

7
24/11



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

"SISTEMA EXPERTO TUTORIAL DE ENSEÑANZA EN INTEGRACION SIMBOLICA (SETEIS)"

T E S I S
QUE PARA OBTENER EL TITULO DE
M A T E M A T I C O
P R E S E N T A

FRANCISCO EDGAR CASTILLO BARRERA

DIRECTOR DE TESIS DE ESTUDIOS PROFESIONALES

M. EN C. GUSTAVO ARTURO MARQUEZ FLORES



1998

TESIS CON
FALLA DE ORIGEN

FACULTAD DE CIENCIAS
SECCION ESCOLAR

262653



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

M. en C. Virginia Abrín Batule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis:
"SISTEMA EXPERTO TUTORIAL DE ENSEÑANZA EN INTEGRACION
SIMBOLICA (SETEIS)"

realizado por FRANCISCO EDGAR CASTILLO BARRERA

con número de cuenta 8955392-5 , pasante de la carrera de MATEMATICAS

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis	M. EN C. GUSTAVO ARTURO MARQUEZ FLORES	<i>Gustavo Marquez Flores</i>
Propietario	M. EN C. MARIA GUADALUPE ELENA IBARGUENGOITIA GONZALEZ	<i>María Guadalupe Elena Ibarguengoitia Gonzalez</i>
Propietario	M. EN C. HORACIO CARVAJAL SANCHEZ-YARZA	<i>Horacio Carvajal Sanchez-Yarza</i>
Suplente	MAT. MARIA CONCEPCION ANA LUISA SOLIS GONZALEZ-COSIO	<i>María Concepción Ana Luisa Solís González-Cosío</i>
Suplente	MAT. HUGO VILLASESOR HERNANDEZ	<i>Hugo Villaseñor Hernández</i>

Consejo Departamental de Matemáticas
DR. MANUEL FALCONI MAGAÑA

FACULTAD DE CIENCIAS
CONSEJO DEPARTAMENTAL DE MATEMÁTICAS

DEDICATORIA

A DIOS

Por que siempre ha estado conmigo en todo momento

A MI MADRE

La señora Leticia Barrera Carrión que siempre ha sido un buen ejemplo a seguir, que ha creído en mí en todo momento y me ha dado todo su apoyo

y

A MEXICO

Por haber me recibido como su hijo

Introducción

La inteligencia artificial (IA) y los sistemas expertos (SE) han tenido aplicación[19] en áreas como la medicina, la industria, los negocios, etc. Sin embargo en el área educativa se han desarrollado menos aplicaciones[19] y no se ha logrado apoyar el gran impacto que se puede tener con la aplicación de la IA y los SE en el proceso educativo.

Como una aportación para la educación, en esta tesis se presentan técnicas de inteligencia artificial y un sistema experto tutorial inteligente denominado SETEIS (Sistema Experto Tutorial de Enseñanza en Integración Simbólica) que busca como objetivo principal, la enseñanza del cálculo diferencial e integral fundamentándolo en la teoría de los sistemas expertos y los sistemas tutoriales inteligentes; las cuales son áreas de aplicación de la inteligencia artificial. El segundo objetivo buscado en este trabajo es el de proporcionar una herramienta computacional que facilite por una parte al profesor la enseñanza del cálculo y por otra, el aprendizaje de los estudiantes de matemáticas, además de hacerlo atractivo.

Este documento esta dividido en cuatro capitulos. En el primer capítulo se desarrollan todos los fundamentos teóricos en que se basan los sistemas expertos. Se explica todo lo referente al diseño, funcionamiento e implementación realizada en su desarrollo, en términos generales.

En el capítulo dos se da la descripción de cómo se desarrolló y diseñó el sistema SETEIS, su arquitectura, la técnica utilizada para representar el conocimiento, su funcionamiento y el software utilizado.

En el capítulo tres se da una explicación de la programación del sistema, explicando cómo se programaron los predicados más importantes del sistema implementados en el lenguaje de programación lógica conocido como Prolog y utilizando el software comercial de Arity/Prolog versión 6.1.27.

En el capítulo cuatro se ejemplifica el uso del sistema con una guía rápida para toda persona interesada en la utilización del sistema.

Se presenta al final de la tesis las conclusiones obtenidas sobre los logros alcanzados por el sistema, así como una tabla de comparación con otros programas comerciales.

Se añade finalmente un apéndice como un complemento matemático para las personas que no tengan una familiarización con los temas de cálculo diferencial e integral que se consideran en este trabajo.

Contenido

Dedicatoria	ii
Introducción	iii

Capítulo I Sistemas Expertos

1.1 Los SE como una rama de la inteligencia artificial	1
1.2 Características de un sistema experto	2
1.3 Funcionamiento de un sistema experto	2
1.4 Diseño de un sistema experto	3
1.5 Ventajas de utilizar un sistema experto	3
1.6 Algunos sistemas expertos conocidos	4
1.7 Lenguajes de programación para SE.	5
1.7.1 Diferencias principales entre Prolog y Lisp.	5
1.8 Herramientas para desarrollo de SE (Shells).	6
1.8.1 Desventajas de utilizar un shell.	6
1.9 El conocimiento de un sistema experto	6
1.9.1 Formas de representación del conocimiento	6

Capítulo II Diseño del sistema SETEIS

2.1 Arquitectura	7
2.2 Representación del conocimiento	8
2.3 Motor de inferencias	9
2.4 Estructuras utilizadas por SETEIS	10
2.4.1 Constantes y Variables	10
2.4.2 Manejo de Listas	10
2.4.3 Hechos y Predicados para SETEIS	11
2.4.4 Funciones	11
2.5 Conjuntos	11
2.6 Notación matemática utilizada por SETEIS	11
2.6.1 Principales operadores utilizados en SETEIS	12
2.6.2 Precedencia de operadores en SETEIS	12
2.6.3 Expresiones Diferenciales e Integrales	12
2.7 Predicados	13

Capítulo III Programación

3.1 Derivación	14
3.1.1 El Predicado derivación	14
3.2 Métodos de Integración	14
3.2.1 El predicado integración	14
3.2.2 El método de integración inmediata y por cambio de variable	15
3.2.3 El método de integración por partes	16
3.2.3.1 Integración por partes iterando	16
3.2.3.2 Integración de expresiones trigonométricas	16
3.3 Utilerias de SETEIS	16
3.3.1 El predicado equivalente	16
3.3.2 El predicado simplifica	18
3.3.3 El predicado multiplica	18
3.3.4 El predicado agrupa	19
3.3.5 El predicado factoriza	19
3.3.6 El predicado polinomio	19

Capítulo IV Implementación del sistema

4.1 El ambiente de MS-DOS	20
4.2 El intérprete de Arity/Prolog	20
4.2.1 Arity vs Turbo Prolog	21
Conclusiones	23

Apéndice A Fórmulas y métodos de integración conocidos por SETEIS

A.1 Tácticas generales de integración	24
A.2 Terminología	24
A.3 Propiedades de las integrales	25
A.4 Integración inmediata	25
A.5 Integración por partes	26
A.6 Integración de funciones trigonométricas	26
A.7 Tácticas para reducción trigonométrica	27
Referencias	30

Capítulo I Sistemas Expertos

1.1 Los SE como una rama de la inteligencia artificial

Como una área de estudio de las ciencias de la computación se encuentra la inteligencia artificial (IA), la cual tiene como objetivo 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 querer definir algún concepto abstracto.

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[19], 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 la IA.



Figura 1: Algunas áreas de la inteligencia artificial.

Un sistema experto se puede describir a grandes rasgos como aquel que se fundamenta en una base de conocimientos llamados hechos, la cual es proporcionada por un conocedor o "Experto" en la materia y una serie de reglas de inferencia que manipulan esta base de datos o de conocimientos; la cual le permite al sistema dar una respuesta a una realidad específica.

1.2 Características de un sistema experto.

Los sistemas expertos deben cumplir con ciertas características[1] las cuales no son fáciles de programar en los lenguajes clásicos de programación para hacer sistemas. Entre las características más importantes tenemos:

- **Alto desempeño:** El sistema debe ser capaz de responder al nivel de competencia de un experto o mejor. Podemos afirmar en base a pruebas que SETEIS lo hace.
- **Tiempo de respuesta adecuado:** El tiempo de respuesta del sistema debe ser mejor o igual, que el tiempo que le tome al experto encontrar la solución. SETEIS cumple con este punto.
- **Confiable:** Debemos tener un buen grado de confiabilidad en los resultados. Lo cual, en el caso de *SETEIS*, se ha probado con más de 300 integrales, verificando sus resultados en libros de cálculo[18].
- **Justificación:** Todo sistema experto debe ser capaz de explicar los pasos de razonamiento que lo llevaron a dar una respuesta. Es aquí donde se saca provecho de *SETEIS* para utilizar al sistema como una herramienta para la enseñanza en la solución de integrales, pues muestra los cálculos y métodos de integración utilizados para obtener la respuesta.

1.3 Funcionamiento de un sistema experto.

El funcionamiento de un sistema experto es mostrado en la figura-3. 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 el motor de inferencias el cual manipula este conocimiento para obtener sus conclusiones y proporcionar una respuesta al usuario.

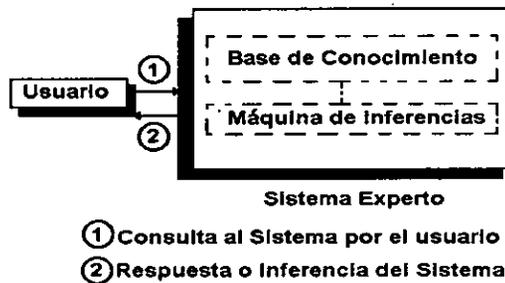


Figura 3: Esquema funcional de un sistema experto.

1.4 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 satisfactoriamente.

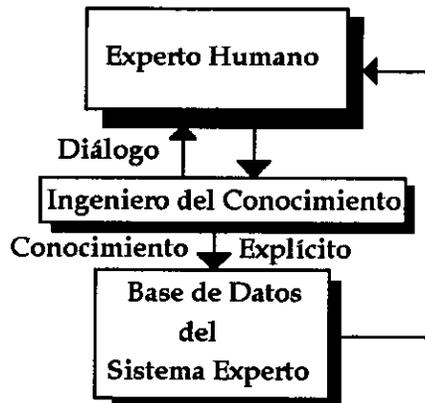


Figura 2: Proceso de Desarrollo de un Sistema Experto.

Para SETEIS el Experto Humano Fue: **M. en C. Miguel Angel Pizaña López**, estudiante de doctorado en ciencias de la computación y catedrático de la Universidad Autónoma Metropolitana, plantel Iztapalapa ciudad de México, D.F.

Se considera "Experto" a SETEIS 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 a partir de hechos básicos y además el sistema SETEIS realiza la integración internamente de manera muy similar a la humana.

Cuando SETEIS no puede evaluar una expresión con el conocimiento que tiene y las reglas de inferencia definidas, este contesta simplemente "no".

1.5 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. Algunas de ellas son las siguientes.

- **Incremento en la productividad.** Esta característica favorece a las industrias al disminuir el tiempo de elaboración de un producto y a la vez tener más control del proceso productivo disminuyendo los posibles errores humanos.

- **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 concimiento de varios expertos puede ser sustituido por un sistema experto.
- **Justificación.** Esta característica es quizás una de las más 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 debe ser 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.

1.6 Algunos sistemas expertos conocidos.

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-1 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	Bucanan, 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	S/R*	[19]
<i>WHY</i>	Causa de las Lluvias	Guiones	Collins A., 1977	S/R*	[19,20]
<i>SCHOLAR</i>	Geografía	Redes semánticas	Carbonell J.R., 1970	S/R*	[19,20]
<i>SOPHIE</i> SOPHisticated Instructional Environment	Localización de fallas electrónicas	Redes semánticas con simulador de circuitos	Brown J.S. y R.J.Bobrow, 1973	LISP y FORTRAN	[19,20]
<i>WEST</i>	Expresiones aritméticas	Reglas de producción	Burton Richard 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]
<i>WUSOR</i>	Relaciones lógicas	Red de gráficas genéticas	Stansfield, Carr y Goldstein, 1976	S/R*	[19,20]
<i>EXCHECK</i>	Lógica y teoría de conjuntos	Reglas de producción con intérprete de lógica	Smith, McDonald J., 1975	S/R*	[19,20]

Tabla 1: Tabla de sistemas expertos.

*No se encontró referencia alguna (S/R).

1.7 Lenguajes de programación para SE.

Existen dos lenguajes en inteligencia artificial utilizados para el desarrollo de sistemas expertos que son LISP y PROLOG. Sin embargo se han desarrollado otros lenguajes como IPL-II, SAIL, CONNIVER, KRL y Smalltalk.

El programar los sistemas expertos en estos lenguajes no implica que no se puedan realizar en otros lenguajes como C o PASCAL, pero éstos no presentan las mismas facilidades para el programador de un sistema experto.

1.7.1 Diferencias principales entre Prolog y Lisp.

El lenguaje Prolog (**P**rogramming in **L**ogic) es un lenguaje de programación basado en la lógica de Horn, mientras que Lisp (**L**ist **P**rocessing) es un lenguaje funcional basado en la idea de "Listas", las cuales son representadas por paréntesis que llegan a ser confusos para el programador en un momento dado. Mientras que en Prolog los programas pueden ser altamente modularizados y fácilmente legibles (siempre y cuando se tengan buenos hábitos de programación).

1.8 Herramientas para desarrollo de SE (Shells).

La creación de un sistema experto se puede realizar utilizando un lenguaje de programación como se explicó anteriormente o utilizando programas especializados llamados "shells" los cuales permiten implementar fácilmente la representación del conocimiento, el motor de inferencias y todos los elementos necesarios que conforman un sistema experto. Como ejemplo de estos programas tenemos a: LEVEL 5 OBJECT, STRESS, JESS, CLIPS, etc.

1.8.1 Desventajas de utilizar un shell.

Existen herramientas ya comerciales para el desarrollo de sistemas expertos como son los shells, sin embargo éstas solo manipulan datos simbólicos y no numéricos, lo cual hace que el desarrollo de sistemas expertos en el área de las matemáticas resulte imposible.

1.9 El conocimiento de un sistema experto.

Puesto que en este tipo de sistemas se tiene que representar el conocimiento de un experto humano, se han desarrollado varias técnicas en inteligencia artificial entre las cuales debe el ingeniero del conocimiento elegir la más apropiada según el tipo de conocimiento a representar.

1.9.1 Formas de representación del conocimiento.

Las técnicas de representación del conocimiento más comunes son:

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. Marcos.
8. Guiones
9. Dependencia conceptual.
10. CYC.

El método 1 es de representación procedural; este método marcó el inicio de la investigación de modelos para representación de conocimiento y hoy en día, prácticamente no se utilizan, por lo que únicamente citaremos las referencias mostradas por Pinson y Autómatas Finitos [3].

Los métodos 3, 4, y 5 son de representación declarativa mientras que los métodos 6 y 7 son de representación mixta. Estas técnicas de representación son las más comunes que se utilizan en la actualidad.

Capítulo II Diseño del sistema

2.1 Arquitectura

El sistema SETEIS está dividido en tres grandes módulos que son: *El módulo Analizador de Expresiones*, *El módulo Integrador* y *el módulo Explicación-Respuesta*. Como se muestra la figura-4. En este diagrama los módulos representan predicados del sistema, las aristas vínculos entre los predicados y los semicírculos indican que el predicado tiene que realizar una acción que lo llevará a tomar alguna decisión sobre que hacer la cual se pueda considerar inteligente.

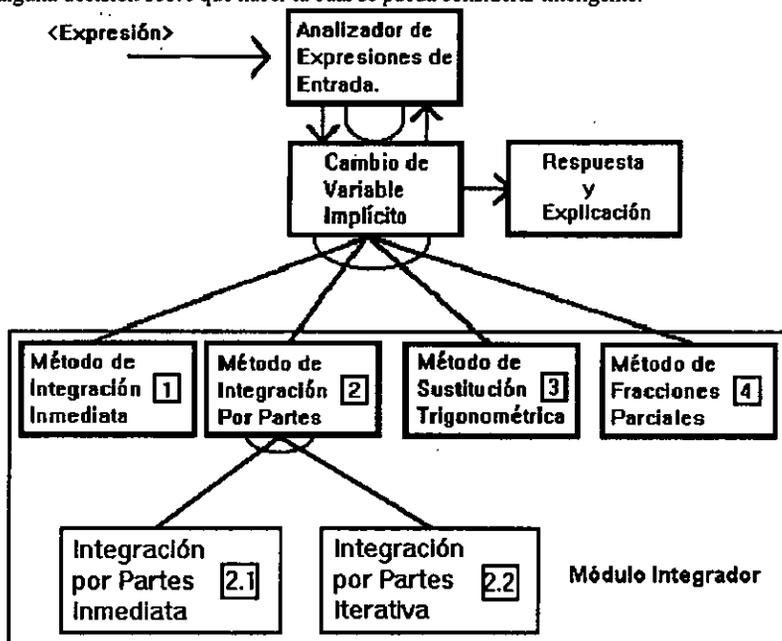


Figura 4: Arquitectura de SETEIS.

El primer módulo sirve como "filtro" para la expresión matemática de entrada, la cual tiene como función principal analizar la expresión y escribirla siempre que se pueda en términos conocidos para 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 ha 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.

El módulo Explicación-Respuesta es el encargado de dar la solución de la integral solicitada y además da una explicación paso a paso de cómo se obtuvo el resultado.

2.2 Representación del conocimiento.

Hace más de 40 años que Post (1943) propuso el formalismo de reglas de producción como mecanismo computacional de carácter general. También lo encontramos en los algoritmos de Markov y en la lingüística por Chomsky (1957), que utiliza reglas de reescritura en el reconocimiento sintáctico de frases de lenguaje natural [3].

Una regla de producción es un par ordenado (A,B), que puede representarse en el lenguaje de la lógica de proposiciones como $A \rightarrow B$. Los elementos del par reciben distintos nombres: antecedente y consecuente, premisa y conclusión o condición y acción.

La sintaxis o escritura de las reglas es diversa y está en función del proceso de selección y de ejecución de las mismas. Esta sintaxis es reconocida por el intérprete que accede a las reglas, que son en sí una cadena de símbolos; algunos sistemas aceptan también el uso de variables.

El intérprete funciona siguiendo una regla de inferencia bien conocida en lógica, la llamada regla de modus ponens: del hecho A y de la regla $A \rightarrow B$, se infiere la conclusión B.

Desde el punto de vista funcional el proceso de inferencia opera en dos fases: una de reconocimiento y otra de acción. La primera se subdivide a su vez en: seleccionar las reglas pertinentes y en resolver el conflicto de resolución en caso de que exista más de una regla aplicable.

Algunos de los criterios que se utilizan para determinar cual acción ejecutar cuando existe un conflicto de resolución son:

- Establecer orden en los datos.
- Clasificar las reglas por prioridad de ejecución.
- Ejecutar la regla más recientemente instanciada.
- Aplicar metareglas.

En la segunda fase se ejecutan las acciones establecidas por las reglas durante el proceso de inferencia. La selección de las reglas depende de la situación en curso de tratamiento.

Si se consideran los hechos establecidos y se verifican con las reglas, se dice que el razonamiento del sistema es hacia adelante, ya que de cumplirse el miembro izquierdo de la regla, se ejecuta la parte derecha de ésta. Por el contrario, si se consideran las metas posibles a alcanzar, es decir el miembro derecho de la regla y se verifican sólo las reglas que concluyen esas metas, se trata de razonamiento hacia atrás.

Una de las características sobresalientes de los sistemas basados en reglas de producción es su potencialidad de aprendizaje; es decir que a partir de su base de conocimiento inicial, el sistema es capaz de aumentar o disminuir el número de reglas de inferencia que rigen su comportamiento y este no se rige de manera algorítmica.

Este esquema es la base de MYCIN [2] - sistema experto desarrollado en Stanford en 1976 para el diagnóstico y tratamiento de enfermedades infecciosas en la sangre -, que, desde el punto de vista de lógica formal, utiliza lógica de proposiciones con algunas extensiones que lo aproximan a la lógica de predicados (existencia de predicados predefinidos y ciertas posibilidades de cuantificar) y a la lógica multivaluada (factores de certidumbre). Un ejemplo de sistemas que utilizan el modelo de reglas de producción es el sistema EURISKO [3].

En suma, los sistemas basados en reglas de producción permiten representar los conocimientos en forma modular y uniforme; son útiles en los casos donde se detecta y trata con una gran cantidad de estados independientes; en sistemas cuyos objetivos son amplios y cuyas acciones son de corto alcance y de toma de decisiones súbita. La mayoría de los sistemas expertos comerciales hoy en día utilizan esta herramienta para la representación de los conocimientos.

SETEIS usa como mecanismo de representación del conocimiento reglas de producción, en donde las aserciones lógicas se ven como representaciones declarativas del conocimiento. SETEIS 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 búsqueda a partir de un estado objetivo.

2.3 Motor de inferencias.

En SETEIS el motor de inferencias trabaja haciendo un razonamiento hacia adelante 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.

2.4 Estructuras utilizadas por SETEIS.

Fue necesario manipular el conocimiento utilizando algunas estructuras que tiene el lenguaje Prolog, como son el manejo de listas entre otras.

2.4.1 Constantes y Variables.

Para SETEIS una constante es un número real, entero natural o una letra diferente a la que aparezca después del operador dos puntos ":" como se muestra en el ejemplo siguiente.

Ejemplo: $(a+x)^2-23.5 : x$.

En donde la letra x es la variable y la letra a es una constante al igual que los números 2 y 23.5.

Es importante hacer notar que en el sistema SETEIS 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 guión bajo "_"; mientras que las variables del lenguaje matemático definido para SETEIS son siempre un átomo.

Ejemplos de tipos de variables

En SETEIS: En el Lenguaje de Prolog:

a, r, x, a, u, t, \dots $X, DV, II, _872F, \dots$

2.4.2 Manejo de Listas.

Para el intérprete de Prolog utilizado para este sistema (Arity/Prolog), una lista es por definición una secuencia de elementos, en donde al primer elemento se le llama cabeza de la lista y los otros 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 "[]".

2.4.3 Hechos y Predicados para SETEIS.

Un hecho es por definición un conocimiento básico que se tiene sobre algún tema, en el caso de la derivada, tenemos como ejemplo el siguiente hecho:

En notación matemática

$$\frac{dk}{dx} = 0$$

Representación interna de un hecho en SETEIS

$d(K, X, 0)$.

Que interpretado por SETEIS es la derivada de una constante con respecto a "X" es cero.

2.4.4 Funciones

Las funciones matemáticas que permite SETEIS 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 SETEIS sólo acepta una escritura lineal, por lo que es necesario seguir una notación que permita la linealidad de las expresiones. Con seguridad al lector se le hará familiar esta notación, que es la más usual en calculadoras y computadoras.

o

Caligrafía matemática: $f(x) = \sqrt{\frac{x^4}{4}}$ caligrafía lineal en SETEIS: raiz((x^4)/4).

2.5 Conjuntos

Las operaciones matemáticas conocidas para el manejo de conjuntos (unión, intersección y pertenencia), las cuales son fundamentales para todo sistema matemático, fueron implementadas utilizando listas y respetando las propiedades de los conjuntos, considerando que un conjunto por definición no tiene elementos repetidos. Existe el conjunto vacío, representado en SETEIS como la lista vacía "[]".

Los predicados implementados bajo las reglas anteriores son el de pertenece y unión. Las demás operaciones para conjuntos (el complemento, conjunto potencia, etc.) se pueden implementar fácilmente.

2.6 Notación Matemática utilizada por SETEIS.

El tipo de integrales que maneja SETEIS son solamente simbólicas o indefinidas, pues para las definidas existen cursos de métodos numéricos en donde se enseñan algoritmos para calcularlas sobre un intervalo.

2.6.1 Principales operadores utilizados en SETEIS.

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 SETEIS, incorpora el operador \int (N), el cual representa el símbolo de la integral que Leibnitz ideó como una "S" deformada.

2.6.2 Precedencia de operadores en SETEIS.

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

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

De la tabla anterior se deduce que el operador ":" que representa la derivada, tiene mayor precedencia que los operadores "*" y "/".

2.6.3 Expresiones Diferenciales e Integrales.

En el caso de las expresiones diferenciales éstas deben utilizar el operador ":" el cual tiene un papel importante como indicador del diferencial.

SETEIS admite todo tipo de expresiones diferenciales como lo muestra la tabla-2.

<u>Expresión SETEIS</u>		
Notación Matemática	Notación SETEIS	Simplificada
1) $d(\text{sen } x)$	1 : $\text{sen}(x)$	$\cos(x) : x$
2) $\cos(x) d(\text{sen}(x))$	$\cos(x) : \text{sen}(x)$	$\cos(x)^2 : x$
3) $(x^2 + 2x + 3) dx$	$x^2 + 2*x + 3 : x$	$x^2 + 2*x + 3 : x$

Tabla 2: Tabla de expresiones diferenciales en SETEIS.

Una expresión en SETEIS 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. (Para llevar a cabo 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 no son necesarios a causa de la precedencia con que fueron definidos los operadores.

2.7 Predicados.

A manera de explicación los predicados en esta tesis están subrayados, escritos con minúsculas según las reglas de Arity/Prolog, y están indicados los parámetros o argumentos correspondientes dentro de un paréntesis y 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". Se entenderá que una expresión se dice ser o estar "semi-instanciada", si ésta se forma de una combinación de átomos y variables del lenguaje Prolog[manual referencia], la cual se denotará con un asterisco.

Ejemplo: deriva(+E, -R).

Este predicado tiene aridad 2 (puesto que tiene dos parámetros 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 III Programación

3.1 Derivación.

Se programaron dos predicados para representar la acción de calcular la derivada de una función. Existe un predicado muy general llamado *derivación*, pero también existen otros predicados muy particulares según los intereses del usuario.

3.3.1 El predicado derivación.

Para poder calcular la integral de una expresión en muchos casos es necesario calcular algunas derivadas. Existen dos formas de realizar la derivada, una es directamente al mandar a llamar al predicado con sus argumentos correspondientes *derivar(E,X,R)* y la otra es utilizando el predicado *derivación* el cual además de simplificar el resultado da una explicación de como obtuvo la respuesta.

Ejemplo:

Notación Matemática

Notación SETEIS

$$\frac{d \operatorname{sen}(3c)}{dc}$$

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

DC = (3*1)*cos(3*c) ->
yes

En el caso muy específico de las fórmulas de derivación implementadas en SETEIS se tiene la regla de la cadena implícita en cada una, sin embargo este predicado puede ser añadido al sistema, independientemente de los predicados ya existentes de derivación.

3.2 Métodos de Integración.

Los métodos de integración en el sistema son implementados mediante predicados, en donde existe un predicado muy general llamado *integración* y varios predicados muy particulares para cada tipo de técnica de integración.

3.2.1 El predicado integración.

El predicado *i(E:U)* intenta integrar una expresión de la forma E:U, en donde E representa la expresión diferencial y U la variable con respecto a la cual se va a integrar. Observaciones: Primero se simplifica la expresión de entrada, después se integra, posteriormente se vuelve a simplificar la expresión de salida, mostrándose en pantalla finalmente.

Ejemplo:

Notación Matemática

$$\int x \exp^2 dx$$

Notación SETEIS

$$?- i(x*\exp(x^2):x).$$

$$2^{+1}*\exp(x^2)$$

yes

$i(E:X, I).$

uso: Integra una expresión "E" con respecto a "X" y devuelve el resultado en "I".

Ejemplo:

Notación Matemática

$$\int x^2 dx$$

Notación SETEIS

$$?- i(x^2:x, I).$$

$$I = 2*x^1 ->$$

yes

porpartes(+E:+X, -U, -DV:-X)

uso: Este predicado es un módulo que decide si la expresión se integra por partes, y en caso afirmativo, determina quién es U y DV:X.

porpartesiter(-E:-X, -U, -DV:-X)

uso: Checa si la expresión se integra por partes, pero de manera iterada, en caso afirmativo se indica quién es U y quiénes DV:X.

Representación Interna

porpartesiter(E:X,K*sec(X)^(N-2),sec(X)^2:X):- equiv(E:X,K*sec(X)^N:X), número(N).

El predicado **porpartesiter**(integración por partes iterando) recibe como primer argumento la expresión diferencial y trata de clasificarla en dos variables que son "E" y "X", en donde la primera es la función a integrar y la segunda es la variable de integración, las cuales son determinadas por el operador dos puntos ":" que las separa (operador diferencial), posteriormente se llama al predicado *equiv* el cual por medio de manipulación algebraica trata de determinar si la expresión recibida es equivalente al tipo de integral que se resuelven aplicando varias veces el método de integración por partes. En caso de cumplirse esto, se devuelve como segundo argumento la expresión que debe ser considerada como "U" y como tercer argumento la expresión que debe ser considerada como "DV", la cual requiere posteriormente ser integrada para obtener el valor de "V" y volver aplicar la fórmula de integración por partes ($uv - \int vdu$).

Ejemplo:

Notación Matemática

Notación SETEIS

$$\int \sec^3(x) dx$$

?- porpartesiter(sec(z)^3:z, U, DV:X).

$$U = 1 * \sec(z)^1$$

$$DV = \sec(z)^2$$

$$X = z \rightarrow$$

yes

3.2.2 El método de integración inmediata y por cambio de variable.

El método de integración inmediata es directamente implementado con un predicado para cada fórmula de las consideradas como básicas. Como ejemplos tenemos las siguientes fórmulas de integración y su predicado correspondiente.

Notación Matemática

Representación interna

$$\int k u^n du = \frac{u^{n+1}}{n+1}$$

i(E:X, K*U^(N+1)*(N+1)^(-1)):- equiv(E:X,K*U^N:U),
numero(N), not(N =: -1), ...

$$\int k \frac{du}{u} = k \ln(u)$$

i(E:X, K*ln(U)):- equiv(E:X, K*U^(-1):U), ...

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

i(E:X, K*C^U/ln(C)):- equiv(E:X, K*C^U:U), numero(C), ...

$$\int k \exp(u) du = k \exp(u)$$

i(E:X, K*exp(U)):- equiv(E:X, K*exp(U):U), ...

$$\int k \sec(u) du = -k \cos(u)$$

i(E:X, -K*cos(U)):- equiv(E:X, K*sen(U):U), ...

En todos los casos anteriores el predicado "I"(de integración) recibe como primer argumento la expresión diferencial clasificandola en dos variables "E" y "X", posteriormente se llama al predicado *equiv* el cual determina si la expresión recibida es equivalente al tipo de integral que resuelve el predicado, en caso de cumplirse esto, se devuelve en su segundo argumento la solución correspondiente a este tipo de integral.

3.2.3 El método de integración 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:

Notación Matemática

$$\int x \operatorname{sen}(x) dx$$

Notación SETEIS

?- porpartes(x*sen(x):x, U, DV:X).

$$U = x^1$$

$$DV = 1*\operatorname{sen}(x)$$

$$X = x \rightarrow$$

yes

3.2.3.1 Integración por partes iterando.

Este módulo resuelve el tipo de integrales que después de haber aplicado el método de integración por partes, dan por resultado del lado derecho de la igualdad una expresión equivalente a la que se desea integrar, y por lo tanto sólo es necesario hacer una manipulación algebraica para obtener la solución.

3.2.3.2 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. SETEIS 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. SETEIS tratar de utilizar esta táctica siempre que le sea posible.

3.3 Utilerías de SETEIS.

Se le llamaron utilerías de SETEIS a todos los predicados que fue necesario programar para realizar los predicados principales del sistema. Entre los principales tenemos: *equiv, simplifica, multiplica, agrupa, factoriza, polinomio, sustituye, unario, binario, cociente, inversos, factores, sumandos, número, producto, suma, resta, desarrolla, ordena, etc.* A continuación explicaremos algunas de ellas.

3.3.1 Equivalente.

En SETEIS 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.*

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 SETEIS.

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($x * \exp(x^2) : x$, $K * \exp(V) : V$).
 $K = 2^2 - 1$
 $V = x^2 \rightarrow$
yes

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

?- equiv($5 + \tan(o)$, $A + B$).
 $A = 5$
 $B = \tan(o) \rightarrow$
yes

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

Ejemplo: ?- equiv($\cos(x) : \sin(x)$, $K * \cos(V) : V$).
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 SETEIS, esto se lleva a cabo en un tiempo finito computable para la máquina[computabilidad] y en un tiempo bastante razonable.

3.3.2 El predicado simplifica.

El predicado simplifica juega un papel muy importante en cuanto eficiencia en el tiempo de respuesta del sistema SETEIS, pues todos los predicados lo utilizan para tratar de emparejar la expresión de entrada. Sus acciones principales son el quitar ceros, reescribir expresiones, calcular diferenciales, 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

Notación SETEIS

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

?- simplifica((x+1)^3*ln(3*x^2):(3*x^2), E2:X)

$$E2 = (x+1)^3 \ln(3*x^2) * 6*x :x \rightarrow$$

yes

$$x^0 + 1x$$

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

$$R = 1+x \rightarrow$$

yes

3.3.3 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. Este predicado es utilizado por el predicado de integración por fracciones parciales.

Ejemplo:

?- multiplica((x+2),(x+2), R).

$$R = x^2 + 2*x + 4 \rightarrow$$

yes

3.3.4 El predicado agrupa.

Existen expresiones matemáticas que crecen mucho al momento de hacer los cálculos (como es el caso cuando se aplica el método de integración por partes) y esto implica que sea necesario 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 tardado.

Ejemplo:

?- agrupa($x^2+1*f-3*x^2$, R).

R = $f-2*x^2$ ->

yes

3.3.5 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:

Notación Matemática

$$\frac{tx + t^2}{tx + t}$$

Notación SETEIS

factoriza(t, $(t*x+t^2)/(t*x+t)$,R).

R= $t^0*(x+t^1)/(t*x+1)$ ->

yes

3.3.6 El predicado polinomio.

El predicado polinomio es utilizado cuando se desea integrar una expresión por el método de fracciones parciales, el cual recibe una expresión de entrada y determina si es un polinomio. En caso de ser cierto se devuelve el grado del polinomio el cual es importante saber para poder descomponer en fracciones la integral para así poder integrar por el método de fracciones parciales. Es importante notar que el predicado polinomio busca las variables en memoria dadas por el usuario y lo demás lo considera constante.

Ejemplo:

polinomio($x*x^2+a*x$,G).

G=3->

yes

Considerando que “x” es la única variable.

Capítulo IV Implementación del sistema

4.1 El ambiente de MS-DOS.

El sistema SETEIS fue diseñado para trabajar en una computadora XT, AT o compatible con IBM-PC, teniendo como mínimo 640 KiloBytes de memoria RAM y con una versión mayor a la 3.0 de MS-DOS.

Existen dos formas de utilizar SETEIS, una es corriendo el archivo ejecutable desde el indicador del sistema operativo. "SETEIS.EXE"

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

La segunda forma de utilizarlo es bajo el intérprete de Arity Prolog.

4.2 El intérprete de Arity/Prolog.



Figura 5: El intérprete de Arity/Prolog.

SETEIS fue diseñado bajo Arity/Prolog versión 6.1.2 como se muestra en la figura-5, por lo consiguiente se puede trabajar en esta versión o en versiones mayores a esta. Arity funciona bajo DOS, y sólo en computadoras PC o compatibles con un mínimo de 640 KiloBytes de memoria RAM.

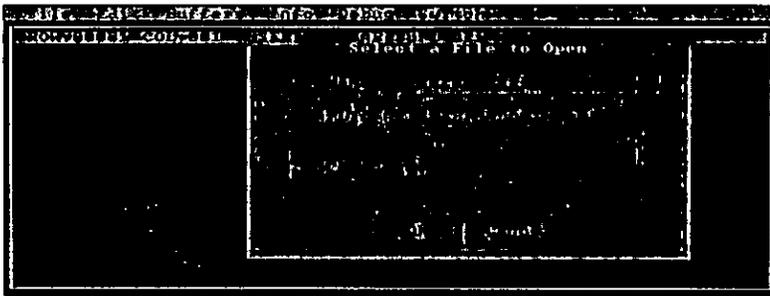


Figura 6: Menú File para abrir archivos en Arity/Prolog.

SETEIS fue hecho utilizando el editor de Arity/Prolog como lo muestra la figura-7.

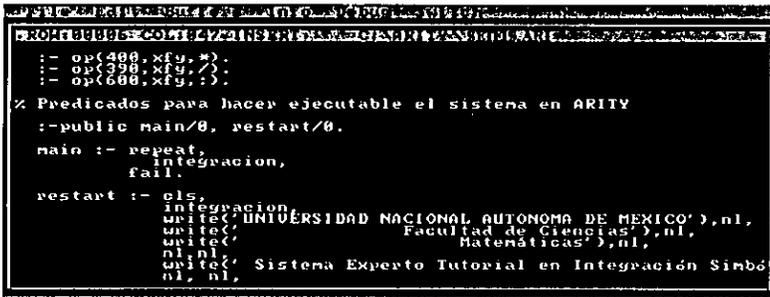


Figura 7: El editor de Arity/Prolog.

Para ejecutar el programa basta con ir a la opción de "File" escoger la opción de "Open File", después ir a la opción de "Buffers" seleccionado la opción de "Reconsult On Exit".

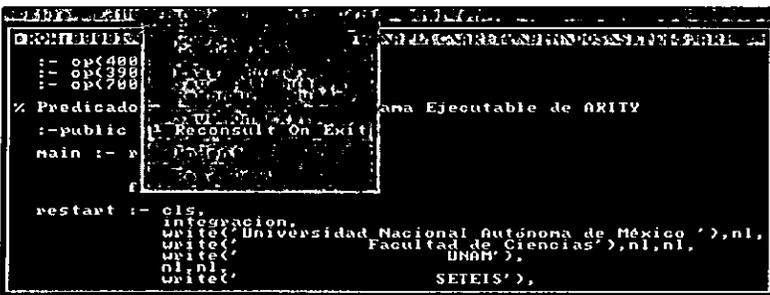
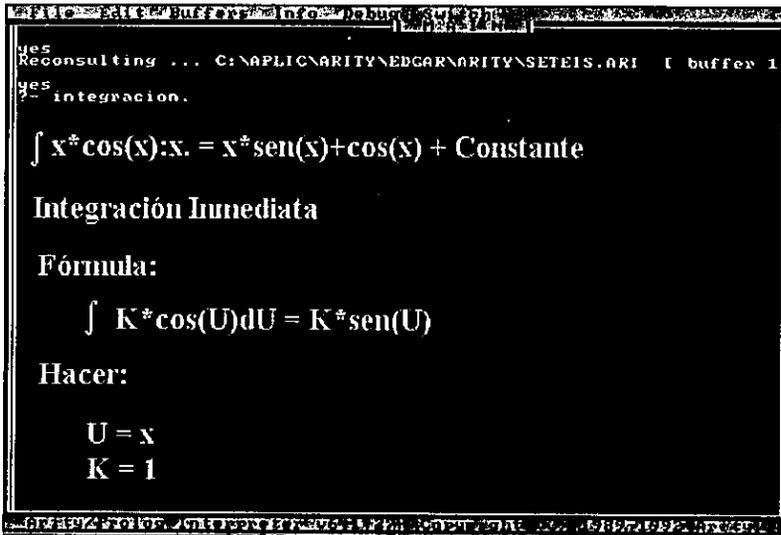


Figura 8: Consultado de un archivo en Arity/Prolog.

Desde este momento se puede invocar a los predicados desde el signo de interrogación que aparece en la ventana como lo muestra la figura-9.

La ejecución del programa se realiza simplemente tecleando la palabra *integración* con punto y pulsando enter, posteriormente se muestra el símbolo de la integral y a partir de ese momento se introduce la integral que se desea resolver. El sistema contesta la respuesta después de poner el símbolo de igual y posteriormente se da la explicación de como resolverla. En la figura-9 se muestra un ejemplo de su uso.



```
yes
ReConsulting ... C:\APLICARITY\EDGAR\ARITY\SETEIS.ARI [ buffer 1
yes
integracion.

∫ x*cos(x):x. = x*sen(x)+cos(x) + Constante

Integración Inmediata

Fórmula:

∫ K*cos(U)dU = K*sen(U)

Hacer:

U = x
K = 1
```

Figura 9: Esquema funcional de SETEIS.

Conclusiones

El desarrollo de sistemas expertos para integración simbólica implica el conocimiento de diferentes áreas y por ende de diversas personas dedicadas a la ciencia como son la Inteligencia Artificial, Diseño de Sistemas Expertos, Programación lógica, Cálculo diferencial e Integral etc.

Para programar el sistema SETEIS se decidió utilizar el lenguaje de Arity/Prolog y no un Shell comercial para sistemas expertos, debido a que estos no permiten la manipulación numérica de manera fácil. Por otra parte no se escogió el lenguaje Lisp por la razón de que es más sencillo utilizar el lenguaje Prolog para representar el conocimiento en base a reglas, además de presentar una notación más sencilla lo cual facilita su legibilidad y modularidad.

En base a las necesidades de éste sistema, se puede concluir que es necesario que los programas para desarrollo de sistemas expertos permitan en un futuro una buena manipulación numérica junto con la manipulación simbólica.

En la comparación de Arity/Prolog y Turbo Prolog se observó que Turbo Prolog pierde parte del carácter simbólico del "Prolog Puro" al tener que definir tipos para sus variables, además de otras características que lo hacen diferir mucho del Prolog original.

SETEIS maneja una notación consistente entre sus predicados y los términos matemáticos, lo cual le permite a cualquier persona con conocimientos básicos de matemáticas una rápida familiarización con los términos utilizados en el sistema, sin tener un conocimiento previo del lenguaje Prolog ni saber programación lógica.

El módulo Explicación-Respuesta no ha sido terminado eficientemente por lo que queda como propuesta de proyecto. Este módulo requiere el desarrollo de un sistema tutorial inteligente que sea capaz de evaluar y supervizar el aprendizaje del estudiante[20].

Software	Espacio en Disco	Portabilidad
Maple V versión 4.0	1,302,528 Bytes	Macintosh, Power Macintosh, Microsoft Windows, Windows NT, IBM OS/2, Sun SPARC, Linux, HP, Digital UNIX y otros.
Mathematica versión 3.0	120,832 Bytes	Macintosh, Power Macintosh, Microsoft Windows, Windows NT, IBM OS/2, Sun SPARC, Linux, HP, Digital UNIX y otros.
SETEIS	30,000 Bytes	Macintosh, Power Macintosh, Microsoft Windows, Windows NT, IBM OS/2, Sun SPARC, Linux, HP, Digital UNIX y cualquier sistema con un intérprete de Prolog.

Tabla 3: Tabla de comparaciones

El sistema SETEIS presenta grandes mejoras en comparación con otros sistemas comerciales para hacer integración simbólica por computadora, entre las que destacan el hecho de requerir muy poco espacio para su almacenamiento en algún medio físico y muy poca memoria para su ejecución; permitiendo la utilización de computadoras personales con poca memoria en el mercado a diferencia de otros programas como lo muestra la tabla-3.

Debido a la modularidad con que fue programado SETEIS, éste permite la posibilidad de añadir nuevos predicados para en un futuro resolver ecuaciones diferenciales ordinarias utilizando los predicados ya existentes; de esta forma se le puede hacer crecer y volverlo más poderoso. Pero aún más que eso todavía, la estructura misma del sistema permite al programador hacer modificaciones a los predicados ya existentes, de tal forma que el resto del sistema no se tenga que enterar de estas modificaciones.

El sistema se probó con más de 200 integrales obtenidas de libros de Cálculo[16,17,18] de las cuales se observó que el 85% de los resultados eran iguales a los del libro y el otro 15% requerían un poco de manipulación algebraica para también serlo.

La aportación más importante de este sistema es el hecho de que puede ser utilizado en un futuro como una herramienta computacional de aprendizaje para los estudiantes de Cálculo diferencial e integral y como un elemento didáctico de apoyo para el profesor que la imparte.

Apéndice A Fórmulas y métodos de integración conocidos por SETEIS

A.1 Terminología

Para las siguientes fórmulas se considerará la siguiente notación: las letras u , v y w representarán funciones cualquiera sobre los reales y la letra n cualquier número entero o real, la letra c una constante definida y la letra k una constante indefinida.

A.2 Propiedades de las integrales.

$$\int d u = u + k$$

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

$$\int k u d u = k \int u d u$$

A.3 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.4 Método de integración por partes.

$$\int u d v = u v - \int v d u$$

A.5 Integración de funciones trigonométricas.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.6 Tácticas para reducción trigonométrica.

Integral	Caso	Aplicar
$\int \text{sen}^n(u) du$	Si $n > 2$ y entero	$-\frac{\text{sen}^{n-1}(u) \cos(u)}{n} + \frac{n-1}{n} \int \text{sen}^{n-2}(u) du$
$\int \text{sen}^2(u) du$	Si $n = 2$ y entero	$\int \left(\frac{1}{2} - \frac{1}{2} \cos(2u) \right) du$
$\int \cos^n(u) du$	Si $n > 2$ y entero	$-\frac{\cos^{n-1}(u) \text{sen}(u)}{n} + \frac{n-1}{n} \int \cos^{n-2}(u) du$
$\int \cos^2(u) du$	Si $n = 2$ y entero	$\int \left(\frac{1}{2} + \frac{1}{2} \cos(2u) \right) du$

$$\int \tan^2(u) du \quad \text{Si } n=2 \text{ y entero} \quad \int (\sec^2(u) - 1) du$$

$$\int \cot^2(u) du \quad \text{Si } n=2 \text{ y entero} \quad \int (\operatorname{cosec}^2(u) - 1) du$$

$\int \sin^m(u) \cos^n(u) du$ m o n positivo e impar decrementar 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, decrementar 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, decrementar 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, decrementar 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, decrementar 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 decrementar 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 decrementar m en m-1 y luego aplicar: $\tan^2(u) = \sec^2(u) - 1$. Conservar $\sec(u)\tan(u)du$.

$\int \sin^m(u) \cos^n(u) du$ Si n y m son enteros positivos y pares.

$$\text{utilizar: } \begin{cases} \sin^2(u) = \frac{1}{2} - \frac{1}{2} \cos(2u) \\ \cos^2(u) = \frac{1}{2} + \frac{1}{2} \cos(2u) \\ \sin(u) \cos(u) = \frac{1}{2} \sin(2u) \end{cases} \quad \text{además } \begin{cases} \sin(2u) = 2 \sin(u) \cos(u) \\ \cos(2u) = \cos^2(u) - \sin^2(u) \end{cases}$$

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 IIMAS-UNAM, Junio 1994.
- [4] Michael A. Covington, Donald Nute and Andre Vellino, *Prolog Programming in DEPTH*, Copyright 1988 Scott, Foresman & Co.
- [5] Deyi Li, Ph. D. Department, *A Prolog Database System*.
- [6] Nilsson, Ulf and Jan Matuszynski, *Logic Programming and Prolog*, Ed Wiley.
- [7] Sterling, Leon & Ehud Shapiro, *The Art of Prolog, Advanced Programming Techniques*, The MIT Press Cambridge, Massachusetts London, England.
- [8] Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Second Edition, Addison Wesley, 1990.
- [9] Winston, Patrick Henry, *Artificial Intelligence*, Third Edition, Addison Wesley, 1992.
- [10] Giannesini, Francis, Henry Kanoui, Robert Pasero, Michel van Caneghem, *Prolog Reading*, Addison-Wesley, 1986.
- [11] Sánchez, J.P. y Beltrán, *Sistemas Expertos, Una metodología de Programación*, Macrobit TM.
- [12] Clark, K.L. and S.-A. Tarnlund, *Logic Programming*, Academic Press.
- [13] Campbell, J.A, *Implementations of Prolog*, New York, John Wiley & Sons, 1984.
- [14] Clark, K.L., S.-A. Tarnlund, *Logic Programming*, New York, Academic Press, 1982.
- [15] Kowalski, Robert, *Logic for Problem Solving*, New York, El-sevier North Holland, 1979.
- [16] Spivak, Michael, *CALCULUS*, Cálculo Infinitesimal, Reverté, S.A. Tomo II.
- [17] Swokowski, Earl W., *Cálculo con geometría Analítica*, Grupo Editorial Iberoamericana.

- [18] Leithold, Louis *El Cálculo con Geometría Analítica*, Cuarta Edición, Editorial Harla.
- [19] Durkin, John, *Expert Systems Catalog of Applications*, Intelligent Computer Systems Inc. Primera Edición, 1993.
- [20] Sleeman, D. and J.S. Brown, *Intelligent Tutoring Systems*, Academic Press, Computer and People Series, Edited by Brian Gaines, 1982.
- [21] Rich, Elaine y Kevin Knight, *Inteligencia Artificial*, McGraw-Hill, Segunda edición, Traducido de la segunda edición en inglés de Artificial Intelligence, 1994.
- [22] *The Arity/Prolog Compiler and Interpreter*, Arity Corporation versión 6.1, Damonmill Square Concord, Massachusetts 01742, 1992.
- [23] *BIM-Prolog Manual 2.4*, BIM SA/NV, Marzo 1989.
- [24] *Quintus-Prolog Reference Manual 2.0*, Quintus Computer Systems, Inc., Marzo 1987.