The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

TRACE-CS: A Synergistic Approach to Explainable Course Scheduling Using LLMs and Logic

Stylianos Loukas Vasileiou, William Yeoh

Washington University in St. Louis
{v.stylianos, wyeoh}@wustl.edu

Abstract

We present TRACE-CS, a novel hybrid system that combines symbolic reasoning with large language models (LLMs) to address contrastive queries in scheduling problems. TRACE-CS leverages SAT solving techniques to encode scheduling constraints and generate explanations for user queries, while utilizing an LLM to process the user queries into logical clauses as well as refine the explanations generated by the symbolic solver to natural language sentences. By integrating these components, our approach demonstrates the potential of combining symbolic methods with LLMs to create explainable AI agents with correctness guarantees.

Introduction

Scheduling systems, which allocate finite resources to multiple agents over time, are ubiquitous in real-world systems, from personnel shift assignments (Van den Bergh et al. 2013) to Mars rover activities (Chi, Chien, and Agrawal 2020). Beyond generating valid and optimal schedules, it is crucial to ensure that both the schedule and the decision-making process are *explainable* to human users. *Explainable scheduling*, therefore, is essential for understanding scheduling decisions, rectifying issues, and providing explanations for specific decisions or schedule generation failures. Most of the work in this space have relied on symbolic, logical methods that generate valid and sound explanations.

At the other end of the spectrum, the emergence of large language models (LLMs) has marked a significant milestone in AI. While LLMs excel at generating coherent and contextually relevant text (Brown et al. 2020), their reliance on statistical inference leads to challenges in maintaining logical consistency and accuracy in reasoning and planning tasks (McCoy et al. 2023; Valmeekam et al. 2023). This limitation is particularly apparent when explanations need to be both linguistically coherent and logically sound. In contrast, symbolic, logical methods provide a robust medium for reasoning and planning due to their ability to perform valid and sound inference. This realization offers an opportunity to combine the strengths of both LLMs and symbolic methods, creating synergistic systems that ensure decisions are not only provably correct and robust, but also communicated in a user-friendly manner.

In this paper, we present **TRACE-CS** (*Trustworthy ReAsoning for Contrastive Explanations in Course Scheduling Problems*), a synergistic system that combines symbolic reasoning with the natural language capabilities of LLMs for generating explanations in course scheduling problems. Particularly, TRACE-CS generates natural language explanations for contrastive user queries (e.g., "Why course X instead of course Y?") by leveraging a state-of-the-art symbolic explainer (Vasileiou, Previti, and Yeoh 2021) together with an LLM-powered user interface for natural language interactions, thus ensuring that the explanations are provably trustworthy as well as communicated to users in natural language. Our demonstration showcases how TRACE-CS handles real-world course scheduling scenarios, illustrating its potential for integrating LLMs into explainable scheduling systems and enabling more intuitive and effective human-AI interaction in scheduling domains.

Related Work. Explainable scheduling research has predominantly relied on logical symbolic methods (Cyras et al. 2019; Agrawal, Yelamanchili, and Chien 2020; Bertolucci et al. 2021; Pozanco et al. 2022; Powell and Riccardi 2022; Vasileiou et al. 2022; Vasileiou, Xu, and Yeoh 2023; Zehtabi et al. 2024). While grounded in sound inference procedures, these approaches often produce explanations that are difficult to communicate to users due to their logic-based nature. Attempts to mitigate this limitation have used templates mapping logical explanations to pre-specified natural language sentences (Pozanco et al. 2022; Vasileiou, Xu, and Yeoh 2023) or visualization interfaces (Čyras, Lee, and Letsios 2021; Kumar et al. 2022; Powell and Riccardi 2022). Concurrently, LLMs have revolutionized natural language processing and found applications across diverse domains, including planning (Kambhampati et al. 2024), code generation (Roziere et al. 2023), and medical applications (Zhou et al. 2023). However, the integration of LLMs with symbolic explainable scheduling systems remains largely unexplored. Our work, TRACE-CS, represents the first attempt to address this gap by presenting a novel hybrid system that synergistically combines a symbolic explainable scheduling module with an LLM module.

TRACE-CS Overview

An overview of TRACE-CS is shown in Figure 1.

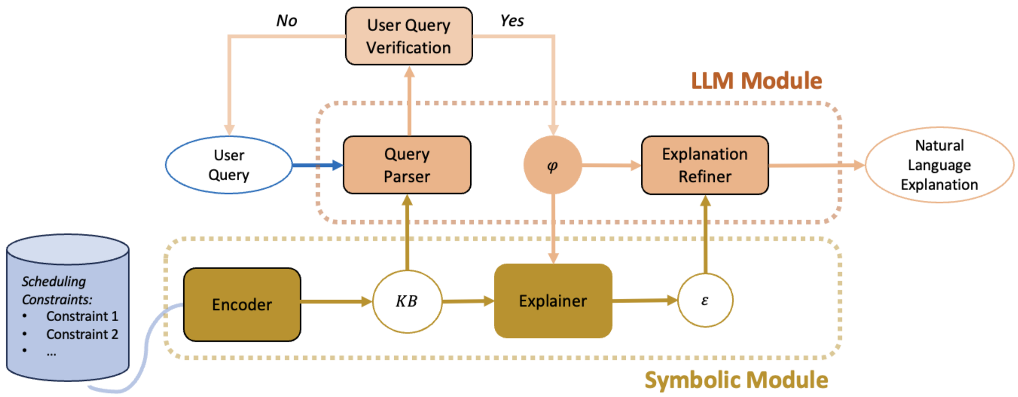

Figure 1: The TRACE-CS workflow.

Symbolic Module. The Symbolic Module forms the core of TRACE-CS, handling the scheduling logic and explanation generation:

- **Encoder:** Encodes specific scheduling constraints into logical formulae, creating a knowledge base KB that represents the scheduling problem. This includes encoding course prerequisites, credit requirements, semester constraints, etc. Importantly, each formula has an associated label attached to it, describing in English the type of scheduling constraint it encodes.
- **Explainer:** Utilizes the state-of-the-art symbolic explanation generation solver by Vasileiou, Previti, and Yeoh (2021). It takes as input the knowledge base KB from the Encoder and a user contrastive query φ (processed by the LLM module), and generates contrastive explanations. The output is a set of logical formulae along with their corresponding labels.

LLM Module. The LLM Module serves as the interface between the user and the Symbolic Module, handling natural language processing tasks:

- **Query Parser:** Interprets a user's contrastive query in natural language and converts it into a symbolic representation φ compatible with the encoded knowledge base KB. This process employs in-context learning to ensure accurate interpretation. However, recent work by Karia et al. (2024) highlights the potential limitations of LLMs in formal interpretation tasks, underscoring the importance of human verification in our system. Thus, TRACE-CS includes a step for user verification of the extracted query information before proceeding to explanation generation.
- **Explanation Refiner:** Takes the symbolic explanation ε from the Explainer and translates it into natural language sentences. This translation process uses in-context learning, utilizing the labels attached to each formula in ε to ensure accurate and coherent explanations.

Figure 1 shows the workflow of TRACE-CS: (1) The user submits a contrastive query in natural language; (2) The Query Parser extracts the information from the query and converts it into a symbolic representation φ consistent with the knowledge base KB created by the Encoder; (3) The user verifies if the extracted query information corresponds to the original query, and proceeds to the next step if it is; (4) The Explainer generates a symbolic explanation ϵ for φ with respect to KB; (5) The Explanation Refiner converts ϵ into natural language and outputs it to the user.

	TRACE-CS	Zero-shot LLM	Few-shot LLM
Explanation Correctness	100%	44%	49%

Table 1: Results from 100 queries comparing TRACE-CS with a zero-shot and a few-shot LLM-only approach.

Proof-of-Concept: Academic Course Schedules

We implemented TRACE-CS in Python as a proof-of-concept for scheduling courses for an undergraduate computer science student across the eight academic semesters at our university. To create a comprehensive and realistic scheduling environment, we scraped the computer science course catalog and degree requirements from the university's official website. The Symbolic Module was implemented using the PySAT library (Ignatiev, Morgado, and Marques-Silva 2018), while the LLM Module uses the GPT-4 model (OpenAI 2023).[1]

To evaluate the effectiveness of TRACE-CS, we conducted a comparative experimental study against a zero-shot and few-shot GPT-4 model. Specifically, we generated 10 distinct schedules and created 10 queries for each schedule, totaling 100 schedule-query pairs. Our evaluation metric was explanation correctness with respect to the scheduling constraints, which was performed manually by the authors. Table 1 shows the results. We observe that TRACE-CS significantly outperformed both zero-shot and few-shot LLM approaches in terms of explanation correctness, achieving 100% accuracy compared to 44% and 49%, respectively. These results underscore the effectiveness of a hybrid approach in providing accurate explanations for course scheduling scenarios.

[1]Code repository: https://github.com/YODA-Lab/TRACE-CS.

Acknowledgements

This research is partially supported by the National Science Foundation under award 2232055 and by J.P. Morgan AI Research. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the United States government.

References

Agrawal, J.; Yelamanchili, A.; and Chien, S. 2020. Using explainable scheduling for the Mars 2020 rover mission. *arXiv preprint arXiv:2011.08733*.

Bertolucci, R.; Dodaro, C.; Galatà, G.; Maratea, M.; Porro, I.; and Ricca, F. 2021. Explaining ASP-based Operating Room Schedules. In *Proceedings of the Workshop on Explainable Logic-Based Knowledge Representation*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 1877–1901.

Chi, W.; Chien, S.; and Agrawal, J. 2020. Scheduling with complex consumptive resources for a planetary rover. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 348–356.

Čyras, K.; Lee, M.; and Letsios, D. 2021. Schedule explainer: An argumentation-supported tool for interactive explanations in makespan scheduling. In *Proceedings of the International Workshop on Explainable and Transparent AI and Multi-Agent Systems*, 243–259.

Cyras, K.; Letsios, D.; Misener, R.; and Toni, F. 2019. Argumentation for Explainable Scheduling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2752–2759.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2018. PySAT: A Python Toolkit for Prototyping with SAT Oracles. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing*, 428–437.

Kambhampati, S.; Valmeekam, K.; Guan, L.; Verma, M.; Stechly, K.; Bhambri, S.; Saldyt, L. P.; and B Murthy, A. 2024. Position: LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. In *Proceedings of the International Conference on Machine Learning*, 22895–22907.

Karia, R.; Dobhal, D.; Bramblett, D.; Verma, P.; and Srivastava, S. 2024. $\forall$uto$\exists$val: Autonomous Assessment of LLMs in Formal Synthesis and Interpretation Tasks. *arXiv preprint arXiv:2403.18327*.

Kumar, A.; Vasileiou, S. L.; Bancilhon, M.; Ottley, A.; and Yeoh, W. 2022. VizXP: A visualization framework for conveying explanations to users in model reconciliation problems. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 701–709.

McCoy, R. T.; Yao, S.; Friedman, D.; Hardy, M.; and Griffiths, T. L. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Powell, C.; and Riccardi, A. 2022. Abstract argumentation for explainable satellite scheduling. In *Proceedings of the International Conference on Data Science and Advanced Analytics*, 1–10.

Pozanco, A.; Mosca, F.; Zehtabi, P.; Magazzeni, D.; and Kraus, S. 2022. Explaining Preference-driven Schedules: the EXPRES Framework. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 710–718.

Roziere, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Remez, T.; Rapin, J.; et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Valmeekam, K.; Marquez, M.; Sreedharan, S.; and Kambhampati, S. 2023. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36: 75993–76005.

Van den Bergh, J.; Beliën, J.; De Bruecker, P.; Demeulemeester, E.; and De Boeck, L. 2013. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3): 367–385.

Vasileiou, S. L.; Previti, A.; and Yeoh, W. 2021. On exploiting hitting sets for model reconciliation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6514–6521.

Vasileiou, S. L.; Xu, B.; and Yeoh, W. 2023. A Logic-based Framework for Explainable Agent Scheduling Problems. In *Proceedings of the European Conference on Artificial Intelligence*, 2402–2410.

Vasileiou, S. L.; Yeoh, W.; Son, T. C.; Kumar, A.; Cashmore, M.; and Magazzeni, D. 2022. A Logic-Based Explanation Generation Framework for Classical and Hybrid Planning Problems. *Journal of Artificial Intelligence Research*, 73: 1473–1534.

Zehtabi, P.; Pozanco, A.; Bolch, A.; Borrajo, D.; and Kraus, S. 2024. Contrastive Explanations of Centralized Multiagent Optimization Solutions. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 671–679.

Zhou, H.; Gu, B.; Zou, X.; Li, Y.; Chen, S. S.; Zhou, P.; Liu, J.; Hua, Y.; Mao, C.; Wu, X.; et al. 2023. A survey of large language models in medicine: Progress, application, and challenge. *arXiv preprint arXiv:2311.05112*.