

Sistema de Suporte Pedagógico para Resolução de Problemas de Construção de Tabelas-Verdade na Lógica Proposicional

Alessandro Fontoura Alves

João Carlos Gluz

Programa de Pós-Graduação em Computação Aplicada (PIPCA) – Unisinos – São Leopoldo – RS – Brazil

bauvesf@gmail.com, jcgluz@unisinos.br

***Abstract.** Logical reasoning is one of the fundamental tools for humans, it is important to structure thinking for problem solving. At the university, students who are entering courses in Computer Science and Information Systems need the discipline of Logic to develop their logical analysis and problem solving skills. The EVTIV tool proposed in this work helps students to solve truth tables problems. This tool offers a friendly interface for the resolution of this kind of problems and exercises that can be integrated with an intelligent tutoring system of propositional logic. This work shows the technical characteristics and features of the tool, and its differentiation from other propositional logic learning tools. The work also present the results of experiments and tests conducted with the tool.*

***Resumo.** O raciocínio lógico é uma das ferramentas fundamentais para os seres humanos, pois é fundamental para estruturação do pensamento para resolução de problemas. Na universidade, os alunos que estão ingressando nos cursos de Ciências da Computação e Sistemas de Informação precisam da disciplina de Lógica para aprimorar suas habilidades na análise lógica e na resolução de problemas. A ferramenta de apoio à resolução de problemas de tabelas-verdade da Lógica Proposicional proposta neste trabalho oferece uma interface amigável para resolução deste tipo de problema que pode ser integrada em um sistema inteligente de tutoria de Lógica Proposicional. O presente trabalho mostra as características técnicas da ferramenta e suas funcionalidades, além de sua diferenciação com outras ferramentas de aprendizagem de Lógica Proposicional. Também são apresentados os resultados dos experimentos e testes conduzidos com a ferramenta.*

1. Introdução

A disciplina de Lógica está presente nos principais cursos de informática das faculdades é a responsável pela primeira etapa do aluno que está ingressando no mundo acadêmico á compreender e desenvolver o raciocínio lógico, que será fundamental para as disciplinas de programação.

A Lógica Proposicional é um dos conteúdos mais difíceis da disciplina de Lógica, pois dados apontam altos índices de reprovação e desistência desta disciplina [Penteado e Gluz, 2011]. Com base nestas premissas, propõe-se neste trabalho a

ferramenta (EVTV – Editor-Verificador de Tabela-Verdade) projetada para auxiliar os alunos nos tópicos iniciais desta disciplina, em particular na resolução de problemas de construção de tabelas-verdade para fórmulas da Lógica Proposicional. Espera-se que através de uma interface amigável, que motiva a compreensão dos conceitos envolvidos, os acadêmicos possam relacionar teoria e prática, na disciplina de Lógica. A ferramenta foi projetada para fornecer um apoio para aprendizagem que possa ajudar os alunos a desenvolver suas capacidades de avaliações de expressões de lógicas e construção da tabela-verdade, utilizando suas funcionalidades específicas como diferencial em relação a outras ferramentas de apoio ao ensino de Lógica Proposicional.

Este trabalho demonstra o resultado do processo de desenvolvimento da ferramenta EVTIV. Também mostra como essa ferramenta se insere no contexto de um sistema maior de tutoria inteligente da disciplina de Lógica.

2. Referencial Teórico

Os conceitos que foram utilizados para a construção da ferramenta foram embasados no estudo da estrutura das fórmulas proposicionais, incluindo seus conectivos lógicos, no processo de construção da tabela-verdade e no método de decomposição de fórmulas proposicionais.

2.1 A Lógica Proposicional

O estudo da Lógica tem como objetivo final verificar se alguma condição ou argumento é válido ou não. A Lógica Formal ou Simbólica estuda os argumentos que estejam representados através de fórmulas. A Lógica Proposicional é uma parte da Lógica Formal que considera como unidade mínima dos argumentos, as *proposições lógicas*. Na Lógica Proposicional os argumentos são compostos por proposições que são afirmações ou sentenças que podem assumir um de dois valores lógicos (ou valores-verdade): *verdadeiro* (V) ou *falso* (F). Proposições simples ou elementares, representadas por letras (variáveis lógicas) como A, B, C,... são as unidades mínimas de formação dos argumentos. Conectivos lógicos podem ser usados para interligar uma proposição simples com outra proposição simples gerando novas proposições compostas conforme a tabela 1. Nesta tabela também é mostrado o valor lógico da proposição composta resultante, dado o valor lógico das proposições de entrada A e B.

Tabela 1 – Conectivos da Lógica Proposicional

A	B	Conjunção $A \wedge B$	Disjunção $A \vee B$	Condicional $A \rightarrow B$	Bicondicional $A \leftrightarrow B$	A	Negação $\sim A$
V	V	V	V	V	V	V	F
V	F	F	V	F	F	F	V
F	V	F	V	V	F		
F	F	F	F	V	V		

Uma *fórmula bem-formada* da Lógica Proposicional (ou apenas *fórmula* no texto a seguir) é gerada a partir das proposições simples e dos conectivos lógicos usando as seguintes regras:

- Todo símbolo de proposição simples A, B, C, ... é uma fórmula;
- Se P é uma fórmula, então $\sim P$ e (P) são fórmulas;

- Se P e Q são fórmulas, então $P \wedge Q$ é uma fórmula;
- Se P e Q são fórmulas, então $P \vee Q$ é uma fórmula;
- Se P e Q são fórmulas, então $P \rightarrow Q$ é uma fórmula;
- Se P e Q são fórmulas, então $P \leftrightarrow Q$ é uma fórmula;
- Nenhuma outra combinação de símbolos é uma fórmula.

A tabela-verdade de uma fórmula mostra em suas colunas mais à esquerda, todas as combinações de valores lógicos que as proposições simples (as variáveis) dessa fórmula podem assumir. A partir destes valores de entrada pode-se calcular os valores que esta fórmula irá ter para cada uma destas combinações de valores. Este cálculo é feito passo a passo, criando-se colunas intermediárias que ficam posicionadas à direita das colunas de entrada e que contêm os valores das sub-fórmulas que compõem a fórmula principal. Na última coluna mais à direita se coloca a coluna que contém os valores finais desta fórmula. Resumindo, para se construir a tabela-verdade de uma fórmula lógica pode-se seguir os seguintes passos:

- a) Colocar nas primeiras colunas à esquerda as variáveis da fórmula original (A, B, \dots).
- b) Caso haja variáveis negadas ($\neg A, \neg B, \dots$), colocá-las nas próximas colunas, sem repetir variáveis negadas se existir mais de uma instância na fórmula original e, por fim.
- c) Seguindo a precedência dos conectivos, criar uma coluna para cada sub-fórmula que é parte da fórmula original. Não é necessário repetir sub-fórmulas iguais.

A última coluna à direita deve conter a fórmula original.

A precedência usual dos conectivos da Lógica Proposicional é a seguinte [Ederton, 2001]:

1. Fórmulas dentro de parênteses (os mais internos primeiro)
2. \neg, \sim (negação)
3. \wedge (conjunção)
4. \vee (disjunção)
5. \rightarrow (condicional)
6. \leftrightarrow (bicondicional)
7. Da esquerda para a direita: \wedge, \vee
8. Da direita para a esquerda: $\rightarrow, \leftrightarrow$

2.2 Árvore Binárias de Decomposição

As árvores binárias podem ser utilizadas para resolução de cálculos proposicionais. A decomposição de uma fórmula proposicional inicia com a identificação do conectivo lógico de maior precedência e segue até o conectivo de menor precedência. A construção de uma árvore binária de sub-fórmulas a partir de uma fórmula dada termina quando todas as folhas contiverem somente letras proposicionais [Abe et al., 2001].

A aplicação do método para decompor as fórmulas na árvore de decomposição permite gerar as colunas de uma tabela-verdade. Os nós folha identificam as variáveis da fórmula, enquanto que os diversos nós internos isolam as sub-fórmulas. As demais colunas, começando pelo lado esquerdo da tabela, são preenchidas através de um caminharmento que começa pelos níveis inferiores do lado esquerdo da árvore de decomposição e vai até os nós inferiores do lado direito, passando por fim aos nós

superiores. O nó raiz, que é o mais alto, será colocado justamente na coluna mais a direita da tabela.

3. Trabalhos Relacionados

Na pesquisa sobre trabalhos e ferramentas de apoio ao processo de ensino da Lógica Proposicional, foram encontradas duas ferramentas que realizam o cálculo de fórmulas proposicionais e construção da tabela verdade:

- A ferramenta *Truth Table Constructor* [Borowski, 2009], é uma ferramenta desenvolvida em Java, que roda em ambiente desktop, sendo possível realizar a construção da tabela verdade completa, linha, coluna e exportar a tabela em arquivo de texto ou em formato de imagem.
- A ferramenta *Truth Table Generator* [Williams, 2010], é uma ferramenta desenvolvida em Ruby, que roda em ambiente web e também como aplicativo para *iPhone* em ambas sendo possível realizar somente a construção da tabela verdade.

Para uma melhor contextualização das contribuições trazidas pela ferramenta (EVTV) foi realizados testes funcionais, com estas duas ferramentas, comparando suas funcionalidades com a ferramenta EVTIV, conforme a Tabela 2.

Tabela 2 – Comparativos entre ferramentas

Ferramentas	EVTV	Truth Table Constructor	Truth Table Generator
Plataforma	Desktop	Desktop	Web
Linguagem	Java	Java	Ruby
Editor	Sim	Não	Não
Construção passo á passo (Tabela)	Sim	Sim	Não
Verificação de Integridade Fórmula	Sim	Sim	Não
Visualização da decomposição da Fórmula em Operações	Sim	Não	Não
Integrável com Sistema de Tutoria	Sim	Não	Não

A ferramenta *Truth Table Constructor* se mostrou mais parecida em relação á ferramenta (EVTV), porém os recursos de edição e demonstração da decomposição da fórmula não estão presentes nessa ferramenta, mostrando o diferencial importante que a ferramenta (EVTV) pode oferecer para apoiar o ensino de Lógica Proposicional.

4. Arquitetura do Protótipo

A ferramenta EVTIV foi desenvolvida para a execução em ambientes desktops (computadores e notebooks), tanto em modo *stand-alone*, dispensando a utilização da internet, ou em modo tutoria, conectado a internet e a um serviço inteligente de tutoria de Lógica. A arquitetura de *software* da ferramenta está dividida em três camadas lógicas: camada de apresentação, camada de aplicação e camada de serviços de comunicação.

- A camada de apresentação trata toda a parte da interface gráfica que o software apresenta, como os campos de entradas de fórmulas proposicionais, botões de

construção da tabela, barra de menu ou seja, toda a parte que fica visível que o usuário interage com o sistema.

- A camada de aplicação é responsável por toda a lógica de processamento dos dados, no nosso caso às fórmulas proposicionais, que são as entradas para realizar todas as rotinas de tratamento até a montagem da tabela verdade.
- A camada de comunicação é responsável pela comunicação com o serviço de tutoria, quando o modo tutoria está habilitado. Esta camada é implementada através de um cliente de serviços web, baseado no protocolo *RESTful*, que provê acesso ao *gateway* de interface aos agentes pedagógicos de tutoria instalados em um servidor remoto.

A ferramenta EVTV foi desenvolvida na linguagem Java, utilizando o ambiente *Netbeans* 8.3, pois os recursos de criação de interface gráfica *swing* são o padrão de mercado para desenvolvimento Java. O sistema de tutoria é baseado no Sistema Heráclito [Galafassi et al., 2015; Gluz et al., 2014; 2015] que está agora sendo adaptado para o suporte à resolução de exercícios de tabela-verdade. A Figura 1 apresenta a arquitetura da ferramenta EVTV, indicando também como esta ferramenta se integra aos agentes pedagógicos do sistema inteligente de tutoria.

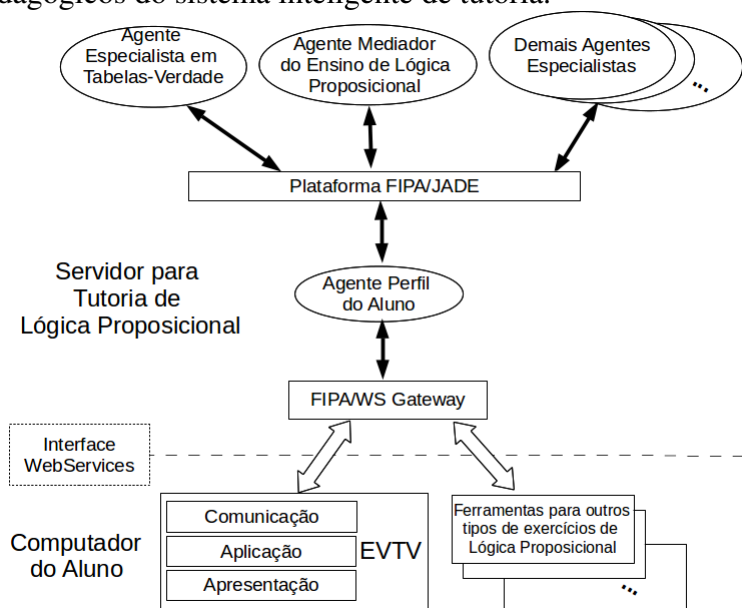


Figura 1. Arquitetura do EVTV e do Sistema de Tutoria

O sistema inteligente de tutoria em Lógica Proposicional é constituído por um sistema multiagente composto por três tipos de agentes pedagógicos: um agente de perfil de aluno que mantém os registros da interação com os alunos, além de controlar o acesso ao sistema, um agente de mediação para tutoria de Lógica Proposicional, projetado especificamente para suportar a resolução passo-a-passo de exercícios de Lógica e um conjunto de agentes especialistas nos vários tipos de exercícios da Lógica Proposicional. Além dos agentes, o sistema de tutoria opera com ferramentas clientes que executam no computador (ou dispositivo móvel, como *tablet*, em alguns casos) do aluno e permitem a resolução interativa do exercício.

Para o EVTV está sendo desenvolvido um agente especialista específico, capaz de acompanhar todo o processo de resolução de uma tabela-verdade, começando pela identificação das variáveis da fórmula, passando pelo processo de identificação das sub-

fórmulas e criação das colunas e chegando ao processo de preenchimento dos valores-verdade.

Todos os agentes pedagógicos usam a plataforma JADE [Bellifemine et al., 2007] e mensagens no padrão FIPA [FIPA, 2001] para comunicação entre si e com as ferramentas de interação com o aluno. Para intercomunicação dos agentes pedagógicos com as ferramentas de interação que executam como clientes nos dispositivos dos alunos, foi desenvolvido um *gateway* de comunicação entre a plataforma JADE e os serviços web. Quando o EVTIV está em modo de tutoria a camada de comunicação da EVTIV estabelece uma conexão com o *gateway* do sistema de tutoria, identificando e autenticando o aluno neste sistema. Logo após informações, na forma de mensagens FIPA, podem ser trocadas entre a EVTIV e o sistema de tutoria. Mensagens e avisos recebidos do sistema de tutoria são direcionados para a camada de apresentação. Mensagens de informações sobre o problema sendo resolvidos, geradas pela camada de aplicação, são enviadas para o agente especialista em tabelas-verdade, enquanto que mensagens com as entradas do aluno, geradas pela camada de apresentação, são enviadas para o agente perfil de aluno.

A camada de apresentação da EVTIV fornece uma tela principal com todos os recursos disponíveis para construção e edição da tabela-verdade (ver Figura 2). Os principais elementos desta tela são: o campo de entrada da fórmula proposicional, as ferramentas de operação, e os painéis de visualização e edição da tabela verdade e de mensagens e avisos do sistema de tutoria, todos identificados na Figura 2.

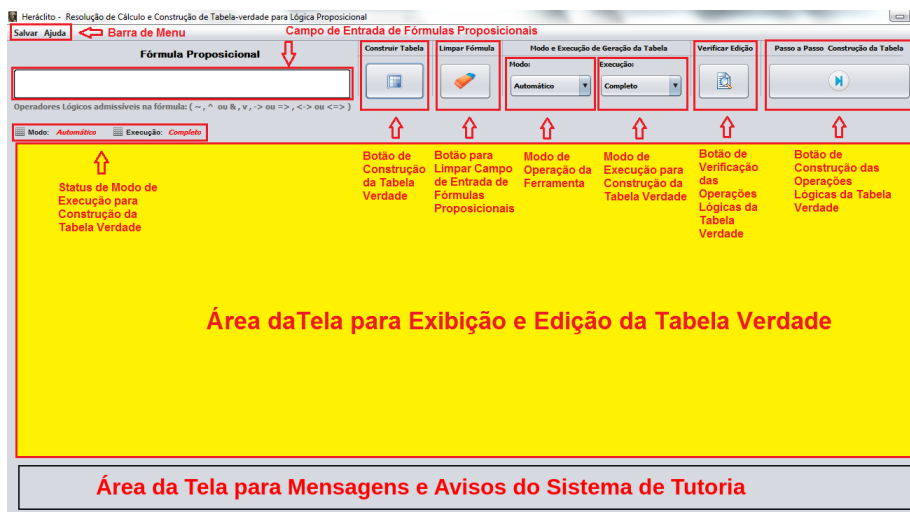


Figura 2 – Tela Principal Ferramenta EVTIV

O início da construção da tabela-verdade começa pela inserção de uma fórmula no campo de entrada de fórmula proposicional. Após clicar no botão de construção da tabela-verdade, o usuário pode escolher entre o modo automático ou modo de edição e também escolher como será a execução: completa, por linha ou por coluna. Mensagens e avisos enviados pelo sistema de tutoria são apresentados em área específica na parte inferior da tela. Problemas e erros detectados por este sistema, além de avisos importantes, também são apresentados ao aluno através de janelas *popup*.

A camada de aplicação é responsável pela validação e decomposição da fórmula proposicional de entrada. Essa fórmula é analisada para validação de suas estruturas sintáticas, de acordo com a seguinte gramática abstrata:

$\langle \text{fórmula} \rangle \rightarrow \langle \text{fórmula} \rangle \leftrightarrow \langle \text{fórmula} \rangle \mid \langle \text{fórmula} \rangle \rightarrow \langle \text{fórmula} \rangle \mid$
 $\langle \text{fórmula} \rangle \vee \langle \text{fórmula} \rangle \mid \langle \text{fórmula} \rangle \wedge \langle \text{fórmula} \rangle \mid$
 $\sim \langle \text{fórmula} \rangle \mid \langle \text{variável} \rangle \mid (\langle \text{fórmula} \rangle)$

Se a sintaxe da fórmula estiver correta, ela é dividida em suas sub-fórmulas gerando uma árvore de decomposição. Por exemplo, a fórmula de entrada $((A \rightarrow B) \wedge C) \leftrightarrow (B \vee (\neg A \vee C) \vee A)$ será decomposta na árvore apresentada na Figura 3.

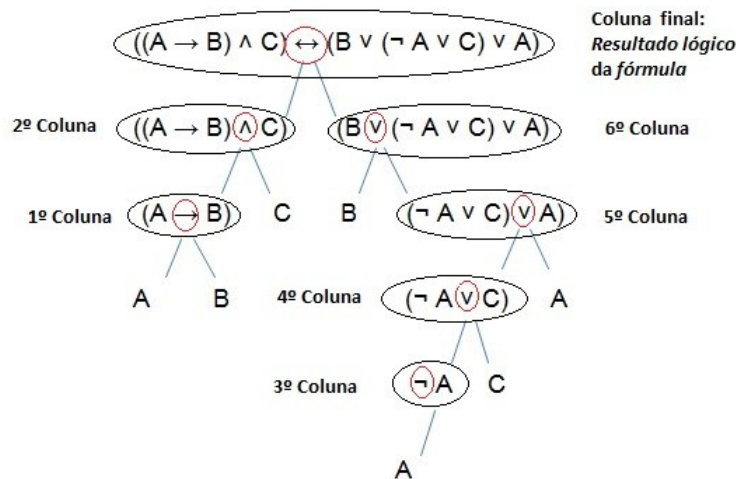


Figura 3 – Exemplo de decomposição de uma fórmula de entrada

A árvore de decomposição resultante pode ser repassada ao agente especialista, quando a ferramenta EVTIV está operando em modo de tutoria, ou se a ferramenta EVTIV estiver em modo *stand-alone* com a opção de geração automática habilitada, essas informações são usadas para gerar as colunas da tabela-verdade, de acordo com o algoritmo apresentado na Seção 2.2. Na Figura 3 também estão indicadas a ordem das colunas das sub-fórmulas.

5. Experimentos em Laboratório

Para validação da ferramenta foram realizados os testes funcionais em laboratório, sendo testadas todas as funcionalidades de interação da ferramenta. Testes de validação lógica foram conduzidos através de comparação com resultados na literatura e com resultados obtidos com as ferramentas *Truth Table Constructor* e *Truth Table Generator*. Um experimento foi conduzido com a ferramenta em modo *stand-alone* para verificar requisitos de usabilidade.

5.1. Testes Funcionais

O principal objetivo dos testes funcionais é verificar se a ferramenta atende os requisitos de interação e edição corretamente. Primeiro foram feitos testes para testar a inspeção da sintaxe da fórmula proposicional, sendo as principais verificações da ordem de parênteses e caracteres inválidos. Após, foram feitos testes para verificar se as colunas correspondentes as variáveis da tabela-verdade são criadas corretamente, incluindo o preenchimento dos valores verdade dessas colunas. Para tanto foram testados casos de duas até quatro variáveis.

Para testar a decomposição de fórmulas proposicionais foi definido um conjunto de teste de fórmulas proposicionais que foram decompostas pelo software com a subsequente geração das colunas da tabela-verdade. Para este requisito foram feitos

testes com fórmulas completamente parentizadas, como por exemplo $(A \vee (B \vee (C <-> D) -> C))$, e também fórmulas com uma quantidade mínima de parênteses, como $(\sim A \wedge B \vee C -> A <-> B)$, para que fosse possível testar o tratamento de precedência pelo software.

Por fim se procedeu aos testes de preenchimento das tabelas-verdade em modo coluna, onde a tabela deve ser preenchida uma coluna de cada vez, e em modo linha, onde a tabela é preenchida linha a linha.

Todos os testes funcionais apresentaram resultados satisfatórios.

5.2 Testes de Validade Lógica

Para um teste de validade lógica da ferramenta EVTIV deve-se avaliar se o resultado lógico final da tabela-verdade está correta. Para tanto foram utilizados exercícios de Lógica Proposicional retirados de [Abe et al., 2001]. Nestes exercícios, o número zero (0) representa o valor falso (F), enquanto que um (1) representa verdadeiro (V). Um dos exercícios utilizados no teste é apresentado na Figura 4.

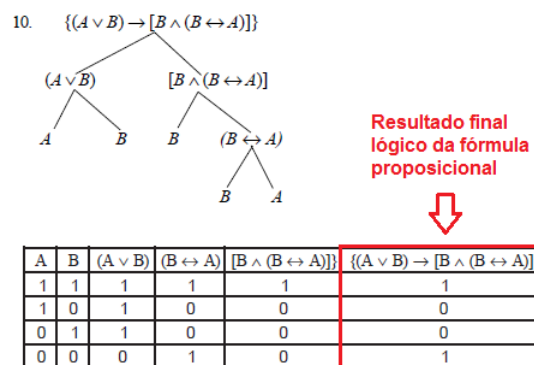


Figura 4 – Exercício de Lógica Proposicional - Fonte [Abe et al., 2001, p.56].

O resultado produzido em modo stand-alone com a opção de geração automática habilitada pode ser visto na Figura 5.

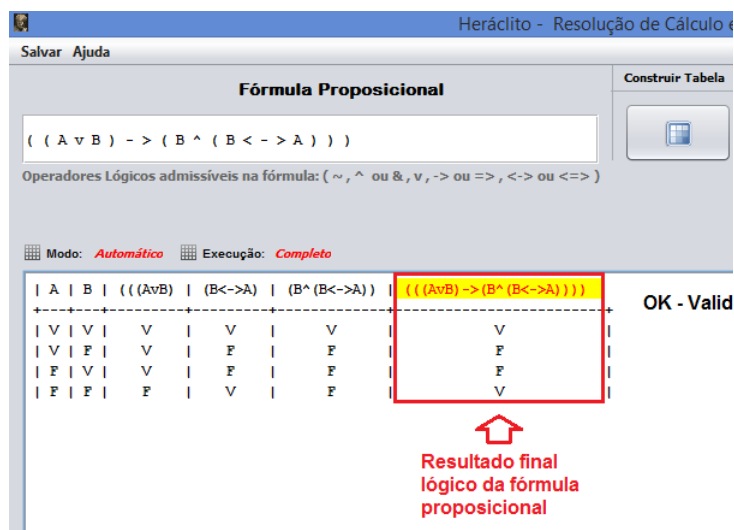


Figura 5 – Tabela-verdade produzida pela EVTIV para o exercício da Figura 4 .

Pode-se observar a conformidade da ferramenta EVTIV com os resultados esperados para a tabela-verdade.

Também foram realizados testes de validade lógica através de uma comparação dos resultados obtidos pela ferramenta EVTV com os resultados das ferramentas *Truth Table Constructor* e *Truth Table Generator*.

Todos os testes de validade lógica realizados com a ferramenta apresentaram resultados corretos, indicando a correção das tabelas-verdade produzidas pela ferramenta.

5.3. Experimento de Usabilidade

Um experimento inicial de usabilidade da ferramenta EVTV foi conduzido em laboratório para uma primeira avaliação do grau com que a ferramenta atende as expectativas dos usuários em relação ao ensino de tabelas-verdade da Lógica Proposicional.

O experimento de usabilidade foi realizado com dois avaliadores, que são professores da disciplina de Lógica, em nível superior, na Universidade XXX. Cada professor possui ao menos quatro anos de experiência ministrando esta disciplina.

Por conta do agente especialista em tabelas-verdade ainda não estar disponível o experimento de usabilidade foi conduzido com a ferramenta EVTV operando em modo *stand-alone*. O roteiro do experimento solicitava que cada avaliador desenvolvesse, a seu critério, as tabelas-verdade de um conjunto de fórmulas.

Após a condução do experimento, foi realizada uma entrevista com cada avaliador para obter uma avaliação e diagnóstico da usabilidade da ferramenta. As avaliações foram qualitativas, sendo formadas por uma lista de pontos positivos e negativos da ferramenta. Também foram sugeridas, pelos avaliadores, melhorias na ferramenta.

Os pontos positivos que podem ser destacados da análise dos avaliadores são: (a) Interface da ferramenta autoexplicativa; (b) Existência de opção de edição livre da tabela verdade; (d) Disponibilidade de criação passo-a-passo na construção da tabela verdade; (e) Possibilidade de gerar a decomposição da fórmula; (f) Mensagens identificando a posição na fórmula proposicional com erro de caracteres inválido ou erro de parênteses.

O principal ponto negativo, ressaltado pelos avaliadores é o fato do EVTV não ser uma ferramenta web.

Dentre as melhorias sugeridas pelos avaliadores, pode-se destacar: (a) Mostrar os números de linhas na construção da tabela verdade; (b) Mostrar quando o resultado final for uma tautologia; (c) Construção desta ferramenta para web.

5.4. Análise dos Experimentos e Testes

Todos os testes funcionais e de validade lógica conduzidos com a ferramenta EVTV em modo *stand-alone* produziram resultados corretos indicando que, ao menos neste modo, a ferramenta pode passar a ser utilizada pelos alunos. O experimento de usabilidade é apenas uma avaliação inicial, mas trás resultados encorajadores para a facilidade de uso e usabilidade da ferramenta. Mais experimentos, realizados em situações reais de ensino devem ser conduzidos com a ferramenta.

A avaliação mais geral da ferramenta como um instrumento de apoio ao ensino de Lógica Proposicional, no contexto de um sistema de tutoria inteligente, depende da conclusão do agente especialista em tabelas-verdade. Porém, os resultados dos testes funcionais e de validade, indicam que a ferramenta EVTV tem o potencial de ser uma boa interface para a execução de exercícios de tabelas-verdade. Questões sobre como migrar a ferramenta para a web estão sendo analisadas levando em consideração a tecnologia HTML5 e *JavaScript*.

6. Conclusões

Este trabalho apresentou a ferramenta EVTV, para o cálculo do valor verdade e construção de tabela-verdade de fórmulas da Lógica Proposicional. Os resultados dos testes e experimentos evidenciam que a ferramenta implementa corretamente os processos e conceitos necessários para a resolução de problemas de construção de tabela-verdade. Em modo *stand-alone* a ferramenta já pode ajudar o aluno a aprender todo o processo de montagem desta tabela e do cálculo de valores-verdade das suas sub-fórmulas até o chegar ao resultado final. O principal trabalho futuro é a integração com o sistema de tutoria inteligente de Lógica Proposicional e o uso efetivo da ferramenta em sala de aula.

Referências

- Abe, J. M., Scalzitti, A. e Silva Filho, J. I. da (2001). *Introdução à lógica para a ciência da computação*. São Paulo: Arte e Ciência.
- Bellifemine, F., Caire, G. e Greenwood, D. (2007) *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology, Liverpool University, UK.
- Borowski, B. (2009) *Truth Table Constructor*. Disponível em :<<http://www.brian-borowski.com>>. Acessado em Out:2014.
- Ederton, H. B. (2001) *A Mathematical Introduction to Logic*. Los Angeles: Harcourt.
- Galafassi, F.P., Santos, A. V., Peres, R. K., Vicari, R.M., Gluz, J. C. (2015). Multi-plataform Interface to an ITS of Proposicional Logic Teaching. Procs. of PAAMS 2015. CCIS, v. 524 , Springer, p. 309-319.
- Gluz, J. C., Penteadó, F., Mossmann, M., Gomes, L., Vicari, R.M. (2014) A Student Model for Teaching Natural Deduction Based on a Prover that Mimics Student Reasoning. Procs. of ITS 2014, Haway. LNCS, v. 8474, Springer, p. 482-489
- Gluz, J. C., Penteadó, F., Mossmann, M., Vicari, R. M. (2013) Heraclito: a Dialectical Tutor for Logic. Procs of EPIA 2013, Açores, Portugal . LNAI, v. 8154, Springer.
- Mortari, C.A. (2001) *Introdução á lógica*. São Paulo: Ed. Da UNESP.
- Souza, J. N. de. (2002) *Lógica para ciência da computação: fundamentos de linguagem, semântica e sistemas de dedução*. Rio de Janeiro: Campus.
- Willians, S. (2010) *Truth Table Generator*. Disponível em :<<http://programming.dojo.net.nz/study/truth-table-generator/index>>. Acessado em Out.2014.