

LSD, una herramienta didáctica para el aprendizaje de la lógica

Josep Humet

Instituto de Informática y Aplicaciones (IIiA)
Departamento de Informática y Matemática Aplicada
Universidad de Girona
Campus Montilivi
E-17071 Girona
josep.humet@ima.udg.es
<http://ima.udg.es/~humet>

Resumen

Se justifica la necesidad de herramientas didácticas para el aprendizaje de la lógica, se analiza algunas de las herramientas existentes específicas para la lógica de predicados de primer orden, y se describe una herramienta desarrollada en el Departamento de Informática y Matemática Aplicada de la Universidad de Girona.

1. Introducción

La lógica es una disciplina que incide en muchas áreas de conocimiento en particular la matemática o la filosofía y, muy especialmente, la informática. Esta *universalidad* nos lleva a considerar la necesidad de una *calidad docente* de la lógica de modo que aparece como absolutamente imprescindible la mejora continuada de su docencia pues ello repercutirá positivamente en el rendimiento global de todas las titulaciones. Alcanzar esta calidad docente no es tarea fácil puesto que la naturaleza abstracta y formal de la lógica dificulta enormemente la comprensión y el aprendizaje de la misma. Por todo ello creemos que es absolutamente necesario llevar a cabo acciones de creación de herramientas para el aprendizaje de la lógica y de diseño de prácticas con autoevaluación asistidas con dichas herramientas.

Actualmente existen muchos programas de apoyo para el aprendizaje de la lógica [8,16]. La mayoría han sido creados en el último decenio y muchos son accesibles libremente a través de internet y ello ha sido posible gracias al auge experimentado por las tecnologías de la información y las comunicaciones pero también a

la propia naturaleza formal de la lógica que la hace idónea para ser tratada con herramientas computacionales.

En el Departamento de Informática y Matemática Aplicada de la Universidad de Girona hemos desarrollado una herramienta para el estudio de la *lógica de predicados de primer orden con funciones* [6] que hemos bautizado con las siglas LSD (Lógica Simbólica Deductiva).

En la sección 2 hacemos un pequeño análisis de los tipos de herramientas existentes actualmente para el aprendizaje de la lógica de primer orden, sugerimos algunos criterios de clasificación, y en particular clasificamos la herramienta que presentamos. En la sección 3 describimos a nivel de usuario el funcionamiento del programa LSD. Finalmente en la sección 4, a modo de conclusión, hacemos un resumen y exponemos líneas futuras de trabajo.

2. Clasificación de herramientas

Las herramientas de apoyo a la docencia de la lógica las podemos clasificar en varios grupos en función de sus prestaciones y en particular de la interactividad que muestran con el usuario y la etapa dentro del aprendizaje en la que el usuario se encuentra:

- Verificadores
- Constructores
- Demostradores
- Ejercicios

Los *verificadores* [1,9,10,14] se caracterizan por una imagen (aparente) de pasividad. El área de trabajo del usuario suele constar de dos elementos:

una zona de texto en la que el usuario escribe el texto a verificar y un botón que dice “Verificar” o algo parecido. La tarea del usuario se realiza sin intervención alguna por parte del programa y sin ningún tipo de consejo (en todo caso acompañado de guías de ayuda). La tarea del programa se limita a localizar errores cuando el usuario solicita explícitamente que se lleve a cabo la verificación del texto. La utilidad de este tipo de herramienta se manifiesta principalmente en las etapas finales del aprendizaje cuando el alumno ya sabe qué es lo que se le exigirá en la evaluación.

Los *constructores* [3,4,5,7,9,11,13] se caracterizan por un elevado grado de interactividad con el usuario. Tienen botones, menús y cuadros de diálogo mediante los cuales hacen participar (trabajar) al usuario, incluso lo aconsejan, pero también parte del trabajo lo realiza el propio programa. Este tipo de herramientas dan al estudiante mayor sensación de utilidad que los verificadores porque efectivamente ayudan a construir la solución: todavía es el usuario el que la construye pero no lo hace solo sino que es ayudado por el programa. Su utilidad se manifiesta principalmente en las etapas iniciales e intermedias del proceso de aprendizaje, cuando el alumno está inseguro aún con respecto a lo que se le exigirá en la evaluación y necesita que le ayuden a resolver los problemas.

Los *demostradores* [7,9,10,13] se caracterizan por su automatismo. En este tipo de herramientas el usuario es el que está en actitud pasiva: simplemente pulsa un botón y el programa construye íntegramente la solución. Su utilidad se manifiesta en el inicio del aprendizaje cuando el alumno todavía no sabe qué se le exigirá y necesita ejemplos que le muestren cómo se resuelven los problemas que se proponen.

Los *ejercicios* [3,4,10,12,13,15] son herramientas que proponen problemas al usuario y le proporcionan un espacio para que introduzca la solución, o bien le ofrecen mecanismos para que la construya. Puede darse el caso, incluso, de ofrecer al usuario la posibilidad de elegir que el programa le de la solución completa (como última salida si el usuario no es capaz de resolver el problema por sí solo). En cierto modo una herramienta de ejercicios es una combinación de los tres tipos expuestos en los párrafos anteriores pero especializados en la resolución de problemas concretos. Los ejercicios propuestos orientan al

alumno acerca de qué es lo que se espera de él y le permiten autoevaluarse de manera objetiva. Las mismas herramientas pero sin el ejercicio propuesto no dan al alumno esta oportunidad de autoevaluarse puesto que al fin y al cabo una evaluación consiste en la tarea de resolver ejercicios propuestos por el (profesor) evaluador y no por el alumno.

3. LSD

El programa LSD es de libre acceso y se puede obtener a través de internet en la dirección <http://ima.udg.es/~humet/lsd.html>. Es un verificador de *deducciones* i de *evaluaciones*. Las deducciones se usan para demostrar la validez y las evaluaciones se usan para demostrar la invalidez.

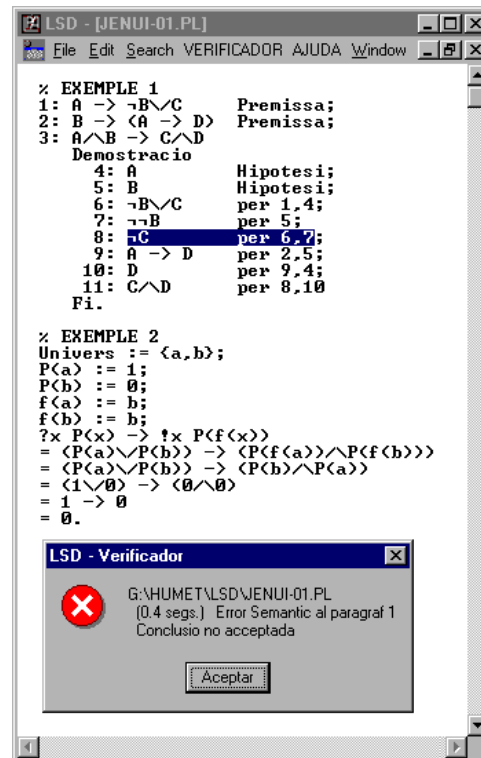


Figura 1. Texto lógico con errores

El funcionamiento del programa es muy simple: la ventana principal del programa dispone de un menú para la edición de textos ascii, una opción denominada VERIFICADOR, y un menú de ayuda. Con el editor el usuario puede escribir textos lógicos de acuerdo con una sintaxis que está descrita íntegramente en una de las opciones del menú de ayuda. Un texto lógico se compone de uno o más párrafos. Cada párrafo puede ser una deducción o una evaluación. Los textos escritos se pueden guardar en archivos y posteriormente se pueden recuperar.

Mediante la opción VERIFICADOR de la barra de menús el programa comprueba si el texto lógico contiene errores; si encuentra alguno se detiene en el primero de ellos indicando donde se ha producido y de qué tipo de error se trata (sintáctico/semántico). De este modo el usuario puede corregir tantas veces como desee el texto hasta conseguir que esté libre de errores.

La figura 1 muestra un texto lógico formado por dos párrafos, una deducción y una evaluación respectivamente. El cuantificador universal se representa mediante el carácter “!” y el cuantificador existencial mediante el carácter “?”. Aunque el texto contiene varios errores el verificador informa sólo acerca del primero que encuentra.

Un aspecto en el que LSD difiere de otras herramientas similares que tratan con deducciones es que la información que acompaña al programa no precisa cuál es el sistema de reglas utilizado y, por tanto, el usuario no puede llegar a conocer completamente cual es el conjunto total de reglas aplicables. Este debe inferirse a partir de los textos de ejemplo que acompañan al programa, de la sintaxis de los textos lógicos descrita en el menú de ayuda, y sobre todo de la experimentación con el propio programa: para saber si el programa posee una determinada regla basta con intentar aplicarla: si el programa rechaza el texto es que no la posee.

El hecho de no tener bien definido el conjunto de reglas aplicables no supone ningún inconveniente, sino al contrario, ya que libera al usuario de la tarea de memorizar las reglas y le permite centrar su atención en las relaciones de validez entre las fórmulas que se van obteniendo a lo largo de la deducción. En cualquier caso es suficiente con saber que LSD incorpora algunas reglas básicas proposicionales y de manejo de

cuantificadores, reglas de estructuración de demostraciones como son la demostración del condicional y la demostración indirecta (por reducción al absurdo), algunas equivalencias algebraicas y de cuantificadores, la regla de Skolem, la regla de substitución (de variables libres), y la regla de resolución. En la figura 2 se muestra una demostración correcta en la que se ha utilizado las reglas deductivas correspondientes al método de resolución.

```

!x?y <P(x)&Q(y)> --> ?y!x <P(x)&Q(y)>
Demostracio
% normalizacion
A1: !x?y <P(x)&Q(y)> hipotesi;
    !x <P(x)&Q(f(x))>;

A2: ~?y!x <P(x)&Q(y)> suposicio;
    !y~!x <P(x)&Q(y)>;
    !y?x ~<P(x)&Q(y)>;
    !y?x <~P(x)&~Q(y)>;
    !y <~P(g(y))&~Q(y)>;

% eliminacion de universales
B1: P(x)&Q(f(x)) per A1;
B2: ~P(g(y))&~Q(y) per A2;

% separacion de clausulas
C1: P(x) per B1;
C2: Q(f(x)) per B1;
C3: ~P(g(y))&~Q(y) per B2;

% resolucion
R1: (<~P(g(y))&~Q(y)>&P(x1)) per C3,C1;
    (<~P(g(y))&~Q(y)>&P(g(y)));
    ~Q(y);

R2: ~Q(y)&Q(f(x2)) per R1,C2;
    ~Q(f(x2)) &Q(f(x2));
    0;

```

Fi.

Figura 2. Demostración por el método de resolución

LSD es una herramienta pasiva, no propone ningún ejercicio sino que es el usuario (o bien el profesor) el que propone, frente a un problema no aconseja qué es lo que el usuario debe hacer, y no interviene en la construcción de la solución: simplemente se limita a detectar los errores cuando el usuario lo solicita.

Los verificadores como LSD aunque sean menos interactivos y atractivos que los constructores, no por ello son menos útiles. Los verificadores cumplen una misión en el proceso de autoevaluación mucho más cercana a la realidad cuando el sistema de evaluación consiste en un examen escrito clásico al que el alumno se deberá enfrentar sin ningún tipo de asistente activo que le ayude a hacer el trabajo.

4. Conclusión

Hemos justificado la necesidad de herramientas para el aprendizaje de la lógica, hemos proporcionado una clasificación de dichas herramientas en función del tipo de interactividad y hemos presentado una de estas herramientas: LSD (verificador) para el aprendizaje de la lógica de predicados de primer orden con funciones.

En cuanto a líneas futuras de trabajo creemos que en este momento tiene gran interés desarrollar herramientas de tipo *ejercicio* interactivas, con niveles variados de intervención por parte de la herramienta, y que en cualquier caso incorporen mecanismos de evaluación de la tarea realizada por el usuario. Deberán estar capacitadas para ayudar al usuario a obtener por sí mismo la solución si así lo solicita, y sobre todo incorporar mecanismos automáticos de evaluación que sean capaces de contabilizar el nivel de creatividad del usuario (+) así como la cantidad de ayuda recibida por el programa (-) en la obtención de la solución.

Referencias

- [1] C. Allen, M. Hand, *Logic Primer*, The MIT Press, 2001
(<http://logic.tamu.edu/daemon.html>)
- [2] E. Artacho, J. Humet, *RESVIS, un sistema de Resolución Visual guiado por el usuario*, Proyecto Final de Carrera, Ingeniería Técnica Informática, UdG, 1998.
- [3] J. Barwise, J. Etchemendy, *Hyperproof*, CSLI, Stanford University
(<http://www-csli.stanford.edu/hp>)
- [4] J. Barwise, J. Etchemendy, *Tarski's World*, CSLI, Stanford University
(<http://www-csli.stanford.edu/hp>)
- [5] K. Broda, S Zappacosta, *New Pandora*, Invited Talk, First International Congress on Tools for Teaching Logic, Salamanca, 2000.
(<http://www.doc.ic.ac.uk/~kb/NewPandora.html>)
- [6] J. Cuenca, *Lógica informática*, Alianza Editorial. 1985.
- [7] U. Endriss, *The Interactive Learning Environment WinKE for Teaching Deductive Reasoning*, First International Congress on Tools for Teaching Logic, Salamanca, 2000
(<http://www.dcs.kcl.ac.uk/staff/endriss/WinKE/>)
- [8] D. Goldson, S. Reeves, R. Bornat, *A Review of Several Programs for the Teaching of Logic*, The Computer Journal, vol. 36, nº4, 1993.
- [9] C. Gottschall, *Gateway to Logic*, Universidad de Viena, 2000
(<http://logik.phl.univie.ac.at/~chris/formular-uk.html>)
- [10] C. S. Layman, *The Power of Logic*, Seattle Pacific University, 1999.
(<http://www.poweroflogic.com/cgi/menu.cgi>)
- [11] F. Llorens, S. Mira, *Herramienta para la enseñanza de la Deducción Natural*, JENUI'2000, VI Jornadas sobre la Enseñanza Universitaria de la Informática, Alcalá de Henares, 25-26 septiembre 2000.
(<http://www.dccia.ua.es/logica/ADN/>)
- [12] J. Mårtensson, *On-line logic book (Interactive exercises)*, 1997.
(<http://home3.swipnet.se/~w-33039/OLL.C.html>)
- [13] A. Moreno, N. Budesca, *Mathematical Logic Tutor - Propositional Calculus*, First International Congress on Tools for Teaching Logic, Salamanca, 2000
(<http://www.etse.urv.es/recerca/banzai/toni/ILO>)
- [14] A. Trybulec, P. Rudnicki, *Using Mizar in Computer Aided Instruction of Mathematics*, Norwegian-French Conference of CAI in Mathematics, Oslo, 1993.
(<http://www.mizar.org/project/>)
- [15] Proyecto ARACNE, *Tools for Teaching Logic (Resultados/Ejercicios)*, 2000
(<http://aracne.usal.es/examenes/indice.html>)
- [16] First International Congress on Tools for Teaching Logic, Salamanca, Junio 2000
(<http://aracne.usal.es/congreso/congress.html>)