

Proyecto
RAZONAMIENTO LÓGICO CON
DIAGRAMAS DE VENN v1.0

Resumen del Proyecto Fin de Carrera



Universidad de Salamanca

*Departamento de Filosofía,
Lógica y Filosofía de la Ciencia*

Realizado por:

M^a Luisa Martín Martín

Tutora:

María Manzano

Fecha de Presentación:

Julio 2001

1. INTRODUCCIÓN

Este proyecto forma parte de uno más general: Summa Logicae en el siglo XXI, que incluye:

1. Recopilación de textos, tanto generales como especializados de lógica.
2. Investigación del razonamiento visual.
3. Elaboración de aplicaciones informáticas de apoyo a la docencia de la lógica.
4. Aprovechamiento de las nuevas tecnologías para la presentación y divulgación de los resultados del proyecto.

Las necesidades de llevar a cabo una investigación acerca del razonamiento visual son varias. Muchas veces es más fácil resolver ciertos problemas y de mostrar ciertos teoremas en teoría de conjuntos utilizando los diagramas de Venn. Por otro lado, la importancia que tiene el hecho de que en la mayoría de nuestros procesos de inferencia hagamos uso de algún tipo de representación visual. Y por último la búsqueda de un modelo de inferencia que sea capaz de incorporar en él una explicación del razonamiento válido a partir de representaciones no lingüísticas.

Tanto filósofos como informáticos, lógicos y psicólogos, han constatado que en muchas ocasiones es más fácil resolver un problema de matemáticas, física, lógica o informática usando diagramas que usando las clásicas representaciones algebraicas. De hecho los profesores de Lógica ya eran conscientes de lo útiles que son, por ejemplo, los diagramas de Venn a la hora de resolver ciertos problemas y de mostrar ciertos teoremas en teoría de conjuntos. Ya Euler, Venn y Peirce se dieron cuenta de la importancia que tienen los diagramas en las pruebas matemáticas.

La lógica de predicados monarios, que se resuelve mediante los diagramas de Venn, constituye un lenguaje intermedio entre el lenguaje de la lógica proposicional y el de la lógica de primer orden. La lógica proposicional se caracteriza por su rigor y precisión, pero carece de matices, no es suficientemente expresiva, existen muchos razonamientos claramente correctos que no son captados por ella, esto se debe a que el análisis realizado es demasiado pobre ya que

sólo intervienen los conectores. En la lógica de predicados se analiza la escritura de las sentencias atómicas y se añaden los cuantificadores. El lenguaje de la lógica de predicados monarios es más expresivo y versátil que el de la lógica proposicional, y permite hacer una escala intermedia antes de pasar al de la lógica de primer orden. En esta última hay predicados de cualquier grado, no solo monarios, y también hay símbolos funcionales.

La lógica proposicional tiene un cálculo correcto, completo y decidible. Es decir, hay un algoritmo que en un número finito de pasos, dice si la fórmula es válida o no. El problema es que su capacidad expresiva es limitada. La lógica de primer orden posee un cálculo correcto y completo, pero no deducible. La lógica de predicados monarios está en una situación intermedia pues retiene las buenas propiedades del cálculo proposicional mientras aumenta su capacidad expresiva.

El cálculo mediante diagramas de Venn es particularmente útil en la pedagogía ya que es fácil de aprender por su inmediata visualización.

2. OBJETIVOS

El objetivo final del proyecto es crear una herramienta interactiva que permita plantear y resolver argumentos en lógica de predicados monarios. La herramienta tendrá que tener una interfaz agradable, fácil de usar e intuitiva. El procedimiento es refutativo: se mostrará la incompatibilidad de las hipótesis junto a la negación de la conclusión.

Para resolver dichos argumentos la aplicación utiliza un procedimiento refutativo; es decir, muestra si son incompatibles las hipótesis junto a la negación de la conclusión.

Cada vez que se quiera plantear un argumento se creará un ejercicio que estará compuesto de dos partes:

- En una de las partes se podrá escribir el enunciado del argumento. Cuando se cree el ejercicio, se crearán tantas zonas como hipótesis más la conclusión que tenga el argumento. Se creará también una zona de escritura adicional para que el usuario pueda introducir las notas que crea oportunas para aclarar el ejercicio. (Por ejemplo, la fecha en la que se realizó)
- En la otra parte se representan en los diagramas las hipótesis y la negación de la conclusión del argumento que el usuario quiera resolver.

Cuando el usuario quiera obtener la solución del argumento, esta se añadirá al ejercicio, en ella se podrá ver la superposición de los diagramas que anteriormente se habían representado y un mensaje textual donde se dice si el argumento es correcto no, dependiendo de si el diagrama con la superposición de las hipótesis y de la negación de la conclusión es consistente o inconsistente.

La aplicación deberá permitir al usuario guardar los ejercicios con los que haya estado trabajando para su posterior recuperación. También sería interesante permitir la impresión de los ejercicios.

El proyecto incluirá toda la ayuda correspondiente al contenido y empleo de la aplicación, para facilitar al usuario la adaptación a la misma. A partir de esta ayuda será muy sencillo trabajar con la aplicación, ya que para utilizar la herramienta sólo habrá que seguir los pasos propuestos en la ayuda.

3. ASPECTOS RELEVANTES

Se ha utilizado el ciclo de vida en espiral basado en prototipos, para el desarrollo del proyecto. Se han ido iterando las distintas fases del desarrollo del software, generándose diversos prototipos a medida que se avanzaba en la espiral y se refinaban los requisitos y las especificaciones del proyecto.

Para la especificación de requisitos se ha seguido la *Metodología para la elicitación de requisitos de sistemas software* Durán y Bernárdez (1999). Este documento servirá como base para el desarrollo del resto de fases del proyecto. Después se ha realizado un análisis del sistema sobre el cual realizar el diseño.

La intención del documento de Análisis de Requisitos es la de obtener un conjunto de objetos que representen al sistema que deseamos obtener. Estos requisitos han sido reflejados mediante la utilización de casos de uso.

Se ha modelado el dominio del problema identificando y especificando dicho conjunto de objetos semánticos que interaccionan y se comportan de acuerdo a los requisitos del sistema. Partiendo de los casos de uso obtenidos en el documento de análisis del sistema se ha realizado la transición hacia los objetos mediante diagramas de interacción de cada caso de uso, en este caso concreto diagramas de secuencia en forma genérica. Se han obtenido las interacciones expresadas en función de secuencias de tiempo, mostrando los objetos participantes y los mensajes que intercambian entre ellos, para satisfacer los casos de uso.

A partir de los diagramas de secuencias se han obtenido las clases para crear el diagrama preliminar, que se ha ido refinando a lo largo del desarrollo del proyecto.

En la fase de diseño se ha refinado el diagrama de clases obtenido en el documento de análisis del sistema, con el fin de ir tomando decisiones estratégicas y tácticas para alcanzar los requisitos funcionales y la calidad esperada.

4. DESCRIPCIÓN

El proyecto consiste en realizar una herramienta interactiva para su uso en la docencia de la lógica. Concretamente, esta herramienta tendrá que comprobar si un argumento de la lógica de predicados monarios es correcto o no.

Los diagramas que construya el usuario tienen que ser consistente, es decir, no puede haber zonas sombreadas y con cruces entrelazadas en la misma zona, tampoco, puede unir una línea una zona que tenga elementos con otra que no los tenga. Si el usuario intentase construir un diagrama inconsistente el sistema no le dejaría

También hay que tener en cuenta que el diagrama se puede corregir. Una zona con elementos puede pasar a ser una zona de la que se carece de información o una zona vacía. Si esto ocurre hay que comprobar si la zona esta unida a otra con elementos mediante una línea, si es así, la línea debe borrarse.

Cuando se calcula la solución, la información que se ha introducido en los diagramas es consistente, por lo que al calcular el resultado final se trabaja con datos correctos.

Los diagramas pueden modificarse en cualquier momento, incluso después de haber calculado la solución. En el momento en el que se vaya a realizar alguna modificación en los diagramas de las hipótesis o el de la negación de la conclusión, la solución se elimina, ya que sino se tendría una solución calculada sobre datos obsoletos y no se correspondería con las hipótesis y negación de la conclusión actuales, esto podría llevar a confusión y a sacar conclusiones erróneas y sería lo contrario de lo que se quiere obtener con el programa.

En la siguiente figura puede observarse la interfaz de la aplicación:

The screenshot shows the DVenn application window. The title bar reads "DVenn" and the menu bar includes "Archivo", "Edición", "Ver", and "Ayuda". The toolbar contains various icons for file operations and editing. The main window is titled "Ejercicio0" and has a "Calcular la solución" button. The text in the window reads:

Hipótesis 1: Algunos sueños (A) son terribles (B).
Hipótesis 2: Ningún borrego(C) es terrible(B).
Conclusión: Algunos sueños(A) no son borregos (C).
Notas: Creado el 13 de Julio.

Below the text are three Venn diagrams, each within a box labeled "U":

- Hipótesis 1:** A Venn diagram with three overlapping circles labeled A, B, and C. The intersection of A and B is shaded.
- Hipótesis 2:** A Venn diagram with three overlapping circles labeled A, B, and C. The intersection of A and B is shaded.
- Negación de la Conclusión:** A Venn diagram with three overlapping circles labeled A, B, and C. The intersection of A and B is shaded.

At the bottom, there is a "SOLUCIÓN" section. On the left is a Venn diagram with three overlapping circles labeled A, B, and C. The intersection of A and B is shaded. To the right of this diagram, the text reads:

EL DIAGRAMA ES INCONSISTENTE
EL RAZONAMIENTO ES CORRECTO

5. ENTORNO DE APLICACIÓN

5.1. Entorno de desarrollo

Esta herramienta ha sido desarrollada utilizando Forte for Java Community Edition Release 2.0.

Se ha elegido Java porque:

- Es orientado a objetos.
- Es independiente de la plataforma. Java se podría correr sobre cualquier plataforma, aunque para explotar todo su potencial, requiere un sistema operativo multihilo, como Unix, Windows9x, OS/2...
El propio lenguaje Java está implementado como la máquina virtual de Java (JVM), que es la aplicación que ejecuta un programa Java. Cuando JVM se instala en un ordenador, este puede ejecutar programas Java. Los programas Java son compilados creando bytecodes y son estos bytecodes la que JVM lee e interpreta para ejecutar el programa.
- Es robusto, no permite el manejo directo del hardware ni de la memoria.
- Gestiona la memoria automáticamente.
- Tiene mecanismos de seguridad incorporados que limitan el acceso a recursos de las máquinas donde se ejecuta.
- Está diseñado específicamente para trabajar sobre una red, de modo que incorpora objetos que permiten acceder a archivos en forma remota (vía URL por ejemplo).

Java también permite la creación de interfaces gráficas fácilmente gracias al paquete Swing.

Swing es un conjunto de paquetes que incluye un conjunto de herramientas y objetos destinados a la construcción de interfaces gráficas de usuario, simplificando el desarrollo de componentes para su posterior inclusión en programas basados en entornos gráficos. Está incluido en el JFC (Java Foundation Classes).

La programación de los componentes Swing se basa en la arquitectura Modelo-Vista-Controlador. El modelo de un

componente está donde están almacenados sus datos, como el estado de un botón o los elementos de una lista. La vista es la representación en pantalla del componente, como la forma en la que aparece un botón o una lista. El controlador es la parte del componente que gestiona la entrada, como los clic del ratón.

Swing basa sus componentes UI en la arquitectura MVC, por lo que es útil separar la vista del modelo, donde se puede cambiar fácilmente la apariencia de los componentes.

5.2. Entorno de explotación

La aplicación Dvenn v1.0 trabaja bajo una plataforma Windows 9x, NT o 2000. El PC en el que se vaya a explotar dicha herramienta deberá tener como mínimo de 16 a 32 Mb. de memoria RAM, para un correcto funcionamiento de la aplicación. La capacidad en disco deberá ser de unos 100 Mb., como mínimo. El ordenador deberá tener instalada la máquina virtual de Java, que se suministra en el CD-ROM adjunto ya que es de libre distribución.

La instalación de la aplicación es muy sencilla si su entorno de explotación contiene los requisitos expuestos, ya que en el CD-ROM adjunto al proyecto, tiene el ejecutable de la aplicación. El ejecutable se denomina DVenn.jar.

6. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURO.

6.1. Conclusiones del resultado del proyecto.

La herramienta que se ha obtenido es familiar y fácil de utilizar de cara al usuario. Éste sólo tendrá que introducir la información de las hipótesis y de la negación de la conclusión en los diagramas.

El sistema proporciona todos los medios para poner y eliminar la información de las distintas zonas en el diagrama. Todo esto se podrá hacer siempre y cuando el diagrama sea consistente.

El sistema es el encargado de detectar e impedir que se introduzca información en los diagramas de manera incorrecta.

Se pueden almacenar en memoria secundaria los ejercicios creados.

El proyecto incluye una ayuda correspondiente al contenido y empleo de la aplicación, para facilitar al usuario la adaptación a la misma. A partir de esta ayuda será muy sencillo trabajar con la aplicación.

6.2. Trabajos futuros.

Cuando un razonamiento es incorrecto se puede definir un modelo en donde la hipótesis del razonamiento sean verdaderas y la conclusión falsa; de esta forma se encuentra un contraejemplo para rebatir la argumentación propuesta. Para este caso se podría crear un generador de modelos.

Crear una página web desde la que se puedan obtener conocimientos de la lógica de predicados monarios y la resolución de argumentos de esta lógica mediante diagramas de Venn. En esta página se podrá ejecutar el programa para poder resolver ejercicios.

Integrarlo con otros procedimientos de prueba mediante diagramas, como por ejemplo, el razonamiento lógico con diagramas de Peirce.