los Sistemas Lógicos con el objetivo de determinar la verdad de una aserción. El Método de Resolución ensaya la prueba de algún Teorema o Meta expresado como una Proposición P es Verdadera, dando un conjunto de axiomas acerca del problema; actualmente, el método ensaya probar que la negación de la meta ~P no puede ser Verdadera, esta Técnica se conoce como *Demostración por Refutación*. El Método de Resolución implica la producción de nuevas expresiones llamadas Resolventes a partir de la Unión de axiomas existentes y la negación del teorema; luego, las Resolventes se adicionan a la lista de axiomas, obteniendo por derivación nuevas resolventes. El proceso se continúa hasta que se produzca una Contradicción. El esquema de esta Técnica se basa en la Lógica Proposicional; el proceso implica la utilización del Algoritmo de Unificación y las Funciones de Skolem (debidas a LUGER y STUBBLEFIELD, en 1989).

La Lógica Clásica dispone de la Propiedad de Monotonicidad, pues considera una visión estática de un mundo a modelizar; en tal instancia, las descripciones no pueden mutar, a excepción que se incorporen nuevas aserciones. Esta Propiedad es incompatible con el accionar humano. En el Año 1980, MC DERMOTT postuló una nueva categoría de lógica, a la que denominó Lógica No-Monotónica, con el propósito de resolver problemas que revisten caracter de excepcionalidad y consecutivamente, en los Años 1980, 1982 y 1987, WINOGRAD, MC DERMOTT y LADKIN presenta formalismos denominados Lógicas Temporales. Para el marco de la resolución de situaciones ordinarias en las cuales surgen incompletitudes, inconsistencias o imperfecciones, en el Año 1979 ZADEH definió el concepto de Lógica Difusa, a partir de "Pertenencia en Mayor o Menor Medida, o bien Grado de Membrecía". La Lógica Clasica es "Extensiva" en cuanto al significado de las proposiciones está dado por el conjunto de todos los objetos para los cuales estas proposiciones son verdaderas. La "Intensividad" constituye un concepto fugaz, asociado con la noción de Mundos Posibles; esta idea está contemplada por la Lógica Modal, debida inicialmente a MC DERMOTT (Año 1982).

INTRODUCCION A LA LOGICA

En el mundo real, para describir estados de situaciones se emplean Sentencias Declarativas, como por caso:

(A) "Los Profesores enseñan a sus alumnos"

(B) "El Profesor es platense y sus alumnos reciben conocimientos de su parte"

Mediante la aplicación de reglas de razonamiento, es posible obtener nuevas conclusiones; por ejemplo, a partir de (A) y (B):

(C) "El Profesor platense enseña a sus alumnos"

El concepto plasmado en el ejemplo constituye la Idea Básica en la Programación Lógica. La sintaxis de las sentencias se debe definir en forma precisa. Las Fórmulas Lógicas proveen una sintaxis formalizada para la escritura de sentencias, tal que cada sentencia se refiere a individuos en algún mundo y a las relaciones entre tales individuos; inicialmente se considera el alfabeto del lenguaje, el cual debe incluir símbolos (denotan individuos) que se denominan *Constantes* y símbolos que describen relaciones, denominados *Predicados*. El lenguaje formal permite la provisión de la posibilidad de expresar sentencias referidas a todos los elementos o bien, a algún elemento del "mundo", *Cuantificador Universal* y *Cuantificador Existencial*, así como también de *Variables* y *Conectivos Lógicos*. El Alfabeto también contiene símbolos denominados *Funcionales* que representan Funciones sobre los dominios de los objetos.

Desde el punto de vista sintáctico de las Fórmulas Lógicas, los conceptos previos pueden formalizarse como secuencias finitas de símbolos, tales como: Variables, Funcionales y Predicados.

En cuanto al formalismo del Paradigma Lógico representado por el Lenguaje PROLOG, en el Año 1995 se postularon las Normas ISO, estableciendo que el *Alfabeto* del Lenguaje de Lógica de Predicados consta de los siguientes símbolos: *Variables*; *Constantes*; *Funcionales* (sobre los cuales se puede establecer la Aridez Asociada, si ésta es "n", será *f/n*); *Predicados* (sobre los cuales se puede establecer la Aridez Asociada, si ésta es "n", será *p/n*); *Conectivos Lógicos* ($\wedge$, Conjunción; $\vee$, Disyunción; $\Leftrightarrow$, Equivalencia Lógica; $\neg$, Negación; $\supset$, Implicación); *Cuantificadores* ($\forall$, Universal; $\exists$, Existencial); *Símbolos Auxiliares* (paréntesis y comas).

En el Lenguaje Formal de la Lógica de Predicados, los objetos pueden ser representados a través de cadenas denominadas *Términos*, cuya sintaxis se define seguidamente:

El Conjunto *T* de Términos sobre un Alfabeto dado *A* es el menor conjunto tal que:

- Alguna Constante en *A* está en *T*;

- Alguna Variable en *A* está en *T*;

- Si *f/n* es un Funcional en *A* y $t_1, t_2, ..., t_n \in T$, entonces: $f\,(t_1, t_2, ..., t_n) \in T$

INTRODUCCION A LA LOGICA

En el Lenguaje Natural solamente ciertas combinaciones de palabras constituyen sentencias significativas; en la Lógica de Predicados, tales combinaciones reciben el nombre de *Fórmulas Bien Formadas (fbf)*, definiendo su sintaxis de la forma:

Sea *T* el conjunto de Términos sobre el Alfabeto *A*. El Conjunto *F* de *fbf* (con respecto a *A*) es el Conjunto más pequeño tal que:

- Si *p/n* es un Predicado en *A* y $t_1, t_2, ..., t_n \in T$, entonces: $p\ (t_1, t_2, ..., t_n) \in F$;

- Si F y G $\in F$, entonces están: $(\neg F)$, $(F \wedge G)$, $(F \vee G)$, $(F \supset G)$ y $(F \Leftrightarrow G)$,

- Si F $\in F$ y X es una variable en *A* entonces $(\forall xF)$ y $(\exists xF) \in F$.

Las Fórmulas $p\ (t_1, t_2,, ..., t_n)$ se denominan *Fórmulas Atómicas o Atomos.*

En términos de adoptar la simbología del Lenguaje PROLOG, la Fórmula $(F \supset G)$ se escribe en la forma $(G \leftarrow F)$.

Sean: $X_1, X_2, ..., X_n$ variables que ocurren libres en una Fórmula F. La Fórmula Cerrada de la forma: $\forall\ X_1\ (\forall\ X_2\ (...\ (\forall\ X_n\ F)...))$ se denomina *Clausura Universal* de F, denotándosela por $\forall$ F. Similarmente, $\exists$ F se denomina *Clausura Existencial* de F y denota la Fórmula F cerrada bajo la cuantificación existencial.

El Significado de una Fórmula Lógica se define relativo a un "mundo abstracto" denominado *Estructura (Algebraica)*, la cual puede ser verdadera o falsa. Definir el significado de las Fórmulas constituye una conexión formal entre el Lenguaje y una Estructura, que debe ser establecida.

Las presentaciones declarativas se refieren a individuos y conciernen a funciones sobre individuos. Las componentes de la construcción del Lenguaje de Fórmulas son constantes, funcionadores y predicados. El enlace entre el Lenguaje y la Estructura se establece así:

.Una *Interpretación I* de un Alfabeto *A* es un Dominio No Vacío *D* (frecuentemente $|D|$) y un Mapeo que asocia:

- Cada constante c $\in A$ con un elemento $c_I \in D$;

- Cada funcional n-ario f $\in A$ con una función: $f_I : D^n \rightarrow D$;

- Cada predicado n-ario p $\in A$ con una relación: $p_I \subseteq \underbrace{D \text{ x ...... x } D}_{n}$

La Interpretación de constantes, funcionales y predicados provee una base para la asignación de valores verdaderos a fórmulas del lenguaje. El significado de una fórmula se define como una función sobre los significados de sus componentes.

Se define una *Valuación* φ como un mapeo desde variables del Alfabeto al Dominio de una Interpretación; se trata de una Función que asigna objetos de una interpretación a variables del lenguaje. $\varphi\ [\ X \rightarrow t\ ]$ denota la Valuación que es idéntica a φ excepto que $\varphi\ [\ X \rightarrow t\ ]$ mapea X a t.

INTRODUCCION A LA LOGICA

Para formalizar la *Semántica de Términos,* se postula:

.Sea I una Interpretación φ de una Valuación y t, un Término. El *Significado* $\varphi_{I\,(t)}$ de t es un elemento en $|I|$ definido de la forma:

-Si t es una constante c, entonces: $\varphi_I\ (t) := c_I$;

-Si t es una variable x, entonces: $\varphi_I\ (t) := x_I$;

-Si t es de la forma: $f\ (t_1, t_2, ..., t_n)$, entonces: $\varphi_I\ (t) := f_I\ (\varphi_I\ (t_1), ..., \varphi_I\ (t_n))$

El Significado de un término compuesto se obtiene por aplicación de la función denotada por su funcional principal a los significados de sus principales sub-términos, los cuales son obtenidos por aplicación recursiva de esta definición.

Para formalizar la *Semántica de Funciones Bien Formadas (fbf),* se postula:

.Sean I una interpretación, φ una valuación y Q una fórmula. El *Significado* de: Q w.r.t. I y φ se define seguidamente:

- $I \models_\varphi p\ (t_1, t_2, ..., t_n)$ sii: $<\varphi_I\ (t_1), ..., \varphi_I\ (t_n)> \in p_I$;

- $I \models_\varphi (\neg F)$ sii: $I \not\models_\varphi F$;

- $I \models_\varphi (F \wedge G)$ sii: $I \models_\varphi F$ y $I \models_\varphi G$;

- $I \models_\varphi (F \vee G)$ sii: $I \models_\varphi F$ o $I \models_\varphi G$ (o ambos);

- $I \models_\varphi (F \supset G)$ sii: $I \models_\varphi G$ siempre que: $I \models_\varphi F$;

- $I \models_\varphi (F \Leftrightarrow G)$ sii: $I \models_\varphi (F \supset G)$ y $I \models_\varphi (G \supset F)$;

- $I \models_\varphi (\forall X\ F)$ sii: $I \models_{\varphi [X \to t]} F$ para cada $t \in |I|$;

- $I \models_\varphi (\exists X\ F)$ sii: $I \models_{\varphi [X \to t]} F$ para algún $t \in |I|$.

donde: $I \models Q$ expresa la Declaración: "Q es verdadera con respecto a I y φ"

$I \not\models Q$ expresa la Declaración: "Q es falsa w.r.t. I y φ"

Las Semánticas de Fórmulas están definidas sobre el concepto auxiliar de valuación que asocia variables de la Fórmula con elementos de un dominio de la interpretación.

Una Interpretación I es un *Modelo* de P si y solo si cada Fórmula de P es verdadera en I. P tiene muchas interpretaciones.
Cuando se utilizan fórmulas para la descripción de "mundos" es necesario asegurar que cada descripción producida sea *Satisfactoria* (tiene al menos un modelo).
En general, un conjunto satisfactorio de fórmulas tiene muchos modelos; esto significa que las fórmulas propiamente describen un mundo particular de interés al mismo tiempo que describen muchos otros mundos.

INTRODUCCION A LA LOGICA

Sea P un Conjunto de Fórmulas Cerradas. Una Fórmula Cerrada F se denomina *Consecuencia Lógica* de P, denotando: $P \models F$, si y solo si F es verdadera en cada modelo de P.

. Sea P un Conjunto Cerrado de Fórmulas y F, una Fórmula Cerrada; entonces: P ‡ F si solo si: $P \cup \{\neg F\}$ es *Insatisfactorio*.

. Dos Fórmulas F y G son *Lógicamente Equivalentes* ($F \equiv G$) si y solo si F y G tienen el mismo valor verdadero para todas las interpretaciones *I* y las valuaciones φ.

. Sean F y G dos Fórmulas arbitrarias y H(X) una Fórmula con ninguna o más ocurrencias de X; entonces:

$$
\begin{array}{rcll}
\neg\neg F & \equiv & F & \\
F \supset G & \equiv & \neg F \vee G & \\
F \supset G & \equiv & \neg G \supset \neg F & \\
F \Leftrightarrow G & \equiv & (F \supset G) \wedge (G \supset F) & \\
\neg (F \vee G) & \equiv & (\neg F \wedge \neg G) & \text{(Ley de De MORGAN)} \\
\neg (F \wedge G) & \equiv & (\neg F \vee \neg G) & \text{(Ley de De MORGAN)} \\
\neg \forall x\, H(X) & \equiv & \exists x\, \neg H(X) & \text{(Ley de De MORGAN)} \\
\neg \exists x\, H(X) & \equiv & \forall x\, \neg H(X) & \text{(Ley de De MORGAN)}
\end{array}
$$

y si no hay ocurrencias libres de x en F, entonces:

$$\forall x\, (F \vee H(x)) \equiv F \vee \forall x\, H(x)$$

La Formalización del Lenguaje y las sentencias expresadas como Fórmulas Lógicas, permiten indicar que el razonamiento puede verse como un proceso de manipulación de fórmulas dadas, de un conjunto de fórmulas llamadas *Premisas*, que producen una nueva fórmula denominada *Conclusión* para tal instancia.

Uno de los objetivos de la Lógica Simbólica está centrado en formalizar los principios del razonamiento como reglas formales reescritas que se pueden emplear para generar nuevas fórmulas, a las cuales se las denomina *Reglas de Inferencia*. Se requiere que las Reglas de Inferencia correspondan a formas correctas de razonamiento, o bien que las Reglas de Inferencia produzcan sólo Consecuencias Lógicas de las Premisas que puedan ser aplicadas, a tales Reglas se las denomina *Sólidas (o Buenas)*.

Las Reglas de Inferencia de Lógica de Predicados de mayor utilización son: *Modus Ponens* (Dadas dos Premisas A y B, si se conoce que A es verdadera y también que la implicación de A a B es verdadera, entonces se concluye que B es verdadero); *Eliminación de Reglas por Cuantificadores Universales; Introducción de Regla para Conjunción*.

Un tema relevante en el uso de las Reglas de Inferencia es conocer si todas las Consecuencias Lógicas de un conjunto arbitrario de premisas P también puede derivarse de P, si esto ocurre, el Conjunto de Reglas recibe el nombre de *Completo*.

Un Conjunto de Reglas de Inferencia es *Sólido (o Bueno)*, si para cada conjunto de fórmulas cerradas P y cada fórmula cerrada F siempre que $P \vdash F$ si tiene cabida que $P \models F$. Las Reglas de Inferencia son *Completas* si: $P \vdash F$ siempre que: $P \models F$.

Un Conjunto de Premisas es *Inconsistente* si alguna fórmula puede ser derivada desde el conjunto.

APROXIMACION FORMAL A LA LOGICA PROPOSICIONAL

En consecuencia, conforme a lo vertido en INTRODUCCION A LA LOGICA, se expresa que para definir un *SISTEMA FORMAL DE LOGICA PROPOSICIONAL* deben especificarse Cuatro Conjuntos de Objetos, a saber:

(1) Un *Alfabeto*: Este Conjunto consiste de *símbolos* utilizados por el sistema; incluye:

- **Un Subconjunto de Variables Proposicionales (p, q, r, ...; o bien: p_1, p_2, ...).**

- **Un Subconjunto de Símbolos de Puntuación (comas, paréntesis).**

- **Un Subconjunto de Operadores Lógicos ($\neg$, $\wedge$, $\vee$, $\Rightarrow$ (o $\supset$) y $\Leftrightarrow$).**

(2) Un Conjunto de *Fórmulas Bien Formadas (fbf)*: Una Fórmula debe ser sintácticamente correcta. Una *Fórmula Bien Formada (fbf)* es una fórmula sintácticamente correcta, que utiliza únicamente los operadores lógicos del alfabeto del sistema, en consecuencia las fórmulas se refieren a *fbf*.

(3) Un Conjunto de *Axiomas*: Se trata de las fórmulas básicas de las cuales los teoremas se derivan. Algunos sistemas lógicos tienen un específico conjunto finito de axiomas, en tanto que otros sistemas lógicos tienen un conjunto infinito de axiomas definidos como Planes de Axiomas.

(4) Un Conjunto de *Reglas de Inferencia (o Procedimiento)*: Estas Reglas disponen nuevas *fbf* a ser derivadas en el sistema a partir de *fbf* existentes.

Algunos sistemas emplean la Regla de Inferencia *Modus Ponens* o bien, una de sus muchas variaciones. La Regla en cuestión se puede expresar en la forma:

> **"Si P y (P $\Rightarrow$ Q) son ambas demostrables en el sistema, entonces Q es demostrable en el sistema (tal que: P, Q constituyen alguna *fbf* del sistema)".**

Hay sistemas que emplean la Regla de Inferencia conocida como *Resolución*, derivada de *Modus Ponens*, este Sistema presenta la particularidad de ser la Base del Lenguaje PROLOG.

INTRODUCCION AL LENGUAJE PROLOG

La Idea Central de la Programación Lógica establece la separación entre la especificación y la ejecución de programas. KOWALSKI (1988) brinda esta definición:

"La Programación Lógica parte de teoremas mecánicos probando el uso de la lógica para representar conocimiento y emplea la deducción para resolver problemas a través de consecuencias lógicas derivadas. Sin embargo, esta difiere de la prueba de los teoremas mecánicos, probando dos alternativas diferentes pero complementarias: a) Explota el hecho que la lógica puede ser usada para expresar definiciones de funciones computables y procedimientos; b) Explota el uso de procedimientos de demostración que ejecutan deducciones en un objetivo direccionado, para correr tales definiciones como programas. El caso más integral de programación lógica ocurre cuando la información es presentada a través de Cláusulas de Horn y la deducción se ejecuta por razonamiento hacia atrás en Resolución"

El Lenguaje PROLOG es el más claro ejemplo del Paradigma de la Programación Lógica. Este Lenguaje está basado sobre la Forma de la Cláusula de Horn de la Lógica de Predicados de Primer Orden (KOWALSKI, 1979) y permite mostrar la misma potencia expresiva que la notación lógica standard. La Programación en PROLOG es narrada como la Prueba de un Teorema: la Existencia de una Solución a un Problema es probada (si una existe), un proceso durante el cual la/s solución/es pueden ser construidas a través de un efecto lateral.

El Primer Intérprete PROLOG fue escrito por A. COLMENAUER en la Universidad de MARSELLA, basado sobre las discusiones mantenidas con R. KOWALSKI; inicialmente, fue empleado en calidad de herramienta para escribir un sistema de aprendizaje de lenguaje natural y luego en una variada cantidad de proyectos en el Laboratorio de Inteligencia Artificial de la Universidad de EDINBURGO. Este primer intérprete fue implementado en ALGOL-W y posteriormente, en FORTRAN y PASCAL. CLARK y McCABE (1980) completaron una Versión de PROLOG para microcomputadoras (Micro-PROLOG), usado básicamente para enseñanza. Posteriormente, en la Universidad de EDINBURGO se escribió un nuevo intérprete PROLOG basado en C, permitiendo ampliar el rango de los sistemas PROLOG, en particular en desarrollos basados en UNIX.

Entre la gran variedad de Intérpretes basados en los Sistemas PROLOG, escritos en FORTRAN o PASCAL, las mejores versiones son: Salford PROLOG (Universidad de SALFORD - FORTRAN 77); York PROLOG (Universidad de YORK - PASCAL); IC PROLOG (Imperial College - PASCAL). IC PROLOG ha servido de base para los Lenguajes Modernos de Programación de Lógica Concurrente, tales como: Concurrent PROLOG, PARLOG (RINGWOOD, 1988), EPILOG (WISE, 1986). PROLOG II (Universidad de MARSELLA: GIANNESINI y Otros, 1986) implementa algunos lenguajes modificados, con un número de hechos adicionales para modularización (mundos) y controla el flujo. Turbo PROLOG (TOWNSEND, 1987) integra un Sistema PROLOG compilado con un desarrollo de programación basado en ventanas; ofrece un rico conjunto de predicados predefinidos, con texto y soportes gráficos y la posibilidad de enlazar programas escritos en otros lenguajes.

INTRODUCCION AL LENGUAJE PROLOG

La programación en PROLOG es distinta a la programación en lenguajes convencionales. En lugar de prescribir un plan paso a paso para la solución de problemas, uno debe definir las propiedades estructurales de un problema en términos de lo que debe ser verdadero y cómo nuevas conclusiones basadas en estos hechos pueden ser obtenidas. En consecuencia, un programa PROLOG es una descripción formal de algún posible “mundo” en términos de objetos relevantes y relaciones. Entonces, la ejecución, resultará en un proceso computacional de deducción lógica a través de esta base de datos, estableciendo ya que las suposiciones “informales” realizadas son consistentes con la descripción formal. Durante el curso de este proceso, las reglas serán aplicadas y las variables pueden ser instanciadas. KOWALSKI (1979) define un Programa Lógico como:

Programa = Lógica + Control

Los *Términos* y las *Cláusulas* forman el nivel más bajo de los Programas PROLOG. Los Términos se refieren a Clases de Individuos, y pueden denotar constantes o variables. Las *Constantes* pueden ser: *Números, Símbolos, Estructuras y Listas,* tanto Enteras como de Punto Flotante. Los Números son escritos en la forma convencional; los Símbolos son *Identificadores* dados por un conjunto de letras, dígitos y caracteres especiales ((), |, [], ‘, “, {}, %), esto significa que los Símbolos representan *Individuos,* que deben comenzar con Letras Minúsculas, en tanto que las *Variables*, que deben comenzar con Letras Mayúsculas. Las Estructuras son colecciones de entidades, que representan relaciones entre objetos. El primer elemento de una estructura se refiere usualmente como un *Funcional (Functor),* el cual nombra la relación; por caso:

hermanos (Neptuno, Plutón)
hermanos (Plutón, Júpiter)

define la relación binaria *hermanos* y afirma dos instancias de ella. El Término “*aridez”* se refiere al número de argumentos de un funcional; en consecuencia, *hermanos* tiene Aridez 2.

Las *Variables* denotan clases de objetos y solamente durante la interpretación serán instanciados a una clase particular de individuos.

Los *Términos* son los bloques de construcción de *Cláusulas,* una colección de ellas forma un programa. Las Cláusulas son separadas por períodos y son interpretadas como otros hechos o reglas. Los *Comentarios* son encerrados por /* y */ o precedidos por % (cabe aclarar que no están permitidos en todos los sistemas) pueden ser incluidos entre cláusulas e ignorados durante la interpretación. Los Hechos son definidos a través de predicados o funcionales de alta aridez. Los *Predicados* y los *Funcionales de n-posiciones* expresan propiedades y relaciones de n-posiciones. Las *Listas* describen clasificaciones.

La suma de todas las aserciones forma la Base de Datos PROLOG, definiendo el estado del “Mundo” que se desea describir; el Lenguaje se desempeña bajo el supuesto de “Mundo Cerrado”. Sólo se pueden probar los hechos (axiomas) almacenados en esta base de datos, de acuerdo a las reglas provistas, es aceptado como “verdadero”. El Lenguaje PROLOG alcanza conclusiones acerca de una descripción, dando la característica de *Completitud,* debiendo ser también, por supuesto *Consistente*. No se pueden expresar directamente inconsistencias en este Lenguaje.

INTRODUCCION AL LENGUAJE PROLOG

El funcionamiento del lenguaje consiste, esencialmente, en un demostrador de teoremas que reduce el problema inicial a demostrar a una conjunción de predicados y así sucesivamente con el objetivo de demostrar cada uno de ellos; esta característica posibilita que su uso para resolver problemas a través de la reducción sea inmediato. PROLOG no emplea información heurística para hacer las reducciones, básicamente funciona con una Estrategia de Búsqueda denominada DEPTH-FIRST. En lo relativo a Reglas de Inferencia, el Lenguaje brinda los medios necesarios para implementar la estructura de la regla (*Sintaxis)*; el mecanismo de apareo y reemplazo de variables (*Unificación)* y la posibilidad de evaluar las condiciones y acciones en forma no determinística (*Intérprete)*. Es posible emplear en forma directa el esquema de regla provisto, a partir de su visión declarativa, interpretando las reglas como axiomas lógicos donde el operador ":- " describe la Implicación Lógica; así, la regla está formada por "una cabeza", que describe el resultado de la regla y "un cuerpo" que impone condiciones acerca de cuándo "la cabeza" es verdadera; se destaca que el núcleo del motor de inferencia lo da el Intérprete PROLOG y además, que el formalismo de Cláusulas de Horn empleado es similar al Formalismo de las Reglas de Producción. Esto posibilita ubicar a PROLOG como una herramienta en sí misma, permitiendo la construcción de intérpretes para lenguajes de definición de estructuras tipo Frame.

INTRODUCCION AL CALCULO DE PREDICADOS

1. Introducción

En general, el *Alfabeto* del Lenguaje de la *Lógica de Predicados* consta de los siguientes símbolos: *Variables*; *Constantes*; *Funcionales* (sobre los cuales se puede establecer la Aridez Asociada, tal que si ésta es "n", será *f/n*); *Predicados* (sobre los que es posible establecer la Aridez Asociada, si ésta es "n", será *p/n*); *Conectivos Lógicos* ($\neg$, Negación; $\wedge$, Conjunción; $\vee$, Disyunción; $\Leftrightarrow$, Equivalencia Lógica; $\supset$, Implicación); *Cuantificadores* ($\forall$, Universal; $\exists$, Existencial) y *Símbolos Auxiliares* (paréntesis y comas).

2. Cálculo de Predicados

La operación con un *Lenguaje de Primer Orden* equivale a concebir al Conjunto de Objetos como el Universo del Discurso, en virtud a considerar a todos los estados que se están discutiendo en el Lenguaje; esto evidencia que para cada lenguaje existen diferentes tipos de "universos de discurso". Considerar el *Alfabeto* del Lenguaje permite indicar que:

a) El rango de las Variables está definido sobre todo el Universo del Discurso;

b) Cada constante corresponde exactamente un miembro distinto del Universo (En el Universo de Enteros, el símbolo constante "c" puede partir del valor 0);

c) Una Función corresponde a una Función sobre el Universo (Dado un Conjunto de Ciudades, la función f(x) se puede interpretar como una función en la cual, dada "una ciudad", f(x) devuelve al "departamento al que pertenece");

d) Un Predicado es un Símbolo que establece una relación, que puede ser vista como una función que delibera entre un valor Verdadero (V) o un valor Falso (F); donde tales argumentos son Términos del Lenguaje de Primer Orden bajo consideración. Cada Predicado asigna una Relación sobre el Universo del Discurso;

e) Los Términos son definidos inductivamente como Variables, Constantes o Funciones, aplicados a términos de mayor simplicidad.

Una Interpretación de un Lenguaje de Primer Orden tiene un Dominio (Universo del Discurso) juntamente con asignaciones de constantes, funciones y predicados a constantes, funciones y relaciones actuales en tal dominio. La consideración de una Interpretación particular para una relación actual, establece que para argumentos dados, la misma sea V o F.

Cada Predicado de un Argumento es un Mapeo: $D \rightarrow \{ V, F \}$, tal que D constituye el Dominio (o Universo del Discurso). Un Predicado de dos Argumentos: $D \times D \rightarrow \{ V, F \}$. Los Valores de Verdad V y F se pueden tomar como Predicados de "Cero Argumento".

Esta Estructura está muy conectada con la noción de Conjuntos; en consecuencia, los objetos del Universo del Discurso debe formar un Conjunto. El "Universo del Discurso" en términos informales, corresponde al Conjunto de Objetos Individuales que se están discutiendo o más formalmente, el Dominio de la Interpretación bajo consideración.

INTRODUCCION AL CALCULO DE PREDICADOS

Los Cuantificadores Existencial (∃) y Universal (∀) también componen el Alfabeto del Lenguaje. Para el Cuantificador Universal es necesario definir el *Dominio de la Cuantificación* (En la expresión: Para Todo x, es necesario indicarse qué valores de x deben incluirse) y el *Alcance de las Variables Cuantificadas*, que corresponde a la parte de la Fórmula que se aplicará.

Es posible afirmar que en el Lenguaje Natural solo ciertas combinaciones de palabras constituyen sentencias significativas; en la Lógica de Predicados a tales combinaciones se las denomina *Fórmulas Bien Formadas (fbf)*.

Una Fórmula debe ser sintácticamente correcta; así, una *Fórmula Bien Formada (fbf)* es una fórmula sintácticamente correcta, que emplea únicamente los operadores lógicos del Alfabeto del sistema; en consecuencia, las fórmulas se refieren a *fbf*.

Sea *T* el conjunto de Términos sobre el Alfabeto *A*. Se define al Conjunto *F* de *fbf* (con respecto a *A*) como el Conjunto más pequeño tal que:

- Si *p/n* es un Predicado en *A* y $t_1, t_2, ..., t_n \in T$, entonces: $p (t_1, t_2, ..., t_n) \in F$;

- Si F y G ∈ *F*, entonces están: $(\neg F)$, $(F \wedge G)$, $(F \vee G)$, $(F \supset G)$ y $(F \Leftrightarrow G)$,

- Si F ∈ *F* y X es una variable en *A* entonces $(\forall x F)$ y $(\exists x F) \in F$.

Las Fórmulas $p (t_1, t_2, ..., t_n)$ se denominan *Fórmulas Atómicas o Átomos.*

3. Conceptos de Semántica de Lógica Proposicional

Un Símbolo Proposicional corresponde a un estado acerca del mundo. Un Símbolo Proposicional puede ser V o F, dado algún estado del mundo; el Valor de Verdad asignado a las sentencias proposicionales constituye una *Interpretación*, una aserción acerca de su valor de verdad en algún *Mundo Posible*.

En términos formales, una Interpretación corresponde a un Mapeo desde los Símbolos Proposicionales en el Conjunto {V, F}. Los Símbolos V y F son parte del Conjunto de Sentencias Bien Formadas del Cálculo Proposicional. Cada posible Mapeo corresponde a un Mundo Posible de Interpretación.

Ejemplo 1: Si: P denota la Proposición: "Estoy estudiando Computación"
y: Q denota la Proposición: "El clima está enrarecido"
y: R denota la Proposición: "La materia es fácil"

Entonces, el Conjunto de Proposiciones: {P, Q, R} tiene ocho mapeos funcionales diferentes en los Valores de Verdad {V, F}, tal que estos mapeos corresponden a cuatro mundos posibles diferentes.

Dada una interpretación de un Conjunto de Proposiciones, es necesario definir el valor de verdad para las expresiones compuestas. La Semántica del Cálculo Proposicional, tal como la Sintaxis se define en forma inductiva.

INTRODUCCION AL CALCULO DE PREDICADOS

4. <u>Introducción a la Semántica de la Lógica de Predicados</u>

En el Cálculo Proposicional, cada Símbolo Atómico (P, Q, R, ...) denota una Proposición de cierta complejidad; no existe forma de acceder a las componentes de una aserción individual; en cambio, el Cálculo de Predicados provee esta capacidad.

<u>Ejemplo 2</u>: Sea P: "He obtenido Ocho en Lógica"

Es posible crear el Predicado "Calificación", que describe una relación entre un dato y la calificación, de la forma:

Calificación(Lógica, Ocho)

Mediante Reglas de Inferencia se pueden manipular expresiones del cálculo de predicados accediendo a sus componentes individuales e infiriendo nuevas sentencias

El Cálculo de Predicados también permite expresiones que contienen variables, tales variables permiten la creación de aserciones generales acerca de clases de entidades. En el Ejemplo 2, para describir materias en las cuales la Calificación ha sido Ocho, se escribe:

Calificación(X, Ocho)

5. <u>Sintaxis de Predicados y Sentencias</u>

Previo a considerar expresiones correctas en el Cálculo de Predicados, deben ser definidos un Alfabeto y una Gramática con el propósito de crear los *Símbolos* del Lenguaje, correspondientes a aspectos lexicográficos de la Definición del Lenguaje de Programación. Los Símbolos del Cálculo de Predicados, como las Señales en los Lenguajes de Programación son elementos sintácticos irreductibles: no pueden partirse en sus partes componentes por las operaciones del lenguaje. Los *Símbolos* del Cálculo de Predicados son Cadenas de Letras y Dígitos, comenzando con una Letra. Los caracteres Blancos y No-Alfanuméricos no pueden aparecer dentro de la cadena, aunque los subrayados, _, pueden ser usados para mejorar la legibilidad.

El *Alfabeto* está compuesto de *símbolos* utilizados por el sistema; consiste de:

- **Conjunto de Letras, sub-índices y supra-índices.**

- **Conjunto de Dígitos: 0, 1, 2, ..., 9.**

- **Subrayados: _.**

Los Paréntesis "( )", las Comas "," y los Períodos "." son empleados únicamente para construir Expresiones Bien-Formadas y no denotan objetos o relaciones en el mundo; estos símbolos se denominan *Símbolos Impropios*.

Los Símbolos del Cálculo de Predicados pueden representar Variables (designan clases generales de objetos o propiedades en el mundo, comenzando con una letra mayúscula), Constantes (nombran objetos específicos o propiedades en el mundo; deben comenzar con una letra minúscula; las constantes verdadera y falsa se reservan como *Símbolos de Verdad*), Funciones (definidas sobre objetos en el mundo del discurso; deben comenzar con una letra minúscula; denotan un mapeo de uno o más elementos en un conjunto, denominado *Dominio* de la Función en un único elemento de otro conjunto, denominado *Rango* de la Función; los elementos del Dominio y del Rango son objetos del Mundo del Discurso) o Predicados.

Cada Función tiene una *Aridez* Asociada que indica el número de elementos en el dominio mapeado sobre cada elemento del rango. Una *Expresión Funcional* es un Símbolo Funcional seguido por sus argumentos, tal que los argumentos son elementos del dominio de la función; el número de argumentos es igual a la aridez de la función; los argumentos son encerrados entre paréntesis y encerrados por comas.

Ejemplo 3: Las Expresiones Funcionales: g(X,Y) ; Suma(1,9) y Profesor(Juan) son Fórmulas Bien-Formadas.

Cada Expresión Funcional denota el mapeo de los argumentos sobre un objeto único en el rango, llamado *Valor* de la Función; en el Ejemplo 3, Profesor es una Función 1-aria, entonces: Profesor(Juan), es una Expresión Funcional cuyo valor en el Mundo del Discurso es: Jorge; en el mismo Ejemplo, Suma es una Función de Aridez 2 con Dominio en los Enteros, entonces: Suma(1,9) es una expresión funcional cuyo Valor es el entero 10. Cuando se reemplaza una Función con su valor, resulta una *Evaluación*.

Los *Símbolos Predicados* son símbolos que comienzan con una letra minúscula. Los Predicados tienen un valor entero positivo asociado referido a la *Aridez* o "Número de Argumentos" para el Predicado. Los Predicados con el mismo número, pero con Aridez diferente son considerados distintos.

Una *Sentencia Atómica* es un Predicado Constante de Aridez n, seguido por n Términos: $t_1, t_2, ..., t_n$, encerrados entre paréntesis y separados por comas. Los Valores de Verdad: V y F, también constituyen Sentencias Atómicas. A las Sentencias Atómicas también se las llama *Expresiones Atómicas, Átomos o Proposiciones*. Pueden combinarse sentencias atómicas usando operadores lógicos ($\wedge$, $\vee$, $\Rightarrow$, $\neg$, $\Leftrightarrow$) para formar *sentencias* en el Cálculo de Predicados.

Cuando una variable aparece como argumento en una sentencia, se refiere a un objeto no especificado en el dominio. El Cálculo de Predicados incluye dos símbolos, los cuantificadores de variables($\forall$, $\exists$) tal que satisfacen el significado de una sentencia que contiene una variable.

Ejemplo 4: $\forall$ X disfruta (X, música) [Es V para todos los valores del Dominio de X]
$\exists$ Y primo (Y, Javier) [Es V para algún valor del Dominio de Y]

Cada Sentencia Atómica es una Sentencia:

1. **Si s es una sentencia, su negación $\neg$s, entonces también es una sentencia.**

2. **Si s_1 y s_2 son sentencias, su conjunción $s_1 \wedge s_2$, entonces también es una sentencia.**

3. **Si s_1 y s_2 son sentencias, su disyunción $s_1 \vee s_2$, entonces también es una sentencia.**

4. Si s_1 y s_2 son sentencias, la implicación $s_1 \Rightarrow s_2$, entonces también es una sentencia.

5. Si s_1 y s_2 son sentencias, su equivalencia $s_1 \Leftrightarrow s_2$, entonces también es una sentencia.

6. Si X es una variable y s es una sentencia, entonces $\forall$X s es una sentencia.

7. Si X es una variable y s es una sentencia, entonces $\exists$ X s es una sentencia.

Ejemplo 5: suma(2, 5) es una Función, No constituye una Sentencia Atómica.
igual(producto(4,5), 20) es una Sentencia Atómica, Verdadera.
$\exists$X cuadrado(X, 196) es una Sentencia Atómica, Verdadera para X = 14.

La definición de Sentencias del Cálculo de Predicados sugiere la necesidad de un Método para verificar que una expresión es una sentencia, que se escribe como un algoritmo recursivo: *verify_sentence*, que toma como argumento a una expresión candidata y retorna *success* si la expresión es una sentencia.

6. Una Semántica del Cálculo de Predicados

La Semántica del Cálculo de Predicados provee una base formal para la determinación del valor de verdad de expresiones bien formadas. La verdad de las expresiones depende del mapeo de constantes, variables, predicados y funciones en objetos y relaciones en el dominio del discurso. La verdad de las relaciones en el dominio determina la verdad de las expresiones correspondientes.

Ejemplo 6: Primos (María, Marisa)
Primos (Marisa, Carlos)
Primos (Carlos, Federico)

} El Valor de Verdad de cada predicado es independiente del resto.

El uso del Cálculo de Predicados como una representación para la Resolución de Problemas, describe objetos y relaciones en el dominio de la interpretación con un conjunto de expresiones bien formadas. Los términos y predicados de estas expresiones denotan objetos y relaciones en el dominio. Esta base de datos del cálculo de predicados, cada una teniendo valor de verdad V, describe el *Estado del Mundo*. El Ejemplo 6 es un caso de tal base de datos.

Para definir formalmente la Semántica del Cálculo de Predicados, en primera instancia debe definirse una *Interpretación* sobre un Dominio D, que determinará la *Asignación de Valores de Verdad* de sentencias en el lenguaje.

Sea el Conjunto D un Conjunto No Vacío, entonces:

Una *Interpretación* sobre D es una asignación de entidades de D a cada símbolo: constante, variable, predicado y función de una expresión del cálculo de predicados, tal que:

1. Cada constante es asignada a un elemento de D.

2. Cada variable es asignada a un subconjunto no vacío de D; siendo sustituciones permitidas para tal variable.

3. **Cada función *f* de aridez *m* está definida sobre m argumentos de D y define un mapeo desde D^m en D.**

4. **Cada predicado *p* de aridez *n* está definido sobre *n* argumentos desde D y define un mapeo desde D^m en el conjunto { V, F }.**

Dada una Interpretación, el significado de una expresión es la asignación de valor de verdad sobre la interpretación.

Asumiendo una Expresión E y una Interpretación I para E sobre un dominio D no vacío; entonces, el *Valor de Verdad* para E se determina a través de:

1. **El valor de una constante es el elemento de D que está asignado a través de I.**

2. **El valor de una variable es el conjunto de los elementos de D, asignado a través de I.**

3. **El valor de una expresión funcional es aquel elemento de D obtenido por evaluación de la función para los valores asignados por la interpretación.**

4. **El valor de verdad del símbolo "verdadero" es V y el del símbolo "falso" es F.**

5. **El valor de una sentencia atómica es V o bien, F, determinado por la interpretación I.**

6. **El valor de la negación de una sentencia es V si el valor de la sentencia es F y es F si el valor de la sentencia es V.**

7. **El valor de la conjunción de dos sentencias es V si el valor de ambas sentencias es V y F en todo otro caso.**

8. **El valor de la disyunción de dos sentencias es F solamente cuando el valor de ambas sentencias es F y es V en todo otro caso.**

9. **El valor de la Implicación de dos sentencias es F solamente cuando la Premisa es V y el Consecuente es F y es V en todo otro caso.**

10. **El valor de la Equivalencia de dos sentencias es V cuando los valores de ambas sentencias son los mismos y es F en todo otro caso.**

11. **El valor de $\forall$ X S es V si S es V para todas las asignaciones a X bajo I, y es F en todo otro caso.**

12. **El valor de $\exists$ X S es V si hay una asignación a X en la interpretación bajo la cual es V; en otro caso, es F.**

Las Asignaciones de Verdad de Proposiciones Compuestas se describen a través de Tablas de Verdad.

La Cuantificación Universal introduce problemas en la computación de los valores de verdad de una sentencia, porque todos los posibles valores de un símbolo variable deben ser testeados para ver las expresiones restantes verdaderas. Si el Dominio de una Interpretación es infinito, el testeo exhaustivo de todas las sustituciones a una variable universalmente cuantificada resulta *computacionalmente imposible*; debido a tal situación al Cálculo de Predicados se lo denomina *Indecidible*.

Para Predicados *p* y *q* y Variables *X* e *Y* se establecen las relaciones entre negación y cuantificadores existencial y universal, las cuales son empleadas en los *Sistemas de Resolución por Refutación*:

1- $\neg\exists X\, p(X) \Leftrightarrow \forall X \neg\, p(X)$
2- $\neg\forall X\, p(x) \Leftrightarrow \exists X \neg\, p(X)$
3- $\exists X\, p(X) \Leftrightarrow \exists Y\, p(Y)$
4- $\forall X\, q(X) \Leftrightarrow \forall Y\, q(Y)$
5- $\forall X\, (p(X) \wedge q(X)) \Leftrightarrow \forall X\, p(X) \wedge \forall Y\, q(Y)$
6- $\exists X\, (p(X) \vee q(X)) \Leftrightarrow \exists X\, p(X) \vee \exists Y\, q(Y)$

En el Lenguaje definido, las variables cuantificadas universal y existencialmente se refieren solamente a objetos (constantes) en el dominio del discurso. Los nombres de los predicados y funciones no pueden ser reemplazados por variables cuantificadas. Así, el Lenguaje se denomina *Cálculo de Predicados de Primer Orden*.

El *Cálculo de Predicados de Primer Orden* permite que las variables cuantificadas se refieran a objetos en el dominio del discurso y no a predicados o funciones.

Ejemplo 7: $\forall$ (Vínculo) Vínculo (Juan, Mirta)

Este ejemplo no constituye una expresión bien formada en el Cálculo de Predicados de Primer Orden. Existe un Cálculo de Predicados de Orden Superior en el cual, expresiones como esta son significativas; Mc Carthy (1968), entre otros ha usado un lenguaje de orden superior para representar conocimiento en programas de aprendizaje de lenguaje natural

7. Teorías de Primer Orden

Un Sistema de Axiomas y Reglas puede ser usado para formar un Sistema Deductivo Lógico (Teoría de Primer Orden) en la Lógica de Predicados. Los axiomas son de dos clases:

(1) Axiomas Deductivos (o Lógicos), los cuales son independientes del alfabeto particular de primer orden.

(2) Axiomas que son relativos al sistema particular de primer orden, conocidos como axiomas propios o hipótesis.

8. Lógicas de Orden Superior

La Interpretación de un término de un L.P.O. es un Objeto Simple, no estructurado en un Dominio No Vacío; ciertamente el objeto en sí mismo puede tener una estructura cuando es considerado desde distintos puntos de vista, por caso: *un vehículo*, pero "el blanco" de un término de un L.P.O. es una unidad entera; además, y como aspecto de mayor relevancia, es necesario expresar que, en tales lenguajes, la cuantificación únicamente se define sobre variables, en consecuencia no puede aplicarse a predicados y funciones; por lo tanto, expresiones de la forma: $(\forall x_i)\ P_1\ (x_i)$ son *fbf*, en tanto que expresiones del tipo: $(\forall P_1)(\exists x_i)\ P_1\ (x_i)$ no lo son.

En el campo de la Aritmética puede verse que el Axioma Inductivo de PEANO no responde a una *fbf* de Primer Orden, dado que:

$$(\forall P)\ (P(0) \wedge (\forall x)\ (P(x) \rightarrow P(s(x))) \rightarrow (\forall y)\ P(y))$$

donde s es la función que denota la sucesión (o siguiente).

En tal instancia, puede afirmarse que una *Lógica de Segundo Orden* permite la cuantificación sobre predicados y funciones de primer orden y provee facilidades para expresar propiedades de funciones o predicados como también representar predicados con predicados y predicados con funciones como argumentos, de aquí puede concluirse que el Axioma Inductivo de PEANO corresponde a una Lógica de Segundo Orden.

La Generalización del concepto previo permite postular que la *Lógica de Tercer Orden* permite la cuantificación sobre predicados y funciones de segundo orden; pudiéndose definir bajo consideraciones similiares *Lógicas de Orden Superior*.

A partir de estos conceptos que posibilitan la postulación de Lógicas de Orden Superior, y en atención a la definición de la Lógica de Primer Orden, puede afirmarse que la Lógica Proposicional es una *Lógica de Orden Cero*.