

Ensino de Lógica através de estratégias de Demonstração e Refutação: A integração do tutorial interativo TryLogic via IMS Learning Tools Interoperability

Patrick Terrematte¹, João Marcos²

¹ Departamento de Filosofia (DFil/CCHLA)

² Departamento de Informática e Matemática Aplicada (DIMAp/CCET)
Universidade Federal do Rio Grande do Norte (UFRN) – Natal – RN – Brasil

terrematte@ppgsc.ufrn.br, jmarcos@dimap.ufrn.br

Abstract. *The present study proposes an interactive tutorial, namely the TRYLOGIC, aimed at teaching to solve logical conjectures either by proofs or refutations. The study also aims at: describing the integration of our infrastructure with the Virtual Learning Environment Moodle through the IMS Learning Tools Interoperability specification; presenting the Conjecture Generator that works for the tasks involving proving and refuting; and, finally evaluating the tool.*

1. Introdução

A Lógica é uma disciplina basilar para a Computação, todavia a dificuldade dos alunos ao aprenderem e apreciarem a relevância das habilidades lógicas sugere na realidade um problema curricular, sobretudo com o papel da Lógica na estrutura dos cursos e na metodologia de apresentação tradicional. A solução proposta por [Barland et al. 2000, p.4] é integrar o ensino de lógica com a utilização de ferramentas no contexto da aprendizagem para assimilar conceitos abstratos através de exemplos concretos. Anteriormente, isto nos motivou a elaborar uma metodologia de ensino para guiar o aluno em tarefas que combinam características do ensino presencial tradicional com recursos da educação à distância [Terrematte et al. 2011]. Nosso trabalho aqui é resultado de uma estratégia de ensino da Lógica para resolver tarefas de demonstrar ou refutar, e da criação de um tutorial interativo para guiar o aluno de forma gradual e sequencial: o TRYLOGIC¹.

2. Metodologia

A nossa metodologia durante a preparação do trabalho foi implementada através da investigação do estado da arte das ferramentas de ensino de Lógica. Para a construção de nossa infraestrutura optamos por utilizar e integrar as seguintes ferramentas:

- ProofWeb²: uma interface web para assistentes de demonstração de teoremas [Hendriks et al. 2010].
- Coq: um assistente de demonstração de teoremas em teorias formais.
- Gerador de Conjecturas: uma ferramenta de geração de exercícios.
- Moodle: um ambiente virtual de aprendizagem para acompanhamento de alunos.

Nossos objetivos específicos envolveram a integração das ferramentas descritas acima através da especificação IMS *Learning Tools Interoperability* (IMS LTI) e a implementação de uma teoria no Coq para refutar conjecturas em Lógica Proposicional.

¹Disponível para acesso através do endereço <http://lolita.dimap.ufrn.br/trylogic>.

²Disponível originalmente para acesso através do endereço <http://prover.cs.ru.nl/>

3. Lógica Proposicional para Demonstrar e Refutar

A Lógica Proposicional (L_{Prop}) é a parte da lógica que trata de proposições declarativas, ou seja, afirmações que podem ser verdadeiras ou falsas, e expressas através de conectivos lógicos e das regras que os manipulam. Sua importância se deve ao fato de que muitos conceitos complexos e sofisticados de lógica matemática aparecem primariamente aqui num contexto mais simples. Todas as sentenças proposicionais podem ser formadas composicionalmente a partir dos seguintes conectivos lógicos (com sua aridade indicada no índice), respectivamente: a conjunção (\wedge_2), a disjunção (\vee_2), a implicação (\rightarrow_2), a bi-implicação (\leftrightarrow_2), a negação (\neg_1) e o absurdo (\perp_0).

A linguagem da lógica proposicional é definida recursivamente, como se segue, considere um conjunto de **letras proposicionais** $Prop = \{A, B, C, D, E, F, A_0, B_0, C_0, D_0, E_0, F_0, A_1, B_1, C_1, D_1, E_1, F_1, \dots\}$; um conjunto de **conectivos** $Conec = \{ \perp_0, \neg_1, \rightarrow_2, \leftrightarrow_2, \wedge_2 \text{ e } \vee_2 \}$; um **alfabeto proposicional** $\Sigma = Prop \cup Conec \cup \{ (,) \}$; e **esquemas de fórmulas** construídas através de letras do proposicionais representadas pelas letras gregas ($\alpha, \beta, \gamma, \delta, \epsilon, \phi, \psi, \dots$) e denotam meta-fórmulas que podem ser substituídas por **instâncias** da meta-fórmula.

Se desejarmos **demonstrar** conjecturas — como a asserção $\Gamma \vdash \alpha$ segundo a que a partir de um conjunto Γ de fórmulas é derivada uma fórmula α —, é preciso tentar construir uma árvore de derivação usando as regras de dedução da teoria N_p , que no caso da conjunção, são expressas pelas seguintes regras:

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} (\wedge I) \quad \frac{\alpha \wedge \beta}{\alpha} (\wedge E_1) \quad \frac{\alpha \wedge \beta}{\beta} (\wedge E_2)$$

Por outro lado, para **refutar** conjecturas — como no caso da asserção $\Gamma \not\vdash \alpha$ segundo a qual existe uma valoração v que satisfaz Γ , mas falsifica α —, é preciso construir árvores de refutação na Teoria Sem_p , por exemplo, aplicando algumas das seguintes regras/táticas da conjunção:

$$\frac{v \Vdash \alpha \quad v \Vdash \beta}{v \Vdash \alpha \wedge \beta} (\wedge T) \quad \frac{v \not\vdash \alpha}{v \not\vdash \alpha \wedge \beta} (\wedge F1) \quad \frac{v \not\vdash \beta}{v \not\vdash \alpha \wedge \beta} (\wedge F2)$$

Através destas e de outras regras, nós implementamos a teoria Sem_p no ProofWeb e podemos mostrar que uma dada sentença não é consequência semântica de um certo conjunto de sentenças dadas. De forma diferente do sistema N_p , as regras de inferência envolvem a manipulação das asserções sobre a interpretação de fórmulas satisfeitas ou falsificadas por uma valoração v .

Neste sistema uma refutação de $(\neg A \rightarrow B) \wedge (\neg A \wedge \neg B)$ é produzida pela exibição, acompanhada da respectiva justificativa de que existe pelo menos uma valoração v que falsifica tal sentença. Uma das vantagens didáticas do TRYLOGIC, em conjunto com o ProofWeb, é sua coerência com ao **Princípio de Representação Múltipla** (*The Modality Effect*) da Teoria da Carga Cognitiva [Sweller et al. 2011, p.129]. Com a Figura 1 podemos observar que sem a sequência de táticas inserida no lado esquerdo (a) não seria possível verificar o objeto apresentado no lado direito (b), pois a sequência de táticas representa uma justificativa para a refutação produzida. Assim, a visualização de (b) possui valor didático útil, através da comunicabilidade da estrutura da refutação em (a).

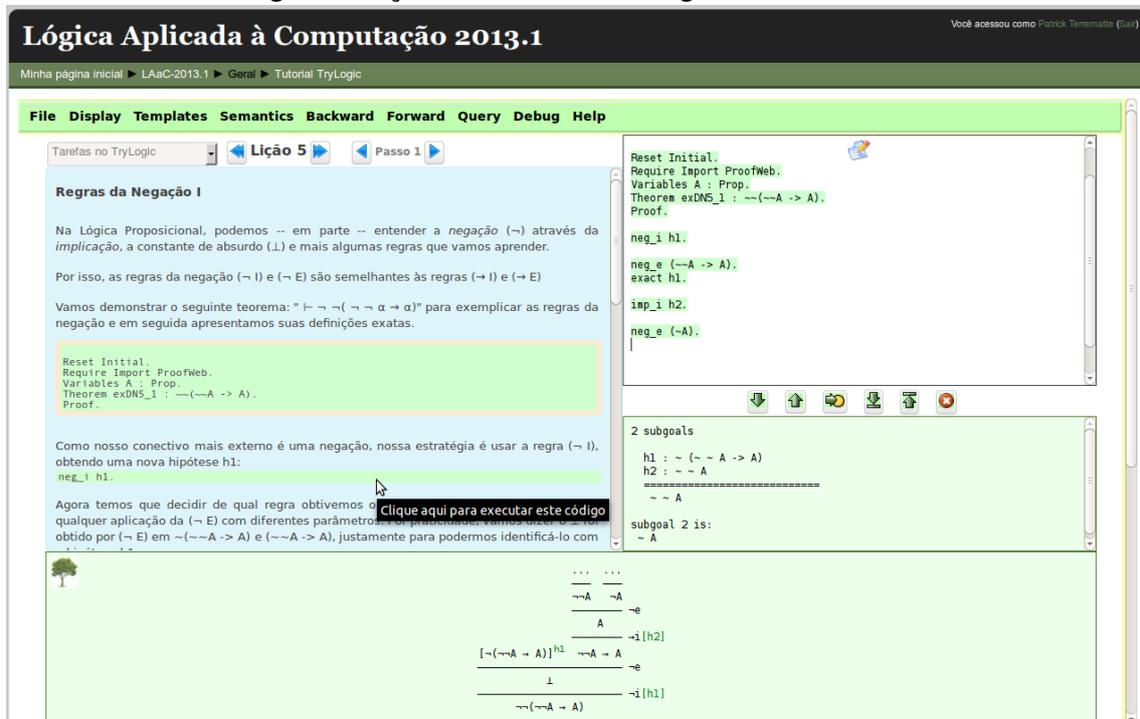
Figura 1. Script para produção de refutação no ProofWeb

<pre> Reset Initial. Require Import Semantics. Parameter A B C D : Prop. Hypothesis f1 : (v -/- A). Hypothesis f2 : (v -/- B). Theorem sem_ex2 : (v -/- ((~A->B) /\ (~A/\~B))). Proof. conjF1. impF. negT. exact f1. exact f2. Qed. </pre>	
<p>a) Script com sequência de táticas</p>	<p>b) Árvore resultante construída no ProofWeb</p>

4. O tutorial interativo TRYLOGIC para Demonstrar ou Refutar

O TRYLOGIC é um tutorial interativo desenvolvido a partir do TryOcaml³ e do ProofWeb. O objetivo principal do TRYLOGIC é diminuir a sobrecarga cognitiva através de um tutorial passo-a-passo de forma sistemática, apresentado diferentes tópicos da lógica relacionados ao processo de DxR conjecturas lógicas. A ideia básica do projeto é fornecer um tutorial interativo para ensinar o aluno a solucionar problemas lógicos.

Figura 2. Lições do TRYLOGIC integradas ao Moodle



O Gerador de Conjecturas em Lógica Proposicional foi desenvolvido através da

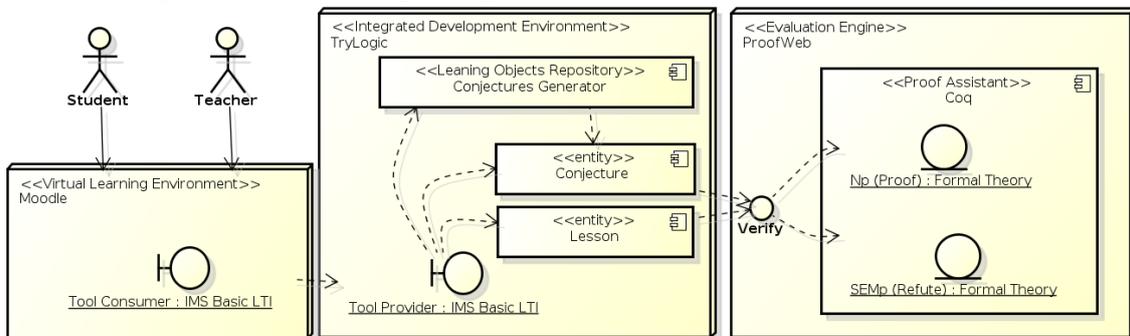
³Disponível para acesso através do endereço <http://try.ocamlpro.com/>

colaboração de alunos de graduação bolsistas de iniciação científica⁴. A implementação foi realizada a partir de um gerador de fórmulas em comunicação com o SAT-*solver* Limboole⁵ para avaliar as fórmulas proposicionais. O Gerador de Conjecturas realiza a criação de conjecturas, no formato de exercício individualizado para o Coq, diretamente na área dos alunos no ProofWeb. Os alunos recebem cada tarefa ou conjectura em um *template* para demonstrar ($\Gamma \vdash \alpha$) teoria N_p e outro para refutar ($\Gamma \not\vdash \alpha$) na teoria SEM_p .

Com o objetivo de centralizar o uso das nossas ferramentas, nós investigamos um conjunto de alternativas de especificação da comunicação entre ferramentas de ensino, e de acordo com [Alario-Hoyos and Wilson 2010] e [Queirós and Leal 2012], uma das alternativas mais representativas para infraestruturas de integração entre plataformas de ensino é a *IMS Learning Tools Interoperability*⁶.

No nosso caso, a ferramenta provedora é o TRYLOGIC e a ferramenta consumidora pode ser qualquer plataforma de aprendizagem como o ATutor, Sakai e Moodle. O nosso objetivo através desta integração é fornecer uma infraestrutura genérica para a comunicação centralizada das nossas ferramentas (Gerador de Conjecturas, ProofWeb e TRYLOGIC) com qualquer uma plataforma de aprendizagem de nossa escolha que contenha uma implementação do *IMS Basic LTI*. A integração permite que um aluno cadastrado em sua plataforma, com nome de ferramenta consumidora, possua uma conta criada numa ferramenta provedora. Com o *IMS LTI*, o acesso ao TRYLOGIC é realizado pelos alunos e tutores no Moodle, pois as contas são criadas automaticamente no ProofWeb e no Moodle a partir do primeiro acesso dos usuários. O resultado da integração do TRYLOGIC é representado pelo diagrama de implantação na Figura 3.

Figura 3. Visão de implantação do TRYLOGIC com o Moodle (*IMS LTI*)



Dessa forma, qualquer colaborador que deseje utilizar o TRYLOGIC em qualquer plataforma como o Moodle poderá adicioná-la como ferramenta externa usando a nossa *Launch URL*, uma *Consumer Key* e uma *Shared Secret* cadastradas. Assim não é necessário instalar o TRYLOGIC, nem são necessárias permissões especiais do servidor.

Para realizar experimentos com a infraestrutura desenvolvida, nos semestres de 2012.2 e 2013.1 da disciplina de Lógica Aplicada à Computação de UFRN, nós utilizamos o TRYLOGIC para tarefas de demonstração e refutação. A tarefa de Demonstrar ou

⁴O Gerador de Conjecturas foi contribuição de alunos de iniciação científica Hudson Geovane e Elias Amaral, com o código disponível em <http://lolita.dimap.ufrn.br/logicamente-ge/>.

⁵Disponível em <http://fmv.jku.at/limboole/>.

⁶O *IMS LTI* foi desenvolvida em 2006 pela *IMS Global Learning Consortium*, e está disponível em <http://www.imsglobal.org/lti/index.html>.

Refutar (DxR) foi atribuída a alunos da graduação que cursaram a disciplina de Lógica Aplicada à Computação (LAaC). Nosso maior desafio foi ensinar a resolução de tarefas de demonstrar ou refutar exclusivamente através do TRYLOGIC. Nas aulas presenciais ensinamos apenas os fundamentos teóricos. As experiências de aplicação serviram para verificar que o TRYLOGIC possibilita o entendimento do procedimento dedutivo no assistente de demonstração Coq para alunos que tiveram um breve contato teórico com o conteúdo de Dedução Natural para Lógica Proposicional. Assim, com o TRYLOGIC o processo de experimentação via tentativa e erro é ensinado em um ambiente controlado.

5. Considerações finais e trabalhos futuros

Neste trabalho foi apresentada uma infraestrutura de convergência de ferramentas de ensino de Lógica com objetivo de: organizar em passo-a-passo a exposição dos conteúdos de Dedução Natural e Semântica Proposicional de forma sequencial e interativa; fornecer ao aluno tarefas interativas de autoavaliação; e integrar o Gerador de Conjecturas e TRYLOGIC com o Moodle via *IMS Basic LTI*. As nossas contribuições sobre o ensino de Lógica fazem parte de uma iniciativa que precisa ser aprimorada, em particular:

- a implementação de métricas de dificuldade para derivações para o Gerador de Conjecturas com o objetivo de garantir a existência de uma métrica efetiva; e
- a geração de conjecturas para a Lógica de Primeira Ordem com o objetivo de produzir novas tarefas e novas lições de Lógica de Primeira Ordem.

Referências

- Alario-Hoyos, C. and Wilson, S. (2010). Comparison of the main alternatives to the integration of external tools in different platforms. In *Proceedings of the International Conference of Education, Research and Innovation, ICERI*, pages 3466–3476.
- Barland, I., Felleisen, M., Fisler, K., Vardi, M. Y., and Kolaitis, P. (2000). Integrating Logic into the Computer Science Curriculum. *Innovation and Technology in Computer Science Education*.
- Hendriks, M., Kaliszyk, C., Raamsdonk, F. V., and Wiedijk, F. (2010). Teaching logic using a state-of-the-art proof-assistant. *Acta Didactica Napocensia*, 3(2):35–48. Disponível em: <http://dppd.ubbcluj.ro/adn/article_3_2_4.pdf>.
- Queirós, R. and Leal, J. P. (2012). Orchestration of e-learning services for automatic evaluation of programming exercises. *Journal of Universal Computer Science*, 18(11):1454–1482.
- Sweller, J., Ayres, P., and Kalyuga, S. (2011). *Cognitive Load Theory*. Explorations in the Learning Sciences, Instructional Systems and Performance Technologies. Springer.
- Terrematte, P., Costa, F., and ao Marcos, J. (2011). Logicamente: A Virtual Learning Environment for logic based on Learning Objects. In *Third International Conference Tools for Teaching Logic.*, volume 6680 of *Lecture Notes in Artificial Intelligence*, pages 223–230. Springer-Verlag, Berlin.