

ENSEÑANZA DE LÓGICA EN LA UNIVERSIDAD: EXPERIENCIAS EN EL USO DE HERRAMIENTAS INFORMÁTICAS

Luis Sierra
sierra@fing.edu.uy

Instituto de Computación, Universidad de la República, Uruguay

Tema: V.4 - Materiales y Recursos Didácticos para la Enseñanza y Aprendizaje de la Matemática

Modalidad: CB

Nivel educativo: Terciario - Universitario

Palabras clave: Lógica; Deducción natural; Herramientas informáticas

Resumen

Este artículo se centra en el uso de herramientas informáticas como apoyo para la enseñanza de la matemática. En particular, presentamos la herramienta Yoda que hemos desarrollado para la enseñanza de pruebas formales en el curso de Lógica de la Facultad de Ingeniería. La noción de prueba formal que usamos es la de un sistema de deducción natural de Gentzen basado en árboles. La herramienta Yoda está codificada en JavaScript y CSS, constituyendo una interfaz liviana que puede visualizarse en navegadores comunes, incluyendo celulares. El artículo también presenta algunos comentarios sobre otras experiencias que hemos realizado, así como algunas perspectivas para extender el uso de herramientas en el apoyo a otros cursos de matemáticas.

Introducción

En la Facultad de Ingeniería de la Universidad de la República se dicta un curso de Lógica que sigue los requerimientos curriculares de la ACM-IEEE para DS/Logic de acuerdo a (ACM-IEEE). El curso atiende a trescientos estudiantes, y cuenta con tres profesores y cinco ayudantes. Nos interesa proporcionar herramientas que permitan al estudiante poder verificar autónomamente su avance en el curso, en particular en lo que hace al desarrollo de las pruebas formales. La representación de estas pruebas se realiza mediante árboles de Gentzen (van Dalen, 1994).

A partir de una experiencia usando ProofWeb (Maxim Hendriks, Cezary Kaliszyk, Femke van Raamsdonk y Freek Wiedijk, 2010) en el 2009 y los comentarios de los estudiantes y docentes involucrados, decidimos implementar una herramienta con una interfaz más sencilla. La propuesta inicial se ha enriquecido con el uso de bibliotecas, cajas de diálogo, y un pequeño manual. Algunos de estos aportes han sido realizados por los propios estudiantes. En esta nota presentamos esta herramienta, Yoda (www.fing.edu.uy/inco/cursos/logica/software/yoda/YodaProp.html, www.fing.edu.uy/inco/cursos/logica/software/yoda/YodaPred.html), que hemos

desarrollado como material adicional para la enseñanza de deducción natural en nuestro curso.

La siguiente sección presenta brevemente las características de las pruebas formales y la deducción natural. Luego comentamos algunos requisitos que esperábamos de Yoda, y un ejemplo de su uso. Más adelante realizamos otras consideraciones acerca del uso de herramientas informáticas en nuestra tarea de enseñanza. Finalizamos la presentación con algunas conclusiones y señalando posibles extensiones de Yoda.

Pruebas formales y deducción natural

Decimos que cierta fórmula φ es consecuencia de un conjunto de fórmulas Γ , y escribimos $\Gamma \Rightarrow \varphi$, cuando tenemos algún argumento que garantiza la validez de φ tomando como válidas las afirmaciones de Γ . Una prueba formal dentro de un sistema lógico L es una descripción paso a paso de dicho argumento, construida de forma tal que permita su verificación por medio de una revisión detallada de la misma. Cada paso de una prueba formal está justificado por una regla bien definida en el sistema L ; si cada paso es correcto, la prueba lo es. En consecuencia, una prueba formal es verificable mediante una computadora.

En una prueba formal podemos usar ciertas fórmulas como axiomas a partir de los cuales llegar a la conclusión; el enfoque de la deducción natural brinda la posibilidad de agregar hipótesis temporarias en el argumento. Además, toda conectiva del lenguaje se explica mediante reglas específicas que permiten introducirlas o eliminarlas en la argumentación. Por ejemplo, en la siguiente imagen de Yoda podemos ver el uso de las reglas de eliminación e introducción de la disyunción para mostrar la conmutatividad de esa construcción.

YODA: Un asistente para derivaciones en PROP

A probar: $p \vee q \vdash q \vee p$

$$\begin{array}{c}
 \begin{array}{ccc}
 & p & q \\
 \hline
 & \text{IV2} & \text{IV1} \\
 p \vee q & q \vee p & q \vee p \\
 \hline
 & & \text{EV} \\
 & & q \vee p
 \end{array}
 \end{array}$$

p V q	Premisas
q V p	Conclusión

Derivar
Reset
Ayuda
Exportar a LaTeX

InCo - Uruguay - 2013

Algunos requisitos de Yoda

Comentamos ahora algunas características que consideramos deseable para nuestra herramienta. Buscamos una herramienta cuyo uso pueda comprenderse con la menor ayuda posible al usuario. Al no disponer de recursos humanos que puedan cubrir consultas de usuarios, estamos obligados a producir un programa con una interfaz simple y de uso fácilmente reconocible.

Un segundo requisito es que la distribución del programa debe abarcar distintos sistemas operativos. Descartamos cualquier solución fuertemente acoplada a algún sistema operativo, emuladores o máquinas virtuales. Optamos por una distribución del software sobre una máquina virtual de distribución universal, los navegadores. Una consecuencia de esta decisión es que parte importante de los problemas de visualización se dejan de lado, confiando en que los navegadores los resolverán.

Otro requisito es el uso de árboles de Gentzen. Nuestro curso usa este estilo de pruebas, y entendemos que las herramientas deben pensarse en función del curso. Francis J. Pelletier (2000) observa que el método de árboles de Gentzen no se usa mucho en libros de lógica elemental, quizá debido a dificultades tipográficas. Esta observación es vigente incluso al referirnos a asistentes para pruebas. Coincidimos con David Bostock (1997) quien afirma que los árboles de Gentzen permiten una buena visualización de las pruebas; en particular, permite entender cuáles son las hipótesis temporales supuestas en cada punto de la prueba.

Yoda es resultado de la búsqueda infructuosa de una herramienta sencilla y portátil para utilizar en nuestro curso; al no encontrarla, decidimos implementarla. En breve, Yoda es un programa sencillo para la asistencia en la construcción de pruebas, sin elementos de automatización, que delega gran parte de la interfaz a un navegador, y que satisface nuestros requisitos al día de hoy.

La herramienta

Yoda es un par de asistentes de prueba que se ejecutan sobre navegadores web, ocupando una única página, implementada sobre JavaScript, CSS y jQuery, con la simplicidad como una de sus características más destacables. El único requisito del sistema es la disponibilidad de un navegador web que cumpla con los estándares.

Para construir una prueba de $\Gamma \Rightarrow \varphi$ es necesario especificar el lenguaje de primer orden a usar, la conclusión y el conjunto de premisas. Luego, la prueba se construye orientada desde la conclusión.

De acuerdo a las reglas de nuestro sistema lógico, se generan dos subárboles. Aplicando distintas reglas se consigue la construcción del árbol de prueba completo. La aplicación de las reglas distingue el uso de premisas del uso de hipótesis auxiliares, que quedan marcadas de diferente manera. La prueba completa se ve a continuación.

YODA: Un asistente para derivaciones en FORM

A probar: $(\forall x)(P(x) \rightarrow Q(x)); (\exists x)P(f(x)) \vdash (\exists x)Q(x)$

	$(\forall x)(P(x) \rightarrow Q(x))$
	$\frac{}{P(f(x)) \rightarrow Q(f(x))} E\forall$
$P(f(x))$	$\frac{}{Q(f(x))} E\rightarrow$
	$\frac{}{(\exists x)Q(x)} I\exists$
$(\exists x)P(f(x))$	$\frac{}{(\exists x)Q(x)} E\exists$

InCo - Uruguay - 2013

Algunos criterios de clasificación

En los últimos treinta años se han implementado distintas herramientas destinadas a la enseñanza de la lógica. En las siguientes líneas proponemos algunos criterios para su clasificación (Sierra, 2008).

El primer criterio refiere a su relación con los cursos. En algunas ocasiones el curso se ata a la herramienta propuesta, como en Tarski's World (Dave Barker-Plummer, Jon Barwise y John Etchemendy, 1999) o Winke (Marcello D'Agostino y Ulrich Endriss, 1998). En otras ocasiones se proporciona una batería de herramientas que cubren una cantidad de aspectos propios de determinado curso, como en Logic Daemon (Colin Allen y Michael Hand, 2000). Finalmente, hay herramientas que toman un problema pequeño y lo tratan de forma independiente a cualquier curso, como Jape (2012).

Otro criterio refiere a la relación entre la herramienta y los estudiantes. En ciertos casos el estudiante tiene una alta interacción con la herramienta, indicando paso por paso cómo resolver el problema. Este enfoque es compartido por ETPS (Peter Andrews, Chad Brown, Frank Pfenning, Matthew Bishop, Sunil Issar y Hongwei Xi, 2004), ProofWeb (Hendriks et al., 2010) y Papuq (Jakub Sakowicz y Jacek Chrzaszcz, 2007), entre otras. Otras herramientas han optado por ofrecer una baja interactividad, proporcionando mecanismos automáticos que completan las tareas.

Perspectivas en el uso de herramientas informáticas: animación

En la última década han aparecido distintos aportes en la web que, en contexto de cursos, se pueden aprovechar como recursos didácticos. Por ejemplo, con respecto a la animación de problemas y algoritmos, a fines de los noventa estudiamos la posibilidad de incorporar elementos de animación para aclarar la construcción de las pruebas formales. En aquel momento, la propuesta más aceptable consistía en la programación en Java de pequeños códigos. Sin embargo, encontramos que el costo de producción de los mismos era demasiado elevado, y el mantenimiento de la propuesta nos resultaba prohibitivo. Hoy hemos experimentado con la biblioteca Raphael, obteniendo a costo razonable la visualización de algoritmos sencillos. Este camino, junto a las nuevas potencialidades de HTML en desarrollo, nos hace pensar que el uso con fines didácticos de tecnologías básicas de la web es una posibilidad cierta.

Conclusiones y perspectivas

Hemos desarrollado Yoda, un programa que permite verificar la construcción correcta de pruebas. Aún cuando esto tiene efectos interesantes sobre el dictado de nuestro curso, nuestra principal contribución es mostrar el potencial de los navegadores como soporte y apoyo en la enseñanza. Esta herramienta ha sido aceptada por los estudiantes, sin necesidad de clases específicas para ello, quienes han propuesto mejoras, detectado errores, y explicado su uso.

El desarrollo próximo previsto para esta herramienta es el desarrollo de una biblioteca de metareglas que permita una forma cómoda de extender esta herramienta a otros sistemas. En particular, nos resulta de interés extender esta herramienta para facilitar el razonamiento ecuacional y algunos estilos de pruebas inductivas.

Referencias bibliográficas

- Allen, C. y Hand, M. (2000). *Logic Primer*. MIT Press.
- Barker-Plummer, D., Barwise, J. y Etchemendy, J. (1999). *Tarski's World*. CSLI Publications.
- Bostock, D. (1997). *Intermediate Logic*. Oxford University Press.
- Van Dalen, D. (1994). *Logic and structure*. Springer-Verlag.
- Pelletier, F.J. (2000). A History of Natural Deduction and Elementary Logic Textbooks. En J. Woods y B. Brown (Eds.), *Logical Consequence: Rival Approaches, Vol. 1*, Capítulo 1, pp. 105-138. Oxford: Hermes Science Pubs.

- Andrews, P. B., Brown, C. E., Pfenning, F., Bishop, M., Issar, S. y Xi, H. (2004). Etps: A system to help students write formal proofs. *Journal of Automated Reasoning*, 32, 75-92.
- D'Agostino, M. y Endriss, U. (1998). WinKE: A Proof Assistant for Teaching Logic. *Proceedings of the First International Workshop on Labelled Deduction*.
- Hendriks, M., Kaliszyk, C., van Raamsdonk, F. y Wiedijk, F. (2010). Teaching logic using a state-of-the-art proof assistant. *Acta Didactica Napocensia*, 3(2), 35-48.
- Machín, B. y Sierra, L. (2011). Yoda: a simple tool for natural deduction. *Third International Congress on Tools for Teaching Logic*.
logicae.usal.es/TICTTL/actas/MachinSierra.pdf
- Sakowicz, J. y Chrzaszcz, J. (2007). Papuq: a Coq assistant. *Proceedings of PATE'07*, 79-96.
- Sierra, L. (2008). Enseñando deducción natural con Coq. *Proceedings of CIESC'08, XVI Congreso Iberoamericano de Educación Superior en Computación*.
- ACM-IEEE. Computer Science Curriculum 2008: An Interim Revision of CS 2001. www.acm.org/education/curricula/ComputerScience2008.pdf Consultado 30/06/13
- CSS. Cascading Style Sheet Snapshot 2010. www.w3.org/TR/CSS/ Consultado 30/06/13
- Raphaël: JavaScript Library. raphaeljs.com/ Consultado 30/06/13
- Jape (2012). users.comlab.ox.ac.uk/bernard.sufrin/jape.html Consultado 29/06/13
- JavaScript. developer.mozilla.org/en-US/docs/Web/JavaScript Consultado 30/06/13
- jQuery. jquery.com/ Consultado 30/06/13