

# LABLOG: UM LABORATÓRIO VIRTUAL DE LÓGICA

Guilherme Vaz França Reis<sup>1</sup>, Silvio do Lago Pereira<sup>2</sup>

<sup>1</sup>Aluno do Curso de Análise e Desenvolvimento de Sistemas – FATEC-SP

<sup>2</sup>Prof. Dr. do Departamento de Tecnologia da Informação – FATEC-SP

guilherme.reis01@fatec.sp.gov.br, slago@fatecsp.br

## Resumo

Lógica simbólica é um conteúdo normalmente abordado em Matemática Discreta, uma disciplina básica em cursos na área de computação. Embora o domínio desse conteúdo seja fundamental na formação de profissionais desses cursos, para muitos estudantes, a lógica simbólica é apenas uma teoria sem muita aplicação prática. Em decorrência deste fato, e devido à dificuldade que os estudantes têm de lidar com conceitos abstratos, o estudo de lógica simbólica acaba sendo negligenciado. Ademais, como os conceitos de lógica simbólica são geralmente difíceis de representar e entender, o desenvolvimento de *software* educacional neste domínio ainda é considerado um grande desafio. Neste contexto, o objetivo deste artigo é descrever um protótipo de laboratório virtual para realização de experimentos em lógica simbólica, que foi desenvolvido em Prolog, e relatar resultados empíricos obtidos com ele.

## 1. Introdução

Lógica simbólica [1] é um formalismo que representa conhecimento e raciocínio usando expressões simbólicas, em vez de sentenças de uma linguagem natural. Há vários sistemas de lógica simbólica, sendo que os principais são a *lógica proposicional*, cujo foco é a proposição atômica (i.e., sentença que pode ser verdadeira ou falsa, mas não as duas coisas simultaneamente), e a *lógica de predicados*, cujo foco é a generalização das proposições, com base em conceitos ainda mais abstratos que permitem, por exemplo, a análise da estrutura interna das proposições. Nestes dois sistemas lógicos, o conhecimento é representado por expressões simbólicas, denominadas *fórmulas*, enquanto o raciocínio pode ser representado por padrões de manipulação sintática de expressões simbólicas, denominados *regras de inferência* (e.g., *modus ponens* e *modus tollens*).

Devido ao seu alto nível de abstração, conceitos de lógica simbólica são geralmente difíceis de representar e entender. De fato, uma pesquisa recentemente publicada, conduzida por psicólogos cognitivos [2], mostrou que boa parte dos estudantes universitários não é capaz de aplicar corretamente as regras de inferência clássicas (Figura 1).

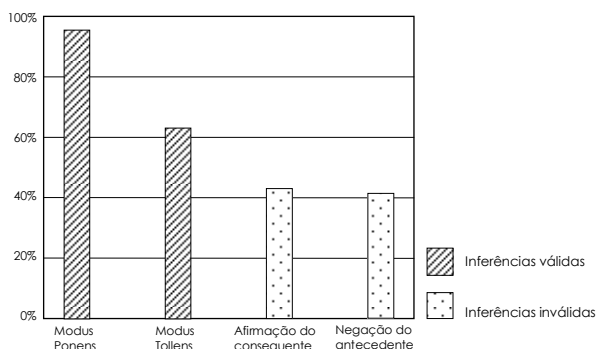


Figura 1 – Inferências feitas por estudantes universitários [2].

O resultado dessa pesquisa evidencia um problema que precisa ser solucionado, pois um profissional que não é capaz de diferenciar inferências válidas e inválidas está propenso a cometer erros de raciocínio que podem ter consequências desastrosas para a sua carreira, ou mesmo para a vida de outras pessoas afetadas por suas decisões. Uma forma de contribuir para a solução desse problema é, justamente, propor *software* educacional para facilitar a aprendizagem de lógica simbólica em Matemática Discreta. Porém, devido à complexidade do assunto, o desenvolvimento de *software* neste domínio ainda é considerado um grande desafio para a área de computação [3].

Neste contexto, este artigo tem como objetivo propor um *laboratório virtual de lógica simbólica*, que permite ao estudante realizar experimentos abstratos em cenários virtuais, que são análogos a cenários concretos. Esse *software* educacional, desenvolvido em Prolog [4] com base em conceitos de aprendizagem ativa [5], é um ambiente de simulação computacional interativo, que visa estimular o estudante a se envolver no seu próprio processo cognitivo.

O restante do artigo está organizado do seguinte modo: a Seção 2 introduz a fundamentação teórica do trabalho; a Seção 3 descreve a arquitetura e as funcionalidades do protótipo de laboratório virtual de lógica que foi desenvolvido; a Seção 4 relata experimentos feitos com o protótipo; e a Seção 5 apresenta as conclusões finais do trabalho.

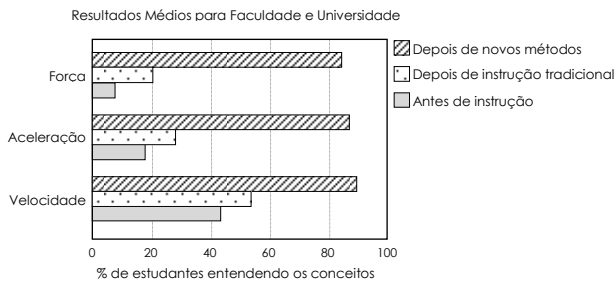
## 2. Fundamentação Teórica

Os fundamentos teóricos desse trabalho são aprendizagem ativa, laboratório virtual e aprendizado multimídia.

### 2.1. Aprendizagem Ativa

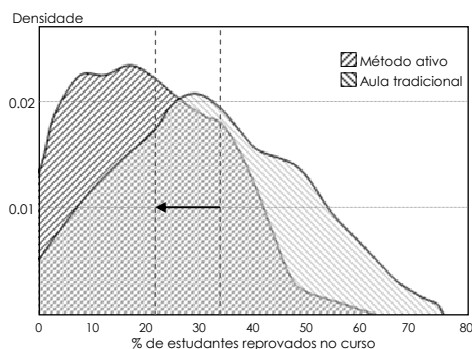
Aprendizagem ativa [5] pode ser definida como qualquer método de ensino que engaja o aluno no seu processo de aprendizagem. Embora essa definição englobe atividades pedagógicas tradicionais como, por exemplo, lição de casa, na prática, ela refere-se às atividades propostas em sala de aula. A aprendizagem ativa requer que os alunos realizem atividades de aprendizagem significativas e que pensem sobre o que estão fazendo. Ao contrário de métodos tradicionais, como aulas expositivas, que admitem um papel passivo do aluno, a aprendizagem ativa exige *ação* e *engajamento* e, portanto, um papel *proativo* do aluno.

Apesar das dificuldades de medir e comparar a eficiência de diferentes métodos de ensino (e.g, definição precisa dos métodos comparados e escolha dos indicadores de eficiência), pesquisas feitas na área de educação têm encontrado fortes evidências empíricas de uma maior eficiência da aprendizagem ativa, quando comparada a métodos de ensino tradicionais [5]. Por exemplo, numa pesquisa feita com milhares de alunos universitários americanos, Law et al. [6] observaram que, enquanto os métodos tradicionais aumentaram a compreensão dos alunos sobre conceitos de cinemática em até 15%, os métodos de aprendizagem ativa puderam aumentar a compreensão em até 90% (Figura 2).



**Figura 2** – Aumento da compreensão de conceitos de física [6].

Outra pesquisa interessante, conduzida por *Freeman et al.* [7], constatou que o aprendizado ativo aumenta o desempenho médio dos alunos universitários dos cursos de ciências, tecnologia, engenharia e matemática (STEM) em cerca de 6% e que os alunos com aulas tradicionais eram 1,5 vezes mais propensos à reprovação do que os alunos submetidos ao aprendizado ativo (Figura 3).



**Figura 3** – Diminuição de reprovação de 33,8% para 21,8% [7].

## 2.2. Laboratório Virtual

Laboratório baseado em computador é um método de aprendizagem ativa que estimula o engajamento dos alunos. Uma comparação dos resultados de 11 aulas ministradas por 6 professores, com e sem laboratórios, mostrou que o uso dos laboratórios computacionais aliados a atividades engajadoras resultou em uma melhoria significativa (medida por meio da aplicação de testes de múltipla escolha cuidadosamente elaborados) em comparação com as aulas expositivas tradicionais [8].

O laboratório para uso de métodos de aprendizagem ativa pode ser *real* (*hands-on*) ou *virtual* (i.e., sistema que realiza simulações computacionais [9] ou que controla remotamente experimentos feitos em um laboratório real).

Um estudo feito por *Hawkins* [10] separou aleatoriamente estudantes de Química Geral II em dois grupos: um deles realizou experimentos de eletroquímica em um laboratório real, enquanto o outro realizou os mesmos experimentos em um laboratório virtual correspondente. Para comparar os dois grupos, foram aplicados testes conceituais antes e depois das aulas em laboratório (real e virtual), além de um teste final *hands-on* para ambos os grupos. Após a análise dos resultados dos testes aplicados, observou-se que não houve diferença estatisticamente significativa entre os dois grupos. Outro estudo feito em duas grandes universidades também mostrou que os laboratórios virtuais foram tão eficazes quanto os tradicionais *hands-on* no estudo de física [11].

## 2.3. Teoria Cognitiva de Aprendizagem Multimídia

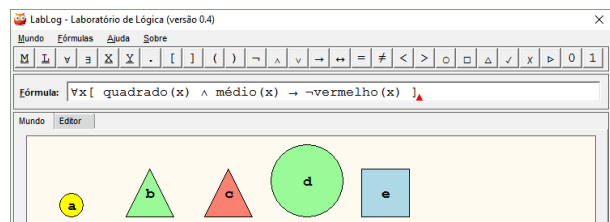
No ensino de lógica simbólica, um laboratório virtual é um recurso pedagógico para sedimentar o conhecimento adquirido pelo aluno em sala de aula, na medida em que: (a) permite que conceitos abstratos da lógica sejam concretizados em representações visuais e (b) exige um papel proativo do aluno no seu próprio processo cognitivo.

De fato, segundo a *Teoria Cognitiva de Aprendizagem Multimídia* [12, 13], o aprendizado é mais profundo e significativo quando os conceitos são concretizados em imagens associadas a palavras do que quando eles são definidos apenas textualmente. Sendo assim, ambientes de aprendizagem computacionais (e.g., laboratórios virtuais) oferecem inúmeras possibilidades para aprimorar o aprendizado, especialmente quando eles permitem que os estudantes manipulem objetos e verifiquem propriedades decorrentes desta manipulação.

Há diversos laboratórios virtuais para estudo de vários assuntos; porém, para lógica, eles ainda são muito raros. Especialmente, para lógica de predicados, *Tarski's World* [14] é um dos poucos laboratórios virtuais existentes. Usando o *Tarski's World*, o estudante pode entender mais facilmente a semântica das proposições, dos conectivos e dos quantificadores universal e existencial. Ele permite ao aluno construir mundos virtuais, compostos por blocos de vários formatos e tamanhos, distribuídos numa plataforma similar a um tabuleiro de xadrez, e analisar a semântica de fórmulas que descrevem propriedades desses mundos. O sistema é bastante interativo e oferece *feedback* sobre eventuais erros, fazendo com que o estudante se sinta mais envolvido e motivado a aprender os conceitos estudados. Apesar disso, *Tarski's World* requer o pagamento de uma licença de uso e exige que o usuário tenha conhecimento da língua inglesa, o que pode ser mais um fator limitante.

## 3. O Sistema LabLog

Tendo em mente a dificuldade dos alunos com lógica (Seção 1) e a efetividade dos métodos de aprendizagem ativa, aliados ao uso de laboratórios virtuais, no processo de aquisição cognitiva (Seção 2), um protótipo de laboratório virtual de lógica simbólica, *LabLog*<sup>\*</sup>, foi criado em SWI-Prolog (*Multi-threaded*, 64 bits, versão 7.4.0) [4]. A janela principal desse sistema é apresentada Figura 4.



**Figura 4** – Janela principal do sistema *LabLog*.

O *LabLog* é um ambiente interativo que oferece recursos para a realização de experimentos em lógica proposicional e lógica de predicados. Com esse sistema, o aluno pode: criar, salvar e carregar *mundos* contendo objetos; criar, salvar e carregar *fórmulas* lógicas envolvendo propriedades e relações entre objetos; avaliar as fórmulas; e esclarecer dúvidas a respeito de seus valores lógicos.

<sup>\*</sup> Disponível em [www.ime.usp.br/~slago/lablog.pl](http://www.ime.usp.br/~slago/lablog.pl).

### 3.1. Arquitetura do Sistema

O sistema *LabLog* é composto de um módulo principal e mais cinco módulos auxiliares, como mostra a Figura 5.

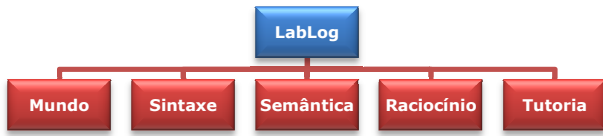


Figura 5 – Arquitetura do sistema *LabLog*.

Segue uma breve descrição de cada módulo:

- O módulo *LabLog* cria a interface gráfica de usuário (com a biblioteca gráfica XPCE [15]) e promove a integração entre os demais módulos do sistema.
- O módulo *Mundo* cria, altera e destrói objetos do mundo virtual. Cada objeto é representado na base de conhecimentos do Prolog por um fato do formato objeto ( $N, F, T, C, P$ ), onde  $N$  é o nome,  $F$  é a forma (círculo, quadrado ou triângulo),  $T$  é o tamanho (pequeno, médio ou grande),  $C$  é a cor (azul, verde, amarelo ou vermelho), e  $P$  é a posição (coordenadas  $x$  e  $y$ ) do objeto no mundo virtual.
- O módulo *Sintaxe* traduz textos digitados pelo usuário, representando fórmulas da lógica de predicados, num formato que pode ser interpretado pelo sistema. Por exemplo, o texto ' $\forall x. \text{quadrado}(x)$ ' é traduzido em  $\forall (\_G4651, \text{quadrado}(\_G4651))$ , sendo  $\_G4651$  o endereço interno da variável  $x$ . A tradução é feita por uma gramática especificada em notação DCG (*Definite Clause Grammar*), que é nativa do Prolog.
- O módulo *Semântica* interpreta as fórmulas traduzidas e avalia seus valores lógicos. Além da semântica dos conectivos e quantificadores, esse módulo define a semântica dos predicados que representam propriedades dos objetos (e.g.,  $\text{quadrado}/1$ ,  $\text{grande}/1$  e  $\text{azul}/1$ ), bem como predicados que representam relações entre objetos (e.g.,  $\text{acima}/2$  e  $\text{esquerda}/2$ ).
- O módulo *Raciocínio* cria, avalia e salva conjuntos de fórmulas digitadas pelo usuário no editor de textos do sistema (usando funcionalidades implementadas nos módulos *Mundo*, *Sintaxe* e *Semântica*).
- O módulo *Tutoria* implementa um algoritmo de explanação de fórmulas baseado no uso de *equivalências lógicas*, *decomposição estrutural recursiva* e busca de *contra-exemplos* para sub-fórmulas. Por exemplo, suponha que no mundo haja apenas um círculo grande  $a$  e um círculo pequeno  $b$ , e que o usuário informe que ele acha que, nesse mundo, a fórmula  $\forall x [\text{círculo}(x) \rightarrow \text{pequeno}(x)]$  é verdadeira. Para encontrar o erro do usuário, o algoritmo aplica a equivalência  $\alpha \rightarrow \beta \equiv \neg \alpha \vee \beta$ , e obtém a fórmula  $\forall x [\neg \text{círculo}(x) \vee \text{pequeno}(x)]$ , cuja decomposição estrutural resulta em  $\neg \text{círculo}(x)$ ,  $\text{círculo}(x)$  e  $\text{pequeno}(x)$ ; depois, ele nota que  $x=a$  é um contra-exemplo para  $\text{pequeno}(x)$ . Finalmente, ele mostra ao usuário que, para  $x=a$ , as fórmulas  $\neg \text{círculo}(a)$  e  $\text{pequeno}(a)$  são falsas; que, consequentemente, a fórmula  $\neg \text{círculo}(a) \vee \text{pequeno}(a)$  é falsa; e que, portanto, a fórmula  $\forall x [\neg \text{círculo}(x) \vee \text{pequeno}(x)]$  não pode ser verdadeira (no mundo considerado).

### 3.2. Funcionalidades do Sistema

As funcionalidades do *LabLog* são disponibilizadas em uma interface gráfica composta por *menu* (com opções para abrir e salvar mundos e fórmulas), *teclado virtual* (com botões para inserção de quantificadores e operadores, além de botões para funções básicas tais como criação de mundos aleatórios, avaliação de fórmulas e ativação de tutoria), *caixa de entrada* (para digitação de fórmulas), e *abas* (uma para exibição do mundo e outra para o editor de textos), como indicado na Figura 6.

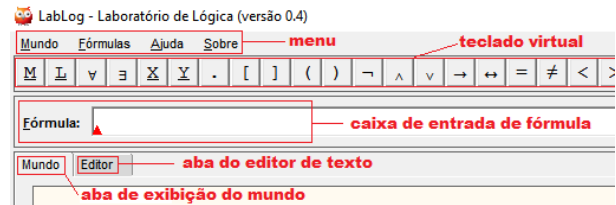


Figura 6 – Componentes da interface gráfica do *LabLog*.

#### 3.2.1. Criação de Objetos e Mundos

Há duas formas de criação de mundos: a primeira, e mais simples, consiste em clicar o botão  $M$  do teclado virtual (nesse caso, o sistema constrói automaticamente um mundo composto de um número aleatório de objetos, com propriedades escolhidas aleatoriamente); a segunda consiste em clicar com o botão direito na aba do mundo, escolher uma forma para o objeto a ser criado (nesse caso um objeto com a forma escolhida é criado; porém, suas demais propriedades são escolhidas aleatoriamente pelo sistema) e, depois, clicar com o botão direito sobre o objeto criado, para alterar suas propriedades. A Figura 7 ilustra o processo de criação de um objeto específico.

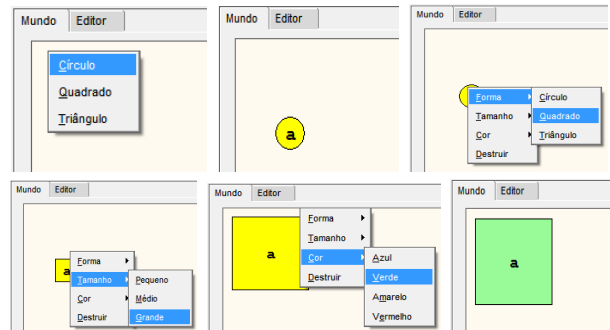


Figura 7 – Etapas para criação de um quadrado grande verde.

Os objetos criados são nomeados automaticamente (de  $a$  a  $z$ ) e distribuídos por linha. Note que a forma inicial do objeto criado também pode ser modificada durante o processo de edição de propriedades. Depois de criado, o mundo pode ser salvo num arquivo com extensão *.mnd*, via menu *Mundo* (depois de salvo, ele pode ser aberto).

#### 3.2.2. Criação e Avaliação de Fórmulas

Para criar uma fórmula, basta digitá-la na caixa de entrada, usando o teclado virtual para inserção de quantificadores e operadores, como mostra a Figura 8.

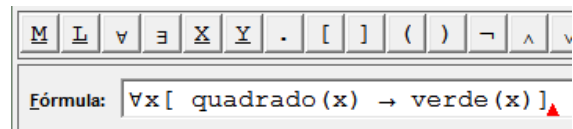


Figura 8 – Criação de fórmula.

Para avaliar uma fórmula, basta clicar o botão ✓ do teclado virtual. Então, o sistema informa se a fórmula é verdadeira, falsa ou se tem erro de sintaxe (Figura 9).

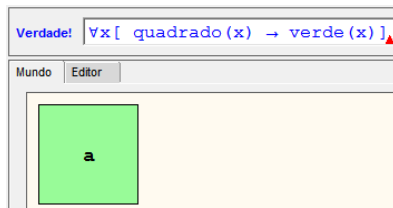


Figura 9 – Avaliação de fórmula no mundo considerado.

Há também a opção de criar um conjunto de fórmulas, usando o editor de textos, e avaliá-las clicando o botão ►, do teclado virtual, como na Figura 10. Depois de criado, o conjunto de fórmulas pode ser salvo num arquivo .frm, via menu Fórmulas.

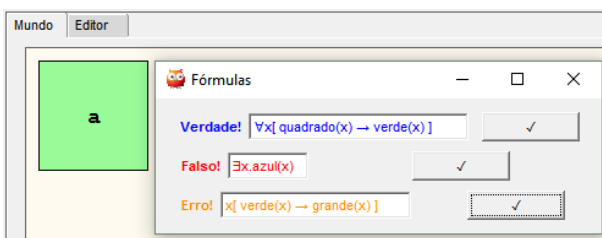


Figura 10 – Avaliação de um conjunto de fórmulas do editor.

Por convenção, todo nome quantificado (universal ou existencialmente) é tratado como uma variável, sendo que os demais são tratados como constantes. Por exemplo, em azul(a), o nome a refere-se ao objeto a; enquanto em  $\exists a. azul(a)$ , o nome a refere-se a um objeto qualquer.

### 3.2.3. Tutoria e Explicação de Semântica

A tutoria visa explicar ao usuário o motivo pelo qual sua intuição a respeito do valor lógico de uma fórmula está errada. Essa funcionalidade é essencial para um software educacional, pois a correção passo a passo de eventuais erros dos alunos permite que eles melhorem seus desempenhos em casos similares no futuro.

Para ativar esse sistema, o usuário precisa digitar uma fórmula e, caso ele ache que a fórmula é falsa (ou verdadeira), clicar no botão 0 (ou 1) do teclado virtual.

Fórmulas não quantificadas podem ser facilmente verificadas pelo aluno, pela simples observação do mundo, que está na aba Mundo. No caso de fórmulas quantificadas, existem quatro casos para correção:

- Se uma fórmula universal ( $\forall$ ) é verdadeira, e o aluno a avaliou como falsa, o sistema pede ao aluno o nome do objeto do mundo que falsifica a fórmula, visando esclarecer que não existe tal objeto.
- Se uma fórmula universal ( $\forall$ ) é falsa, e o aluno a avaliou como verdadeira, o sistema mostra ao aluno o nome de um objeto que torna a fórmula falsa.
- Se uma fórmula existencial ( $\exists$ ) é verdadeira, e o aluno a avaliou como falsa, o sistema mostra ao aluno o nome de um objeto que a torna verdadeira.
- Se uma fórmula existencial ( $\exists$ ) é falsa, e o aluno a avaliou como verdadeira, o sistema pede ao aluno o nome do objeto do mundo que falsifica a fórmula, visando esclarecer que não existe tal objeto.

A Figura 11 mostra um exemplo de tutoria, considerando um mundo onde existem apenas os seguintes objetos: um círculo a, um quadrado b e um triângulo c. A fórmula do aluno é  $\exists x[circulo(x) \wedge \forall y[\neg circulo(y) \rightarrow quadrado(y)]]$  (i.e., “existe um círculo e tudo que não é círculo é quadrado”), e sua intuição é que ela é verdadeira.

Você acha que  $\exists x[circulo(x) \wedge \forall y[\neg(circulo(y)) \rightarrow triângulo(y)]]$  é verdadeira. Por favor, informe um objeto que torna essa fórmula verdadeira: [a-z] a

Você acha que:

(1) círculo(a) e

(2)  $\forall x[\neg(circulo(x)) \rightarrow triângulo(x)]$  são verdadeira? [s/n] s

Considere que:

(1)  $\neg(circulo(b)) \rightarrow triângulo(b)$  e

(2)  $\neg(\neg(circulo(b))) \vee triângulo(b)$  são fórmulas equivalentes.

Você acha que:

(1)  $\neg(\neg(circulo(b)))$  ou

(2) triângulo(b) é verdadeira? [s/n] s

Você está errado,  $\neg(\neg(circulo(b)))$  é verdadeira.

Você acha que  $\neg(circulo(b))$  é falsa.

Você acha que círculo(b) é verdadeira? [s/n] s

Você está errado, círculo(b) é falsa.

Portanto,  $\neg(circulo(b))$  é verdadeira.

Portanto,  $\neg(\neg(circulo(b)))$  é falsa.

Você acha que triângulo(b) é verdadeira? [s/n] s

Você está errado, triângulo(b) é falsa.

Portanto,  $\neg(\neg(circulo(b))) \vee triângulo(b)$  também é falsa.

Portanto,  $\neg(circulo(b)) \rightarrow triângulo(b)$  também é falsa.

Assim,  $\forall x[\neg(circulo(x)) \rightarrow triângulo(x)]$  é falsa para  $x=b$ .

Portanto,  $\forall x[\neg(circulo(x)) \rightarrow triângulo(x)]$  é falsa.

Portanto,  $\exists x[circulo(x) \wedge \forall y[\neg(circulo(y)) \rightarrow triângulo(y)]]$  também é falsa.

De fato, não existe objeto que torne  $\exists x[circulo(x) \wedge \forall y[\neg(circulo(y)) \rightarrow triângulo(y)]]$  verdadeira.

Portanto,  $\exists x[circulo(x) \wedge \forall y[\neg(circulo(y)) \rightarrow triângulo(y)]]$  é falsa.

Você perdeu!

Figura 11 – Explicação de semântica pelo tutor.

A lógica do sistema de tutoria já está completa; porém, ainda será desenvolvida uma interface gráfica para essa funcionalidade. Por enquanto, a interação com o usuário é feita através da janela do interpretador Prolog. A ideia é que esse sistema funcione como um jogo, no qual o usuário pode acumular pontos ao longo do seu aprendizado e ficar ainda mais motivado (em resolver desafios propostos pelo instrutor, que poderá também ser automatizado).

## 4. Experimentos Realizados com o LabLog

Nesta seção, descrevemos alguns dos experimentos realizados para testar as funcionalidades do protótipo do sistema desenvolvido e analisamos os resultados obtidos.

### 4.1. Experimento I: Veracidade por Vácuo

Este primeiro experimento consiste em perguntar ao aluno se a fórmula  $\forall x[circulo(x) \rightarrow azul(x)]$  (i.e., “todo círculo é azul”) é verdadeira, no mundo da Figura 12. Em geral, o aluno tem a intuição de que essa fórmula é falsa; porém, ao obter o feedback do sistema, ele vê que ela é verdadeira. Depois, pede-se que ele avalie a fórmula  $\exists x[circulo(x) \wedge \neg azul(x)]$  (i.e., “existe um círculo que não é azul”). Nesse caso, a intuição do aluno corresponde ao feedback do sistema, que indica que a fórmula é falsa (Figura 13), pois isso pode ser facilmente constatado no mundo considerado. Finalmente, para que o aluno entenda porque a primeira fórmula é verdadeira (o que, em geral, não é evidente para ele), basta mostrar que a segunda fórmula é a negação da primeira. Logo, se a segunda fórmula é falsa, a primeira tem que ser verdadeira.

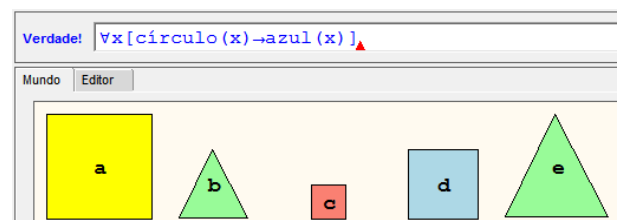


Figura 12 – Veracidade por vácuo.



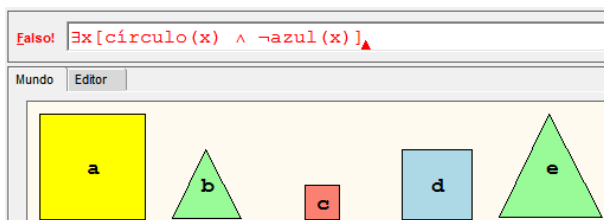


Figura 13 – Negação da fórmula na Figura 12.

## 4.2. Experimento II: Ordem de Quantificação

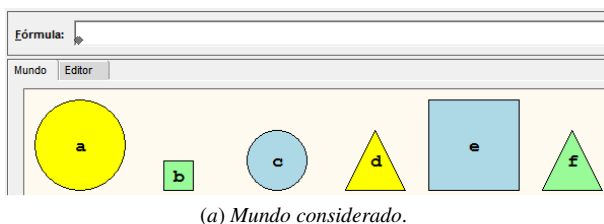
O segundo experimento visa ressaltar a importância da correta ordenação de quantificadores universais e existenciais, para que uma fórmula lógica corresponda, de fato, a uma determinada sentença em português.

Supondo que a fórmula  $\text{ama}(x, y)$  codifica a sentença “ $x$  ama  $y$ ”, nota-se que as fórmulas (i)  $\forall x \exists y . \text{ama}(x, y)$  e (ii)  $\forall y \exists x . \text{ama}(x, y)$  não são equivalentes: a fórmula (i) estabelece que “*todos amam alguém*” (o que é verdade nas Figuras 14-a e 14-b), enquanto (ii) estabelece que “*alguém é amado por todos*” (o que é verdade só na Figura 14-b).

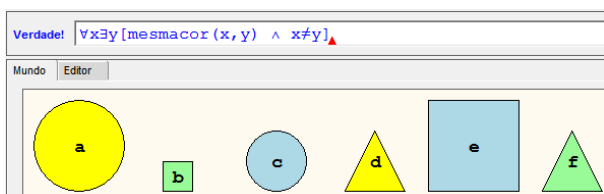


Figura 14 – Dois mundos possíveis.

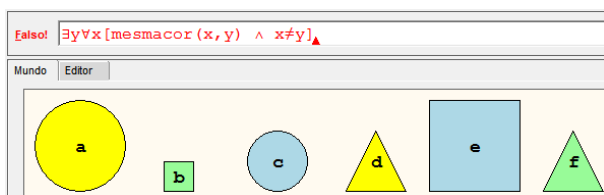
O experimento consiste em perguntar ao aluno se ele acha que as fórmulas  $\forall x \exists y [\text{mesmacor}(x, y) \wedge x \neq y]$  e  $\exists y \forall x [\text{mesmacor}(x, y) \wedge x \neq y]$  são ambas verdadeiras no mundo da Figura 15-a e, em seguida, pedir que ele avalie essas fórmulas no sistema. Caso a intuição do aluno sobre os valores dessas fórmulas não corresponda às respostas do sistema (Figuras 15-b e 15-c), ele deve ser encorajado a acessar o sistema de tutoria para esclarecer suas dúvidas.



(a) Mundo considerado.



(b) Valor da fórmula  $\exists x \exists y [\text{mesmacor}(x, y) \wedge x \neq y]$ .



(c) Valor da fórmula  $\exists y \forall x [\text{mesmacor}(x, y) \wedge x \neq y]$ .

Figura 15 – Importância da ordem dos quantificadores.

## 4.3. Experimento III: Inferências Clássicas

Este último experimento visa mostrar o uso das regras de inferência clássicas em provas por dedução.

A primeira regra de inferência clássica, denominada *modus ponens*, é definida pelo esquema  $MP(\alpha, \alpha \rightarrow \beta) \equiv \beta$ . Por exemplo, a partir das sentenças “*todo triângulo é pequeno*” e “*o objeto ‘a’ é um triângulo*”, o MP deriva “*o objeto ‘a’ é pequeno*”. Para testar essa regra, o aluno deve usar o editor de textos do sistema para criar um conjunto de fórmulas correspondente e avaliá-las. Note que essa regra de inferência é *válida* e, portanto, a sua conclusão é verdadeira sempre que suas premissas são verdadeiras. A Figura 16 mostra o conjunto de fórmulas e a Figura 17 mostra a avaliação dessas fórmulas em quatro mundos possíveis.

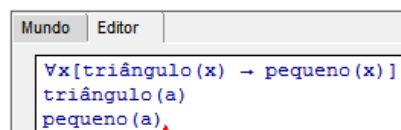


Figura 16 – Fórmulas para uso de *modus ponens*.

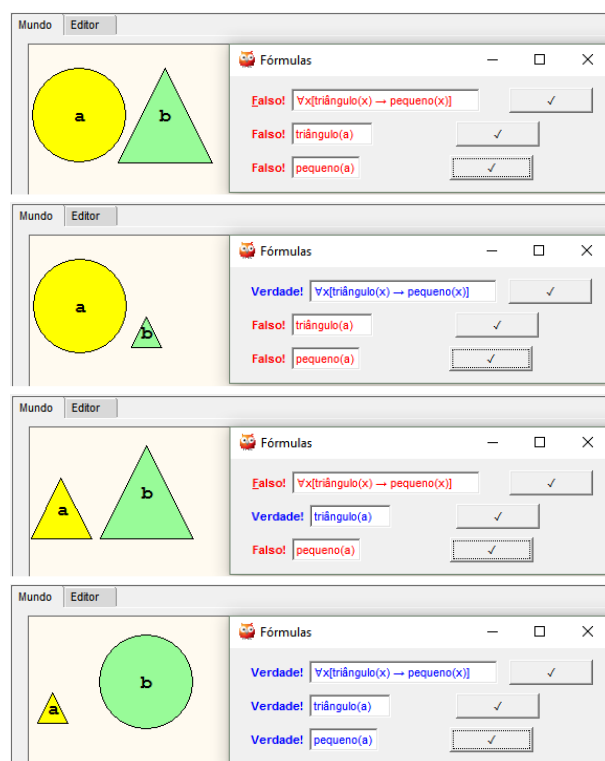


Figura 17 – Inferência com *modus ponens*.

A segunda regra de inferência clássica, denominada *modus tollens*, é definida como  $MT(\alpha \rightarrow \beta, \neg \beta) \equiv \neg \alpha$ . Por exemplo, a partir das sentenças “*todo triângulo é pequeno*” e “*o objeto ‘a’ não é pequeno*”, o MT deriva “*o objeto ‘a’ não é triângulo*”. Assim como o *modus ponens*, o *modus tollens* é uma regra de inferência válida. A Figura 18 mostra o conjunto de fórmulas e a Figura 19 mostra a avaliação dessas fórmulas em quatro mundos possíveis.

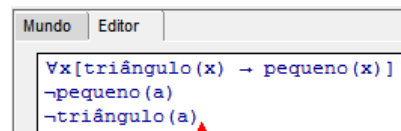


Figura 18 – Fórmulas para uso de *modus tollens*.

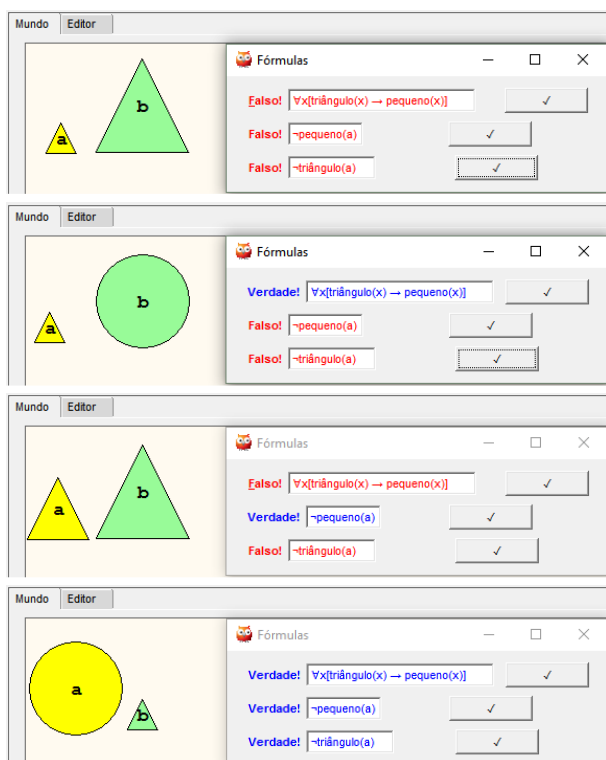


Figura 19 – Inferência com *modus tollens*.

A terceira regra de inferência clássica, *silogismo hipotético*, é definida por  $SH(\alpha \rightarrow \beta, \beta \rightarrow \gamma) \equiv \alpha \rightarrow \gamma$ . Por exemplo, a partir das sentenças “todo triângulo é pequeno” e “tudo que é pequeno é verde”, o *SH* deriva “todo triângulo é verde”. Assim como as duas regras de inferências anteriores, o *silogismo hipotético* é uma regra de inferência válida e os resultados dos experimentos realizados com ela são similares àqueles já apresentados.

## 5. Conclusões

Com base nos fundamentos de aprendizagem ativa, laboratório virtual e teoria cognitiva de aprendizado multimídia, este artigo propõe um laboratório virtual de lógica simbólica, desenvolvido em Prolog, descreve detalhes de sua implementação, discute suas principais funcionalidades e relata resultados de experimentos realizados.

Os resultados dos experimentos mostraram que o sistema funciona de acordo com suas especificações e produz resultados que são coerentes com a teoria de lógica de predicados. Os testes com o módulo de tutoria, baseado em técnicas de prova automática e correção de erros, também produziram resultados corretos.

O laboratório virtual criado é um ambiente de simulação computacional interativo, que pode estimular o aluno a se envolver ativamente no seu processo de aprendizagem. Os experimentos mostraram que o protótipo permite a criação de mundos virtuais e a execução de experimentos que não podem ser facilmente feitos no mundo real. Ademais, ao guiar o aluno durante os experimentos, por meio de *feedbacks* e explicação de erros, ele o incentiva a adotar uma estratégia de investigação baseada em tentativa e erro, que exige uma postura mais proativa do aluno.

Como continuidade desse trabalho será desenvolvida uma interface gráfica para o sistema de tutoria; novos mundos e conjuntos de fórmulas serão projetados para

realização de novos experimentos que explorem conceitos de lógica que são difíceis para os alunos entenderem; as funcionalidades existentes serão aprimoradas e possíveis extensões do trabalho serão investigadas (como, por exemplo, experimentos didáticos com alunos do curso de Análise e Desenvolvimento de Sistemas da Fatec-SP).

## Agradecimentos

Ao CNPq pela bolsa de Iniciação Científica<sup>1</sup> (Processo Nº 800321/2016-2).

## Referências Bibliográficas

- [1] G. Bezhanishvili; W. Fussner. **An Introduction to Symbolic Logic**. Disponível para download em: [www.cs.nmsu.edu/historical-projects/Projects/symbolic\\_logic8.pdf](http://www.cs.nmsu.edu/historical-projects/Projects/symbolic_logic8.pdf), acessado em 11/05/16.
- [2] J. R. Anderson. **Cognitive Psychology and Its Implications**. 8th ed., Worth Publishers, p. 239-241, 2015.
- [3] M. A. Brändle. **GraphBench: Exploring the Limits of Complexity with Educational Software**. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, 2006.
- [4] I. Bratko. **Prolog Programming for Artificial Intelligence**, 4<sup>th</sup> ed., Pearson, 2011.
- [5] M. Prince. **Does active learning work? A review of the research**. Journal of Engineering Education, v. 93(3), p. 223-231, John Wiley & Sons, Inc., 2004.
- [6] P. Laws; D. Sokoloff; R. Thornton. **Promoting Active Learning Using the Results of Physics Education Research**. UniServe Science News, v. 13, 1999.
- [7] S. Freeman et al. **Active learning increases student performance in science, engineering, and mathematics**. Proceedings of the National Academy of Sciences, v. 111(23), p. 8410-8415, 2014.
- [8] E. Redish; J. Saul; R. Steinberg. **On the Effectiveness of Active-Engagement Microcomputer-Based Laboratories**. American Journal of Physics, v. 65(1), p. 45-54, 1997.
- [9] L. Mercer; P. Prusinkiewicz; J. Hanan. **The Concept and Design of a Virtual Laboratory**. In: Proc. of Graphics Interface, Halifax, Nova Scotia, 14-18, 1990.
- [10] I. Hawkins. **Virtual laboratory vs. traditional laboratory: which is more effective for teaching electrochemistry?**. Chem. Educ. Res. Pract., v. 14, p. 516-523, 2013.
- [11] M. Darrah et al. **Are Virtual Labs as Effective as Hands-on Labs for Undergraduate Physics? A Comparative Study at Two Major Universities**. J. Sci. Educ. Technol., v. 23(6), p. 803-814, 2014.
- [12] S. D. Sorden. **The Cognitive Theory of Multimedia Learning**. Mohave Community College, Northern Arizona University, 2012.
- [13] R. E. Mayer. **Cognitive Theory of Multimedia Learning**. The Cambridge handbook of multimedia learning, 43-71, 2014.
- [14] D. Barker-Plummer; J. Barwise; J. Etchemendy. **Tarski's World: Revised and Expanded**. University of Chicago Press, 2004.
- [15] J. Wielemaker; A. Anjewierden. **Programming in XPE/Prolog**. Disponível para download em: <http://info.ee.pw.edu.pl/Prolog/Download/userguide.pdf>, acessado em 20/08/16.