

03063

2
24.



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

**U.A.C.P. y P. DEL C.C.N.
I.I.N.A.S.**

S i e t e - D o

**Sistema Experto Tutorial en
Ecuaciones Diferenciales Ordinarias**

T E S I S

**QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACION**

P R E S E N T A:

FRANCISCO EDGAR CASTILLO BARRERA

Director de Tesis:

M. EN C. HORACIO CARVAJAL SANCHEZ-YARZA

MAYO 1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

Agradezco en primer lugar a DIOS

Por estar siempre conmigo.

Agradezco a mi Madre

Que siempre ha creído en mí y me ha dado todo su apoyo.

Agradezco a la Doctora Hanna Oktaba

Por su entrega y dedicación al pueblo de México en su desempeño como coordinadora de la maestría en ciencias de la computación que se imparte en el edificio IIMAS Ciudad Universitaria. Así como su apoyo recibido durante mis estudios en esta institución.

Agradezco a los Profesores

Por su atención recibida y en particular al M.en C. Horacio Carvajal Sánchez-Yarza por asesorarme en esta tesis.

Agradezco a las Personas

Que de alguna forma colaboraron y me aportaron algo cuando realizaba este trabajo de tesis

Agradezco a MEXICO

Por haberme recibido como su hijo.

Introducción

La inteligencia artificial (IA) y los sistemas expertos (SE) han tenido gran aplicación en distintas áreas [5] como la medicina, la producción y los negocios, etc. Sin embargo en el área educativa se han desarrollado pocas aplicaciones[5] y no se ha logrado visualizar el gran impacto que puede tener la utilización de técnicas de IA y los SE en el proceso educativo.

Con este trabajo de tesis se presenta una propuesta de apoyo para el proceso educativo utilizando técnicas de IA y presentándose un sistema computacional que se le ha denominado SIETEDO (Sistema Experto Tutorial en Ecuaciones Diferenciales Ordinarias) el cual es considerado como un sistema experto tutorial inteligente que está dirigido a la enseñanza del cálculo diferencial e integral y en particular para el área de las matemáticas conocida como ecuaciones diferenciales ordinarias. El desarrollo de este sistema está fundamentado en la teoría de los sistemas expertos y los sistemas tutoriales inteligentes; las cuales son áreas de estudio de la inteligencia artificial.

En este trabajo de tesis se busca como primer objetivo *la enseñanza de las ecuaciones diferenciales ordinarias* utilizando técnicas de IA, ya que al momento de hacer esta tesis no se conocía ningún sistema computacional inteligente comercial que lo hiciera. El utilizar la computadora como una herramienta de aprendizaje presenta más ventajas que el simple hecho de querer aprender al leer un libro sobre algún tópico específico, ya que los sistemas expertos permiten tener un *diálogo directo* con el estudiante por medio de la computadora permite manipularlo gráficamente, utilizar voz y audio al mismo tiempo logrando despertar en él un mayor interés por el aprendizaje y por ende el uso de la computadora para este fin, así como el lograr que el estudiante ponga mayor atención y concentración al sistema al momento en que es presentado el conocimiento. También se logra *una enseñanza personalizada*(particular) por este medio debido a que el uso de la computadora es individual y esto es muy difícil de hacer en un salón de clases en donde el profesor tiene que atender a varios alumnos a la vez.

Como segundo objetivo en esta tesis, se busca que el sistema SIETEDO sea por una parte, el que proporcione al profesor una herramienta didáctica computacional que le facilite la enseñanza del cálculo a sus alumnos y por otra que los estudiantes puedan tener un fácil aprendizaje de esta materia; además de hacerla atractiva para cualquier usuario. Es importante notar que se propone este sistema como un sustituto del profesor.

El alcance de este trabajo está limitado a la solución simbólica de cuatro tipos de ecuaciones diferenciales ordinarias que son: las homogéneas, exactas, de variable separable y las ecuaciones de Bernoulli.

Esta tesis está dividida en tres partes que la conforman de cinco capítulos. En la primera parte se dan todos los fundamentos teóricos en que se basan los sistemas expertos y los sistemas tutoriales inteligentes correspondiendo a los capítulos I y II.

La segunda parte comprende los capítulos III y IV en donde se explica el diseño e implementación del sistema SIETEDO.

La tercera parte de este documento comprende el capítulo V que corresponde a las posibilidades de uso del sistema.

Se presenta como última parte del documento las conclusiones obtenidas sobre los logros alcanzados por el sistema, así como una tabla de comparación con otros softwares comerciales.

Se añaden finalmente dos apéndices A y B sobre la teoría matemática utilizada por el sistema. Los cuales pueden servir de consulta para quién no este familiarizado con los conceptos matemáticos del cálculo diferencial e integral así como los de ecuaciones diferenciales ordinarias.

Contenido

Agradecimientos	ii
Introducción	iii

Capítulo I Sistemas Expertos

1.1. Inteligencia artificial y sistemas expertos	1
1.2. Características de un sistema experto	2
1.3. Diseño de un sistema experto	3
1.4. Ingeniería de software para el diseño de sistemas expertos	3
1.5. Funcionamiento de un sistema experto	4
1.6. Ventajas de utilizar un sistema experto	5
1.7. Aplicaciones	6
1.8. El Conocimiento de un sistema experto	7
1.8.1. Representación del conocimiento	7

Capítulo II Sistemas tutoriales inteligentes (ITS)

2.1. Antecedentes	8
2.2. Componentes de un sistema ITS	8
2.2.1. Modelo del experto	9
2.2.2. Modelo del estudiante	9
2.2.3. Modelo instruccional	9
2.3. Tipos de sistemas ITS	10
2.4. Algunos sistemas tutoriales inteligentes	10

Capítulo III Diseño del sistema SIETEDO

3.1.Descripción general	13
3.2.Arquitectura	13
3.3.Representación del conocimiento.	14
3.3.1.Hechos y predicados para SIETEDO	14
3.4.Motor de inferencias	16
3.5.Estructuras utilizadas	16
3.5.1.Constantes y variables	16
3.5.2.Manejo de listas.	17
3.5.3.Conjuntos	17
3.5.4.Funciones	17
3.6.Notación matemática	17
3.6.1.Principales operadores utilizados en SIETEDO	18
3.6.2.Precedencia de operadores en SIETEDO	18
3.6.3.Expresiones Diferenciales e Integrales.	18
3.6.4.Ecuaciones Diferenciales Ordinarias	19
3.7.Predicados.	19
3.8.1.Formato de lectura en SIETEDO	19

Capítulo IV Programación del sistema SIETEDO

4.1.El lenguaje Prolog	20
4.2.El predicado ecuación	20
4.3.El predicado integración	21
4.3.1.Integración inmediata y por cambio de variable	21
4.3.2.El predicado por partes.	21
4.3.2.1.El predicado por partes iterando	21
4.3.3.Integración de expresiones trigonométricas	22
4.4.El Predicado derivación	22
4.5.Utilerias de SIETEDO.	22
4.5.1.El predicado equivalente	23
4.5.2.El predicado simplifica.	24
4.5.3.El predicado multiplica.	24
4.5.4.El predicado agrupa	25
4.5.5.El predicado factoriza	25
4.5.6.El predicado polinomio.	25

Capítulo V Uso del sistema

5.1.Requerimientos de trabajo	26
5.2.El ambiente de MS-DOS	26
5.3.Compilación del sistema	26
5.4.El intérprete de Arity/Prolog	26

Conclusiones	29
--------------	----

Apéndice A	Tácticas y métodos de integración utilizados por SIETEDO	
	A.1. Tácticas generales de integración	33
	A.2. Terminología	33
	A.3. Propiedades de las integrales	33
	A.4. Fórmulas de integración inmediata	33
	A.5. Método de integración por partes	34
	A.6. Integración de funciones trigonométricas	35
	A.7. Tácticas para reducción trigonométrica	35
Apéndice B	Ecuaciones diferenciales ordinarias	
	B.1. Conceptos básicos de ecuaciones diferenciales ordinarias	37
	B.2. Ecuaciones con variables separables y ecuaciones reducibles a ellas	38
	B.3. Ecuaciones homogéneas y reducibles a ellas	40
	B.4. Ecuaciones lineales de primer orden. Ecuaciones de Bernoulli	42
	B.5. Ecuaciones diferenciales exactas. Factor integrante	44
Referencias		46

Capítulo I Sistemas Expertos

1.1. Inteligencia artificial y sistemas expertos

Como una área de estudio de las ciencias de la computación se encuentra la inteligencia artificial (IA) la cual estudia cómo lograr que las máquinas realicen tareas que por el momento, son realizadas mejor por los seres humanos. Sin embargo el intentar dar una definición de lo que es la IA es tan difícil como el querer definir algún concepto abstracto. En la tabla-1 se muestran algunas tareas que son objetivo de estudio de la IA.

Tareas de la vida diaria			
<i>Percepción</i>		<i>Lenguaje natural</i>	
Visión		Comprensión	
Habla		Traducción	
Tareas formales			
<i>Juegos</i>		<i>Matemáticas</i>	
Ajedrez		Geometría	
Backgammon		Lógica	
Damas		Cálculo integral	
Go			

Tabla-1: Áreas de trabajo en IA

Existen áreas específicas de estudio en IA que están prosperando como disciplinas prácticas para aplicaciones muy comunes, como por ejemplo el diagnóstico médico que hace un doctor a su paciente; este se trata de imitar lo mejor posible utilizando un sistema inteligente llamado *sistema experto*; como es el caso real del sistema MYCIN (Shortliffé, 1976), un sistema que trata de imitar el conocimiento de un médico para la detección de enfermedades infecciosas. En la figura-1 se muestran algunas áreas de estudio de la IA.

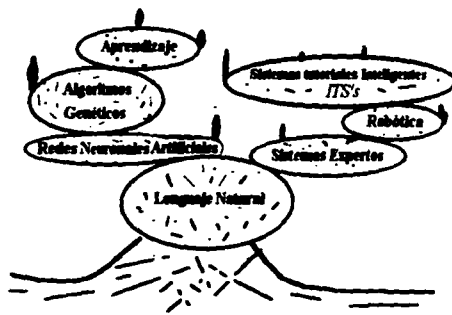


Figura-1: Algunas áreas de la inteligencia artificial

Los sistemas expertos también conocidos como sistemas basados en reglas (Post, 1943) son sistemas por computadora que tratan de emular el comportamiento de un experto humano en algún área del conocimiento. El conocimiento del sistema es proporcionado por uno o varios expertos en la materia el cual es almacenado en una base de datos la cual es manipulada por el sistema por medio de un motor lógico de inferencias que le permiten al sistema dar respuestas.

El primer programa que dio origen a esta área fue un sistema llamado META-DENDRAL (Mitchell, 1978) el cual usó técnicas de aprendizaje para construir automáticamente reglas para un sistema experto. En él se construyen reglas para ser usadas por DENDRAL (Hayes-Roth, 1983) un programa que determina la estructura de los componentes de compuestos químicos muy complejos.

1.2. Características de un sistema experto

Para que un sistema computacional sea considerado como sistema experto se debe cumplir con ciertos requisitos como los que se describen en la tabla-2.

Alto desempeño	Tiempo de respuesta adecuado	Confiable	Justificación
El sistema debe ser capaz de responder al nivel de competencia de un experto humano o mejor.	El tiempo de respuesta del sistema debe ser mejor o igual, que el tiempo que le tome al experto humano encontrar la solución.	Se debe tener un buen grado de confiabilidad en los resultados	Todo sistema experto debe ser capaz de explicar los pasos de su razonamiento que lo llevaron a dar una respuesta

Tabla-2: Características de un sistema experto

1.3. Diseño de un sistema experto

El proceso de desarrollo de un sistema experto es el que se muestra en la figura-2. Es el ingeniero del conocimiento el que establece un diálogo con el experto humano y extrae el conocimiento del experto. Este conocimiento es representado en la base de datos o base de conocimiento por medio de alguna técnica de IA. El experto humano evalúa el sistema y le da sus observaciones al ingeniero del conocimiento el cual hace las modificaciones pertinentes. Este proceso se repite hasta que el experto humano considera que el sistema trabaja bien.



Figura-2: Proceso de desarrollo de un Sistema Experto

1.4. Ingeniería de software para el diseño de un sistema experto

La ingeniería de software utilizada para el desarrollo de los sistemas inteligentes se basa en los mismos modelos de desarrollo (modelo de cascada, modelo de espiral, etc.) que se emplean para cualquier otro sistema computacional.

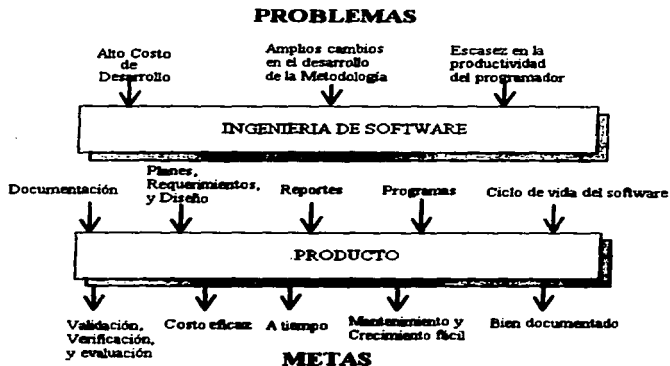


Figura-3: Proceso de desarrollo de un sistema experto

Es importante notar que la parte del sistema en donde se debe tener más cuidado en su desarrollo es la representación del conocimiento, pues es donde más comúnmente se presentan errores en su elaboración.

1.5. Funcionamiento de un sistema experto

El funcionamiento de un sistema experto es muy sencillo en general y es mostrado en la figura-4. El usuario hace una consulta al sistema sobre algún hecho o información que desea saber y el sistema da respuesta al usuario según sus resultados obtenidos. Internamente el sistema consiste de dos componentes principales: la base de conocimiento la cual contiene el conocimiento del experto y la máquina de inferencias la cual manipula este conocimiento para obtener sus conclusiones y proporcionar una respuesta al usuario.

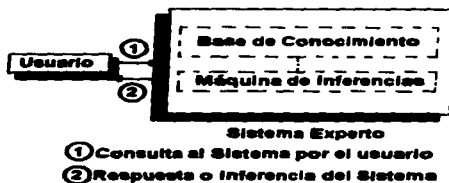


Figura-4: Esquema funcional de un sistema experto

1.6. Ventajas de utilizar un sistema experto

Los sistemas expertos presentan muchas ventajas en su uso, a diferencia de los sistemas de cómputo que no utilizan técnicas de inteligencia artificial.

- **Incremento en la productividad.** Esta característica favorece a las industrias al aumentar en tiempo el número de productos elaborados y a la vez tener más control del proceso productivo disminuyendo los posibles errores humanos debidos al cansancio.
- **Reducción de Costos.** El costo de uso de un sistema experto es en general muy bajo comparado con el costo de servicio de un empleado.
- **Permanencia.** Este tipo de sistemas tienen la característica de ser extendidos más que de ser eliminados, lo cual les da un tiempo de vida muy alto.
- **Múltiples Expertos.** El conocimiento de varios expertos puede ser sustituido por un sistema experto.
- **Justificación.** Esta característica es quizás una de las fuertes que tienen todos los sistemas expertos, cualquier resultado obtenido es justificado por el sistema.
- **Tiempo de respuesta rápido.** El tiempo de respuesta dado por el sistema experto es menor comparado con el tiempo que le llevaría a un experto humano encontrar la solución.
- **Comportamiento estable en cada momento.** En comparación con un experto humano el sistema puede trabajar por varias horas sin modificarse la calidad de sus respuestas mientras que las respuestas dadas por el humano se pueden ver afectadas según su estado físico.
- **Tutor inteligente.** El sistema al momento de justificar sus resultados puede actuar como un instructor al mostrar su forma de razonamiento.
- **Base de Datos Inteligente.** Los sistemas expertos pueden ser utilizados para acceder de una forma inteligente una base de datos.

1.7. Aplicaciones

Los sistemas expertos tienen aplicación en todas las áreas del quehacer humano como son la medicina, la agricultura, la milicia, los simuladores de vuelo, etc., todavía hoy en día existen muchas otras áreas en las que no se han desarrollado sistemas expertos pero sería muy beneficioso su uso. En la tabla-3 se muestran algunos sistemas expertos muy populares con aplicaciones en diferentes áreas.

Sistema Experto	Uso	Base de Conocimientos	Autores	Lenguaje	Ref.
<i>MYCIN</i>	Diagnóstico de enfermedades infecciosas	Reglas de producción	Shortliffe, 1976	Lisp	[21]
<i>DENDRAL</i>	Deducción de estructuras químicas a partir de los datos de masa espectral y test magnético nucleares	Reglas de producción	Buchanan, B. G. y E. A. Feigenbaum 1978	Lisp	[19]
<i>PROSPECTOR</i>	Aconseja sobre la exploración minera	Reglas de producción	Duda et al., 1979; Hart et al., 1978	Lisp	[19]
<i>MACSYMA</i>	Integración Simbólica	Reglas de producción	Martin W.A. y R.J. Fateman, 1981	Lisp	[3,19]
<i>INTEGRATE</i>	Integración Simbólica	Reglas de producción	Kimball, R.A., 1981	-----	[19]
<i>WHY</i>	Causa de las Lluvias	Guiónes	Collins A., 1977	-----	[19,20]
<i>SCHOLAR</i>	Geografía	Redes semánticas	Carbonell J.R., 1970	-----	[19,20]
<i>SOPHIE</i> SOPHisticated Instructional Environment	Localización de fallas electrónicas	Redes semánticas con simulador de circuitos	Brown J.S. y R.J. Brown, 1973	LISP y FORTRAN	[19,20]
<i>WEST</i>	Expresiones aritméticas	Reglas de producción	Burton R. ichard R. y John Seely Brown, 1979	INTERLISP	[19,20]
<i>BUGGY</i>	Operaciones de restas	Red basada en procedimientos	Brown J.S. et al., 1977	INTERLISP	[19,20]

Tabla-3: Algunos sistemas expertos conocidos

1.8.El conocimiento de un sistema experto

La parte más importante que constituye un sistema experto es *el conocimiento que tiene y la forma en que lo maneja*. Es el ingeniero del conocimiento la persona que tiene la tarea de obtener el conocimiento para el sistema y decidir cual es la mejor forma de representarlo.

Existen varios tipos de representaciones del conocimiento pero en todas ellas se tienen dos tipos de entidades:

- **Hechos:** verdades en un cierto mundo (aquello que se quiere representar).
- **Representaciones de los hechos en un determinado formalismo.** Estas son las entidades que se manipulan.

Para poder tener una estructuración del conocimiento se tiene que clasificar estas dos entidades en dos niveles distintos:

- **El nivel del conocimiento:** donde se describen los hechos.
- **El nivel simbólico:** donde se describen los objetos del nivel del conocimiento en términos de símbolos manipulables por programas.

Existen dos tipos básicos de representaciones del conocimiento que son la *declarativa* y la *procedimental*. Una representación declarativa, es aquella en la que el conocimiento está especificado, pero en las que la forma en que el conocimiento debe ser usado no se tiene. Una representación procedimental, es aquella en la que la información de control necesaria para utilizar el conocimiento se encuentra contenida en el mismo conocimiento. Para utilizar este tipo de representación se necesita añadir al intérprete las reglas para que siga las instrucciones dadas por el conocimiento.

1.8.1.Representación del conocimiento

Existen varias técnicas de representación de conocimiento[13] pero entre las más utilizadas tenemos:

- 1.-Autómatas finitos.
- 2.-Lógica simbólica y de predicados.
- 3.-Sistemas basados en reglas de producción.
- 4.-Sistemas no monótonos.
- 5.-Sistemas de razonamiento estadístico.
- 6.-Redes semánticas.
- 7.-Dependencia conceptual.
- 8.-CYC.
- 9.-Marcos.
- 10.-Guiones

El método 1. es de representación procedimental, los métodos 2,3,4,5,6,7 y 8 son de representación declarativa, y los métodos 9 y 10 son de representación mixta.

Una de las formas de representación del conocimiento en IA más popular es sin duda mediante reglas. Este tipo de representación tiene sus fundamentos en la lógica en donde las afirmaciones lógicas del mundo son consideradas como un programa y no como los datos de entrada al programa.

Capítulo II Sistemas tutoriales inteligentes (ITS)

2.1. Antecedentes

Los sistemas tutoriales inteligentes (ITS) son sistemas computacionales que utilizan técnicas de inteligencia artificial y su principal objetivo es el de imitar a un instructor el cual da enseñanza personalizada a su alumno, teniendo que supervisar y evaluar el desarrollo académico de su aprendiz.

Los precursores de los sistemas tutoriales fueron los sistemas CAI's (Computer-Assisted Instruction). A mediados de los sesentas Uhr y sus colaboradores implementaron una serie de sistemas los cuales generaban problemas en el área de la aritmética (Uhr, 1969). Se siguieron desarrollando sistemas para la aritmética eligiéndose un conjunto de problemas selectos a un nivel de dificultad apropiado para que los estudiantes los pudieran resolver (Suppes, 1967; Woods & Hartley, 1971).

Al agregar las técnicas de la inteligencia artificial en el diseño de los sistemas de instrucción asistida por computadora (CAI's), se dio origen a un nuevo tipo de sistemas tutoriales, que fueron denominados como ICAI's (Intelligent Computer-Assisted Instruction). En el diseño de los ICAI's se introduce un nuevo objetivo, que consiste en incluir material instruccional que permita analizar el desempeño del estudiante para aplicar estrategias tutoriales individualizadas. Sin embargo, debe puntualizarse que los primeros ICAI's se enfocaron principalmente en la representación del material a presentar y se olvidaron de que, para realizar su tarea educativa, un buen maestro debe comprender, y no sólo suponer lo que está haciendo el estudiante.

Actualmente el término ICAI ha sido cambiado por el de ITS's (Intelligent Tutoring System) para hacer referencia a los sistemas instruccionales por computadora que involucran el uso de técnicas de IA.

Los ITS son una herramienta más de ayuda para el estudiante en la adquisición de destrezas y de conocimientos sobre un tema, a través de una actitud análoga a la de un profesor que lo atiende en forma personalizada [45].

2.2. Componentes de un sistema ITS

Los sistemas ITS han tomado muchas formas, pero esencialmente han separado los componentes principales de un mecanismo de instrucción, de tal forma que le permite al estudiante y al sistema, la flexibilidad de un ambiente de aprendizaje que se asemeja a lo que actualmente ocurre cuando el estudiante y el profesor se sientan juntos e intentan enseñar y aprender. A pesar de la diversidad de arquitecturas que se manejan alrededor de los sistemas ITS, es posible distinguir tres elementos básicos comunes que lo forman. El modelo del experto, el modelo del estudiante y el modelo instruccional; mismos que se describen a continuación.

2.2.1. Modelo del Experto

Esta parte que constituye a un sistema tutorial inteligente, es la que está formada de los conocimientos sobre un dominio específico del cual el sistema intenta enseñar al estudiante.

Este módulo del sistema se encarga de generar el contenido instruccional, así como para evaluar el desempeño del estudiante. Los métodos de inteligencia artificial más representativos son las redes semánticas, sistemas de regla de producción, representación procedimental y la construcción de marcos o guiones.

2.2.2. Modelo del Estudiante

El cual se utiliza para evaluar el nivel de conocimientos del estudiante y hacer hipótesis acerca de las estrategias de razonamiento utilizadas para alcanzar el estado actual de conocimientos.

Como la mayoría de los sistemas ITS representan el conocimiento del estudiante como un subconjunto de la base de conocimientos de un sistema experto, el modelo se construye comparando el desempeño del estudiante con el desempeño de un sistema experto en la misma tarea. A esta técnica se le llama el "modelo de sobreposición" (Carr).

2.2.3. Modelo Instruccional

El modelo instruccional es el conjunto de especificaciones que se le dan al sistema acerca de cual es el material que debe de presentar al estudiante, cuando y como hacerlo.

En los sistemas ITS se conocen básicamente dos estrategias de instrucción que son: el método Socrático y el método por entrenamiento(coaching).

El método Socrático

El cual hace al estudiante se cuestione sobre el conocimiento y así llevarlo a un proceso de reconocimiento de sus malas interpretaciones. Este método mejor conocido como el de la mayéutica (el arte de dar luz) era el medio por el cual Sócrates se valía para enseñar a sus seguidores. En el proceso de depuración se considera que el estudiante razona acerca de lo que sabe y de lo que no sabe, y de esta forma modifica sus interpretaciones (Carbonell, 1970 y Stevens 1977).

El método por entrenamiento(coaching)

En el cual se ejercita al estudiante para hacer que desarrolle ciertas habilidades, como el utilizar juegos por computadora o el mostrar varios ejemplos de soluciones de problemas matemáticos y hacer que resuelva ejercicios similares a los mostrados.

Algunos autores (Burton 1979) consideran que este método debe de mantener entretenido al estudiante, de tal forma que como resultado del entretenimiento, que puede ser considerado diversión se de aprendizaje.

En la actualidad en muchos de los sistemas ITS no se han desarrollado completamente los tres módulos del sistema, debido a la complejidad y al tamaño que implica el desarrollo de cada parte del sistema.

2.3. Tipos de sistemas ITS

Con base en los primeros sistemas tutoriales inteligentes que aparecieron se da una clasificación con la principal aportación que estos dieron

1. Tutores mezclados con indicativas	Este tipo de sistemas involucran al estudiante en tener una conversación bilateral tratando de enseñarlo via el método Socrático y con un descubrimiento del conocimiento guiado. Un ejemplo lo tenemos en los sistemas SCHOLAR y SOPHIE.
2. Tutores con entrenador(Coach)	Un entrenador (coach) observa el desempeño del estudiante y le asesora para lograr mejores resultados. Este tipo de sistemas lo encontramos en juegos como son los sistemas WEST y WUSOR.
3. Tutores de errores conceptuales	Estos sistemas tienen una base de conocimiento con los errores conceptuales más comunes que el estudiante puede tener al momento de intentar resolver un problema. Este tipo de sistemas los encontramos en BUGGY y PROUST.
4. El concepto de micromundos	Este tipo de sistemas tienen desarrollado todo un marco teórico sobre algún tema como la geometría, física o la música, el cual le permite al estudiante conocer sobre estos temas. Como ejemplo tenemos al lenguaje LOGO(Papert's 1971).
5. Sistemas AICAI	En este tipo de sistemas CAI se incluyen técnicas de inteligencia artificial dentro del mismo sistema CAI. AI-based expertise y se introduce el concepto de utilizar una <i>gráfica-genética</i> como marco para la representación procedimental.

2.5. Algunos sistema tutorial inteligentes

Se han desarrollado varios sistemas tutoriales inteligentes[5] en diferentes áreas y con diferentes objetivos como lo muestra la tabla-7 en donde se menciona el nombre del sistema, el uso que se le da, una breve descripción del mismo, los autores, el tipo de representación del conocimiento utilizada, el lenguaje en que fue programado y su referencia.

ITS	Uso	Descripción	Autores	Base de Conocimiento	Lenguaje	Ref.
Algebraic Tutoring	Algebra	Consiste de 4 módulos: dominio del Experto, modelo del estudiante, módulo tutorial real y interfaz del usuario	Ghoeffi A. y G.Moccaldi, 1990.	Modelo de Objetos	SmalltalkV	[5]
PSOLVE	Ecuaciones Algebraicas	El sistema es capaz de mostrar paso a paso la solución de un problema	Lantz B.S., W.S. Bregar y A.M. Farley, 1983	Reglas de producción	LISP	[5]
INTEGRATE	Integración Simbólica	El sistema enseña integración	Kimball, R.A., 1981	Reglas de producción	-----	[3]
MACSYMA Symbolic Manipulation of Algebraic equations	Integración Simbólica	Resuelve límites, ecuaciones e integración simbólica.	Martin W.A. y R.J. Fateman, 1981	Reglas de Producción	LISP	[3,5]
WHY	Meteorología, aprender las causas de las lluvias	El sistema enseña al estudiante el proceso de las precipitaciones	Collins A., 1977	Guiones	-----	[42]
SCHOLAR	Geografía	El sistema enseña la geografía de Sudamérica	Carlsonell J.R., 1970	Redes Semánticas	-----	[42]
SOPHIE I Sophisticated Instructional Environment	Electrónica Localización de fallas eléctricas y electrónicas	El sistema enseña al estudiante a detectar fallas en los circuitos electrónicos	Brown J.S. y R.J. Bobrow, 1973	Redes Semánticas con simulador de circuitos	LISP y FORTRAN	[42]
WEST	Expresiones aritméticas con base en un juego.	Se presenta un tablero con tarjetas graduadas	Burton Richard R. y John Seely Brown, 1979	Reglas de producción	INTERLISP	[42]
BUGGY	Operaciones de restas	El sistema desarrolla las habilidades aritméticas del estudiante.	Brown J.S. et al., 1977	Módulo basado en procedimientos	INTERLISP	[42]
WUSOR-I	Relaciones lógicas. Con base en el juego WUMPUS	El sistema guía al jugador para salir de las cavernas sin ser devorado por el monstruo.	Stansfield, Carr y Goldstein, 1976	Red de gráficas genéticas	-----	[42]

EXCHECK	Logica y Teoria de conjuntos	El sistema guía al estudiante en el estudio de la lógica y conjuntos	Smith, McDonald J., 1975	Reglas de producción con intérprete de lógica		[42]
ITS Intelligent Tutoring System	Algebra	Enseña al estudiante a factorizar expresiones algebraicas	Girolfi A. y G. Moccaldi, 1990	Modelo de Objetos	Smalltalk/V	[5]
Statistics Tutor	Estadística	Enseña al estudiante estadística	Saleem N. y A.N. Azad, 1992	Reglas de producción	VP-Expert	[5]
QUADRATIC	Algebra	El sistema selecciona problemas para le enseñanza de ecuaciones cuadráticas	O'Shea T., 1979	El sistema utiliza reglas de producción basadas en el axioma de la raíz general de Vieta.		[4]
GUIDON	Medicina	Enfermedades infecciosas	Cheney W.J., 1983	Reglas de producción	INTERLISP	[3]
ADVISOR	Matemática	Enseña como utilizar mejor el programa de MACSYMA	Grosseroh M.R., 1977	Reglas de producción	LISP	[2]
MEND-II	Enseñanza del lenguaje PASCAL	El sistema detecta y diagnostica los errores de programación	Soloway E.M., et al., 1983	Reglas de producción		[2]
LISP-ITS	Enseñanza del lenguaje LISP	El sistema monitorea y detecta errores del estudiante	Harmon P.R. Mann y W. Morriset, 1988	Reglas de producción	LISP	[2]
STREAMER	Equipos de propulsión a vapor	El sistema muestra válvulas, interruptores y el usuario debe ajustarlos y observa lo que sucede	Williams M.D., Holland J.D. and A.L. Stevens, 1983	Esquema orientado a objetos utilizando FLAVORS	FLAVORS	[2]

Tabla-7: Tabla de sistemas tutoriales inteligentes

Capítulo III Diseño del sistema SIETEDO

3.1. Descripción general

El sistema SIETEDO es un sistema creado para enseñar a resolver ecuaciones diferenciales ordinarias de manera simbólica, las cuales se imparten en los primeros curso de cálculo a nivel superior. Este sistema tiene como base para su desarrollo un sistema experto que resuelve integrales en forma simbólica y un sistema experto en ecuaciones diferenciales ordinarias.

3.2. Arquitectura

El sistema SIETEDO está compuesto de dos grandes módulos que son el módulo experto que lo conforma un sistema experto en ecuaciones diferenciales ordinarias y el módulo instruccional el cual es otro sistema inteligente que se encarga del proceso de enseñanza y evaluación del aprendizaje del alumno. En la figura-5 se muestran los elementos que componen cada módulo; los rectángulos y los óvalos denotan predicados del programa, las aristas indican vínculos entre los predicados y los semicírculos significan que el predicado debe tomar alguna decisión que representa una acción inteligente para la solución del problema.

El módulo simplificación mostrado en la figura-5 sirve como "filtro" para la expresión matemática de entrada el cual tiene como función principal analizar la expresión y escribirla siempre que se pueda en términos conocidos por algún predicado del sistema. En muchos casos basta con simplificar la expresión o reescribirla para tener una expresión matemática conocida y fácil de resolver por algún método de integración. En otros casos se hace un cambio de variable para reescribir la expresión en términos conocidos para algún método de integración o fórmula.

El módulo integrador está compuesto por varios predicados que representan un método específico de integración (integración por fórmula inmediata, integración por partes, integración por sustitución trigonométrica e integración por fracciones parciales). Es importante notar que el método de integración por partes está compuesto por dos métodos que se les han denominado Integración por partes inmediata, el cual aplica la fórmula directamente y el otro reconoce expresiones en las cuales sólo basta una manipulación algebraica para obtener el resultado después de haber aplicado una vez la fórmula de integración por partes.

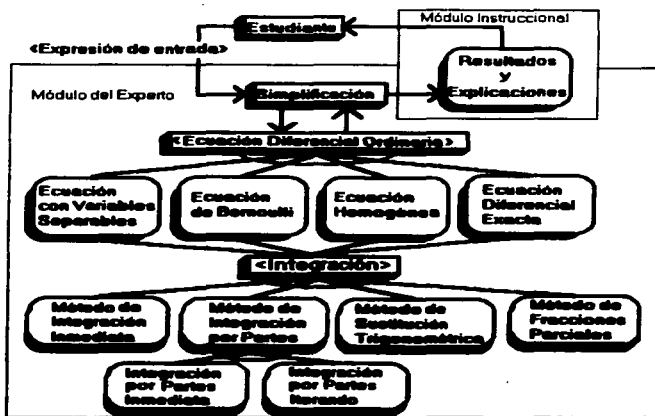


Figura 5: Arquitectura de SIETEDO

El módulo instruccional representado en la figura-5 como Resultados y Explicaciones es el encargado de dar la solución de la ecuación diferencial solicitada por el usuario y además da una explicación paso a paso de cómo se obtuvo el resultado. Esta propiedad es aprovechada por SIETEDO para mostrar el proceso que realizó para obtener una integral específica.

3.3. Representación del conocimiento

El sistema SIETEDO tiene su representación del conocimiento mediante reglas de producción en donde las afirmaciones lógicas se ven como representaciones declarativas del conocimiento.

El conocimiento que maneja el sistema está fundamentado en fórmulas y axiomas matemáticos del los primeros son representados por predicados en el sistema y los segundos por hechos que representan axiomas en matemáticas.

3.3.1. Hechos y predicados para SIETEDO

Un hecho para el sistema SIETEDO es por definición un conocimiento básico en el cálculo diferencial e integral, el cual se da como verdadero siempre. En el caso particular para obtener la derivada de una función constante con respecto a "x's" tenemos el siguiente ejemplo.

Notación matemática $\frac{dk}{dx} = 0$ (1)

Representación interna del hecho "d" de la fórmula(1) en SIETEDO

d(K, X, 0).

Que es interpretado por SIETEDO como la derivada de una constante con respecto a "X" es cero; la letra K denota una constante y la letra X representa la variable de derivación. Un predicado para SIETEDO es una representación procedimental de alguna fórmula matemática del cálculo diferencial e integral. Como ejemplo en SIETEDO para la representación de la integral de la suma de dos funciones, se tiene un predicado llamado "i", el cual denota el símbolo de la integral y cinco letras mayúsculas F,G,X,IF e IG que denotan los argumentos del predicado.

La letra F representa la primera función a integrar, la G la segunda función a integrar, la letra X la variable de integración y por último las letras IF e IG las cuales denotan el resultado de la integración de la primera y segunda función respectivamente. En lenguaje Prolog las letras mayúsculas son variables y las letras minúsculas o números son constantes.

Notación matemática $\int (f + g) dx = \int f dx + \int g dx$ (2)

Representación interna del predicado "i" de la fórmula(2) en SIETEDO

i(F+G:X,IF+IG) :- i(F:X,IF),!, i(G:X,IG).

Para el caso de las ecuaciones diferenciales ordinarias tenemos como ejemplo el predicado ecuación.

```
ecuacion(E):-ecuacion_diferencial_exacta(E,Mxy:X,Op,Nxy:Y),
  deriva(MXY:Y,Dy),
  deriva(NXY:X,Dx),
  equiv(Dy,Dx),
  titulo2,
  assertz(titulo('Ecuaciones de la forma: M(x,y)dx + N(x,y)dy = 0,')).
assertz(titulo("","")),
assertz(titulo(' Considerar a Mxy como => ',Mxy,"")),
assertz(titulo(' Considerar a Nxy como => ',Nxy,"")),
assertz(titulo(' Derivar a Mxy = ',Mxy,' con respecto a ',Y)),
assertz(titulo(' Derivar a Nxy = ',Nxy,' con respecto a ',X)),
assertz(titulo(' La derivada de ',Mxy,' es ',Dy)),
assertz(titulo(' La derivada de ',Nxy,' es ',Dx)).
```

```
ecuacion_diferencial_exacta( Mxy:X + Nxy:Y = 0, Mxy:X,+,Nxy:Y ).
ecuacion_diferencial_exacta( Mxy:X - Nxy:Y = 0, Mxy:X,-,Nxy:Y ).
```

3.4. Motor de inferencias

SIETEDO hace un proceso de búsqueda hacia adelante, haciendo búsqueda a través de un espacio problema partiendo de un estado inicial y finalizando en un estado objetivo. Cabe mencionar que el lenguaje Prolog únicamente efectúa búsquedas a partir de un estado objetivo. La técnica que utiliza para hacer las búsquedas es utilizando *razonamiento hacia delante* a partir de los estados iniciales. Se comienza por construir un árbol de secuencias de movimientos que se pueden representar como soluciones por la configuración inicial en la raíz del árbol. Se generará el siguiente nivel del árbol encontrando todas las reglas cuyos lados izquierdos se relacionen con el nodo raíz, y que utilicen sus lados derechos para crear nuevas configuraciones. Se creará el siguiente nivel tomando cada nodo que se haya generado en el nivel anterior, y aplicándolo a todas las reglas cuyos lados izquierdos se relacionen con éste. Se continuará así hasta que se consiga una configuración que se empareje con el estado objetivo.

Por otro lado, considerando el problema de las ecuaciones diferenciales, el espacio de búsqueda es un conjunto de fórmulas las cuales contienen expresiones diferenciales. El estado inicial es una ecuación diferencial concreta. Se comienza con un estado inicial y varios estados objetivos posibles.

Cuando el sistema SIETEDO no puede encontrar la solución a una ecuación diferencial con el conocimiento que tiene en su base de datos y las reglas de inferencia definidas, este contesta al usuario que no tiene todo el conocimiento para dar su respuesta.

3.5. Estructuras utilizadas

Las estructuras de datos utilizadas por el sistema SIETEDO son las usuales en programación como son constantes, variables y listas.

3.5.1. Constantes y variables

Para SIETEDO una constante es un número real, entero natural o una letra diferente a la letra que aparezca después del operador dos puntos.

Constante variable

Ejemplo: $I((a+x)^2-23.5 * dx), a, 2, -23.5, dx$

Es importante hacer notar que en el sistema SIETEDO coexisten dos lenguajes: El de Prolog y el matemático. Esta situación se presenta muy claramente en las variables. En Prolog, las variables siempre comienzan con mayúsculas o con "_"; mientras que las variables del lenguaje matemático definido para SIETEDO son siempre un átomo.

Ejemplos de Variables

En SIETEDO
a, r, x, a, u, t, k,

En Arity/Prolog
X, DV, 11, _872F,

3.5.2. Manejo de Listas

Para Arity/Prolog una lista es por definición un ordenación lineal de elementos, en donde al primer elemento se le llama cabeza de la lista y a los demás elementos se les conoce como el resto de la lista. Existe una lista muy especial, que es la lista vacía la cual no tiene elementos y se representa con dos corchetes cuadrados "[]".

3.5.3. Conjuntos

Al igual que algunos sistemas matemáticos, SIETEDO tiene sus bases matemáticas en la teoría de conjuntos, los cuales son representados en el sistema por medio de listas.

Las operaciones con conjuntos fueron programadas utilizando los predicados usuales de Prolog para el manejo de listas y respetando las propiedades de los conjuntos:

- Existe el conjunto vacío, representado en SIETEDO como la lista vacía "[]".
- Un conjunto por definición no tiene elementos repetidos.

Los predicados implementados bajo las reglas anteriores son el de *pertenece* y *unión*. Las demás operaciones se pueden implementar fácilmente.

Ejemplos:

?- pert(paty, [miguel,edgar,paty]).
yes

?- union([1,juan,6], [2,pedro,6], L3).
L3 = [1,juan,6,2,pedro] ->
yes

3.5.4. Funciones

Las funciones matemáticas que permite SIETEDO son las más usuales, entre las cuales encontramos las trigonométricas, las polinomiales, exponenciales, logarítmicas, lineales, etc.

Es importante hacer notar, que las funciones matemáticas tienen una caligrafía no lineal y SIETEDO sólo acepta una escritura lineal, por lo que es necesario seguir la notación clásica para computadoras que permite la linealidad de las expresiones. Con seguridad si se han utilizado calculadoras de bolsillo y computadoras se estará familiarizado con esta notación.

3.6. Notación Matemática

La notación matemática utilizada para la programación de SIETEDO es muy similar a la notación utilizada en los libros de cálculo[29,31], salvo algunas excepciones.

3.6.1.Principales operadores utilizados en SIETEDO

Se clasifican los operadores en dos tipos: binarios y unarios. Entre los que tenemos la suma "+", resta "-", multiplicación "*", división "/" y demás operadores usados en computación sin embargo cabe mencionar que SIETEDO, incorpora el operador $\int(N)$, el cual representa el símbolo de la integral que Leibnitz ideó como una "S" deformada.

3.6.2.Precedencia de operadores en SIETEDO

La precedencia de la mayoría de los operadores (el orden en que se evalúan los operadores) en SIETEDO es la misma que maneja Arity/Prolog (Para mayor información ver el manual de "The Arity/Prolog Language Reference Manual"), sin embargo se modificó en cuatro operadores su jerarquía, así como su asociatividad.

Fue importante modificar la asociatividad de los operadores en Arity/Prolog, debido a que el sistema SIETEDO incorpora un nuevo operador denotado por ":" el cual representa el operador diferencial el cual debe tener la mayor precedencia entre los operadores.

Operador	Precedencia	Asociatividad
*	400	De derecha a izquierda
/	390	" " "
:	600	" " "
!	550	" " "

De la tabla anterior se deduce que el operador ":" tiene mayor precedencia que los operadores "*" y "/" lo cual es un factor importante al momento en que Arity/Prolog evalúa los operadores.

3.6.3 Expresiones Diferenciales e Integrales

En el caso de las expresiones diferenciales, estas deben utilizar el operador ":" que tiene un papel importante, como indicador del diferencial. SIETEDO admite todo tipo de expresiones diferenciales, como por ejemplo las que se muestran en la tabla-8.

Expresiones para SIETEDO

Notación Matemática	Notación SIETEDO.	Simplificada
1) $d(\sin x)$	1 : sen(x)	cos(x) : x
2) $\cos(x) d(\sin(x))$	cos(x) : sen(x)	cos(x)^2 : x
3) $(x^2 + 2x + 3)dx$	(x^2 + 2*x + 3) : x	x^2 + 2*x + 3 : x
4) y'	y :	y : x
5) $\frac{\cos(x)}{dx} + x : \frac{e^n}{dp} = 3$	cos(x) : x + x^2 * exp(p) : p = 3	cos(x) : x + x^2 * exp(p) : p = 3

Tabla-8: Tabla de expresiones diferenciales

Cuando se recibe de entrada una expresión para el sistema SIETEDO, esta es inmediatamente simplificada a una expresión diferencial equivalente en donde el segundo operando de ":" es sólo un átomo, tal y como se muestra en la columna 3 de la tabla-8. (Para llevar acabo este proceso es indispensable que el segundo operando de ":" contenga una sola variable). Nótese que los paréntesis en el tercer ejemplo de la columna 2 de la tabla-8 no son necesarios a causa de la precedencia con que fueron definidos los operadores.

3.6.4. Ecuaciones Diferenciales Ordinarias

Un ecuación diferencial para SIETEDO, es una expresión que contiene sumas o diferencias de funciones diferenciales igualadas a una constante, como por ejemplo:

$$\begin{array}{ll} \cos(x):x + x*\text{sen}(p):p = 3 & (1) \\ m*g*x^2:g+g*\exp(p):p = 0 & (2) \\ y! + x = 5 & (3) \end{array}$$

Observese que en la ecuación (3) del ejemplo anterior se utilizó el símbolo de admiración ":" para denotar la derivada de "y" con respecto a "x".

3.7. Predicados

SIETEDO está formado por un gran número de predicados los cuales pueden ser invocados directamente desde el intérprete de Arity/Prolog después del indicador del intérprete (el signo de interrogación y el guión).

$$?-\text{identidad}(\text{sen}(x)^2,R).$$

3.7.1. Formato de lectura en SIETEDO

Los predicados en este manual están subrayados, escritos con minúsculas según las reglas de Arity/Prolog, y están indicados los parámetros o aridades correspondientes dentro de un paréntesis en el cual son separados por comas. Se utilizarán los símbolos de "-", "+", "?" y "*" para indicar si el parámetro debe estar vinculado a un valor específico, a esta acción le llamaremos instanciar un parámetro, indicándolo con el signo "+". No instanciado con el símbolo "-" y con la posibilidad de estar en cualquier caso de los anteriores con el signo "?", y el asterisco "*", en caso de que el parámetro deba estar "semi-instanciado".

Como ejemplo tenemos el predicado *derivada*(+E, -R) el cual tiene aridad 2 (puesto que tiene dos argumentos E y R), en donde el parámetro "E", debe estar instanciado puesto que lo indica el signo "+" y el parámetro "R" no, por estar precedido de un signo "-".

Capítulo IV Programación de SIETEDO

4.1.El lenguaje Prolog

SIETEDO fue programado utilizando el paradigma de programación lógica en el cual las afirmaciones lógicas del mundo se consideran como programas. Actualmente existen diferentes sistemas de programación lógica, el más popular de ellos es el Prolog(Clocks in y Mellish, 1984; Bratko, 1986). Un programa escrito en lenguaje Prolog está descrito como una serie de afirmaciones lógicas en las que cada cláusula es una cláusula de Horn. Una cláusula de Horn es una cláusula que tiene como mucho una literal positiva. Así $p \vee q$, y $p \rightarrow q$ son cláusulas de Horn. Un programa en prolog está constituido básicamente por dos partes *las hechas* los cuales contienen constantes y representan sentencias acerca de objetos específicos y *las predicados* los cuales contienen variables y representan sentencias acerca de las diferentes clases de objetos.

4.2.El predicado ecuación

Existe un predicado general para resolver ecuaciones diferenciales ordinarias llamado *ecuación*, el cual tiene cuatro casos diferentes de ecuaciones posibles que son:separables,homogéneas,exactas y de Bernoulli.

El predicado ecuación	Este predicado determina si la ecuación es homogénea e intenta resolverla por alguno de los métodos conocidos por el sistema.
El predicado homogénea	Este predicado en SIETEDO es el que tiene el nivel máximo de complejidad con respecto a los demás, pues involucra a todos los predicados del sistema, desde la unión de conjuntos hasta la solución de un tipo de ecuación diferencial ordinaria.
El predicado exacta	Este predicado reconoce si la ecuación es una ecuación diferencial exacta e intenta resolverla por este método.
El predicado separable	Este predicado resuelve ecuaciones diferenciales ordinarias por el método de variables separables las cuales se integran directamente.
El predicado Bernoulli	En este predicado se determina si la ecuación es una ecuación de Bernoulli y se intenta resolverla.

Tabla 5: Principales predicados de SIETEDO

4.3.El predicado integración

El sistema SIETEDO conoce cuatro métodos de integración utilizados en cálculo para resolver integrales que son el aplicar fórmula inmediata, por cambio de variable, integrando por partes e integración de funciones trigonométricas.

Se puede hacer la integración directamente llamando al predicado i(E.I), o se puede invocar al predicado integración el cual recibe la expresión de entrada, la simplifica, la analiza y posteriormente da una justificación de sus resultados.

Ejemplo:

```
?- i( x^2;x, 1 ).  
I = 2*x^1 ->  
yes
```

4.3.1.Integración inmediata y por cambio de variable

El método de integración inmediata se implementó utilizando un predicado por cada fórmula de integración de las consideradas en el apéndice A, utilizando la notación equivalente para cada expresión matemática en SIETEDO.

4.3.2.Por partes

El predicado por partes toma en cuenta las situaciones más usuales que se pueden presentar al intentar obtener una integral por este método.

Cabe mencionar que en este método de integración existen integrales que se tienen que evaluar volviendo a aplicar este método, así como aquellas en que en cierto momento se presentan iguales a la integral original, por lo que el sistema necesita reconocer el instante en el que se presenta esta situación, lo que hace ver a este predicado como una parte realmente "inteligente" del sistema.

Ejemplo:

```
?- porpartes( x*sen(x);x, U, DV:X ).  
U = x^1  
DV = 1*sen(x)  
X = x ->  
yes
```

4.3.2.1.El predicado por partes iterando

Este módulo le indica al método de integración por partes, que debe ser necesario conocer la integral para poder después hacer la manipulación algebraica necesaria.

Ejemplo:

?- porpartesite(sec(x)^3:x, U, DV:X).

U = 1*sec(x)^1

DV= sec(x)^2

X = x ->

yes

4.3.3.Integración de expresiones trigonométricas

Existen algunas integrales trigonométricas que se pueden resolver utilizando el método de integración por partes, con sólo disminuir la potencia de la función y utilizar alguna identidad trigonométrica apropiada. SIETEDO., utiliza este método en expresiones como seno, coseno y secantes elevadas a potencias. Existen otro tipo de funciones trigonométricas, en donde la integral es "casi directa" pues sólo basta hacer una transformación adecuada. SIETEDO. tratar de utilizar esta táctica siempre que le sea posible.

4.4.El predicado derivación

Para poder calcular cierto tipo de integrales es necesario muchas veces obtener la derivada de ciertas expresiones, por lo que es necesario que el sistema sepa derivar, además de ser el fundamento matemático necesario para hacer una integral.

En el caso muy específico de las fórmulas de derivación implementadas en SIETEDO, se tiene la regla de la cadena implícita en cada una (en caso de ser composición de funciones). También se puede invocar explícitamente a este predicado de la regla de la cadena.

Se puede realizar la derivada directamente al mandar a llamar al predicado derivar(E,X,R) e utilizar el predicado derivación, el cual además de simplificar da una explicación de como obtuvo la respuesta.

Ejemplo:

?- derivar(sen(3*c), c, DE).

DE = (3*1)*cos(3*c) ->

yes

4.5.Utilerías de SIETEDO

En la programación de SIETEDO se le considero utilerías a los predicados que eran necesarios para algunas operaciones algebraicas al momento de resolver la ecuación diferencial dada como son: *agrupa, polinomio, simplifica, reescribe*, etc.

4.5.1. Equivalente

En SIETEDO existe un predicado que se considera el más importante. Este predicado mejor conocido como "equiv" (de equivalente), presenta varias modalidades en su uso, entre la que destaca una modalidad diferente en la definición de argumentos en Prolog.

Se entenderá que una expresión se dice ser o estar "Semi-instanciada", si esta se forma de una combinación de átomos y variables del lenguaje Prolog[manual referencia], la cual se denotará con un astedisco.

Principales modalidades del equiv:

- (1) equiv(E1, E2).
- (2) equiv(E1, F*G).
- (3) equiv(E1, A+B).
- (4) equiv(E1, K*E2:V).

El caso más simple que resuelve el predicado equiv es hacer la comparación(caso(1)), esta se puede realizar con átomos, con funciones y con cualquier expresión en general válida para SIETEDO.

Ejemplo:

?- equiv(x^2, (x+0)^2).
yes

La instanciación de variables (casos (2),(3) y (4)), es uno de los pasos más delicados que se llevan a cabo dentro del predicado equiv, pues es en este momento cuando se tratan de identificar esquemas de expresiones, según los valores asignados a cada variable, por lo que una mala asignación de valores ocasionaría una respuesta errónea por parte del equiv.

Ejemplos:

?- equiv(x*exp(x^2):x, K*exp(V):V).
K = 2^~ 1
V = x^2 ->
yes

?- equiv(x^2, K*X^2).
K = 1
X = x ->
yes

?- equiv(5-tan(o), A+B).
A = 5
B = tan(o) ->
yes

Una de las modalidades más importantes del equív, es la de reconocer diferenciales, cuando es necesario equív se encarga de completar el diferencial, sin embargo es necesario que en la expresión $K \cdot E \cdot V$, en donde K es una constante y E una expresión, la variable V , se encuentre presente en la expresión E .

Ejemplos:

?- equív(cos(x):sen(x), K*cos(V):V).
yes

Aunque en primera instancia podría parecer que el problema de determinar por medios computables si dos expresiones matemáticas son equivalentes, es un problema sin solución finita. Para los tipos de expresiones que maneja SIETEDO, esto se lleva a cabo en un tiempo finito computable para la máquina[computabilidad] y en un tiempo bastante razonable.

4.5.2.El predicado simplifica

El predicado simplifica juega un papel muy crítico en cuanto eficiencia en el tiempo de respuesta de SIETEDO, pues se encarga de determinar en gran parte el tiempo que le lleva a cabo a los predicados que lo utilizan realizar su tarea. Sus acciones principales son el quitar ceros, reescribir expresiones etc.

Este predicado es capaz de simplificar diferenciales de tal forma que el segundo operando del ":", sea un átomo (la variable de integración) siempre que en la expresión original sea claro cual es la variable de integración.

Ejemplos:

Notación Matemática

$(x+1)^3 \ln(3x^2) / (3x^2)$

$x^3 + 1x$

Notación SETEIS

?- simplifica((x+1)^3*ln(3*x^2):(3*x^2), E2:X)
E2 = (x+1)^3*ln(3*x^2)*6*x;x ->
yes

?- simplifica(x^0+1*x, R)
R = 1+x ->
yes

4.5.3.El predicado multiplica

Al igual que simplifica, este predicado tiene una función crítica en los predicados que lo invocan, pues es en gran parte el que determina el tiempo de ejecución de dichos predicados. Su tarea es la de efectuar productos, Binomios de Newton etc. En el caso de binomios de Newton, para potencias mayores que 10, la eficiencia del predicado depende de la memoria de la computadora.

Ejemplo:

```
?- multiplica((x-2).(x+2), R ).
R = x^2-2*x+4 ->
yes
```

4.5.4.El predicado agrupa

Así como hay expresiones que crecen mucho al momento de hacer los cálculos es importante ir agrupando términos semejantes resultantes de la operación para después simplificarlos. En la realización de productos de polinomios este predicado juega un papel muy importante en el tiempo de su ejecución.

Su forma de trabajar, es descomponer la expresión en partes, después ordenarlas y finalmente agrupar los términos - semejantes, lo cual en ocasiones puede ser bastante demoroso.

Ejemplo:

```
?- agrupa( x^2+1*f-3*x^2, R ).
R = f-2*x^2->
yes
```

4.5.5.El predicado factoriza

Este predicado es uno de los más complejos además del "equiv" por que realiza la tarea de factorizar expresiones matemáticas, que en muchos casos llegan a ser difícil de realizar hasta para el mismo ser humano.

Ejemplo:

```
factoriza( 1, (t*x+t^2)/(t*(x+t),R).
R= t^0 *(x+t^1)/(t*x+1)->
yes
```

4.5.6.El predicado polinomio

El predicado polinomio recibe una expresión de entrada y determina si es un polinomio, en caso de serlo devuelve el grado del polinomio. Es importante notar que busca las variables en memoria dadas por el usuario y lo demás lo considera constante.

Ejemplo:

```
polinomio(x*x^2+n*x,G). %Considerando que "x" es la única variable
G=3->
yes
```

Capítulo V Uso del sistema

5.1.Requerimientos de trabajo

El sistema SIETEDO puede para trabajar en una computadora XT, AT o compatible con PC,bajo DOS teniendo como mínimo 640 KiloBytes de memoria RAM y con una versión mayor a la 3.0 de MS-DOS, así como un mínimo de 20 megas en disco. Sin embargo para mejor funcionalidad del mismo se recomienda 1 mega de memoria RAM.

5.2.El ambiente de MS-DOS

Existen dos formas de utilizar SIETEDO una es utilizando una computadora compatible con IBM-PC e invocarlo directamente desde el indicador del DOS, por medio del prompt del sistema. En el caso de tener la versión ejecutable que es el archivo con el nombre de "SIETEDO.EXE" sólo es necesario indicarlo así:

```
A>SIETEDO <- Pulsar Enter -|
```

La segunda forma de utilizarlo es bajo el intérprete de Arity/Prolog, bajo el ambiente MS-DOS (cualquier versión).

5.3.Compilación del sistema

Para compilar el sistema es necesario tener el programa *opc.exe* y las librerías necesarias de Arity versión 6.1.27. Si se desea tener el programa ejecutable(SIETEDO.ENE)después de compilarlo es necesario ligarlo con el programa *link.exe* de Microsoft.

5.5.El intérprete de Arity/Prolog

En la figura-5 se muestra la pantalla de entrada del intérprete de Arity/Prolog.

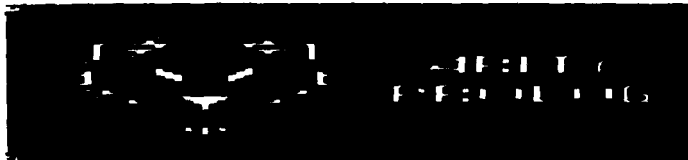


Figura 5: El intérprete de Arity/Prolog

SIETEDO fue diseñado bajo Arity/Prolog versión 6.1.27, por lo consiguiente se puede trabajar en esta versión o en versiones mayores a esta. Arity consta de varios "Buffers", donde se pueden editar programas, el número de "Buffers", queda determinado por la opción de FILES, contenida en el archivo de CONFIG.SYS del Sistema Operativo MS-DOS.

Para ejecutar el programa basta con ir a la opción de "File" escoger la opción de "Open File", elegirla y después ir a la opción de "Buffers" y indicar la opción de "Reconsult On Exit". Desde este momento puede invocar a los predicados desde el signo de interrogación que aparece en la ventana del intérprete así:



Figura 8: Consultado de un archivo en Arity/Prolog

Después de que se ha consultado el archivo, se tiene uno que cambiar al intérprete pulsando la tecla de **F8** y posteriormente se escribe "main" y así Prolog ejecutará inmediatamente el sistema.

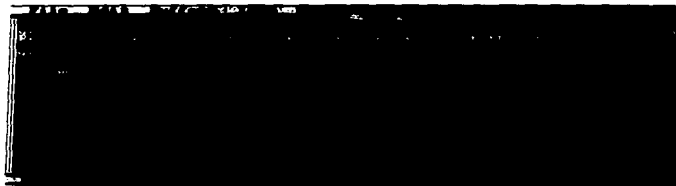


Figura 9: Ejecución del sistema desde el intérprete de Arity/Prolog

En la figura-10 se muestra el menú principal del sistema SIETEDO, el cual permite obtener ayuda sobre la notación utilizada para dar una expresión de entrada al sistema y sobre información general de como utilizar los menús. El sistema no esta restringido sólo a resolver y enseñar ecuaciones diferenciales ordinarias, también permite realizar integrales o derivadas.

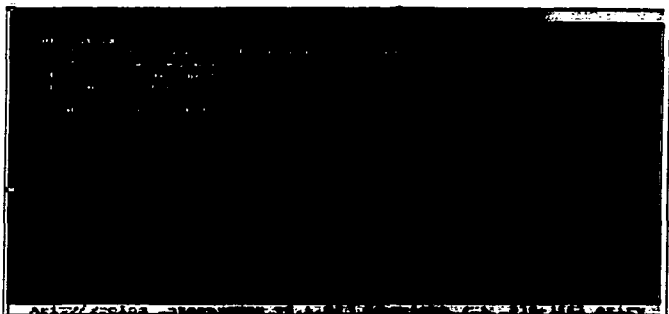


Figura 10: Menú principal de SIETEDO

En la figura-11 se le da como entrada al sistema la ecuación diferencial ordinaria $(\sin(xy) + xy \cos(xy))dx + x^2 \cos(xy)dy = 0$ la cual es resuelta en la ventana que tiene como título "Ecuaciones Diferenciales Ordinarias" y la explicación de la solución obtenida se da en otra ventana con título "Justificación".

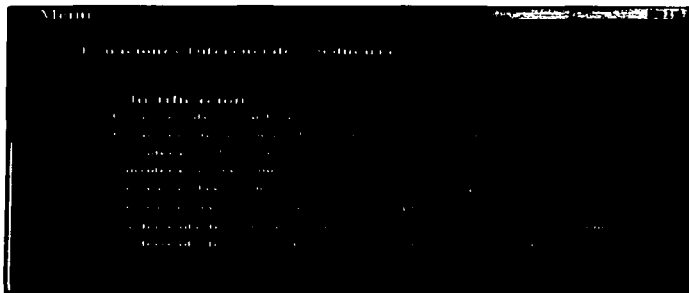


Figura 11: Respuesta del sistema

Conclusiones

El sistema SIETEDO es un sistema que se considera "Inteligente" o "Basado en Conocimiento" entre otras razones debido a que el objetivo de este sistema, no es aumentar sus fórmulas de integración, si no el hecho de que sea capaz de deducirlas partiendo de hechos básicos y además SIETEDO integra internamente de manera muy similar a la humana realiza el mismo proceso que se indica en los libros de cálculo[29,30,31,32]).

Otra de las razones por la cual se considera *inteligente* a este sistema radica en el hecho de que *no existe un algoritmo conocido* que resuelva este tipo de ecuaciones.

El tipo de integrales que maneja SIETEDO son solamente simbólicas e indefinidas, debido a que en los cursos de Métodos Numéricos que se imparten en las áreas de ingeniería o ciencias de cualquier universidad, se estudian algoritmos para calcular las integrales definidas sobre un intervalo que son fácilmente programables por cualquier estudiante de computación a nivel licenciatura. Es importante mencionar que SIETEDO hace cálculos numéricos muy elementales, debido a las restricciones que presenta el intérprete de Arity/Prolog. Se queda como un problema abierto el programar en algún otro lenguaje o el de utilizar un programa para hacer los cálculos numéricos y encontrar la forma de ligarlo con los predicados hechos para cálculos numéricos desde Arity/Prolog

En lo que se refiere a la interfaz del sistema esta es todavía muy sencilla y utilizando modo texto, por lo que en trabajos futuros es necesario hacer una "buena interfaz gráfica" que sea agradable para el usuario y le permita un buen desarrollo en el proceso de aprendizaje!

En la realización de SIETEDO se observó que los SHELL's para desarrollo de sistemas expertos necesitan considerar elementos numéricos y no únicamente simbólicos. Puedo concluir que el diseño de sistemas inteligentes no debe ser totalmente simbólico ni numérico, debe existir una flexibilidad en la representación del conocimiento del mundo tanto real como abstracto. Este problema se presenta de forma muy clara en el diseño de redes neuronales artificiales en donde se utilizan cálculos numéricos y no se consideran elementos simbólicos.

Una propuesta que dejo abierta en este trabajo es que se programe una RED NEURONAL ARTIFICIAL la cual se encargue de hacer un *preprocesamiento* de las funciones de entrada y las clasifique para después darselas de entrada al sistema experto y le sea más fácil encontrar la solución deseada.

Es importante denotar que se ha elegido el lenguaje de Arity/Prolog por ser el intérprete del lenguaje Prolog el más parecido al lenguaje estandar de Prolog de Edinburgo[1].

Turbo Prolog a diferencia de Arity/Prolog destruye el concepto simbólico que se maneja en los lenguajes de inteligencia artificial, el programador tiene que definir "tipos" para las variables utilizadas dentro del programa. Por otro lado Lisp aunque es un lenguaje muy popular en los Estados Unidos[5], no permite una fácil modularización de los programas por el uso excesivo de paréntesis y por consiguiente llega a ser muy confuso su código a diferencia de Arity Prolog.

La implementación en Prolog permitió un tiempo de respuesta mucho menor en comparación a otros programas comerciales que también resuelven ecuaciones diferenciales ordinarias de manera simbólica como es el caso del programa Maple V versión 4.0 y Mathematica versión 3.0.

Al igual que en el área de las matemáticas cuando se propone un nuevo conjunto, este debe de ser consistente con los ya existentes como es el caso de los números complejos.

SIETEDO maneja una notación consistente entre sus predicados y los términos matemáticos, lo cual le permite a cualquier aficionado a las matemáticas y a la notación utilizada por los paquetes matemáticos por computadora (como Maple V versión 4.0 y Matemática versión 3.0), una rápida familiarización con los términos utilizados en el sistema, sin tener un conocimiento del lenguaje Prolog ni saber programación lógica.

También queda abierta la opción de añadir nuevos predicados a SIETEDO utilizando los ya existentes, haciendolo crecer y volviendolo más robusto y poderoso., pero aún más que eso todavía, la estructura misma permite hacer modificaciones a los predicados ya existentes, de tal forma que el sistema no se tenga que enterarse de dichas modificaciones.

En el sistema SIETEDO no se desarrolló el modelo del estudiante, ya que esto implicaría la creación de otro sistema y queda como propuesta de tesis para trabajos futuros.

Los ITS se encuentran aún en pleno desarrollo por lo que aún pueden identificarse algunas fallas metodológicas; Rosenberg[4] menciona dos muy importantes: la primera es que los ITS's no han sido bien fundamentados en un modelo de aprendizaje; y la segunda se refiere al desarrollo de procedimientos de evaluación de la eficiencia del propio sistema.

Por otra parte, el hecho de que la investigación de los ITS se extienda sobre tres diferentes disciplinas (Ciencias de la computación IA, Psicología y Educación), significa que hay diferencias muy grandes en terminología, metas de investigación, marcos teóricos y distinto énfasis entre investigadores, además de que estos requieren de un entendimiento mutuo de las tres disciplinas involucradas. Los ITS están teniendo mayor aceptación como una disciplina práctica dentro de la inteligencia artificial.

La utilización de los sistemas ITS's sólo se logrará si se logran resolver los problemas anteriores (Sleeman y Brown[43]), para lo cual será necesario integrar a dichos sistemas, las teorías psicológicas del aprendizaje y la instrucción, y enfocar sus objetivos también a mejorar la adquisición, retención y recuperación de la información (por parte del estudiante), haciendo conexiones directas entre las variables instruccionales y las del aprendizaje (Hajovi y Christensen 1987).

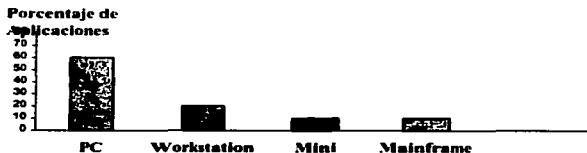


Figura-7: Plataformas utilizadas para sistemas inteligentes

En lo que se refiere al interés de que ha sido el sistema SIETEDO podemos mencionar que ha sido aceptado en simposiums internacionales de inteligencia artificial y de enseñanza de las matemáticas, como el "IX International Symposium on Artificial Intelligence and VI International Conference on Industrial Fuzzy Control and Intelligent Systems", celebrado en la ciudad de CANCUN, México, del 12 al 15 de Noviembre de 1996 y el "V Simposio Internacional en Educación Matemática Elfriede Wenzelburger", celebrado en la ciudad de México del 16 al 18 de Octubre de 1995.

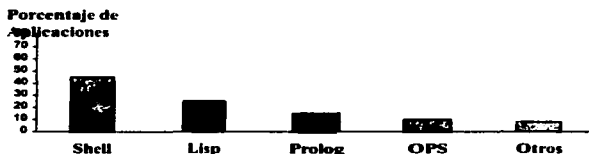


Figura-8: Software utilizado en el desarrollo de sistemas inteligentes

En la figura-8 se muestra el porcentaje de aplicaciones realizadas en diferentes softwares, de donde se observa que sólo el 20% de las aplicaciones han sido hechas en lenguaje Prolog y la mayoría utilizando un Shell para sistemas expertos.

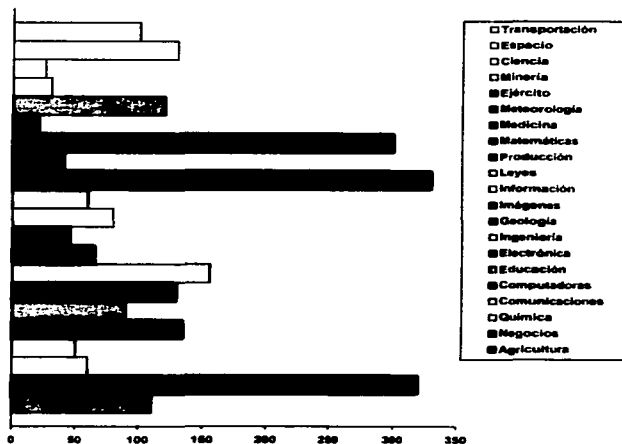


Figura-9: Número de sistemas expertos utilizados en diferentes áreas

En la figura-9 se observa que se han desarrollado pocas aplicaciones en matemáticas y en el área educativa todavía se han realizado menos, lo cual manifiesta la necesidad de desarrollar más programas educativos inteligentes. En el caso particular del sistema SIETEDO se ha puesto a prueba con alumnos de licenciatura (Centro de Estudios Universitarios Londres) y se ha observado que el sistema ha logrado motivar al alumno en su proceso de aprendizaje en el área del cálculo diferencial e integral y motivado el uso de la computadora como una herramienta personal de aprendizaje, por lo que puedo concluir que se debe continuar trabajando en el mejoramiento y crecimiento de este sistema.

Apéndice A Tácticas y métodos de integración utilizados por SIETEDO

A.1. Tácticas generales de integración

Las tácticas más utilizadas por *SIETEDO* para intentar resolver una integral son:

- Simplificar al máximo la expresión de entrada antes de realizar cualquier otra acción.
- Utilizar exponentes fraccionarios o negativos.
- Factorizar lo más posible.
- Realizar operaciones algebraicas (división de polinomios, reducción de términos numéricos, etc.).
- Aplicar identidades trigonométricas.
- Método de sustitución de variables.

Hacer $u = ?$ y obtener du , después hacer el cambio correspondiente.

A.2. Terminología

Para las fórmulas que se presentan en este documento se considerará la siguiente notación: a las letras u , v y w como funciones cualquiera sobre los números reales y a la letra n como cualquier número entero o real, a la letra e como una constante definida y a la letra k como una constante indefinida.

A.3. Propiedades de las integrales

$$\int d u = u + k$$
$$\int (u d u + v d v - w d w) = \int u d u + \int v d v - \int w d w$$
$$\int k u d u = k \int u d u$$

A.4. Fórmulas de integración inmediata

$$\int d u = u + k$$
$$\int u^n d u = \frac{u^{n+1}}{n+1} + k, \quad \forall n \neq -1$$
$$\int \frac{d u}{u} = \ln(u) + k$$
$$\int c^{u d u} = \frac{c^u}{\ln(c)} + k$$
$$\int e^{u d u} = e^u + k$$

A.5. Método de integración por partes

$$\int u \, dv = uv - \int v \, du$$

A.6. Integración de funciones trigonométricas

$$\int \operatorname{sen}(u) \, du = -\cos(u) + k$$

$$\int \cos(u) \, du = \operatorname{sen}(u) + k$$

$$\int \tan(u) \, du = -\ln|\cos(u)| + k = \ln|\sec(u)| + k$$

$$\int \sec^2(u) \, du = \tan(u) + k$$

$$\int \sec(u) \tan(u) \, du = \sec(u) + k$$

$$\int \operatorname{cosec}(u) \cot(u) \, du = -\operatorname{cosec}(u) + k$$

$$\int \cot(u) \, du = \ln|\operatorname{sen}(u)| + k$$

$$\int \sec(u) \, du = \ln|\sec(u) + \tan(u)| + k$$

$$\int \operatorname{cosec}(u) \, du = \ln|\operatorname{cosec}(u) - \cot(u)| + k$$

$$\int \operatorname{cosec}^n(u) \, du = -\frac{1}{n-1} \operatorname{cosec}^{n-2}(u) \cot(u) + \frac{n-2}{n-1} \int \operatorname{cosec}^{n-2}(u) \, du$$

$$\int \operatorname{cosec}^2(u) \, du = -\cot(u) + k$$

$$\int u^n \cos(cu) \, du = \frac{u^n}{c} \operatorname{sen}(cu) - \frac{n}{c} \int u^{n-1} \operatorname{sen}(cu) \, du$$

$$\int u^n \operatorname{sen}(cu) \, du = -\frac{u^n}{c} \cos(cu) + \frac{n}{c} \int u^{n-1} \cos(cu) \, du$$

$$\int \operatorname{sen}^m(u) \cos^n(u) \, du = \frac{\operatorname{sen}^{m+1}(u) \cos^{n-1}(u)}{m+n} + \frac{n-1}{m+n} \int \operatorname{sen}^m(u) \cos^{n-2}(u) \, du$$

Fórmulas que involucran $\sqrt{c^2 + u^2}$

$$\int \frac{du}{c^2 + u^2} = \frac{1}{c} \operatorname{angtan} \frac{u}{c} + k$$

$$\int \sqrt{c^2 + u^2} \, du = \frac{u}{2} \sqrt{c^2 + u^2} + \frac{c^2}{2} \ln|u + \sqrt{c^2 + u^2}| + k$$

$$\int u^2 \sqrt{c^2 + u^2} \, du = \frac{u}{8} (c^2 + 2u^2) \sqrt{c^2 + u^2} - \frac{c^4}{8} \ln|u + \sqrt{c^2 + u^2}| + k$$

$$\int \frac{\sqrt{c^2 + u^2}}{u} du = \sqrt{c^2 + u^2} - c \ln \left| \frac{c - \sqrt{c^2 + u^2}}{u} \right| + k$$

$$\int \frac{du}{\sqrt{c^2 + u^2}} = \ln |u + \sqrt{c^2 + u^2}| + k$$

$$\int \frac{du}{u\sqrt{c^2 + u^2}} = -\frac{1}{c} \ln \left| \frac{\sqrt{c^2 + u^2} + c}{u} \right| + k$$

Fórmulas que involucran $\sqrt{c^2 - u^2}$

$$\int \frac{du}{c^2 - u^2} = \frac{1}{2c} \ln \left| \frac{c+u}{c-u} \right| + k$$

$$\int \frac{du}{\sqrt{c^2 - u^2}} = \text{ang sen} \left(\frac{u}{c} \right) + k$$

$$\int \sqrt{c^2 - u^2} du = \frac{u}{2} \sqrt{c^2 - u^2} + \frac{c^2}{2} \text{ang sen} \left(\frac{u}{c} \right) + k$$

$$\int \frac{du}{c^2 - u^2} = \frac{1}{2c} \ln \left| \frac{c+u}{c-u} \right| + k$$

Fórmulas que involucran $\sqrt{u^2 + c^2}$

$$\int \frac{du}{u^2 + c^2} = \frac{1}{c} \text{ang tan} \left(\frac{u}{c} \right) + k$$

$$\int \sqrt{u^2 + c^2} du = \frac{u}{2} \sqrt{u^2 + c^2} + \frac{c^2}{2} \ln(u + \sqrt{u^2 + c^2}) + k$$

Fórmulas que involucran $\sqrt{u^2 - c^2}$

$$\int \frac{du}{u^2 - c^2} = \frac{1}{2c} \ln \left| \frac{u-a}{u+a} \right| + k$$

$$\int \frac{du}{u\sqrt{u^2 - c^2}} = \frac{1}{c} \text{ang sec} \left(\frac{u}{c} \right) + k$$

$$\int \sqrt{u^2 - c^2} du = \frac{u}{2} \sqrt{u^2 - c^2} - \frac{c^2}{2} \ln |u + \sqrt{u^2 - c^2}| + k$$

A.7. Tácticas para reducción trigonométrica

Tipo de Integral

Caso

Aplicar

$$\int \sin^n(u) du$$

Si $n > 2$ y entero

$$-\frac{\sin^{n-1}(u) \cos(u)}{n} + \frac{n-1}{n} \int \sin^{n-2}(u) du$$

$\int \sin^2(u) du$	Si $n=2$ y entero	$\int \left(\frac{1}{2} - \frac{1}{2} \cos(2u)\right) du$
$\int \cos^2(u) du$	Si $n=2$ y entero	$-\frac{\cos^{n-1}(u) \sin(u)}{n} + \frac{n-1}{n} \int \cos^{n-2}(u) du$
$\int \cos^2(u) du$	Si $n=2$ y entero	$\int \left(\frac{1}{2} + \frac{1}{2} \cos(2u)\right) du$
$\int \tan^2(u) du$	Si $n=2$ y entero	$\int (\sec^2(u) - 1) du$
$\int \cot^2(u) du$	Si $n=2$ y entero	$\int (\operatorname{cosec}^2(u) - 1) du$
$\int \sin^n(u) \cos^m(u) du$	Si m o n positivo e impar disminuir en uno el exponente impar y aplicar la identidad trigonométrica: $\sin^2(u) + \cos^2(u) = 1$	
$\int \cot^n(u) du$	Si n es entero y positivo, par o impar, disminuir n en $n-2$ y aplicar la identidad trigonométrica: $\cot^2(u) = \operatorname{cosec}^2(u) - 1$.	
$\int \tan^n(u) du$	Si n es entero y positivo, par o impar, disminuir n en $n-2$ y aplicar la identidad trigonométrica: $\tan^2(u) = \sec^2(u) - 1$.	
$\int \sec^n(u) du$	Si n es positivo y par, disminuir n en -2 y aplicar la identidad trigonométrica: $\sec^2(u) = 1 + \tan^2(u)$.	
$\int \operatorname{cosec}^n(u) du$	Si n es positivo y par, disminuir n en -2 y aplicar la identidad trigonométrica: $\operatorname{cosec}^2(u) = 1 + \cot^2(u)$.	
$\int \tan^m(u) \sec^n(u) du$	Si n es positivo y par utilizar el caso anterior. Si m es positivo e impar disminuir m en $m-1$ y luego aplicar: $\tan^2(u) = \sec^2(u) - 1$. Conservar $\sec(u) \tan(u) du$.	
$\int \cot^m(u) \operatorname{cosec}^n(u) du$	Si n es positivo y par utilizar el caso anterior. Si m es positivo e impar disminuir m en $m-1$ y luego aplicar: $\tan^2(u) = \sec^2(u) - 1$. Conservar $\sec(u) \tan(u) du$.	
$\int \sin^n(u) \cos^m(u) du$	Si n y m son enteros positivos y pares.	
utilizar:	$\begin{cases} \sin^2(u) = \frac{1}{2} - \frac{1}{2} \cos(2u) \\ \cos^2(u) = \frac{1}{2} + \frac{1}{2} \cos(2u) \\ \sin(u) \cos(u) = \frac{1}{2} \sin(2u) \end{cases}$	$\text{además } \begin{cases} \sin(2u) = 2 \sin(u) \cos(u) \\ \cos(2u) = \cos^2(u) - \sin^2(u) \end{cases}$

Apendice B Ecuaciones diferenciales ordinarias

B.1. Conceptos básicos de ecuaciones diferenciales ordinarias

Se llama ecuación diferencial a una ecuación que liga la variable independiente x , la función incógnita $y = y(x)$ y sus derivadas $y', y'', \dots, y^{(n)}$. Es decir una ecuación de la forma:

$$F(x, y, y', y'', \dots, y^{(n)}) = 0$$

En otras palabras, se llama *ecuación diferencial* a una ecuación en la que figura la derivada o diferencial de la función incógnita. Si la función incógnita $y = y(x)$ depende de una sola variable independiente x , la ecuación diferencial se llama *ordinaria*. Por Ejemplo:

$$1) \frac{dy}{dx} + xy = 0.$$

$$2) y'' + y' + x = \cos(x).$$

$$3) (x^2 + y^2)dx - (x + y)dy = 0.$$

El orden de una ecuación diferencial es el de la derivada de mayor orden que figura en la ecuación. Por ejemplo: la ecuación diferencial $y' + xy = e^x$ es de primer orden; la ecuación diferencial $y'' + p(x)y = 0$, en donde $p(x)$ es una función dada, es de 2º orden; la ecuación diferencial $y^{(n)} + x y^{(n-1)} + x = x^2$ es de n º orden.

Se dice que una solución de la ecuación diferencial es una función $y = \phi(x)$ determinada en el intervalo (a, b) junto con sus derivadas sucesivas hasta el orden n inclusive, tal que al hacer la sustitución $y = \phi(x)$ en la ecuación diferencial, ésta, se convierte en una identidad con respecto a x en el intervalo (a, b) .

Por ejemplo, la función $y = \sin x + \cos x$ es solución de la ecuación $y'' + y = 0$. En efecto, derivando dos veces esta función, se tiene:

$$y' = \cos x - \sin x, \quad y'' = -\sin x - \cos x$$

Sustituyendo en la ecuación diferencial y'' e y por sus expresiones, resulta la identidad:

$$-\sin x - \cos x + \sin x + \cos x \equiv 0.$$

La gráfica de una solución de la ecuación diferencial se denomina curva integral de la ecuación

La forma general de una ecuación de primer orden es:

$$f(x, y, y') = 0 \quad (1)$$

Si en la ecuación (1) es posible despejar y' , resulta

$$y' = f(x, y),$$

que representa una ecuación de primer orden, resuelta con respecto a la derivada.

B.2. Ecuaciones con variables separables y ecuaciones reducibles a ellas

La ecuación diferencial de la forma:

$$\varphi(y)dy = f(x)dx \quad (1)$$

se llama *ecuación con variables separadas*.

Las ecuaciones de la forma:

$$\varphi_1(x)\psi_1(y)dx = \varphi_2(x)\psi_2(y)dy$$

en las que los coeficientes de las diferenciales se descomponen en factores que dependen solamente de x o solamente de y , se llaman *ecuaciones con variables separables*.

Dividiendo por el producto $\psi_1(y)\varphi_2(x)$ éstas se reducen a ecuaciones con variables separadas:

$$\frac{\varphi_1(x)}{\varphi_2(x)} dx = \frac{\psi_2}{\psi_1} dy.$$

La integral general de esta ecuación

$$\int \frac{\varphi_1(x)}{\varphi_2(x)} dx - \int \frac{\psi_2(y)}{\psi_1(y)} dy = C.$$

Observación. La división por $\psi_1(y)\varphi_2(x)$ puede dar lugar a que se pierdan las soluciones particulares que anulan al producto $\psi_1(y)\varphi_2(x)$.

La ecuación diferencial de la forma:

$$\frac{dy}{dx} = f(ax + by + c),$$

donde a , b y c son constantes, se reduce a una ecuación con variables separables haciendo la sustitución

$$z = ax + by + c.$$

Ejemplo. Resolver la ecuación

$$3e^{-x} \operatorname{tg} y \, dx + (2 - e^{-x}) \sec^2 y \, dy = 0.$$

Solución. Dividimos ambos miembros de la ecuación por el producto $\operatorname{tg} y \cdot (2 - e^{-x})$

$$\frac{3e^{-x} dx}{2 - e^{-x}} + \frac{\sec^2 y \, dy}{\operatorname{tg} y} = 0.$$

Ha resultado una ecuación con variables separadas. Integrándola hallamos:

$$-3 \ln|2 - e^{-x}| + \ln|\operatorname{tg} y| = C_1.$$

Efectuando la potenciación, obtenemos:

$$\frac{|\operatorname{tg} y|}{|2 - e^{-x}|^3} = e^{C_1}, \text{ o bien,}$$

De aquí.

$$\frac{\operatorname{tg} y}{(2 - e^{-x})^3} = \pm e^{C_1}.$$

Designando $\pm e^{C_1} = C$, se tiene:

$$\frac{\operatorname{tg} y}{(2 - e^{-x})^3} = C, \text{ o bien, } \operatorname{tg} y - C(2 - e^{-x})^3 = 0.$$

Hemos obtenido la integral general de la ecuación dada.

Al dividir por el producto $\operatorname{tg} y (2 - e^{-x})$ se suponía que ninguno de los factores se convertía en cero. Igualando cada factor a cero, obtenemos, respectivamente:

$$y = k\pi \quad (k = 0, \pm 1, \pm 2, \dots), \quad x = \ln 2.$$

Sustituyendo en la ecuación inicial comprobamos que $y = k\pi$ y $x = \ln 2$ son soluciones de esta ecuación. Estas pueden obtenerse formalmente de la integral general haciendo $C = 0$ y $C = \infty$. Esto significa que la constante C se sustituye por $\frac{1}{C_2}$, después de lo cual, la integral general toma la forma:

$$\operatorname{tg} y - \frac{1}{C_2} (2 - e^{-x})^3 = 0, \text{ o bien, } C_2 \operatorname{tg} y - (2 - e^{-x})^3 = 0;$$

haciendo en la última igualdad $C_2 = 0$, lo que corresponde a $C = \infty$, tendremos $(2 - e^{-x})^3 = 0$; de aquí obtenemos la solución $x = \ln 2$ de la ecuación inicial. En consecuencia, las funciones $y = k\pi$ ($k = 0, \pm 1, \pm 2, \dots$) y $x = \ln 2$ son soluciones particulares de la ecuación dada. Por consiguiente, el resultado final es:

$$\operatorname{tg} y - C(2 - e^{-x})^3 = 0.$$

B.3. Ecuaciones homogéneas y reducibles a ellas

Una función $f(x, y)$ es homogénea de grado n en sus argumentos si se cumple la identidad:

$$f(tx, ty) = t^n f(x, y)$$

Por ejemplo, $f(x, y) = x^2 + y^2 - xy$ es una función homogénea de segundo grado, puesto que $f(tx, ty) = (tx)^2 + (ty)^2 - (tx)(ty) = t^2(x^2 + y^2 - xy) = t^2 f(x, y)$.

Para $n = 0$, se tiene una función de grado cero. Por ejemplo, $f(x, y) = \frac{x^2 - y^2}{x^2 + y^2}$ es una función homogénea de grado cero, puesto que

$$f(tx, ty) = \frac{(tx)^2 - (ty)^2}{(tx)^2 + (ty)^2} = \frac{t^2(x^2 - y^2)}{t^2(x^2 + y^2)} = \frac{x^2 - y^2}{x^2 + y^2} = f(x, y)$$

Una ecuación diferencial de la forma $\frac{dy}{dx} = f(x, y)$ se llama homogénea si $f(x, y)$ es una función homogénea de grado cero en sus argumentos. La ecuación homogénea siempre se puede representar en la forma:

$$\frac{dy}{dx} = \varphi\left(\frac{y}{x}\right). \quad (1)$$

Introduciendo una nueva función incógnita $u = \frac{y}{x}$, la ecuación (1) se reduce a la ecuación con variables separables:

$$x \frac{du}{dx} = \varphi(u) - u.$$

Si $u = u_0$ es una raíz de la ecuación $\varphi(u) - u = 0$, la solución de la ecuación homogénea es, $u = u_0$ o bien, $y = u_0 x$ (recta que pasa por el origen de coordenadas).

Observación. Al resolver las ecuaciones homogéneas no es indispensable reducirlas a la forma (1). Se puede hacer inmediatamente la sustitución $y = ux$.

Las ecuaciones de la forma:

$$\frac{dy}{dx} = f\left(\frac{a_1 x + b_1 y + c_1}{a_2 x + b_2 y + c_2}\right)$$

Se reducen a homogéneas trasladando el origen de coordenadas al punto (x_0, y_0) de intersección de las rectas:

$$a_1 x + b_1 y + c_1 = 0 \quad \text{y} \quad a_2 x + b_2 y + c_2 = 0.$$

Esto se consigue haciendo la sustitución de las variables:

$$x = \xi + x_0, \quad y = \eta + y_0$$

El método indicado no es aplicable cuando las rectas $a_1x + b_1y + c_1 = 0$ y $a_2x + b_2y + c_2 = 0$ son paralelas. Pero, en este caso,

$$\frac{a_2}{a_1} = \frac{b_2}{b_1} = \lambda,$$

y la ecuación (2) se puede escribir en la forma

$$\frac{dy}{dx} = f\left[\frac{a_1x + b_1y + c_1}{\lambda(a_1x + b_1y) + c_2}\right] = F(a_1x + b_1y),$$

Que se puede reducir a una ecuación con variables separables haciendo la sustitución $z = a_1x + b_1y$.

Si la ecuación diferencial viene expresada en la forma:

$$P(x, y)dx + Q(x, y)dy = 0,$$

Será homogénea si $P(x, y)$ y $Q(x, y)$ son funciones homogéneas de un mismo grado.

A veces, la ecuación se puede reducir a homogénea mediante la sustitución de la variable

$y = z^\alpha$. Esto ocurre cuando todos los términos de la ecuación son de un mismo grado, atribuyendo el grado 1 a la variable x , el grado α a la variable y , y el grado $\alpha - 1$ a la derivada $\frac{dy}{dx}$.

Ejemplo. Resolver la ecuación

$$xy' = \sqrt{x^2 - y^2} + y.$$

Solución. Escribamos la ecuación en la forma:

$$y' = \sqrt{1 - \left(\frac{y}{x}\right)^2} + \frac{y}{x}.$$

Como la ecuación es homogénea, hacemos $u = \frac{y}{x}$ o bien, $y = ux$. Entonces, $y' = xu' + u$.

Sustituyendo en la ecuación las expresiones para y e y' , obtenemos

$$x \frac{du}{dx} = \sqrt{1 - u^2}.$$

Separamos las variables:

$$\frac{du}{\sqrt{1-u^2}} = \frac{dx}{x}.$$

De aquí, integrando obtenemos:

$$\arcsen u = \ln|x| + \ln C_1 \quad (C_1 > 0), \text{ o bien, } \arcsen u = \ln C_1 x.$$

Como $C_1 x = \pm C_1 x$, haciendo la notación $\pm C_1 = C$, obtenemos $\arcsen u = \ln Cx$, donde

$|\ln Cx| \leq \frac{\pi}{2}$, o bien, $e^{-\frac{\pi}{2}} \leq Cx \leq e^{\frac{\pi}{2}}$. Sustituyendo u por $\frac{y}{x}$ tendremos la integral general

$$\arcsen \frac{y}{x} = \ln Cx.$$

Por consiguiente, la solución general es: $y = x \operatorname{sen} \ln Cx$.

Al separar las variables dividimos ambos miembros de la ecuación por el producto $x\sqrt{1-u^2}$, por lo cual, se podrían perder las soluciones que convierten en cero sus factores. Pongamos ahora $x = 0$ y $\sqrt{1-u^2} = 0$.

Pero $x = 0$ no es solución de la ecuación, debido a lo cual resulta, $1 - \frac{y^2}{x^2} = 0$, de donde $y = \pm x$.

Con una prueba directa nos convencemos de que las funciones $y = -x$ e $y = x$ son soluciones de la ecuación. Estas son soluciones singulares de la ecuación dada.

B.4. Ecuaciones lineales de primer orden. Ecuaciones de Bernoulli

Se llama ecuación diferencial lineal de primer orden a la que es lineal con respecto a la función incógnita y su derivada. Esta tiene la forma:

$$\frac{dy}{dx} + p(x)y = q(x), \quad (1)$$

en donde $p(x)$ y $q(x)$ son funciones continuas de x en la región en que se pida integrar la ecuación (1).

Si $q(x) = 0$, la ecuación (1) se llama lineal no homogénea. Si $q(x) \neq 0$, se dice que la ecuación (1) es lineal homogénea. Esta última es una ecuación con variables separables y posee la solución general:

$$y = ce^{-\int p(x)dx} \quad (2)$$

La solución general de la ecuación lineal no homogénea se puede hallar por el método de variación de la constante, según el cual se busca una solución de la ecuación (1) de la forma:

$$y' = c(x)e^{-\int p(x)dx}$$

donde $c(x)$ es una función incógnita nueva de x .

La ecuación (1) se puede integrar también del modo siguiente. Hacemos :

$$y' = u(x)v(x). \quad (3)$$

Poniendo (3) en (1), después de las transformaciones obtenemos:

$$u'v + u(pv + v') = q(x) \quad (4)$$

Determinando $v(x)$ de la condición $v' + pv = 0$, hallamos después la función $u(x)$ resolviendo la ecuación (4), obteniendo, por consiguiente, la solución $y = uv$ de la ecuación (1). En este caso, $v(x)$ es una solución particular cualquiera de la ecuación $v' + pv = 0$ (distinta de la solución trivial $v = 0$).

Observación. Puede ocurrir que la ecuación diferencial sea lineal respecto a y , considerada esta variable como función de x .

La forma normal de tal ecuación es: $\frac{dy}{dx} + p(y)x = \varphi(y)$.

Ejemplo: Resolver la ecuación:

$$y' + 2xy = 2xe^{-x^2}.$$

Solución. Aplicando el método de variación de la constante, consideramos la ecuación homogénea:

$$y' + 2xy = 0,$$

correspondiente a la ecuación no homogénea dada. Esta es una ecuación con variables separables. Su solución general tiene la forma:

$$y = ce^{-x^2}$$

Buscamos la solución general de la ecuación no homogénea en la forma

$$y = c(x)e^{-x^2} \quad (6)$$

donde $c(x)$ es una función incógnita de x .

Poniendo (6) en (5), obtenemos $c'(x) = 2x$. De donde $c(x) = x^2 + c$. Resumiendo, la solución general de la ecuación no homogénea es:

$$y = (x^2 + c)e^{-x^2}$$

donde C es la constante de integración.

Ecuación de Bernoulli

Una ecuación de la forma:

$$\frac{dy}{dx} + p(x)y = q(x)y^n \quad (8)$$

donde $n \neq 0; (1)$, puesto que para $n = 0$ y $n = 1$ esta ecuación es lineal. La ecuación (8) se reduce a una ecuación lineal haciendo la sustitución $z = \frac{1}{y^{n-1}}$. Pero resulta más conveniente resolver la ecuación de Bernoulli haciendo la sustitución (sin reducirla a lineal) $y = u(x)v(x)$.

Ejemplo. Resolver la ecuación de Bernoulli:

$$xy' + y = y^2 \ln x.$$

Solución. Hagamos $y = u(x)v(x)$. Tendremos

$$xuv' + u(xv' + v) = u^2v^2 \ln x.$$

Hallamos la función $y = u(x)v(x)$ como solución particular de la ecuación $xv' + v = 0$.

Resulta, $v(x) = \frac{1}{x}$. Entonces, $u' = \frac{u^2}{x} \ln x$. Separando las variables e integrando, obtenemos:

$$-\frac{1}{u} = \int \frac{\ln x}{x^2} dx = -\frac{\ln x}{x} - \frac{1}{x} - c,$$

o sea,

$$u = \frac{x}{1 + cx + \ln x}$$

La solución general de la ecuación es:

$$y = \frac{1}{1 + cx + \ln x}.$$

B.5. Ecuaciones diferenciales exactas. Factor integrante

La ecuación diferencial de la forma:

$$M(x, y)dx + N(x, y)dy = 0 \quad (1)$$

Se llama ecuación diferencial exacta, si su primer miembro es la diferencial total de una función $u(x, y)$:

$$M dx + N dy = du = \frac{\partial u}{\partial x} dx + \frac{\partial u}{\partial y} dy$$

La condición necesaria y suficiente para que la ecuación (1) sea una ecuación diferencial exacta: es que se cumpla la condición:

$$\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x} \quad (2)$$

(en un recinto simplemente conexo D de variación de x, y). La integral general de la ecuación (1) tiene la forma $u(x, y) = C$, o bien,

$$\int_{x_0}^x M(x, y) dx + \int_{y_0}^y N(x_0, y) dy = C.$$

Ejemplo. Resolver la ecuación diferencial:

$$(\operatorname{sen} xy + xy \cos xy) dx + x^2 \cos xy dy = 0.$$

Solución. Comprobemos que la ecuación dada es una ecuación diferencial exacta.

Se tiene que:

$$\frac{\partial M}{\partial y} = \frac{\partial}{\partial y} (\operatorname{sen} xy + xy \cos xy) = x \cos xy + x \cos xy - x^2 y \operatorname{sen} xy = 2x \cos xy - x^2 y \operatorname{sen} xy.$$

$$\frac{\partial N}{\partial x} = \frac{\partial}{\partial x} (x^2 \cos xy) = 2x \cos xy - x^2 y \operatorname{sen} xy.$$

o sea,

$$\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}.$$

Como vemos, se cumple la condición (2). Por consiguiente,

$$u(x, y) = \int (\operatorname{sen} xy + xy \cos xy) dx + \int x_0^2 \cos x_0 y dy =$$

$$= x \operatorname{sen} xy \Big|_{x_0}^x + x_0 \operatorname{sen} x_0 y \Big|_{x_0}^x = x \operatorname{sen} xy - x_0 \operatorname{sen} x_0 y_0;$$

de modo que

$$x \operatorname{sen} xy = C + x_0 \operatorname{sen} x_0 y_0, \text{ o bien, } x \operatorname{sen} xy = C_1.$$

Al resolver algunas ecuaciones diferenciales se puede agrupar los términos de tal modo que resulten combinaciones fáciles de integrar.

Referencias

- [1] Giarratano, Joseph (University of Houston, Clear Lake), and Gary Riley (NASA, Johnson Space Center), "*Expert System, Principles and Programming*". PWS Publishing Company: 20 Park Plaza, BOSTON, MA 02116, CLIPS Version 6.0, Second Edition, 1995.
- [2] Clocksin & Mellish, "*Programming in Prolog*". Berlin, Springer-Verlag, 3rd. Edition.
- [3] Obregón, Sánchez Arturo y Sergio Marcellin Jacques, "*Inteligencia Artificial y Sistemas Expertos*", Notas de la maestría en ciencias de la computación de la UACPyP del CCH con sede en el IMAS-UNAM, Junio 1994.
- [4] Marcellin, Jacques Sergio et al., "*Conocimiento*". Notas de la maestría en ciencias de la computación de la UACPyP del CCH con sede en el 4to. piso del edificio IMAS, UNAM, Ciudad Universitaria, Enero, 1996.
- [5] Durkin, John, "*Expert Systems Catalog of Applications*". Intelligent Computer Systems Inc., University of Akron Printing Department, Akron, OH, 1993.
- [6] Covington, Michael A, Donald Nute and Andre Vellino, "*Prolog Programming in DEPTH*". Copyright 1988 Scott, Foresman & Co.
- [7] Deyi Li, Ph. D. Department, "*A Prolog Database System*".
- [8] Nilsson, Ulf and Jan Matuszyn'ski, "*Logic Programming and Prolog*", Ed Wiley.
- [9] Sterling, Leon & Ehud Shapiro, "*The Art of Prolog, Advanced Programming Techniques*", The MIT Press Cambridge, Massachusetts London, England.
- [10] Bratko, Ivan, "*Prolog Programming for Artificial Intelligence*". Second edition, Addison Wesley, 1990.
- [11] Winston, Patrick Henry, "*Artificial Intelligence*". Addison Wesley, Third Edition, 1992.
- [12] Nilsson, Nils J., "*Principles of Artificial Intelligence*". Tioga publishing company palo alto, california
- [13] Rich, Elaine and Kevin Knight, "*Inteligencia Artificial*", McGraw-Hill, Segunda edición, 1994.
- [14] Slagle, James R., "*A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus*", in Computers and Thought, edited by Edward A. Feigenbaum and Julian Feldman, McGraw-Hill, New York, 1965. Based on a PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1961. (Also in JACM, 1963, vol. 10, 507-520).

- [15] Giannesini, Francis, Henry Kanoui, Robert Pasero, Michel van Caneghem. "*Prolog Reading*". Addison-Wesley, 1986.
- [16] Marcus, "*Prolog Programming*". Addison-Wesley, 1986.
- [17] Rogers Jean B., "*A Prolog Primer*". Addison-Wesley, 1986.
- [18] J.P. Sánchez y Beltrán, "*Sistemas Expertos, Una metodología de Programación*". Macrobit TM.
- [19] K.L. Clark and S.-A. Tammlund, "*Logic Programming*". Academic Press.
- [20] Campbell, J.A. "*Implementations of Prolog*", New York. John Wiley & Sons, 1984.
- [21] Caneghem, Michelvan, David H.D. Warren, Norwood, "*Logic Programming and Its Applications*". New Jersey: Ablex Publishing, 1986.
- [22] G. Cleary John, "*Logical Arithmetic*", Future Computing Systems, Volume 2, Number 2, 1987.
- [23] Lee, J.H.M. and M.H. Van Emden, "*Numerical Computation can be Deduction in CHIP*". Logic Programming Laboratory, Technical Report LP-19 (DCS-184-IR), January 1, 1992. Submitted to the Journal of Logic Programming.
- [24] Clark, K.L., S.-A. Tammlud, "*Logic Programming*", New York. Academic Press, 1982.
- [25] Kowalski, Robert, "*Logic for Problem Solving*", New York, El-sevier North Holland, 1979.
- [26] Wolfram, Sihepen "*MATHEMATICA*", A system for doing mathematics by computer, Redwood city, California, Addison-Wesley, 1988.
- [27] L. Abell, Martha and James P. Braselton, "*Differential Equations with MATHEMATICA*". AP Profesional, Academic Press Order Fulfillment Department, Sea Harbor, Orlando, 1993.
- [28] Argabright, L.N., R.C. Busby, "*CALCULUS WORKBOOK USING MAPLE*". MAPLE V versión 4.0, Kendall-Hunt, 1994. ISBN, 0-7872-0081-6.
- [29] Michael Spivak, "*CALCULUS*", Cálculo Infinitesimal. Reverté, S.A. Tomo II.
- [30] Haaser, LaSalle y Sullivan, "*Análisis Matemático*". Curso de Introducción Vol. I, Trillas.
- [31] Swokowski, Earl W., "*Cálculo con geometría Analítica*". Grupo Editorial Iberoamericana.
- [32] Leithold, Louis, "*El Cálculo con Geometría Analítica*". Cuarta edición. Harla.

- [33] A. Kiseliov, M.Krasnov, G.Makarenko, "*Problemas con Ecuaciones Diferenciales Ordinarias*", Tercera edición, Ediciones Quinto Sol.
- [34] Boyce, William E. and Richard C. DiPrima, "*Elementary Differential Equations*", Rensselaer Polytechnic Institute, John Wiley & Sons, Third edition, 1977.
- [35] Kimball, R., "*Self-optimizing computer-assisted tutoring: Theory and practice*", Psychology and Education Series, Technical Report No. 206, Institute of Mathematical Studies in the Social Science, Stanford, CA, 1973.
- [36] Stevens, A. L., & Collins, A., "*The goal structure of a Socratic tutor*", Proceedings of the Annual Meeting of Man-Machine Studies, Seattle, Washington, 1977.
- [37] Carbonell, J.R., AI in CAI: "*An artificial intelligence approach to computer assisted instruction*", IEEE Transactions on Man-Machine Systems, 11, 1970.
- [38] Brown, J.S., & Burton, R. R., & de Kleer, J., "*Pedagogical, natural language and knowledge engineering in SOPHIE I, II and III, Intelligent Tutoring Systems*", Sleeman, 1982.
- [39] Burton, R.R. & Brown, J.S., "*An investigation of computer coaching for informal learning activities*", International Journal of Man-Machine Studies, 11, 1979.
- [40] Goldstein, I.P. & Carr, B., "*The computer as coach: An athletic paradigm for intellectual education*", Proceedings of the Annual Meeting of the Association for Computer Machinery, Seattle, Washington, 1977.
- [41] Suppes, P. & Jerman, M., & Brian, D., "*Computer-Assisted Instruction: The 1965- 66 - Stanford Arithmetic Program*", Academic Press, New York, 1968.
- [42] Felipe Lara, Felipe y Nicolás Kemper Valverde et., Carolina De la Paz Alva y Julia Alcántara Zavala, "*Sistema Experto Tutorial para la Capacitación en el Diseño de Plantas de Cogeneración de Energía*", memorias del simposium internacional de computación, XXX aniversario de la maestría en ciencias con especialidad en computación, centro nacional de cálculo, "Tendencias de la computación hacia el nuevo milenio", Instituto Politécnico Nacional, celebrado el 8,9, y 10 de Noviembre de 1995, México D.F.
- [43] Sleeman, Dereck and John S. Brown, eds., "*Intelligent Tutoring Systems*", Academic Press; Troy, Mo., 1982, 345 pp.
- [44] Clancey, William J., "*Knowledge-Based Tutoring: The GUIDON Program*", MIT Press; Cambridge, Mass., 1987, 337 pp.
- [45] Robert W. Lawler and Masoud Yazdani, eds., "*Artificial Intelligence and Education: Learning Environments and Tutoring Systems*", Vol. 1, Ablex Publishing Co., Norwood, N.J., 1987, 448 pp.

- [46] C. Polson Martha and J. Jeffrey Richardson, eds., "*Foundations of Intelligent tutoring Systems*", Lawrence Erlbaum Associates, Hillsdale, N.J., 1988, 296 pp.
- [47] Wenger, Etienne, "*Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*", Morgan Kaufmann, Palo Alto, Calif., 1987, 486 pp.
- [48] Hernández L. Liliána, Monroy A. Carmen del P., Cervantes P. Francisco, "*STIN: Sistema Tutorial Inteligente en Neurociencias*", (Lab. de Bioingeniería, Inst. de Fisiología Celular, UNAM), México, 2do. Congreso Iberoamericano de Inteligencia Artificial IBERAMIA 90, Morelia, México, Julio 9 a 13 de 1990, LIMUSA, 729-742 pp.
- [49] Urretavizcaya Maite, Verdejo M. Felisa, "*Sistema Interactivo de Detección y Diagnóstico Adaptado para un Tutor Inteligente De Programación*", (Facultad de Informática, Universidad Politécnica de Cataluña, Facultad de Informática, Universidad del País Vasco), España, 2do. Congreso Iberoamericano de Inteligencia Artificial IBERAMIA 90, Morelia.
- [50] Claude Frasson, Thierry Mengelle, Esma Aimeur, Guy Gouardères, "*An Actor-based Architecture for Intelligent Tutoring Systems*", Third International Conference on Intelligent Tutoring Systems, ITS '96, Montreal, June 12-14, 1996, Paper #117.
- [51] Ljubomir Jerenic, Vladan Devedzic, "*OBOA Knowledge Model for EduSof-Intelligent Tutoring Shell*", Third International Conference on Intelligent Tutoring Systems, ITS '96, Montreal, June 12-14, 1996, Paper #130.
- [52] Mitrovic, Antonija, "*SINT-a Symbolic Integration Tutor*", Computer Science Department, University of Canterbury, Christchurch, New Zealand, Third International Conference on Intelligent Tutoring Systems, ITS '96, Montreal, June 12-14, 1996, Paper #041.
- [53] Stefano A. Cerri, "*Computational Mathematics Toolkit: architecture for modelling dialogues*", Dipartimento di Scienze dell' Informazione, Università di Milano, Italy, Third International Conference on Intelligent Tutoring Systems, ITS '96, Montreal, June 12-14, 1996, Paper #014.
- [54] Friedland, "*Acquisition of Procedural Knowledge from Domain Experts*", IEEE Principles of Experts Systems, International Joint Conference on Artificial Intelligence, 1981.
- [55] Gaines, B.R., "*An Overview of Knowledge-Acquisition and Transfer*", Knowledge-Based Systems Vol. 1, Academic Press Ltd., 1988.
- [56] Macdonald, B.A. and Witten, I.H. 1989, "*A Framework for Knowledge Acquisition*" through Techniques of Concept Learning", IEEE Vol. 19 No. 3.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

- [57] Sanchez, Sandra V., "*Adquisición del Conocimiento*". Tesis de Maestría, IIMAS-UNAM, México, D.F. 1993.
- [58] Kahan, G., Nowlan S. and McDermott J., "*Strategies for Knowledge Acquisition*", IEEE Trans. Vol. PAMI-7, No. 5 (Sept. 1985).
- [59] Aikins, Janice S., "*Prototypical Knowledge for Expert Systems*", Artificial Intelligence 20, North-Holland, 1983
- [60] Allen, "*Natural Language Understanding*", James Allen, The Benjamin/Cummings Publishing Company, Inc., 1987
- [61] Bobrow, D.G., "*GUS: A frame-driven dialog system*", R.M. Kaplan, D.A. Norman, H. Thompson and T. Winograd, Artificial Intelligence, Vol. 8 no 1., Feb 1977
- [62] Cuenca, J., "*Inteligencia Artificial: Sistemas Expertos*", G. Fernández, R. López y M. Verdejo, Alianza Editorial, Madrid, 1986.
- [63] Davis, R., "*Production rules as a representation for a knowledge-based consultation program*", B. Buchanan and E. Shortffe Artificial Intelligence, Vol. 8 no 1., Feb 1977.
- [64] Duda, "*Model Design in the Prospector Consultant System for Mineral Exploration*", R. Duda, J. Gaschnig, P. Hart, Expert System for the Microelectronic Age D. Michie Edinburgh University, Academic Press, 1979
- [65] Ermden, M.H. and R.A. Kowalski, "*The Semantics of Predicate Logic as a Programming Language*", ACM 23, no. 4., 1976
- [66] Findler, N.V., "*Associative Networks: Representation and use of Knowledge by Computers*", Academic Press, New York, 1979.
- [67] Fisher, D.A., "*Computer structures for programming languages*", Computer Science Dept., PhD thesis, Pittsburg Carnegie Mellon Univ., 1970.
- [68] Frey, Peter, "*A bit-mapped classifier*", Byte, Nov 1986
- [69] Goldstein, "*Using frames in scheduling*", I.P. Goldstein and B. Roberts, Artificial Y Intelligence: An MIT Perspective, Vol. 1, 1979
- [70] Holland, J., "*Escaping Brintleness: The Possibilities of Genera Purpose Learning Algorithms Applied to Parallel Rule-Based Systems*", Machine Learning II, Los Altos Ca., Morgan Kaufmann Publ., 1986
- [71] Kowalski, 1974, "*Predicate logic as a programming language*", Kowalski, R.A., North Holland Publishing Co., Amsterdam, 1974

- [72] Lenat, "*An artificial intelligence approach to discovery in mathematics as heuristic search*"; Lenat D.B., SAIL AIM-286 A.I. Lab. Stanford University, 1976
- [73] Lenat, "*EURISKO: A program that learns new heuristics and domain concepts*"; Lenat D.B., Artificial Intelligence 21, 1983
- [74] Machado, "*ESPLA: Entendimiento Simbólico Por medio de la Inteligencia Artificial*"; Machado A.,García H. y Barrera E. Tesis de Licenciatura, Universidad de las Américas, Puebla 1985.
- [75] Martínez . Ana Ma., "*Representación del Conocimiento y su Mecanismo de Inferencia*"; Primer Curso Internacional de Sistemas Expertos, Nov. 1987
- [76] Minsky,Marvin, "*A framework for representing knowledge*".The Psychology of Computer Vision, McGraw-Hill, 1975.
- [77] Nilsson,N.J. "*Problem solving methods in Artificial Intelligence*". McGraw-Hill, 1971.
- [78] Norman,D.A., "*A System for Perception and Memory*".Models of Human Memory, Academic Press, New York, 1971
- [79] Pinson,Suzanne, "*Représentation des Connaissances dans les Systèmes Experts*"; R.A.I.R.O Informatique/Computer Science, Vol. 15 no 4., 1981
- [80] Post,E., "*Formal reductions of the general combinatorial decision problem*"; American Journal of Mathematics 65, 1943
- [81] Quinlan,J.Ros, "*Learning Efficient Classification Procedures and their Application to Chess and Games*"; In Machine Learning An Artificial intelligence Approach, Tioga Publishing co., 1983
- [82] Quillian,M.R. "*Semantic Memory*",Semantic information Processing, Cambridge Ma., 1968
- [83] Scheffler,Israel, "*Las Condiciones del Conocimiento. Una introducción a la Epistemología y a la Educación*". Instituto de Investigaciones Filosóficas. UNAM No.29, 1979
- [84] Schrodtt,Philip A. "*Predicting International Events*"; Byte, Nov 1986.
- [85] Schubert,L. "*Extending the expressive power of Semantic Networks*", Artificial Intelligence, Vol. 7 no. 2, 1976
- [86] Shortliffe,E. "*Computer Based Medical Consultations, MYCIN*". Shortliffe,Elsevier Computer Sc. Library N.J. USA, 1976