

# Sistema Tutor Baseado em Inteligência Artificial Explicável para Ensino de Autômatos

Elvis Samir Gannem Silva  
Universidade do Vale do Itajaí –  
Mestrado em Computação  
Aplicada,  
elvis.silva@edu.univali.br

Anita Fernandes  
Universidade do Vale do Itajaí  
Mestrado em Computação  
Aplicada  
anita.fernandes@univali.br

Rudimar Dazzi  
Universidade do Vale do Itajaí  
Mestrado em Computação  
Aplicada  
rudimar@univali.br

## ABSTRACT

Formal Languages and Automata constitute a fundamental domain in Computer Science; however, courses addressing this subject are often associated with high academic failure rates. Intelligent Tutoring Systems (ITS) have been employed to support student learning yet providing effective and meaningful feedback remains a significant challenge. Recent research has explored the integration of Machine Learning techniques and Large Language Models (LLMs) to enhance instructional personalization, performance diagnosis, and automated feedback generation. Despite their potential, the increasing sophistication of these models has raised concerns regarding the lack of transparency in pedagogical decision-making processes. Such opacity undermines trust, fairness, and ethical accountability in ITS, thereby fostering the development and adoption of Explainable Artificial Intelligence (XAI) approaches. In this context, this paper proposes the development of an XAI-based Intelligent Tutoring System for teaching Automata.

## KEYWORDS

Explainable Artificial Intelligence, Intelligent Tutoring Systems, Automata.

## 1 Introdução

Os Sistemas Tutores Inteligentes (STI) são sistemas computacionais que utilizam técnicas de Inteligência Artificial para oferecer ensino individualizado, avaliando o estado de conhecimento do aprendiz, adaptando instruções e fornecendo feedback em tempo real com o objetivo de replicar os benefícios da tutoria [1]. Em termos gerais, tais sistemas são concebidos para simular um comportamento instrucional que, se fosse realizado por um ser humano, seria descrito como o de um “bom professor”, ou seja, alguém que sabe ensinar, a quem ensinar, e como ensinar [2].

Os STI vêm incorporando técnicas de Machine Learning e LLM (Large Language Models) para aprimorar a personalização do ensino, o diagnóstico de desempenho e a geração automática de feedback. Contudo, a crescente sofisticação desses modelos tem sido acompanhada por um déficit de transparência em suas decisões pedagógicas, dificultando a compreensão de como e por

que determinadas recomendações, avaliações ou intervenções são realizadas [3]. Apesar dos avanços na acurácia dos modelos de aprendizado de máquina, muitos deles permanecem como “caixas pretas”, dificultando a sua adoção em domínios sensíveis [4].

A IA Explicável (XAI) surge como resposta a essa problemática buscando tornar compreensíveis as decisões dos sistemas tutores inteligentes para seus diferentes públicos (tutores humanos, alunos e desenvolvedores). Neste sentido, este trabalho apresenta a proposta de um STI baseado em XAI para ensino de autômatos.

## 2 Visão Geral do Sistema Proposto

O sistema proposto tem sua arquitetura baseada no modelo clássico para STIs descrito por Nwana [5], estruturada em quatro componentes interdependentes:

- Modelo de domínio: armazena o conhecimento especialista e é responsável pelo diagnóstico de erros através de um motor de inferência.
- Modelo de aluno: estima o conhecimento do aluno sobre os conceitos através de uma abordagem probabilística (redes bayesianas).
- Modelo pedagógico: num primeiro momento, este componente traduz os diagnósticos técnicos em feedbacks explicativos e rastreáveis.
- Interface de usuário: o ponto de partida do usuário é a interface gráfica onde as interações de construção de autômatos ocorrem.

### 2.1 Modelo de Domínio

Para o modelo de domínio foi criada uma base de conhecimento que armazena uma solução ideal para cada problema e separamos os problemas por tópicos. A base focou nos seguintes tópicos: autômatos finitos determinísticos (DFA), autômatos finitos não determinísticos (NFA), minimização de autômatos e transformação de NFA para DFA. A Figura 1 mostra a estrutura dos dados em JSON que representa um autômato e a sua solução ideal na base de conhecimento, onde: (i) *difficulty* representa a complexidade do exercício; (ii) *meta* contém o alfabeto que o aluno deve levar em

consideração para a construção do autômato, e a chave *kc* se refere ao conceito pedagógico principal avaliado pelo exercício; (iii) *ideal\_solution* armazena a solução ideal e minimizada do exercício; (iv) as chaves *test\_suite* e *constraints* são usadas pelo motor de inferência.

```

{
  "title": "DFA que aceita strings terminadas em 'a'",
  "description": "Construa um AFD que aceite todas as strings (alfabeto (a, b))
  que terminam com o caractere 'a'.",
  "difficulty": "easy",
  "meta": {
    "alphabet": ["a", "b"],
    "kc": "K_DFA"
  },
  "ideal_solution": {
    "states": [
      {
        "id": "q0",
        "isInitial": true,
        "isFinal": false
      },
      {
        "id": "q1",
        "isInitial": false,
        "isFinal": true
      }
    ]
  },
  "transitions": [
    {
      "from": "q0",
      "to": "q1",
      "symbol": "a"
    },
    {
      "from": "q0",
      "to": "q0",
      "symbol": "b"
    },
    {
      "from": "q1",
      "to": "q1",
      "symbol": "a"
    },
    {
      "from": "q1",
      "to": "q0",
      "symbol": "b"
    }
  ],
  "test_suite": {
    "accept": ["a", "ba", "aba", "bbba"],
    "reject": ["", "b", "ab", "bbb"]
  },
  "constraints": [
    {
      "code": "MISSING_INITIAL_STATE",
      "type": "structural",
      "severity": "error",
      "message": "O autômato deve ter um estado inicial.",
      "hint": "Você precisa marcar um estado como inicial."
    }
  ]
}

```

Figura 1: Exemplo de autômato representado na base de conhecimento.

### 2.2 Motor de Inferência

O motor de inferência é o interpretador das regras do sistema e faz parte do modelo de domínio do tutor. A função do motor de inferência é aplicar as regras da base de conhecimento [6]. Executa dois tipos de validação: JSON que representa um autômato e a sua solução ideal na base de conhecimento, na qual tem-se a validação de restrições e a validação funcional.

As restrições são um conjunto de regras que o aluno deve seguir para que a solução seja válida dentro do domínio [7]. O motor de inferência valida se as restrições estão seguindo seguidas (chave *constraints* da Figura 1), e proporciona um feedback seguindo uma classificação pela chave *severity*, onde restrições que impedem a execução do autômato são prioritárias. Na validação funcional tem-se a chave *test\_suite* da Figura, que representa o conjunto de cadeias que devem ser aceitas e recusadas para considerar a resolução do problema correto.

### 2.3 Modelo do Aluno

O objetivo do modelo de aluno é estimar o domínio do aluno sobre conceitos fundamentais na construção de autômatos. O modelo considera que diferentes tipos de erro indicam que o aluno possui dificuldades específicas. Por exemplo, um exercício correto, porém com estados desnecessários, indica que o aluno entende o conceito de autômatos finitos, mas, que precisa entender melhor a minimização de autômatos. Para isso, utilizou-se uma rede bayesiana (RB) [8].

A RB é um grafo probabilístico que representa as relações de dependência entre as habilidades do domínio. Cada nó corresponde a um conceito binário (domina/não domina), e as arestas representam os pré-requisitos (o conhecimento da definição formal da teoria dos autômatos é um pré-requisito para a construção de um DFA e de um NFA).

Após a resolução de cada exercício, o motor de inferência fornece um diagnóstico. Baseado nesse código, a RB calcula a probabilidade de o aluno dominar o conceito relacionado. Depois da resolução de cada exercício, a probabilidade é atualizada. O resultado é um perfil dinâmico que representa o conhecimento atual do aluno sobre cada conceito, permitindo ao modelo pedagógico tomar decisões de adaptação. A estrutura da rede reflete a hierarquia pedagógica da teoria de autômatos, onde o domínio de um conceito básico é um pré-requisito para o domínio de um conceito mais avançado. A Figura 2 é um grafo da RB que ilustra a estrutura do modelo, onde os vértices com prefixo K são as variáveis que serão medidas, e os vértices com prefixo OBS são as evidências.

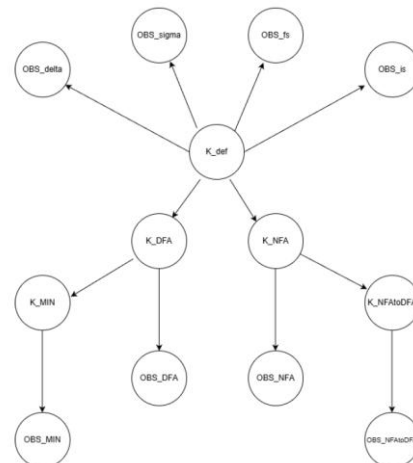


Figura 2: Grafo que representa a estrutura da RB.

Como a lógica da RB é quantificada através das tabelas de probabilidade condicional (CPTs), todas as CPTs são armazenadas no banco de dados. A Tabela 1 mostra a CPT para a variável *K<sub>DFA</sub>*.

Tabela 1. CPT para a variável *K<sub>DFA</sub>*

<i>K<sub>DFA</sub></i>	<i>K<sub>def</sub> = 0</i>	<i>K<sub>def</sub> = 1</i>
<i>K<sub>DFA</sub> = 0</i>	0.65	0.25
<i>K<sub>DFA</sub> = 1</i>	0.35	0.75

## 2.4 Modelo Pedagógico

É o responsável por interpretar as probabilidades do modelo de aluno para aplicar estratégias personalizadas de acordo com o conhecimento do aluno [9]. O modelo pedagógico decide: (a) o que o aluno deve aprender, (b) quando intervir, e (c) o feedback apresentado.

Na geração de explicações: o sistema prioriza o feedback baseado na severidade (chave *severity* da Figura 1) do erro diagnosticado pelo motor de inferência. Por exemplo: um erro que impede o funcionamento do autômato (*severity: error*) é corrigido antes de um erro de otimização, já que facilita a interpretação da resolução do problema para o aluno.

O modelo pedagógico recebe o código de diagnóstico do motor de inferência e usa a base de conhecimento para recuperar a mensagem relevante, que atua como fonte das explicações locais. Isso garante que a explicação seja transparente. A Figura 3 apresenta um diagrama de sequência que representa o cenário de diagnóstico e geração de feedback explicativo.

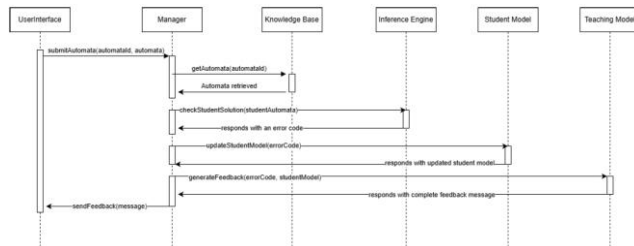


Figura 3: Cenário de diagnóstico e geração de feedback explicativo.

A adaptação de conteúdo: guiada pelo perfil do aluno, conforme estimado pela RB no modelo de aluno. O sistema define um limiar de proficiência (por exemplo, 80% de probabilidade de domínio) para cada habilidade. O aluno só consegue avançar para um novo tópico se a probabilidade de domínio da habilidade pré-requisito for maior ou igual a esse limiar. Dessa forma, o modelo pedagógico usa a probabilidade para tomar duas decisões principais de adaptação: (i) Reforço, se a probabilidade de uma habilidade fundamental (como a definição formal da teoria dos autômatos) cair abaixo do limiar definido, o sistema interrompe a progressão e seleciona exercícios de reforço focados especificamente naquela habilidade; (ii) Progressão, se o aluno atingir ou exceder o limiar definido em um tópico, o tutor libera o acesso ao próximo conceito.

## 2.5 Interface do Usuário

O aluno interage com o tutor através de um editor visual interativo, projetado para simplificar a construção de grafos de estados. Os controles do editor visual permitem: (i) a construção de grafos (Fig. 4), onde o aluno pode adicionar estados, definir funções de transição entre estados, excluir elementos e alterar estados para ser inicial ou final; (ii) conversão, na qual o desenho do grafo é convertido em um objeto JSON que representa a solução do aluno, permitindo o diagnóstico simbólico pelo motor de inferência.

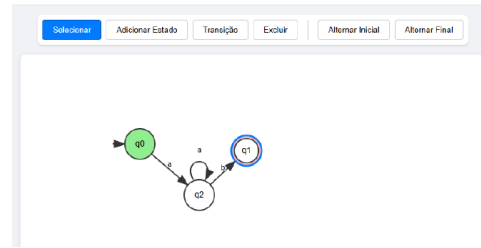


Figura 4: Canvas para construção de autômato.

## 3 Considerações Preliminares

O sistema está ainda em sua fase de construção dos módulos. A parte referente a XAI está sendo implementada considerando técnicas como LIME, SHAP, e camadas de atenção visualizáveis serão adaptadas para identificar quais variáveis, como por exemplo tempo de resposta, histórico de erros, tipos de tarefa, influenciaram as decisões do tutor.

O STI será testado em alunos de graduação de um curso de Ciência da Computação.

## REFERÊNCIAS

- [1] Ali Alkhatlan and Jugal Kalita, 2019. T Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments. *International Journal of Computer Applications* 181. 1-20. DOI: <https://doi.org/10.5120/ijca2019918451>.
- [2] Brent Morgan, Marie Hogan, Andrew J. Hampton, Anne Lippert, Arthur C. Graesser. (2020). The need for personalized learning and the potential of intelligent tutoring systems. In P. Van Meter, A. List, D. Lombardi, & P. Kendeou (Eds.), *Handbook of learning from multiple representations and perspectives* (pp. 495–512). Routledge/Taylor & Francis Group. DOI: <https://doi.org/10.4324/9780429443961-32>
- [3] Kenneth Holstein, Vicent Alevén, Nikol Rummel. 2020. A Conceptual Framework for Human–AI Hybrid Adaptivity in Education. In: Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds) *Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science*, vol 12163. Springer, Cham. DOI: [https://doi.org/10.1007/978-3-030-52237-7\\_20](https://doi.org/10.1007/978-3-030-52237-7_20).
- [4] Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, Peter M. Atkinson. 2021. Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery*, v. 11, n. 5, p. e1424, 2021. DOI: <https://doi.org/10.1002/widm.1424>.
- [5] Hyacinth S. Nwana. 1990. Intelligent tutoring systems: An overview. *Artificial Intelligence Review*, Springer, v. 4, n. 4, p. 251-277, 1990. systems. In: FARMANBAR, M. et al. (Ed.). *Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications*. Singapore: Springer Nature Singapore, 2024. p. 59–70. ISBN 978-981-99-9836-4.
- [6] William J. Clancey and Robert R. Hoffman. 2021. Methods and standards for research on explainable artificial intelligence: Lessons from intelligent tutoring systems. *Applied AI Letters* 2, 4 (2021), e53. DOI: <https://doi.org/10.1002/aii2.53>.
- [7] Jaxton Winder, Erik Falor, Seth Poulsen, and John Edwards. 2024. The Shell Tutor: An Intelligent Tutoring System For The UNIX Command Shell And Git. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (Milan, Italy) (ITICSE 2024). Association for Computing Machinery, New York, NY, USA, 548–554. DOI:10.1145/3649217.3653631.
- [8] Hugo Gamboa and Ana L. N. Fred. 2001. Designing Intelligent Tutoring Systems: A Bayesian Approach. In *International Conference on Enterprise Information Systems*. <https://api.semanticscholar.org/CorpusID:5957011>.
- [9] Kostas Karpouzis. 2024. Explainable AI for Intelligent Tutoring Systems. In *Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications*, Mina Farmanbar, Maria Tzamtzi, Ajit Kumar Verma, and Antorweep Chakravorty (Eds.). Springer Nature Singapore, Singapore, 59–70..