A implementação colaborativa de uma suíte de ferramentas on-line de apoio ao ensino de Lógica

Thiago M. Barros, Alexandre E. F. De Araújo
João Marcos (orientador)
medeiros@engcomp.ufrn.br, topo@lcc.ufrn.br, jmarcos@dimap.ufrn.br

Resumo: *O* **Logicamente** *(cf. [1]) é uma suíte de programas implementados por alunos da disciplina de Lógica Aplicada à Computação da UFRN e disponibilizada on-line, com o objetivo de auxiliar o aprendizado dos alunos e interessados na área de Lógica. Parcialmente financiado pela Pró-Reitoria de Graduação como um Projeto de Apoio à Melhoria da Qualidade do Ensino, o principal objetivo é criar uma suíte de programas que coloque em prática os conceitos teóricos estudados em sala de aula tanto no que diz respeito à sintaxe da Lógica Clássica quanto à sua semântica e à aplicação do método computacional de resolução. O grande diferencial deste projeto é ter, como diretriz principal, a participação dos alunos em seu desenvolvimento, fazendo com que os alunos envolvidos estimulem sua curiosidade e buscando proporcionar assim outra abordagem ao estudo da Lógica, complementar àquela que foi apresentada em sala de aula. Pretende-se, para o futuro, transformar o* **Logicamente** *em um Ambiente Virtual de Aprendizagem, onde o aluno possa realmente praticar e aprender de forma dirigida, para além da sala de aula, os assuntos relacionados a Lógica.*

Palavras-chave: lógica, ambiente virtual de apoio ao ensino, interatividade.

Introdução

Tornar o conteúdo da disciplina de Lógica mais fácil de assimilar pelos alunos é sem dúvida um desafio para a maioria dos instrutores que se dispõem a tentar ministrar o conteúdo da mesma. Mais difícil ainda é a tarefa de tornar o conteúdo tão interessante quanto possível de ser estudado sem a clássica presença do professor em sala de aula escrevendo na lousa.

Pensando nisso, a suíte de ferramentas **Logicamente** foi projetada como um conjunto de módulos integrados a fim de que o aprendizado do conteúdo do curso possa ser ministrado com outra perspectiva daquela apresentada em sala de aula (geralmente realizada apenas de uma forma expositiva). Disponível on-line, o aluno do curso de Lógica pode utilizar o sistema de forma interativa, atacando diversos dos problemas propostos em sala de aula, como por exemplo:

- na geração de fórmulas aleatórias dentro de especificações pré-definidas,
- no uso de tabelas de verdade,
- no uso do método de resolução,
- na visualização de forma mais amigável da árvore de sintaxe abstrata correspondente a uma expressão bem formada da linguagem da Lógica,

bem como muitas outras comodidades que estudantes e até mesmo instrutores podem utilizar para facilitar o aprendizado dos principais conceitos de Lógica em um ambiente virtual. Note-se em especial que todo esse aparato auxiliar ao ensino da disciplina está acessível a qualquer aluno que possua apenas o acesso a internet e um browser.

É importante ressaltar que esse projeto só pôde ser desenvolvido devido à característica singular da disciplina de que boa parte dos conceitos apresentados em sala de aula pode ser representada algoritmicamente. Assim, a possibilidade de implementá-los e utilizá-los como material didático torna-se um importante recurso de apoio ao aprendizado do conteúdo do curso. Além disso, o processo de implementação por parte dos alunos envolvidos no projeto faz uso prático de conceitos aprendidos em diversas outras disciplinas usualmente ministradas nos cursos de computação, como por exemplo a disciplina de Estrutura de Dados.

Características técnicas

Em sua primeira versão o **Logicamente** foi desenvolvido na linguagem de programação JAVA, utilizando applets. Entretanto, hoje o **Logicamente** é desenvolvido na linguagem de programação PHP5 em sistema operacional Linux e no servidor Apache2.0. A escolha da linguagem PHP5 foi feita devido a esta ser tida como uma linguagem de programação de alto-nível e de fácil aprendizagem, e que pode ser utilizada na web, permitindo, assim, que todo o sistema implementado seja acessado por qualquer um que possua acesso à internet. Todo esse ambiente também prefigura a possibilidade do uso de tecnologias consolidadas para web tais como HTML, JavaScript e CSS. Desde o início houve a preocupação de que esse trabalho de discussão e implementação das especificações pré-definidas pelo professor orientador fosse

compartilhado por todos os envolvidos —no caso, estudantes de graduação que se interessaram pelo projeto—, e que a suíte de módulos tivesse também a função de auxiliar no aprendizado por parte dos outros alunos —donde o requisito com relação à *interatividade* do sistema com o usuário (o aluno ou acadêmico interessado em Lógica) se torna tão importante quanto o programa estar correto. Assim sendo, foram utilizados algumas tecnologias web, tal como o AJAX, a fim de possibilitar o máximo de interação do usuário com o sistema.

Justificativa

A habilidade de raciocínio lógico é fundamental aos aos mais diversos perfis profissionais, e até mesmo ao exercício pleno da cidadania Diversos cursos superiores, com efeito, levam esta necessidade em consideração, incluindo em seu currículo básico disciplinas relacionadas à Lógica. Por exemplo: em **Filosofia** a Lógica está na base do estudo da argumentação, convencional, crítica e retórica, em **Matemática** a Lógica está na base dos seus fundamentos sintáticos e semânticos, em **Direito** a Lógica fundamenta a própria ordenação jurídica, a qual, no nosso sistema românico, é moldada segundo o formato axiomático euclidiano.

Em **Computação**, vale ressaltar, a Lógica é tão fundamental que podemos dizer que ela cumpre o mesmo papel da Análise na Física ou do Cálculo nas Engenharias. Com efeito, enquanto a própria *Arquitetura de Computadores* projeta estas máquinas pela combinação de portas lógicas, na *Engenharia de Software* a Lógica tem um papel fundamental na especificação e verificação de projetos e componentes, na área de *Linguagens de Programação* a Lógica fundamenta as abordagens declarativas e cuida em geral da semântica das linguagens artificiais para diálogo com a máquina, na área de *Bancos de Dados* a Lógica está por trás das abordagens via álgebra relacional e SQL, na área de *Inteligência Artificial* a Lógica contribui para a representação do conhecimento, o aprendizado de máquina e a demonstração automática de teoremas, na área de *Algoritmos* a Lógica se inscreve em particular no estudo das questões de complexidade e expressividade, e na área de *Teoria da Computação* a Lógica se preocupa com as noções gerais de computabilidade. É imediato concluir assim que os currículos dos cursos de computação precisam prover aos estudantes uma base sólida nas habilidades necessárias para planejar e desenvolver artefatos computacionais robustos e seguros.

Dito isto, é forçoso reconhecer, contudo, que os currículos convencionais privilegiam os alunos de estilos de aprendizado intuitivo, verbal, dedutivo, refletivo e sequencial, em oposição àqueles alunos, muitas vezes a maioria, que aprendem de outras formas (cf. [2]). Somando isto à má consolidação da formação básica sólida dos estudantes que entram hoje nas nossas universidade e aos recursos precários de aprendizagem não-convencional presentemente disponíveis em nossas instalações, não surpreende tanto assim observar que o índice de trancamentos e reprovações dos alunos matriculados no curso de *Lógica Aplicada à Computação* mantenha-se consistentemente elevado.

Com o objetivo de colaborar na reversão desta cena, agregamos a partir de novembro de 2006 uma equipe de estudantes interessados para dar início ao desenvolvimento da suíte **Logicamente**. O objetivo foi investir, assim, em um estilo de aprendizagem mais sensorial, visual, indutivo, ativo e global, visando formar profissionais mais aptos a atender de forma eficiente as necessidades reais da comunidade.

Desenvolvimento

Partindo da idéia de que as expressões da linguagem da Lógica podem ser representadas como árvores, os módulos desenvolvidos possuem essa estrutura de dados como a principal. O trabalho com árvores facilitou o percurso das fórmulas e o casamento de padrões nas mesmas, permitiu escrevê-las sem ambigüidade e orientou a definição de procedimentos recursivos. Com relação a esta última, a razão de utilizá-la está na própria natureza dos assuntos envolvidos na Lógica.

Figura l: Tela inicial do **Logicamente**

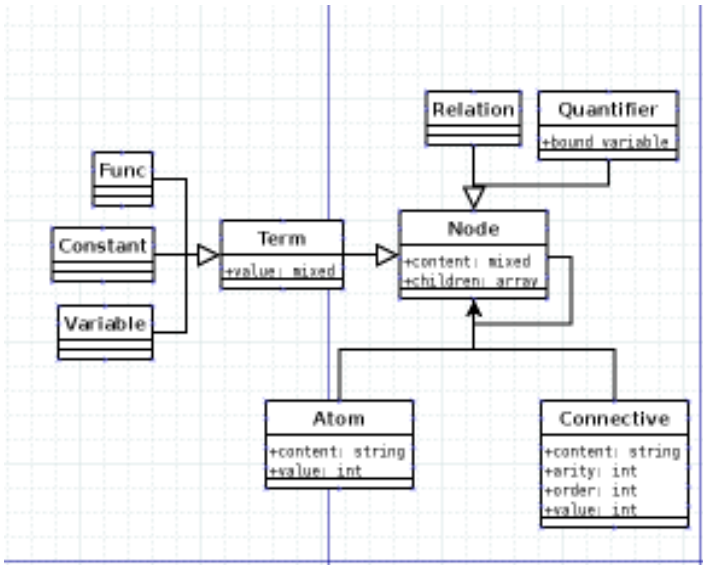

Figura 2: Estrutura interna das fórmulas

A árvore utilizada no projeto do **Logicamente** se configura segundo a estrutura que se exibe na Figura 2.

Ajuda

Já na página inicial da suíte **Logicamente** disponibiliza-se ao usuário a opção de ajuda, bastando apertar as teclas **Ctrl+Shift+h.** A ajuda desta página inicial contém diversos atalhos úteis para tornar mais ágil o acesso aos diversos módulos do programa.

Área de transferência

A área de transferência no projeto serve para integrar todos os módulos, de maneira que um dado inserido em um módulo possa ser utilizado em outro. Esta funciona segundo o princípio de que todos os módulos trabalham sobre as árvores descritas na seção anterior.

Uma vez que as árvores são implementadas como estruturas de dados no PHP, podemos usar sobre elas técnicas como *serialização* e *armazenamento em sessões* (cf. [3] e [4]), de modo a armazenar as modificações que forem sendo feitas sobre estas estruturas ao longo dos diferentes módulos do sistema.

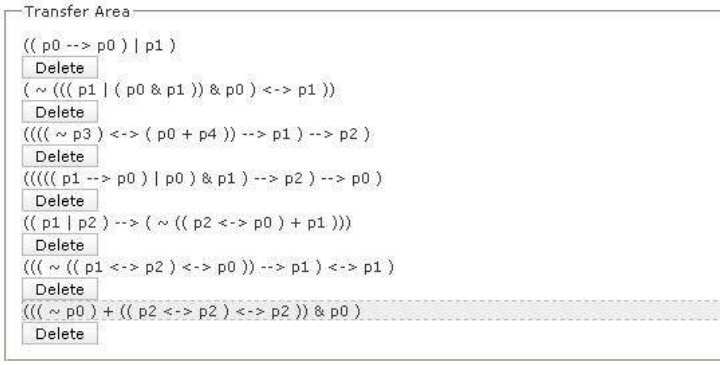

Figura 4: Área de Transferência

Figura 3: Ajuda

Figura 5: Aba Settings

Aba Settings

Essa aba não representa um módulo propriamente dito, mas sim uma opção ao usuário de poder manipular algumas configurações do **Logicamente**. As opções de configuração são:

- **Fórmula Style**: onde o usuário define qual o formato com que irá trabalhar com as fórmulas no sistema. As opções são:
  - **Infix**: Formato geralmente mais usual, no qual os conectivos binários são inseridos entre os átomos (ex: a **&** b).
  - **Polish**: Notação polonesa, na qual os conectivos são representados de forma infixa (ex: **K** a b).
  - **Functional**: Formato funcional, no qual os conectivos são apresentados em uma notação mais usual no curso de circuitos (ex: **xor**(a,b)).

Deve ser observado que nem todos os módulos aceitam ainda o formato Polish e Functional, sendo o uso do formato Infix fortemente aconselhado. Para o futuro este módulo deverá dar suporte para a definição de novos símbolos pelo usuário.

Aba Generator

É o módulo responsável por gerar fórmulas proposicionais bem formadas (fbf's) arbitrárias de acordo com as especificações do usuário (número de ocorrências de conectivos e seus tipos, número de variáveis sentenciais). O retorno do gerador será uma fórmula bem formada, e será transferida para a área de

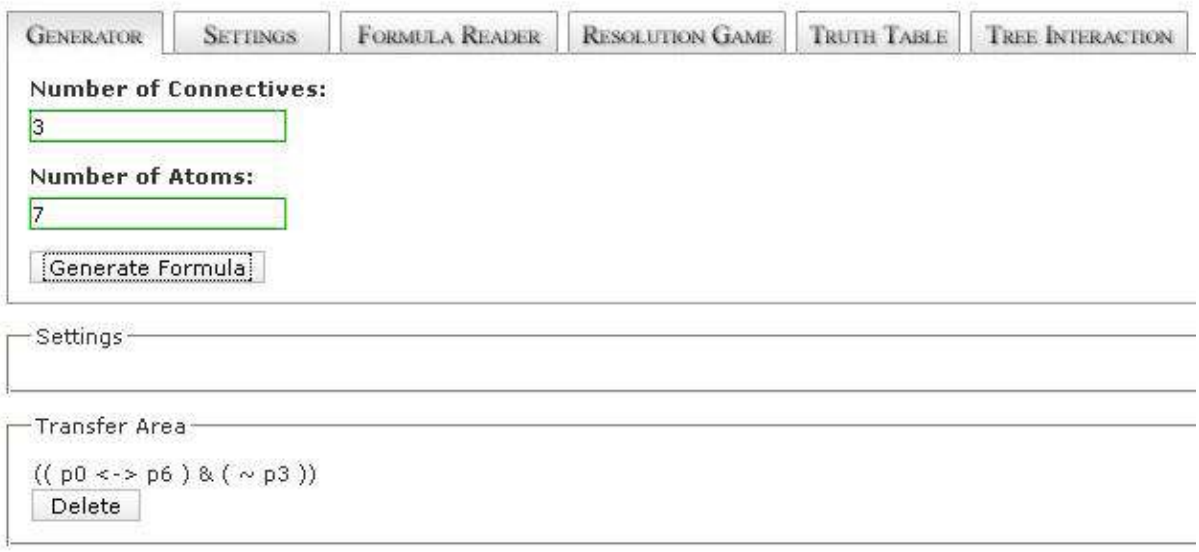

Figura 6: Aba Generator

transferência, a fim de que o usuário possa manipulá-la em outros módulos ou excluí-la, caso não tenha deseje utilizá-la. Input: **array** com número de *Connectives* (que é introduzido pelo usuário na aba *Settings*)**,** **integer** com número de átomos, **integer** com número de ocorrência de conectivos. Output**: fórmula** em formato de árvore.

Figura 7: Aba Formula Reader

Aba Formula Reader

Para usar algumas funcionalidades do **Logicamente** é necessário passar uma expressão para a área de transferência ("Transfer Area"). Para este fim foi implementado o módulo Formula Reader. Para utilizá-lo é preciso primeiro preencher o formulário com a expressão desejada, que será copiada para a área de transferência. As expressões podem ou não ser parentetizadas, os átomos devem ser constituídos de uma sequência de letras, que podem ser seguidas de uma sequência de números. Para adicionar a expressão à área de transferência, basta clicar no botão "*Read Formula*". Exemplos de fórmulas:

- a & (b | c)
- p3 <-> (p4 & ~p5)
- chove | ~chove

Para utilizar a fórmula inserida no Formula Reader em outro módulo, basta clicar na expressão, já na área de transferência. Caso não se deseje mais utilizar uma certa expressão, a mesma pode ser apagada da área de transferência apertando no botão "*Delete*".

Aba Resolution Game

O módulo Resolution Game implementa o método computacional de eliminação de literais complementares. O aluno deve entrar com as cláusulas já na Forma Normal Clausal, para adicionar mais cláusulas, basta clicar em "*Add more clauses"*, e caso queira remover a última cláusula, basta clicar em "*Remove clause"*, depois de inseridas as cláusulas, o usuário deve clicar no botão "*Start Game"* para iniciar o jogo.

No jogo tem-se duas opções, você pode pedir para o programa estabelecer a derivação automaticamente, apertando no botão "*Solve automatically*", ou clicar no botão "*Do selected elimination*", para selecionar as cláusulas sobre as quais se pretende utilizar a regra da eliminação dos literais complementares até chegar a uma cláusula vazia, que representa um absurdo, ou, caso restem cláusulas não-vazias, pode-se usá-las para produzir contra-modelos para a fórmula inicial.

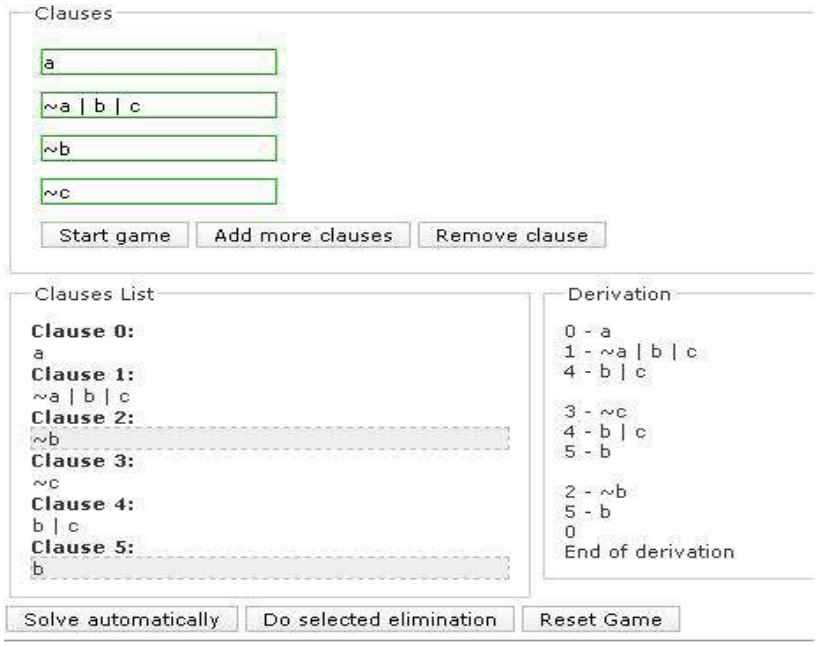

Figura 8: Resolution Game

Aba Truth Table

Este módulo permite o acompanhamento de todos os estados possíveis das variáveis sentenciais na lógica proposicional e consequentemente da fórmula como um todo. Para o futuro deverá dar suporte para a

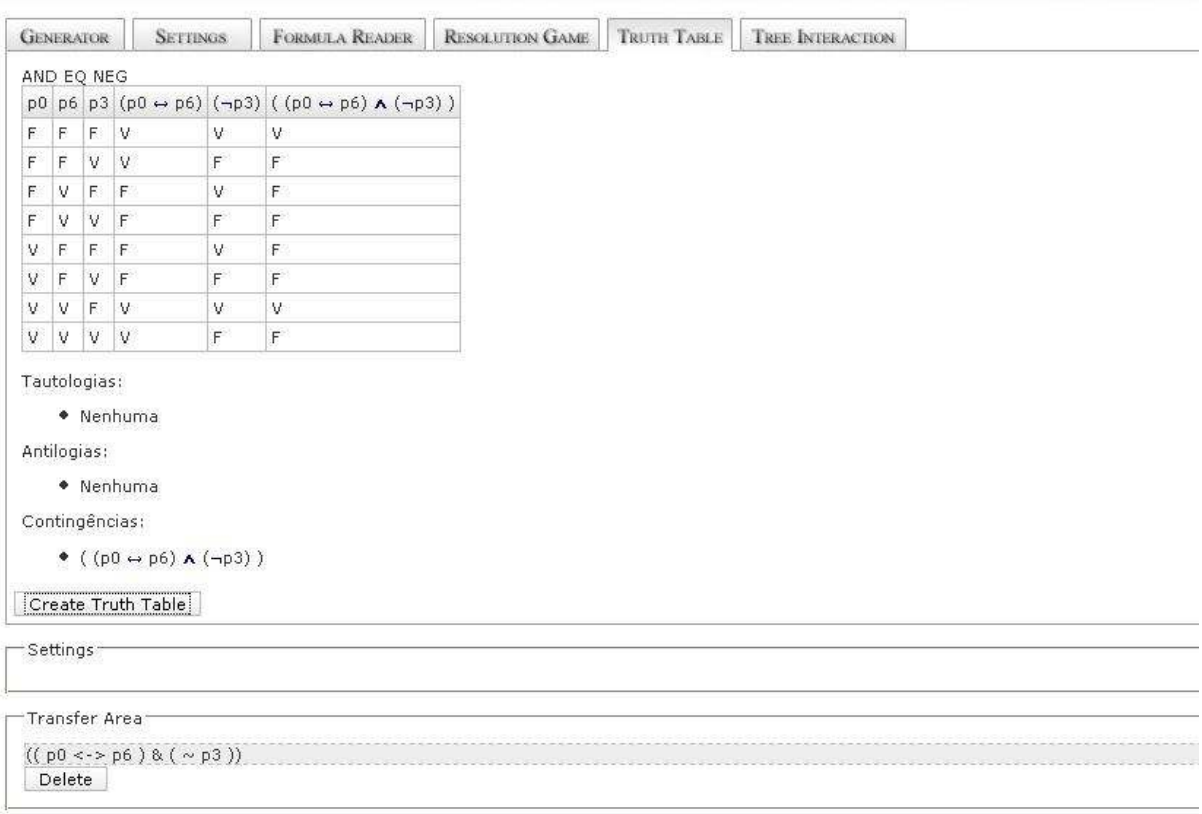

Figura 9: Aba Truth Table

definição de novos operadores pelo usuário. O usuário entrará com uma string com a fórmula, o programa transformará em árvore (internamente) e começará a atribuição de valorações para os átomos a fim de verificar se a fórmula é verdadeira ou não. Caso o usuário queira uma valoração específica, ele deverá entrar com os valores respectivos dos átomos da fórmula, a fim de que o programa só verifique as valorações dadas. Assim, a tabela de verdade consiste numa função que verifica recursivamente a satisfação de um conjunto de valorações para átomos, e uma função que gera essas valorações.

Input: string com a fórmula desejada, **optional** os respectivos valores **0, 1** de uma determinada valoração para um conjunto de átomos

Output: array de boolean

Aba Tree Interaction

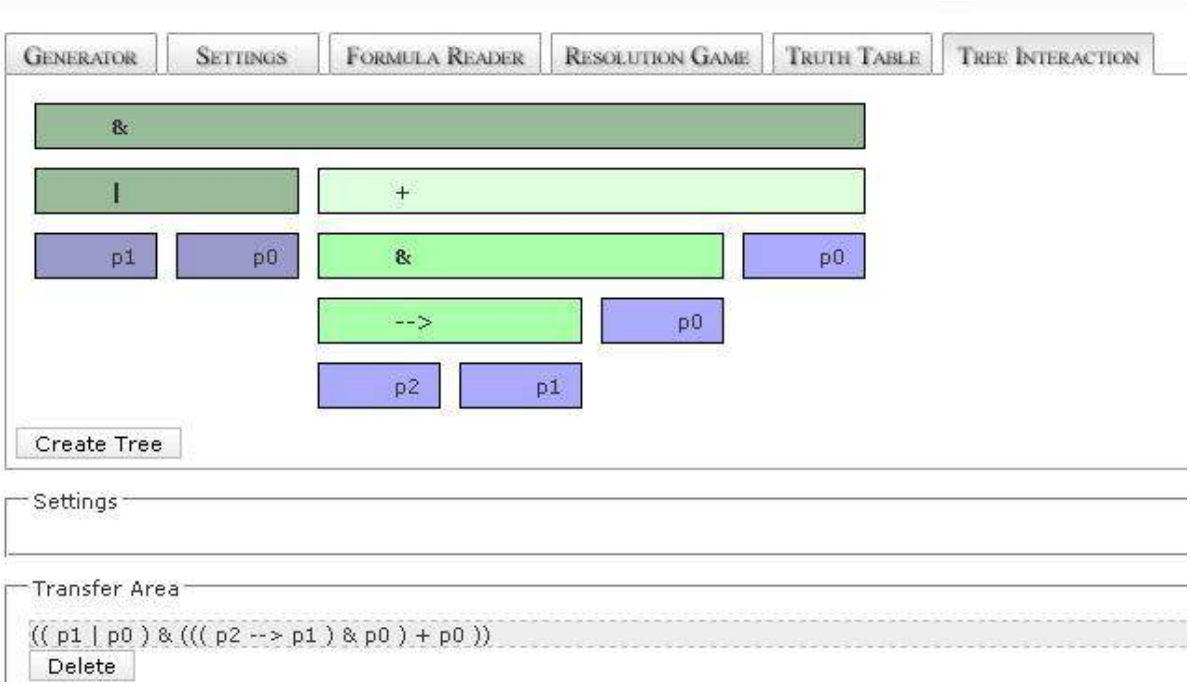

Figura 10: Aba Tree Interaction

Este módulo permite ao usuário visualizar as fórmulas em um formato mais amigável, proporcionando ao usuário uma melhor visualização das fórmulas em formato de árvores. Para tanto, basta selecionar uma fórmula que se encontra na área de transferência e então clicar no botão "*Create Tree*". Caso selecione alguma folha da árvore, será destacada a cor para todas as folhas que tenham o mesmo conteúdo. Caso selecione um nó interno, será destacado toda a sub-árvore do nó. Em um próximo estágio, ao selecionar um quantificador, será possível visualizar todo o escopo do mesmo.

Pretensões Futuras

A principal meta a ser atingida pela suíte de ferramentas **Logicamente** é que esta se torne um *Ambiente Virtual de Aprendizado*, a fim de que estudantes do mundo inteiro tenham informações disponíveis sobre diversos tópicos de Lógica e pratiquem através de exercícios e jogos interativos o conteúdo ministrado em sala de aula. Todavia, para que esse objetivo se torne realidade, outras metas ainda necessitam ser concluídas, como a implementação de outros módulos.

Em curto prazo, pretende-se integrar os módulos já implementados. Um deles é o *Small Counter-Model Builder* (SCMBuilder) que implementa um buscador exaustivo de contra-modelos paa a lógica clássica de primeira ordem, com um domínio de quantificação com cardinalidade limitada *a priori* pelo usuário. Este módulo receberá uma coleção de fórmulas de primeira ordem como premissas e uma fórmula como conclusão. Outros módulos não integrados são S*ubstitution Master,* que recebe uma fórmula, uma variável que se pretende substituir sintaticamente por um certo termo nesta fórmula, e o termo em questão. Após verificar se esta substituição proposta pode realmente ser efetuada, sem que ocorra a captura de variáveis do termo que se pretende colocar no lugar da variável escolhida na fórmula, o programa efetuará a substituição caso a verificação dê um resultado

positivo; se negativo o resultado, será enviado ao usuário um aviso de que o a variável não pode ser substituída pelo termo na fórmula. e proposta uma renomeação das variáveis ligadas de forma a permitir que a substituição seja efetuada. Deve-se corrigir e integrar ainda, entre outras coisas, o módulo *Drawing Tableaux*, o qual implementa o método de tableaux proposicionais de forma automática e interativa; o módulo *CNF Converter*, o qual consiste em o usuário a cada passo escolher uma regra de uma lista de regras de conversão permitidas qual modificação na árvore deve ser feita, a fim de se obter a fórmula normal conjuntiva; E o módulo de conseqüência semântica proposicional, o qual verifica que uma dada fórmula segue de N fórmulas também arbitrárias pelo usuário. Outra preocupação para o futuro são modificações de módulos que trabalham apenas com fórmulas proposicionais para aceitar fórmulas de primeira.

Agradecimentos

Agradecemos a todos os alunos de graduação dos cursos de Ciência da Computação e Engenharia da Computação da UFRN, que participaram desse projeto direta ou indiretamente.

Referências

[l] **Logicamente**: http://www.dimap.ufrn.br/logicamente

[2] I. Barland, M. Felleisen, K. Fisler, M. Y. Vardi, P. Kolaitis, "Integrating Logic into the Computer Science Curriculum", manifesto apresentado no evento anual *Innovation and Technology in Computer Science Education*, Helsinki, 2000.

[3] *Serialização:* http://br2.php.net/manual/pt_BR/function.serialize.php

[4] *Armazenamento em sessões* :(http://br2.php.net/manual/pt_BR/ref.session.php