EvoLogic: Sistema Tutor Inteligente para Ensino de Lógica

Cristiano Galafassi[1], Fabiane F. P. Galafassi[2], Eliseo B. Reategui[1], Rosa M. Vicari[1]

[1]Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação –
Universidade Federal do Rio Grande do Sul (UFRGS)
90.040-060 – Porto Alegre – RS – Brasil

[2]Universidade Federal do Pampa (UNIPAMPA) – Campus Itaqui –
970650-000 – Itaqui – RS – Brasil

{cristianogalafassi,fabiane.penteado,eliseoreategui}@gmail.com,
rosa@inf.ufrgs.br

Abstract. *This article presents the cognitive model of the EvoLogic Intelligent Tutoring System, developed to assist in the teaching-learning process of Natural Deduction in Propositional Logic. EvoLogic consists of 3 agents, among which, the Pedagogical agent (treated here as the student model) and the Specialist agent (based on a Genetic Algorithm) compose the cognitive model. The purpose of the article, in addition to presenting the EvoLogic, is to analyze the efficiency of the ITS in a known exercise that has already been studied in the literature (applied to 57 students). The results show that the EvoLogic obtained all the solutions presented by the students, allowing it to follow the student's steps, providing real-time feedback, based on the steps that the students are taking, known as model tracing.*

Resumo. *Este artigo apresenta o modelo cognitivo do Sistema Tutor Inteligente EvoLogic, desenvolvido para auxiliar no processo de ensino-aprendizagem da Dedução Natural na Lógica Proposicional. O EvoLogic consiste em 3 agentes, entre os quais o agente Pedagógico (tratado aqui como o modelo do aluno) e o agente Especialista (baseado em um algoritmo genético) compõem o modelo cognitivo. O objetivo do artigo, além de apresentar o EvoLogic, é analisar a eficiência do STI em um exercício conhecido que já foi estudado na literatura (aplicado a 57 alunos). Os resultados mostram que o EvoLogic obteve todas as soluções apresentadas pelos alunos, permitindo seguir os passos individuais de cada aluno, fornecendo feedback em tempo real, com base nos passos que os alunos estão seguindo, conhecido como model tracing.*

1. Introdução

Nos últimos anos, com o avanço da capacidade de processamento dos computadores, a Inteligência Artificial (IA) tem sido utilizada em diversos campos. Os avanços da IA associados ao surgimento de novas tecnologias impactam diretamente na Educação.

Dentre estas tecnologias temos: Sistemas Tutores Inteligentes (STI), Jogos Sérios, Sistemas Tutores Inteligentes Afetivos, *Learning Management Systems*,

Robótica Inteligente Educacional e *Massive Online Open Courses*. Cada uma destas aplicações, no entanto, faz uso de tecnologias da IA de formas distintas.

No campo da Educação, a IA tem sido utilizada, prioritariamente, em STIs com o objetivo de ampliar o acesso ao conhecimento, bem como favorecer a personalização do processo de ensino-aprendizagem. O surgimento dessas ferramentas é resultado do interesse dos pesquisadores em entender e simular cada vez mais o processo de ensino e aprendizagem com o intuito de melhorar a qualidade do ensino, de forma que seja possível alcançar melhores níveis de proficiência.

STIs são programas de computador concebidos para incorporar técnicas de IA comumente utilizadas na educação. Uma característica da IA e da educação é usar a inteligência para raciocinar sobre o ensino e a aprendizado, representando o que, quando e como ensinar determinado conteúdo. Essas características vêm de encontro com Nwana (1990), que afirma que a produção de um sistema de ensino-aprendizagem bem delineado apresenta três aspectos: 1) Conhecem o que ensinam; 2) Sabem como ensinar; e 3) Detectam como os alunos estão aprendendo.

Ainda no contexto educacional, o presente trabalho aborda a temática de Lógica para Computação. Lógica é uma componente encontrada em matrizes curriculares de cursos de Bacharelado e Licenciatura em Computação (Resolução CNE/CES nº 5, de 16 de novembro de 2016), sendo ministrada tipicamente no primeiro ou segundo semestre destes cursos. A disciplina possui como conteúdos básicos os tópicos: Lógica Proposicional – Proposições, Fórmulas e Tabelas-Verdade (Proposições e operadores lógicos, Implicação material e equivalência lógica, Fórmulas e precedência, Construção de tabelas-verdade para fórmulas Proposicional, entre outros) e Dedução Natural na Lógica Proposicional (DNLP) – Argumentos, Regras de Inferência e Provas (Argumentos válidos, Tabela-verdade para argumentos, Demonstrações formais, Regras de dedução natural, entre outros).

Dentre as diversas ferramentas de apoio para o ensino de lógica tem-se: provadores de teoremas (Coq, 2020; Hol, 2020), verificadores e/ou editores de provas formais (Prover9/Mace4, 2020; E-Prover, 2020; SPASS, 2020; JAPE, 2020; Pandora, 2020; Isabelle, 2020) e STIs (Lesta e Yacef, 2002; Yacef 2005; Lukins et al. 2007; Sieg, 2007; Galafassi et al. 2013, 2019a).

Nesse sentido, o presente trabalho tem como objetivo apresentar o modelo cognitivo do STI EvoLogic desenvolvido para lidar com problemas de Dedução Natural na Lógica Proposicional. Esse modelo cognitivo compõe o *model tracing* do EvoLogic, utilizado para acompanhar os passos individuais de cada aluno fornecer *feedback* importantes ao longo da realização de um exercício. Para tanto, apresentam-se as diversas soluções obtidas pelo EvoLogic para um exercício resolvido por 57 alunos em um experimento já apresentado em (Galafassi et al., 2019a), destacando a possibilidade de acompanhar os alunos por diferentes linhas de raciocínio previamente identificadas.

2. Fundamentação Teórica

Das ferramentas encontradas na literatura que dão apoio ao ensino de Lógica, vale o destaque de alguns STIs: Logic-ITA (Lesta e Yacef, 2002; Yacef 2005), P-Logic Tutor (Lukins et al. 2007) e AProS (Sieg, 2007), além do ambiente Heráclito (Galafassi et al., 2013, 2019a, 2019b).

O Logic-ITA é um assistente de ensino/aprendizagem de lógica proposicional baseado na Web. Seu domínio de aplicação é a construção de provas formais em lógica. O sistema atua como um intermediário entre o professor e os alunos: por um lado, ele fornece aos alunos um ambiente para a prática de provas formais com *feedback* e, por outro lado, permite aos professores monitorar o progresso e os erros da classe. O sistema é adaptado para dois tipos diferentes de usos: para os alunos, é um STI autônomo, enquanto que para os professores, inclui a funcionalidade para configurar níveis de aprendizagem, ajustar os parâmetros para progredir através destes níveis, monitorar o progresso de classe e recolher dados. O objetivo do P-Logic Tutor é ensinar aos alunos conceitos fundamentais da lógica proposicional e técnicas de prova de teoremas. O P-Logic Tutor desempenha um duplo papel como um instrumento educativo e um ambiente de pesquisa. Ele apresenta aos alunos os conceitos fundamentais da lógica proposicional e também disponibiliza a prática na resolução de teoremas. O programa também fornece um ambiente no qual é possível acompanhar o aprendizado dos alunos, explorar as questões cognitivas de resolução de problema, e investigar as possibilidades de aprendizado. O projeto AProS iniciou em 2006 buscando abranger os conteúdos de lógica. Contudo, o projeto se expandiu e atualmente conta com diversas ferramentas que se interconectam possibilitando buscar provas, resolver teoremas e acompanhar os alunos no processo de prova. Dentre elas destaca-se o Proof Tutor que faz a ponte entre as ferramentas Proof Lab e Truth Lab de prova. Ele permite que os estudantes que estão com dificuldades de avançar uma prova recebam sugestões, obtidas dinamicamente a partir de outras provas que já foram geradas.

O Ambiente Heráclito é um STI voltado para o ensino de Lógica, possibilitando que os alunos resolvam exercícios desde tabela-verdade até em prova de argumentos por meio das regras da Dedução Natural. Para tanto disponibiliza um Caderno Eletrônico de Exercícios de Lógica – LOGOS que permite criar e editar fórmulas, tabelas-verdade e provas da Lógica Proposicional (Galafassi et al., 2019b).

Além de STIs, existem os provadores automáticos, tais como: Coq e HOL, que fornecem linguagens de especificação baseados em lógicas avançadas (Lógicas de Alta Ordem), capazes de oferecer um sofisticado apoio para a construção de provas formais nestas Lógicas. Já Prover9/Mace4, EProver e SPASS são provadores automáticos de Lógica de Primeira Ordem; além dos editores/verificadores de provas JAPE, Pandora e Isabelle.

Dentre os STI apresentados (Logic-ITA, P-Logic Tutor, AproS e Heráclito), todos fazem uso de uma demonstração formal que oferece uma estrutura simbólica apropriada para acompanhar o processo de ensino-aprendizagem de dedução na Lógica. Contudo, apenas o AproS e o Heráclito se utilizam de demonstrações similares às utilizadas pelos professores e alunos em sala de aula. Ainda no que tange a demonstração, apenas o Heráclito apresenta a possiblidade de continuar a prova do teorema de onde o aluno parou, bem como pode fornecer o próximo passo (com base na prova atual). Nesse contexto, o EvoLogic se assemelha ao Heráclito, por possuir todas essas características. Contudo, a diferença se dá no processo em como a solução é gerada. Enquanto o Heráclito continua a prova do ponto que lhe foi apresentada (exercício novo ou parcialmente resolvido pelo aluno), diferentemente do EvoLogic. Por possuir um AG como especialista, o EvoLogic obtém inúmeras soluções para o mesmo exercício no momento em que o aluno o inicia. Essas diversas soluções podem levar a diferentes linhas de raciocínio possibilitando que o *model tracing* forneça

feedback (sugerindo novos passos ao aluno) de acordo com o comportamento que ele vem apresentando ao longo da resolução do exercício.

3. Materiais e Métodos

Dentre os materiais, citam-se:

- Exercícios de DNLP: consistem em uma sequência de passos (aplicação de regras de dedução) que buscam provar um teorema. Esses exercícios são tipicamente ministrados nos semestres iniciais de cursos de computação, tendo como objetivo, desenvolver o raciocínio lógico dos alunos. Os exercícios de DNLP aqui tratados se baseiam no livro escrito por Gluz e Py (Gluz e Py, 2015).
- Algoritmo Genético (AG): o agente especialista do EvoLogic consiste em um AG adaptado para resolver problemas de DNLP. Esse agente segue o funcionamento clássico dos AG com adaptações na representação e interpretação da solução, operadores de cruzamento e mutação que devem evitar a geração de soluções inviáveis e na função de aptidão.

Quanto ao método, essa pesquisa se classifica como exploratória, pois busca-se identificar como os alunos se comportam ao resolver exercícios de DNLP, avaliando o modelo cognitivo do EvoLogic.

4. EvoLogic

O STI EvoLogic consiste em um sistema multiagente que busca acompanhar o aluno no processo de resolução de um problema. Mais especificamente, o foco do STI se dá na resolução de problemas da lógica Proposicional, onde faz-se necessário a realização de múltiplos passos para obter uma solução viável ou recaiam em problemas de análise combinatória.

O EvoLogic é apresentado como uma arquitetura composta por 3 agentes: Interface, Pedagógico e Especialista. Neste trabalho, o foco se dá no modelo cognitivo, composto pelos agentes Pedagógico e Especialista. Cabe salientar que o modelo pedagógico tradicional em um STI compreende: estratégias de ensino aprendizagem, táticas pedagógicas e modelo do aluno. No âmbito deste trabalho, o foco do agente Pedagógico está no modelo do aluno. A comunicação entre os agentes se dá através da troca de simples mensagens, suportadas pelas estruturas de memória, a qual também armazena o Modelo de Aluno e o Conhecimento acerca dos exercícios resolvidos pelo agente Especialista. A organização do ponto de vista de um sistema multiagente pode ser visto na Figura 1, onde são destacados os elementos que compõem o sistema e o Web Service, utilizado para receber as mensagens provenientes da interação do aluno com seu ambiente de trabalho.

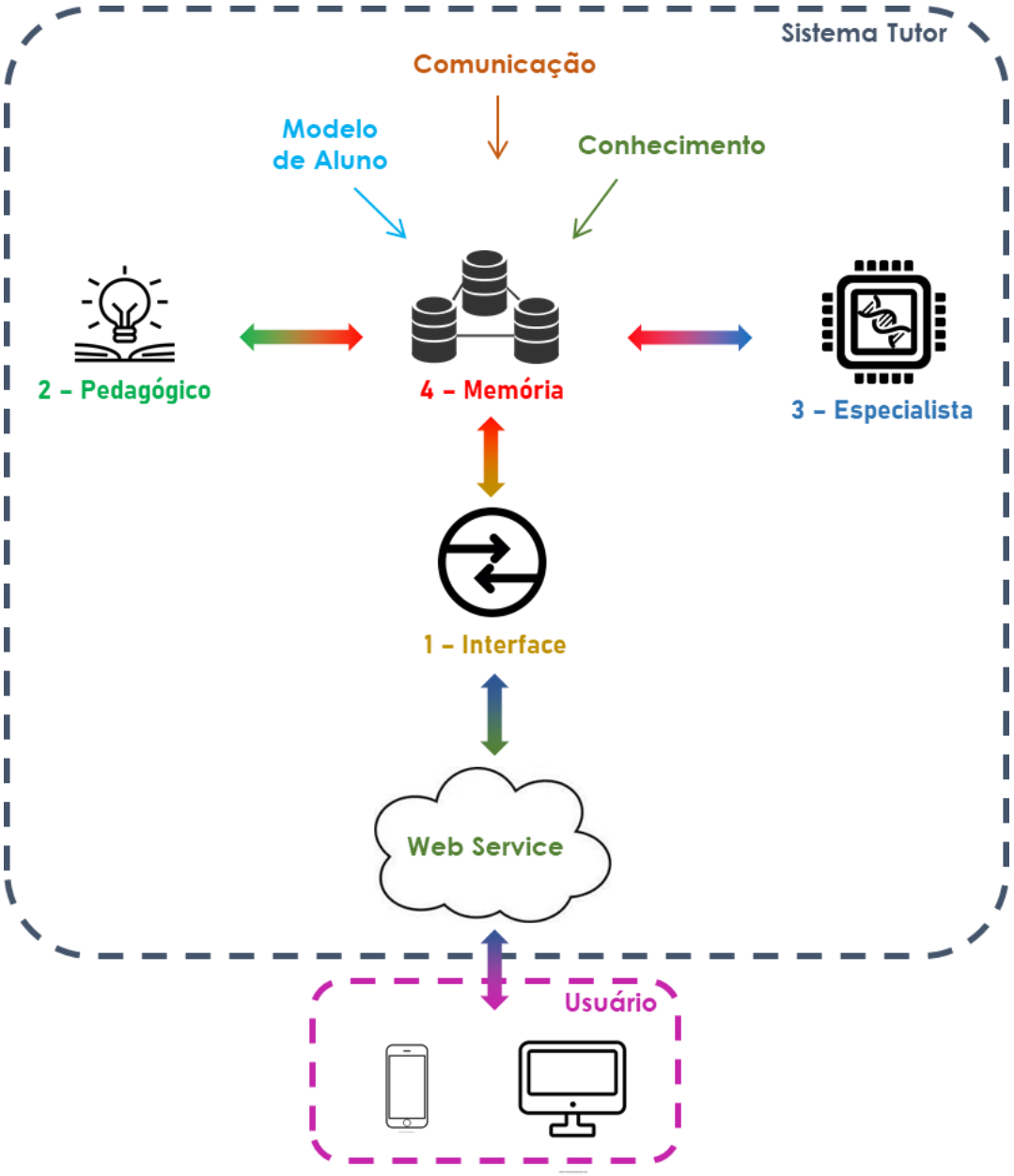

Figura 1. Arquitetura Multiagente do STI EvoLogic.

O Agente Especialista consiste em um Algoritmo Genético adaptado para resolver problemas de DNLP, inspirados nos modelos clássicos da literatura (Holland, 1975). Desse modo, aqui serão destacados os elementos diferenciais, que tornam o Agente Especialista capaz de lidar com a resolução dos problemas em questão: interpretação da solução, representação do indivíduo, operadores de cruzamento e mutação e função de aptidão.

A interpretação da solução consiste em identificar se a solução para o problema é correta, em outras palavras, se o teorema pode ser provado, bem como avaliar os passos que foram realizados para obtenção da solução. A Figura 2 mostra duas soluções corretas para o exercício A, A->B, B->C |- C, onde A, A->B, B->C são as hipóteses e deseja-se provar C, identificando os passos que foram realizados.

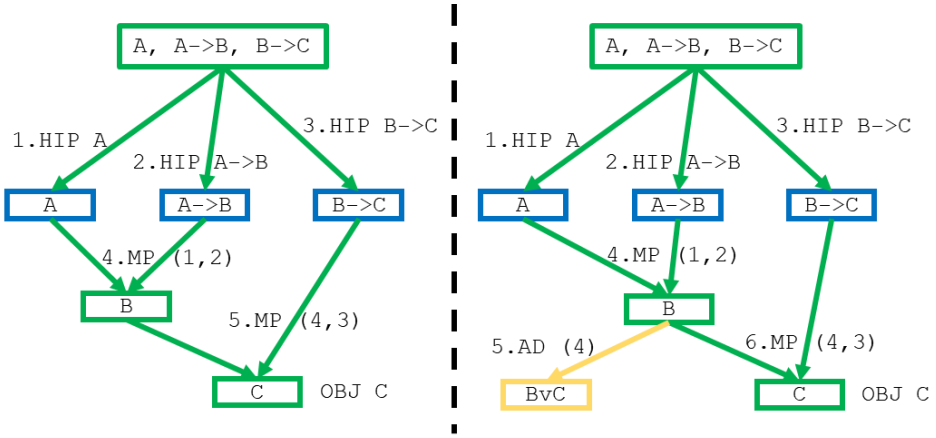

Figura 2. Exemplo de duas soluções para um exercício de DNLP.

Ao lado esquerdo, vemos uma solução composta com 5 passos, todos levando diretamente ao objetivo C. Em contra partida, do lado direito, também temos uma solução correta, atingindo o objetivo C, contudo pode ser visto que foi realizado um passo extra (5.AD (4)). Esse passo não inviabiliza a solução, uma vez que a regra de Adição pode ser aplicada durante o processo. Entretanto, esses passos podem indicar que o aluno está tendo alguma dificuldade e pode ser conveniente uma intervenção por parte do tutor. Vale mencionar que em alguns exercícios, o aluno pode optar por utilizar apenas regras básicas, evitando o uso de regras derivadas. Nesses casos, é importante que o tutor identifique claramente a diferença em as linhas de raciocínio e passos extras.

A representação do indivíduo, no AG, é feita através de um vetor bidimensional, onde armazenma-se a regra aplicada e seus dependentes. Um exemplo pode ser visto na Tabela 1 onde acrescentou-se o índice para que fosse possível visualizar as dependências dos passos 4 e 5.

Tabela 1. Representação de um indivíduo.

Índice	0	1	2	3	4	5
Passos	OBJ C	HIP A	HIP A->B	HIP B->C	MP B	MP C
Dependência	-	-	-	-	1,2	4,3

O operador de cruzamento foi desenvolvido para evitar que soluções inviáveis sejam geradas (i.e., soluções que contenham erros). Para gerar um novo indivíduo, copia-se o objetivo e as hipóteses de um dos pais (essa informação será sempre igual em todos os indivíduos) e seleciona-se aleatoriamente um dos demais passos desse mesmo pai e copia-se o passo e suas dependências para o novo indivíduo. Em seguida, realiza-se o mesmo processo com o segundo pai, copia-se um passo e seus dependentes para o novo indivíduo (selecionado aleatoriamente). O processo se repete até que o indivíduo esteja completo. Por fim, o operador de mutação substitui um passo por um novo ou inclui um novo passo na solução corrente.

A Função de Aptidão é dada pela Equação 1.

$$f = (1 - x) \times t_i + (x \times (E_i \times c)) \quad (1)$$

Onde:

- x é uma variável inteira assumindo valor 1 se o indivíduo i resolve o problema e 0 caso contrário;

- t_i é o tamanho do indivíduo i (i. e., o número de genes)
- E_i é a eficiência do indivíduo i, calculada apenas quando $x = 1$ através da Equação 4.2;
- c é uma constante que determina o peso dado a uma solução que resolve o problema. Nesta proposta será utilizado $c = 25$, sendo 25 o tamanho máximo de um indivíduo.

Utilizando $(1 - x)$ e x, é possível zerar uma parte da equação, priorizando os indivíduos que possuem mais características genéticas (maiores indivíduos) ou os indivíduos mais eficientes. Isso permite que os indivíduos que possuem maior carga genética passem adiante parte de seus genes, além de permitir que os indivíduos que apresentem soluções também tenham um lugar de destaque na população através do cálculo da eficiência da solução (Equação 2).

$$E_i = t'_i / t_i \tag{2}$$

Onde:

- t_i é o tamanho do indivíduo i;
- t'_i é o tamanho do indivíduo considerando apenas as características que compõem o a solução do problema.

A Equação 2 apresenta a relação entre o tamanho da solução e as partes que compõem a solução do problema. Em diversas situações, um indivíduo pode representar a solução para um problema que demanda diversos passos. Alguns desses passos podem ser simplificados, não sendo necessário que eles apareçam no indivíduo. Nesse contexto, apenas os passos que estão diretamente ligados com a solução são considerados na variável t'_i.

Ainda na Equação 1, a constante c serve como um viés para priorizar soluções corretas em detrimento a soluções que não resolvem o problema. Vale citar que todas os indivíduos são aceitáveis durante o processo evolutivo, uma vez que o problema abordado neste trabalho é trivial e demanda que seja realizada uma abrangente exploração do espaço de soluções.

Por fim, o agente Pedagógico, no que tange o modelo cognitivo é responsável pelo modelo aluno, acompanhando os passos realizados pelo aluno e categorizando-o pela sua qualidade e pela linha de raciocínio seguida ao longo da resolução do exercício.

5. Resultados

Em 2019 (Galafassi et al., 2019a) foi proposto um novo modelo de aluno para o Ambiente Heráclito, no contexto de NDLP, onde foram apresentados os resultados acerca da resolução de 10 exercícios, de modo que todos os passos realizados pelos alunos foram armazenados. Com isso, é possível reproduzir os passos de cada aluno, individualmente, no EvoLogic buscando a identificar as linhas de raciocínio de cada aluno. Para evidenciar as características do modelo cognitivo, optou-se pelo uso dados referentes ao exercício: A<->Q,F<->R,A∧R|-F∧Q onde as hipóteses são A<->Q, F<->R e A∧R e deseja-se provar F∧Q. Dentre os 57 alunos que participaram do experimento, 44 resolveram corretamente o exercício enquanto que os demais (13 alunos), iniciaram a resolução, mas não a concluíram com sucesso (seja por dificuldade ou falta de tempo).

Ao iniciar o exercício, tanto ambiente Heráclito quanto o EvoLogic resolvem o exercício a fim de identificar se o mesmo possui soluções viáveis. Dada a característica de funcionamento do Ambiente Heráclito, apenas uma solução é obtida incialmente (sendo que novas soluções são geradas com base nos passos do aluno). A solução obtida pelo Ambiente Heráclito é mostrada na Tabela 2, onde OBJ define o que se deseja provar, HIP representam as Hipóteses do problema e -EQ, SP, MP e CJ representam as regras Eliminação da Equivalência, Simplificação, Modus Ponens e Conjunção respectivamente. A primeira coluna identifica os passos enquanto que a segunda coluna mostra os passos realizados pelo aluno, onde tem-se a regra aplicada, o resultado da aplicação da regra e as linhas que foram utilizadas para se obter o determinado resultado. Esse formato de apresentação será adotado no restante do trabalho.

Tabela 2. Solução obtida pelo ambiente Heráclito.

Heráclito	
0	OBJ F∧Q
1	HIP A<->Q
2	HIP F<->R
3	HIP A∧R
4	-EQ R->F (2)
5	-EQ A->Q (1)
6	SP A (3)
7	SP R (3)
8	MP Q (6, 5)
9	MP F (7, 4)
10	**CJ F∧Q (9, 8)**

Em contra partida, o EvoLogic, por utilizar um mecanismo evolutivo como especialista, obtém diversas soluções para o mesmo problema ao longo das gerações populacionais. Ao aplica-lo no exercício em questão, foram obtidas 8 soluções diferentes (ao remover os passos extras), sendo que os alunos seguiram por 5 delas, mostradas nas Tabelas 2 e 3, onde mostra-se também a geração populacional em que foram obtidas. Os parâmetros utilizados para obtenção desses resultados foram: Tamanho da População 50, Taxa de Mutação 5% e Total de Geração Populacional 500 (critério de parada).

Tabela 2. Soluções 1 e 2 obtidas pelo EvoLogic.

EvoLogic – 1 (135)		EvoLogic – 2 (112)	
0	OBJ F∧Q	0	OBJ F∧Q
1	HIP A<->Q	1	HIP A<->Q
2	HIP F<->R	2	HIP F<->R
3	HIP A∧R	3	HIP A∧R
4	-EQ R->F (2)	4	-EQ A->Q (1)
5	-EQ A->Q (1)	5	-EQ R->F (2)
6	SP A (3)	6	SP A (3)
7	SP R (3)	7	SP R (3)
8	MP Q (6, 5)	8	MP F (5, 7)
9	MP F (7, 4)	9	MP Q (4, 6)
10	**CJ F∧Q (9, 8)**	**10**	**CJ F∧Q (8, 9)**

Tabela 3. Soluções 3, 4 e 5 obtidas pelo EvoLogic.

EvoLogic – 3 (193)		EvoLogic – 4 (189)		EvoLogic – 5 (203)	
0	OBJ F∧Q	0	OBJ F∧Q	0	OBJ F∧Q
1	HIP A<->Q	1	HIP A<->Q	1	HIP A<->Q
2	HIP F<->R	2	HIP F<->R	2	HIP F<->R
3	HIP A∧R	3	HIP A∧R	3	HIP A∧R
4	-EQ A->Q (1)	4	SP R (3)	4	SP R (3)
5	SP A (3)	5	SP A (3)	5	-EQ R->F (2)
6	SP R (3)	6	-EQ A->Q (1)	6	MP F (4, 5)
7	MP Q (4, 5)	7	MP Q (6, 5)	7	SP A (3)
8	-EQ R->F (2)	8	-EQ R->F (2)	8	-EQ A->Q (1)
9	MP F (8, 6)	9	MP F (8, 4)	9	MP Q (7, 8)
10	**CJ F∧Q (8, 7)**	**10**	**CJ F∧Q (9, 7)**	**10**	**CJ F∧Q (6, 9)**

Do ponto de vista do AG, verifica-se que a última solução obtida foi na geração 203 (considerando as 8 obtidas), apesar de terem sido geradas 500 populações. Esse fenômeno pode ocorrer devido a falta de variabilidade genética, impedindo que o AG obtenha novas soluções. Nesse contexto, sempre que o aluno apresentar uma linha de raciocínio que seja contemplada, uma nova instância do AG é executada, inicializando a população com características similares as apresentadas pelo aluno. Ressalta-se que neste exercício não foi necessário realizar esse procedimento, sendo que todas as soluções foram obtidas na primeira execução do AG.

Com base nos passos realizados pelos 57 alunos, realizou-se uma simulação, utilizando o modelo cognitivo do EvoLogic, onde foram realizados (de forma simulada) os passos individuais de cada aluno.

Tendo como base os passos dos alunos, verificou-se que alguns deles foram objetivamente até a solução enquanto que outros acabaram realizando alguns passos extras. Esses passos extras podem indicar que, para o aluno, o processo de resolução não está claro. Quando isso ocorre, além de saber o quanto o aluno conhece sobre as regras que ele deve aplicar, é importante identificar qual linha de raciocínio ele vinha seguindo para fornecer *feedback* adequados. Nesse sentido, é importante detectar diversas soluções possíveis para o problema pois elas podem empregar linhas de raciocínio diferentes. Considerando as soluções 1 e 2, os alunos optaram primeiramente por manipular as Hipóteses para então combinar os resultados e chegar na solução. Pode-se observar que os primeiros passos (4 ao 7) aplicam regras diretamente nas hipóteses e, somente nos passos seguintes (8 e 9) buscam derivar as proposições para atingir a solução (passo 10).

No caso da solução 5, pode-se verificar um comportamento diferente, onde o aluno aplica as regras em duas Hipóteses (passos 4 e 5) e já os manipula (passo 6). Em seguida, retorna para a hipóteses (passos 7 e 8) e novamente os manipula (passo 9), culminando na obtenção da solução.

Comparando as duas estratégias, percebe-se duas linhas de raciocínio distintas que necessitam de acompanhamento separado. No primeiro caso (soluções 1 e 2), dos 28 alunos que apresentaram essa solução, 5 deles realizaram passos extras, o que pode indicar que eles estavam explorando as possiblidades para entender como proceder com a resolução. No segundo caso (solução 5), 4 alunos seguiram os passos na mesma

ordem, indicando que poderiam ter identificado a linha de solução antes de começar a interagir com ambiente (nenhum desses alunos realizou passos extras).

Além disso, os outros 12 alunos seguiram os passos das soluções 3 e 4, sendo que 11 deles realizaram um ou mais passos extras. Analisando essas soluções, pode-se perceber que ocorre uma alternância entre a aplicação de regras nas hipóteses e em suas derivadas. Essa alternância se difere da apresentada na solução 5 pela quantidade de passos extras. Esses passos indicam que os alunos identificaram que algumas regras podem ser aplicadas, contudo não identificou uma regra leve à resolução do exercício.

Vale ressaltar que todos os passos realizados pelos alunos foram acompanhados e categorizados, tanto no que tange a qualidade (se faz ou não parte da solução) quanto ao caminho pela sua linha de raciocínio.

4.1. Discussões

A diferença entre o Heráclito e o EvoLogic se dá na forma de lidar com as soluções dos exercícios. Enquanto o Heráclito realiza a prova a partir do ponto onde o aluno está, o EvoLogic já possui a solução e apenas acompanha o raciocínio do aluno através da solução.

Em alguns casos, no trabalho de (Galafassi et al., 2019a), os participantes apontaram que acabaram se confundindo ao longo da resolução do exercício pois estavam seguindo um raciocínio que demandaria 3 passos a conclusão da prova do teorema enquanto que o Heráclito apontou que faltavam apenas 2 passos (mensagem enviada como forma de incentivo ao aluno). Por ter continuado a prova parcial do aluno, do ponto onde ele estava, o ambiente Heráclito identificou um caminho com menos passos. Isso pode ter ocorrido pois o Heráclito concluiu a prova do teorema utilizando alguma regra derivada, enquanto que o aluno estaria utilizando apenas regras básicas (obtendo o mesmo resultado com mais passos). É importante salientar que ambas as soluções estão corretas e que a diferença se dá na linha de raciocínio seguida pelo aluno. Diferentemente do Heráclito, o EvoLogic poderia ter verificado que ainda existiram mais de um caminho possível, evitando que tal mensagem fosse enviada antecipadamente.

Essa diferença sutil permite que estratégias de mediação pedagógica mais precisas sejam exploradas, bem como a criação de um mecanismo de *model tracing* automatizado, que estime por qual linha de raciocínio o aluno está seguindo, fornecendo *feedback* direcionados ao seu perfil.

O mecanismo de identificação da linha de raciocínio do aluno pode se utilizar de diversas características como: continuações possíveis a partir da solução parcial, histórico de comportamento do aluno (como e quais passos ele vem realizando), estimativas de conhecimento de regras que envolvem cada caminho (modelo de aluno) e também a probabilidade de conhecer determinada solução (também com base no modelo de aluno).

6. Conclusões

Este artigo apresentou o Sistema Tutor Inteligente EvoLogic, em particular, seu modelo cognitivo, composto por 2 agentes, Pedagógico e Especialista. O agente Pedagógico representa o modelo do aluno, enquanto o especialista é responsável por resolver os

exercícios DNLP iniciados pelos alunos. O modelo cognitivo visa seguir cada passo do aluno, identificando quando ele apresenta alguma dificuldade e fornecendo *feedback* relevantes ao seu aprendizado.

Para avaliar as capacidades do agente Especialista, um exercício conhecido, aplicado a 57 alunos e registrado passo a passo, foi aplicado no EvoLogic, em um processo simulado. Cada passo realizado pelos estudantes foi simulado no STI, onde as soluções foram analisadas e comparadas.

O EvoLogic obteve 8 soluções diretas para o problema (excluindo os passos extras), enquanto que os 44 estudantes que provaram o teorema com sucesso, apresentaram 5 dessas soluções. Pode ser observado que os alunos adotaram diferentes abordagens, optando por explorar todas as hipóteses antes de considerar suas derivações ou derivando os resultados ao máximo e retornando as hipóteses quando necessário. Nesse processo, verificou-se que em algumas linhas de raciocínio (soluções 3 e 4), os alunos apresentaram vários passos extras. Isso pode indicar que os alunos estavam tendo dificuldades, dando subsídios para que estratégias de mediação pedagógica voltadas para cada comportamento possam ser desenvolvidas.

Dessa forma, os objetivos deste estudo são alcançados, abrindo uma série de estratégias possíveis para futuras aplicações. Portanto, como trabalhos futuros, busca-se uma análise mais detalhada do agente Especialista, principalmente no que se refere a problemas com regras de dedução mais complexas, bem como a formalização de um modelo de *model tracing* automatizado, suportado pelas soluções geradas pelo agente Especialista.

Referências

Coq (2020) The Coq Proof Assistant. https://coq.inria.fr/

E-Prover (2020) The E Theorem Prover. https://wwwlehre.dhbw-stuttgart.de/~sschulz/E/E.html.

GALAFASSI, P. F. F.; GLUZ, J. C. ; GALAFASSI, C. (2013) Análise Crítica das Pesquisas Recentes sobre as Tecnologias de Objetos de Aprendizagem e Ambientes Virtuais de Aprendizagem. REVISTA BRASILEIRA DE INFORMÁTICA NA EDUCAÇÃO, v. 21, p. 85-99.

Galafassi, P. F. F.; Galafassi, C.; Vicari, R. M.; Gluz, J. C.; (2019a) Identifying Knowledge from the Application of Natural Deduction Rules in Propositional Logic. In: Demazeau Y., Matson E., Corchado J., De la Prieta F. (eds) Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection. p. 66-77.

Galafassi, P. F. F.; Galafassi, C.; Gluz, J. C.; Vicari, R. M. (2019b) LOGOS - Caderno de Estudos e Exercícios de Lógica do Ambiente de Ensino Heráclito. In: Karina Durau. (Org.). Demandas e Contextos da Educação no Século XXI. 1ed.Ponta Grossa: Atenas Editora, p. 1-8.

Gluz, J. C., Py, M. X. (2015) Introdução à Lógica Proposicional. Grupo TAOS3 - PIPCA – UNISINOS.

Hol (2020) Interactive Theorem Prover. https://hol-theorem-prover.org/

Holland, J. H. (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan.

Isabelle (2020) Isabelle proof assistant. https://isabelle.in.tum.de/.

JAPE. (2020) http://japeforall.org.uk/

Lesta L., Yacef K. (2002) An Intelligent Teaching Assistant System for Logic. In: Cerri S.A., Gouardères G., Paraguaçu F. (eds) Intelligent Tutoring Systems. ITS 2002. Lecture Notes in Computer Science, vol. 2363. p. 421-431.

Lukins, S.; Levicki, A. and Burg, J. (2002) A Tutorial Program for Propositional Logic with Human/Computer Interactive Learning. Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education. p. 381-385.

Nwana, H. S. (1990) Intelligent Tutoring Systems: an overview. Artificial Intelligence Review. Springer,

Pandora (2020) Proof Assistant for Natural Deduction using Organised Rectangular Areas. http://www.doc.ic.ac.uk/pandora/newpandora/quick_start.html.

Prover9/Mace4 (2020) Prover9 and Mace4 https://www.cs.unm.edu/~mccune/prover9/

Resolução CNE/CES nº 5, de 16 de novembro de 2016. http://portal.mec.gov.br/component/content/article?id=12991

Sieg, W. (2007) The AProS project: Strategic thinking & computational logic. Logic Journal of IGPL, 15(4): 359-368.

SPASS (2020) Classic SPASS: An Automated Theorem Prover for First-Order Logic with Equality. https://www.mpi-inf.mpg.de/departments/automation-of-logic/software/spass-workbench/classic-spass-theorem-prover/.

Yacef, K. (2005) The Logic-ITA in the classroom: a medium scale experiment. International Journal of Artificial Intelligence in Education (IJAIED), vol. 15, p. 41-62.