Towards Large Language Model Architectures for Knowledge Acquisition and Strategy Synthesis

Paolo Giorgini, Andrea Mazzullo, Marco Robol and Marco Roveri

University of Trento

Abstract

To address the bottlenecks of knowledge acquisition and strategy synthesis, in the development of autonomous AI agents capable of reasoning and planning about dynamic environments, we propose an architecture that combines large language model (LLM) functionalities with formal verification modules. Concerning knowledge acquisition, we focus on the problem of learning description logic concepts to separate data instances, whereas, in a process mining setting, we propose to leverage LLMs to extract linear temporal logic specifications from event logs. Finally, in a strategy synthesis context, we illustrate how LLMs can be employed to address realisability problems in linear temporal logic on finite traces.

Keywords

Large Language Models, Knowledge Acquisition, Strategy Synthesis, Learning from Examples, Description Logics, Linear Temporal Logic

1. Introduction

The combination of machine learning methods, based on stochastic black-box architectures, with logic-based techniques, symbolic and explainable in nature, is considered of critical importance for developing AI-based autonomous agents that can evolve strategies and plans, or reason about their surroundings in the presence of newly acquired information [1, 2, 3, 4]. In this direction, the integration of *large language models* (LLMs) with knowledge representation features is receiving significant attention in the literature [5, 6, 7, 8, 9]. One approach aims at combining LLMs, which are known to perform well in natural language generation tasks, with integrated *reasoning* modules, used to address and solve formal problems in a provably correct way [10, 11, 12]. Another area that is recently gaining traction is concerned with the enhancement of LLMs with *planning* capabilities, to perform explainable scheduling tasks [13, 14, 15, 16, 17, 18].

When dealing with knowledge-intensive structured domains, or in the presence of data evolving over time in dynamic environments, autonomous AI agents face the following important challenges: (1) *knowledge acquisition*, that is, the task of extracting structured information from raw data in a given domain, in turn allowing for domain-specific or time-dependent conceptual modelling and reasoning; (2) *strategy synthesis*, i.e., the task of devising sequences of actions, possibly in response to environmental conditions or other agents' choices, in order to reach a

OVERLAY 2023: 5th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, November 7, 2023, Rome, Italy

paolo.giorgini@unitn.it (P. Giorgini); andrea.mazzullo@unitn.it (A. Mazzullo); marco.robol@unitn.it (M. Robol); marco.roveri@unitn.it (M. Roveri)

given goal, for automatic programming and planning purposes. From a foundational viewpoint, two formalisms can be arguably considered sufficiently expressive for these problems: *description logics* (DLs), a well-known family of knowledge representation languages devised to be computationally well-behaved [19, 20]; and *linear temporal logic* (LTL), which extends classical propositional logic with time modalities interpreted on linear structures [21, 22], and is widely applied in computer science and AI [23, 24, 25, 26, 27, 28].

By relying on these formalisms, we propose an integrative AI architecture based on LLMs to address both knowledge acquisition and strategy synthesis tasks. We first illustrate, in Section 2, our framework within the knowledge extraction context. This approach is related to: *ontology* and *concept learning* or *separability* in DLs [29, 30, 31, 32, 33, 34, 35], as well as *reverse engineering of formulas* in LTL [36, 37, 38, 39]; problems in *inductive* (and *abductive*) *reasoning* [40, 41]; the model of *active learning with membership queries* in machine learning [42, 43, 44, 45]; and the *query-by-example* approach from database theory [46]. In Section 3, we present our architecture for the strategy synthesis setting. This shares connections with *LLM-based planning* approaches [47, 48], and to *counterexample-guided inductive synthesis* [49] in the field of automatic programming. We briefly discuss in Section 4 future research directions.

2. LLM-Driven Knowledge Acquisition

Concept Learning in DLs. In DLs, *concept learning* is the task of automatically generating, from a set of examples, a concept description that correctly represents them (see also [35] and references therein). Related to this question, for a given DL language, the following *concept separability problem* has been investigated (for space limitations, we present here a simplified version, and assume familiarity with DL concepts; see [35] for detailed preliminaries). First, given a DL $\mathcal{L}$, let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be an $\mathcal{L}$ *knowledge base*, containing both the background axioms in the *ontology* $\mathcal{O}$, as well as (ground) facts stored in the *dataset* $\mathcal{D}$. The *positive individuals*, P, and *negative individuals*, N, are subsets of $\mathsf{ind}(\mathcal{D})$, i.e., of the *individuals* occurring in $\mathcal{D}$. The (*weak*) *concept separability problem* asks the following: given $\mathcal{K}$, P, and N, is there an $\mathcal{L}$ concept C that *separates* the positive from the negative examples? That is: $\mathcal{K} \models C(e^+)$, for every $e^+ \in P$; and $\mathcal{K} \not\models C(e^-)$, for every $e^- \in N$. (A *strong* version of the separability problem requires that $\mathcal{K} \models \neg C(e^-)$, for every $e^- \in N$: we omit it for space reasons). This problem has been investigated both as a decision problem, from a theoretical perspective [31, 35], and addressed by concrete tool implementations for separating concept generation [50, 51, 52, 53]. Towards LLM-assisted generation of separating concepts, we propose the following architecture, illustrated in the box below and summarised in Fig. 1 (left).

```
Input $(\mathcal{K}, P, N)$.
Output Separating $\mathcal{L}$-concept $C$ for $(\mathcal{K}, P, N)$, if it exists.
Procedure
• Prompt input $(\mathcal{K}, P, N)$ to the LLM-based DL concept generation module.
• While no separating concept is found, repeat the following steps:
  – ask the LLM module to candidate a separating $\mathcal{L}$ concept $C$;
  – check with a DL $\mathcal{L}$ reasoner if $\mathcal{K} \models C(e^+)$, for all $e^+ \in P$, and $\mathcal{K} \not\models C(e^-)$, for all $e^- \in N$:
      ⋆ if a counterexample $\bar{e}$ is found, i.e., $\mathcal{K} \not\models C(\bar{e})$, with $\bar{e} \in P$, or $\mathcal{K} \models C(\bar{e})$, with $\bar{e} \in N$, pinpoint $\bar{e}$ to the LLM module;
      ⋆ otherwise, return $C$ as separating concept.
```

Process Mining in LTL. A challenge in *process mining* [54, 55] consists in the identification, from sets of event logs, of a formal specification that captures the underlying process structure. In an LTL setting (we assume familiarity with its basic notions), such process mining task can be connected to the problem of finding a temporal specification, in the form of an LTL formula, that is capable of discerning a set of "positive" logs, i.e., examples of successful processes, from a set of "negative" ones, instantiating instead undesired dynamics. This problem has also received attention in the literature from a theoretical standpoint [37]. Similarly to the concept separability problem presented above, the task here is, given a set of propositional letters Σ, a set $P \subseteq (2^{\Sigma})^{\omega}$ of *positive traces*, and a set $N \subseteq (2^{\Sigma})^{\omega}$ of *negative traces*, to determine a corresponding *LTL process* Σ*-formula* φ that: $\sigma^{+} \models \varphi$, for all $\sigma^{+} \in P$; and $\sigma^{-} \not\models \varphi$, for all $\sigma^{-} \in N$. An analogous problem would consider LTL on *finite traces* (often denoted by LTL^{f}), which is interpreted on finite sequences in $(2^{\Sigma})^{+}$. Our LLM-driven architecture to address such a process formula generation is described in the following box, and illustrated in Fig. 1 (right).

Input (Σ, P, N).
Output LTL *process* Σ*-formula* φ for (Σ, P, N), if it exists.
Procedure

- Prompt input (Σ, P, N) to the LLM-based LTL formula generation module
- While no LTL process Σ-formula is found, repeat the following steps:
  - ask the LLM module to candidate an LTL process Σ-formula φ;
  - check with a model-checking tool whether: $\sigma^{+} \models \varphi$, for all $\sigma^{+} \in P$; and $\sigma^{-} \not\models \varphi$, for all $\sigma^{-} \in N$:
    - if a *counterexample* $\bar{\sigma}$ is found, i.e., $\bar{\sigma} \not\models \varphi$, with $\bar{\sigma} \in P$, or $\bar{\sigma} \models \varphi$, with $\bar{\sigma} \in N$, pinpoint $\bar{\sigma}$ to the LLM module;
    - otherwise, return φ as LTL process Σ-formula.

The procedures sketched above are not guaranteed to terminate, as the LLM module might be incapable of finding successful candidates, and it is possible that no separating concept [35] or formula [56] exist in the formalisms. Further analysis on soundness, completeness, termination, and explainability issues, involving heuristic search techniques and reinforcement learning, or controlled loops with time-outs and generation of failed attempt explanations, is left as future work. For another recent approach integrating LLMs and declarative process mining, cf. [57].

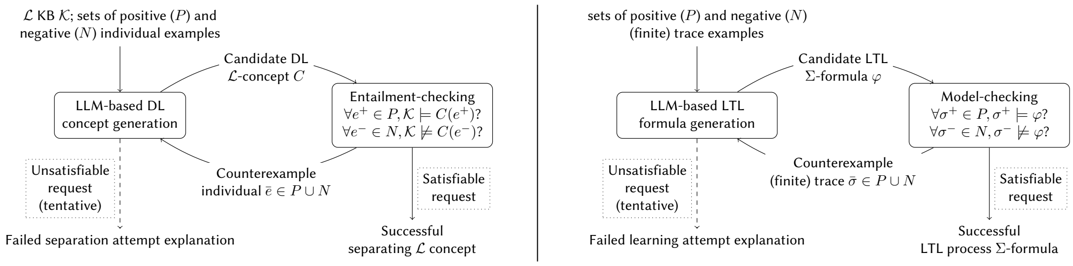
$\mathcal{L}$ KB $\mathcal{K}$; sets of positive (P) and negative (N) individual examples
Candidate DL $\mathcal{L}$-concept C
LLM-based DL concept generation
Entailment-checking
$\forall e^{+} \in P, \mathcal{K} \models C(e^{+})$?
$\forall e^{-} \in N, \mathcal{K} \not\models C(e^{-})$?
Unsatisfiable request (tentative)
Counterexample individual $\bar{e} \in P \cup N$
Satisfiable request
Failed separation attempt explanation
Successful separating $\mathcal{L}$ concept
sets of positive (P) and negative (N) (finite) trace examples
Candidate LTL Σ-formula φ
LLM-based LTL formula generation
Model-checking
$\forall \sigma^{+} \in P, \sigma^{+} \models \varphi$?
$\forall \sigma^{-} \in N, \sigma^{-} \not\models \varphi$?
Unsatisfiable request (tentative)
Counterexample (finite) trace $\bar{\sigma} \in P \cup N$
Satisfiable request
Failed learning attempt explanation
Successful LTL process Σ-formula

Figure 1: Architecture of LLM-driven DL concept (left) and LTL formula (right) learning.

3. LLM-Driven Strategy Synthesis

With *strategy synthesis*, we refer to the problem of identifying a user strategy providing a sequence of actions to reach a goal, or of operations capable of satisfying a given specification, possibly in response to uncontrollable choices of other agents or environments. As such, it can encompass problems both in the fields of *planning*, as well as in *automatic programming*.

For *planning* purposes, in a purely LTL setting, the so-called *realisability* and *synthesis problems* have attracted considerable attention [58, 59, 60], particularly in the *finite trace* case of LTL^f. Here, we slightly modify the standard setting [58, 61], and adopt the following definitions. Let φ be an LTL^f formula, with its proposition letters from Σ partitioned in sets of *controllable* (C) and *Environment* (E) ones. A *strategy* for φ is a function $s : (2^E)^+ \to 2^C$ such that, for any finite sequence $\mathsf{E} = (\mathsf{E}_0, \ldots, \mathsf{E}_n) \in (2^E)^+$ of *Environment choices*, it determines a *Controller choice* $s(\mathsf{E}) \in 2^C$. Moreover, let $A \subseteq (2^E)^\omega$ be a *finite* set of *admissible* infinite sequences of Environment choices. A strategy is *winning* if, for any admissible $\mathsf{E} \in A$, there exists $k \in \mathbb{N}$ such that $\mathsf{react}(s, \mathsf{E})_{[0,k]} \models \varphi$, where $\mathsf{react}(s, \mathsf{E}) = (\mathsf{E}_0 \cup s((\mathsf{E}_0)), \mathsf{E}_0 \cup s((\mathsf{E}_0, \mathsf{E}_1)), \ldots)$ is the trace obtained by reacting to E according to s, and $\mathsf{react}(s, \mathsf{E})_{[0,k]}$ denotes its *prefix* from 0 to k. An LTL^f formula φ is *realisable* with respect to (C, E, A) if there exists a winning strategy. The *realisability* problem asks whether φ is realisable with respect to (C, E, A), while the *synthesis* problem requires to provide such a winning strategy if it exists.

In the box below and in Fig. 2 we illustrate our architecture, aiming at synthesising LTL^f formulas via combined interactions between an LLM-based module and a model-checking tool.

Input (φ, C, E, A, N), with N (possibly empty) set of traces not satisfying φ.
Output Winning strategy for φ under A, if it exists.
Procedure

- While no winning strategy for φ is found, repeat the following steps:
  - prompt input (φ, C, E, N) to the LLM-based LTL formula synthesis module;
  - ask the LLM module to candidate a *strategy* for φ;
  - check with a model-checking tool whether, for all $\mathsf{E} \in A$, $\mathsf{react}(s, \mathsf{E}) \models \varphi$:
    - if a counterexample $\bar{\mathsf{E}}$ is found, i.e., $\mathsf{react}(s, \bar{\mathsf{E}}) \not\models \varphi$, assign $N \leftarrow N \cup \{\mathsf{react}(s, \bar{\mathsf{E}})\}$;
    - otherwise, return s as winning strategy.

Observe that the set of admissible sequences of Environment choices is a restriction imposed to limit the search space in the formal verification module. Moreover, the LLM module should provide a finite presentation of the Controller strategy, by means e.g. of a finite-state transducer.

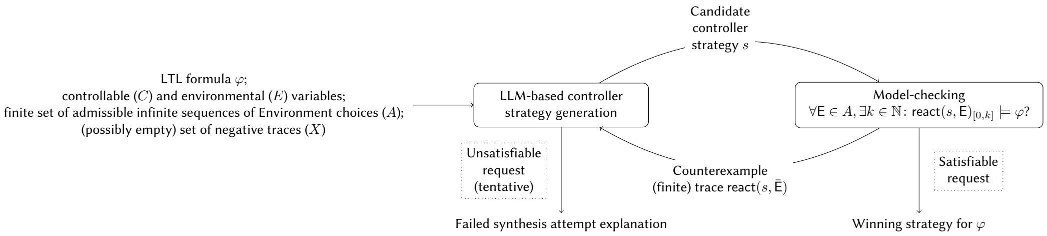
Candidate controller strategy s
LTL formula φ;
controllable (C) and environmental (E) variables;
finite set of admissible infinite sequences of Environment choices (A);
(possibly empty) set of negative traces (X)
LLM-based controller strategy generation
Model-checking
$\forall \mathsf{E} \in A, \exists k \in \mathbb{N}$: $\mathsf{react}(s, \mathsf{E})_{[0,k]} \models \varphi$?
Unsatisfiable request (tentative)
Counterexample (finite) trace $\mathsf{react}(s, \bar{\mathsf{E}})$
Satisfiable request
Failed synthesis attempt explanation
Winning strategy for φ

Figure 2: Architecture of LLM-driven LTL strategy synthesis.

4. Discussion and Future Work

We proposed architectures for the development of AI agents capable of performing complex knowledge acquisition and strategy synthesis tasks, combining the generative capabilities of LLMs with logic-based formalisms and techniques. As future work, we plan to both improve the definition and the understanding of the formal properties of the proposed architectures, as well as to develop dedicated tools based on state-of-the-art LLMs, comparing their performances over suitable benchmarks with other systems from the literature.

Acknowledgments

M. Robol and M. Roveri are partially supported by the project MUR PRIN 2020 - RIPER - Resilient AI-Based Self-Programming and Strategic Reasoning - CUP E63C22000400001. P. Giorgini, A. Mazzullo and M. Roveri are partially supported by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by NextGenerationEU.

References

[1] B. G. Humm, P. Archer, H. Bense, C. Bernier, C. Goetz, T. Hoppe, F. Schumann, M. Siegel, R. Wenning, A. Zender, New directions for applied knowledge-based AI and machine learning, Inform. Spektrum 46 (2023) 65–78.

[2] J. P. Delgrande, B. Glimm, T. Meyer, M. Truszczynski, M. S. Teixeira, F. Wolter, Current and future challenges in knowledge representation and reasoning (dagstuhl seminar 22282), Dagstuhl Reports 12 (2022) 62–79.

[3] K. Hamilton, A. Nayak, B. Bozic, L. Longo, Is neuro-symbolic AI meeting its promise in natural language processing? A structured review, CoRR abs/2202.12205 (2022). arXiv:2202.12205.

[4] A. P. Sheth, K. Roy, M. Gaur, Neurosymbolic artificial intelligence (why, what, and how), IEEE Intell. Syst. 38 (2023) 56–62.

[5] H. Liu, R. Ning, Z. Teng, J. Liu, Q. Zhou, Y. Zhang, Evaluating the logical reasoning ability of chatgpt and GPT-4, CoRR abs/2304.03439 (2023). arXiv:2304.03439.

[6] M. Trajanoska, R. Stojanov, D. Trajanov, Enhancing knowledge graph construction using large language models, CoRR abs/2305.04676 (2023). arXiv:2305.04676.

[7] F. Moiseev, Z. Dong, E. Alfonseca, M. Jaggi, SKILL: structured knowledge infusion for large language models, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022, Association for Computational Linguistics, 2022, pp. 1581–1588.

[8] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, CoRR abs/2306.08302 (2023). arXiv:2306.08302.

[9] H. Zhang, L. H. Li, T. Meng, K. Chang, G. V. den Broeck, On the paradox of learning to reason from data, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, ijcai.org, 2023, pp. 3365–3373.

[10] L. Pan, A. Albalak, X. Wang, W. Y. Wang, Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning, CoRR abs/2305.12295 (2023). arXiv:2305.12295.

[11] A. Creswell, M. Shanahan, I. Higgins, Selection-inference: Exploiting large language models for interpretable logical reasoning, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net, 2023.

[12] J. Huang, K. C. Chang, Towards reasoning in large language models: A survey, in: Findings

of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 1049–1065.

[13] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W. Chao, Y. Su, Llm-planner: Few-shot grounded planning for embodied agents with large language models, CoRR abs/2212.04088 (2022). arXiv:2212.04088.

[14] K. Valmeekam, A. O. Hernandez, S. Sreedharan, S. Kambhampati, Large language models still can't plan (A benchmark for llms on planning and reasoning about change), CoRR abs/2206.10498 (2022). arXiv:2206.10498.

[15] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, A. Garg, Progprompt: Generating situated robot task plans using large language models, in: IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023, IEEE, 2023, pp. 11523–11530.

[16] V. Pallagani, B. Muppasani, K. Murugesan, F. Rossi, B. Srivastava, L. Horesh, F. Fabiano, A. Loreggia, Understanding the capabilities of large language models for automated planning, CoRR abs/2305.16151 (2023). arXiv:2305.16151.

[17] K. Valmeekam, S. Sreedharan, M. Marquez, A. O. Hernandez, S. Kambhampati, On the planning abilities of large language models (A critical investigation with a proposed benchmark), CoRR abs/2302.06706 (2023). arXiv:2302.06706.

[18] W. Huang, P. Abbeel, D. Pathak, I. Mordatch, Language models as zero-shot planners: Extracting actionable knowledge for embodied agents, in: International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 9118–9147.

[19] The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.

[20] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[21] A. Pnueli, The temporal logic of programs, in: 18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977, IEEE Computer Society, 1977, pp. 46–57.

[22] Z. Manna, A. Pnueli, The temporal logic of reactive and concurrent systems - specification, Springer, 1992.

[23] Z. Manna, A. Pnueli, Verification of concurrent programs: Temporal proof principles, in: Logics of Programs, Workshop, Yorktown Heights, New York, USA, May 1981, volume 131 of *Lecture Notes in Computer Science*, Springer, 1981, pp. 200–252.

[24] E. M. Clarke, E. A. Emerson, A. P. Sistla, Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach, in: Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 1983, ACM Press, 1983, pp. 117–126.

[25] Z. Manna, P. Wolper, Synthesis of communicating processes from temporal logic specifications, ACM Trans. Program. Lang. Syst. 6 (1984) 68–93.

[26] C. Baier, J. Katoen, Principles of model checking, MIT Press, 2008.

[27] S. Goedertier, J. Vanthienen, F. Caron, Declarative business process modelling: principles and modelling languages, Enterp. Inf. Syst. 9 (2015) 161–185.

[28] Handbook of Temporal Reasoning in Artificial Intelligence, volume 1 of *Foundations of*

Artificial Intelligence, Elsevier, 2005.

[29] B. Konev, C. Lutz, A. Ozaki, F. Wolter, Exact learning of lightweight description logic ontologies, J. Mach. Learn. Res. 18 (2017) 201:1–201:63.

[30] V. Gutiérrez-Basulto, J. C. Jung, L. Sabellek, Reverse engineering queries in ontology-enriched systems: The case of expressive horn description logic ontologies, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 1847–1853.

[31] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 1682–1688.

[32] A. Ozaki, C. Persia, A. Mazzullo, Learning query inseparable $\mathcal{ELH}$ ontologies, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 2959–2966.

[33] A. Ozaki, Learning description logic ontologies: Five approaches. where do they stand?, Künstliche Intell. 34 (2020) 317–327.

[34] M. Funk, J. C. Jung, C. Lutz, Actively learning concepts and conjunctive queries under elr-ontologies, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 1887–1893.

[35] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Logical separability of labeled data examples under ontologies, Artif. Intell. 313 (2022) 103785.

[36] D. Neider, I. Gavran, Learning linear temporal properties, in: 2018 Formal Methods in Computer Aided Design, FMCAD 2018, Austin, TX, USA, October 30 - November 2, 2018, IEEE, 2018, pp. 1–10.

[37] M. Fortin, B. Konev, V. Ryzhikov, Y. Savateev, F. Wolter, M. Zakharyaschev, Unique characterisability and learnability of temporal instance queries, in: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel. July 31 - August 5, 2022, 2022.

[38] J. C. Jung, V. Ryzhikov, F. Wolter, M. Zakharyaschev, Temporalising unique characterisability and learnability of ontology-mediated queries, CoRR abs/2306.07662 (2023). arXiv:2306.07662.

[39] J. Gaglione, R. Roy, N. Baharisangari, D. Neider, Z. Xu, U. Topcu, Learning temporal logic properties: an overview of two recent methods, CoRR abs/2212.00916 (2022). arXiv:2212.00916.

[40] S. H. Muggleton, Inductive logic programming: Issues, results and the challenge of learning language in logic, Artif. Intell. 114 (1999) 283–296.

[41] M. Denecker, A. C. Kakas, Abduction in logic programming, in: Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I, volume 2407 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 402–436.

[42] D. Angluin, Queries and concept learning, Mach. Learn. 2 (1987) 319–342.

[43] B. Settles, Active Learning, Synthesis Lectures on Artificial Intelligence and Machine

Learning, Morgan & Claypool Publishers, 2012.

[44] N. H. Bshouty, Exact learning from membership queries: Some techniques, results and new directions, in: Algorithmic Learning Theory - 24th International Conference, ALT 2013, Singapore, October 6-9, 2013. Proceedings, volume 8139 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 33–52.

[45] S. Blum, R. Koudijs, A. Ozaki, S. Touileb, Learning horn envelopes via queries from large language models, CoRR abs/2305.12143 (2023). arXiv:2305.12143.

[46] D. M. L. Martins, Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities, Inf. Syst. 83 (2019) 89–100.

[47] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, P. Stone, LLM+P: empowering large language models with optimal planning proficiency, CoRR abs/2304.11477 (2023). arXiv:2304.11477.

[48] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net, 2023.

[49] R. Alur, R. Singh, D. Fisman, A. Solar-Lezama, Search-based program synthesis, Commun. ACM 61 (2018) 84–93.

[50] L. Bühmann, J. Lehmann, P. Westphal, Dl-learner - A framework for inductive learning on the semantic web, J. Web Semant. 39 (2016) 15–24.

[51] N. Fanizzi, G. Rizzo, C. d'Amato, F. Esposito, Dlfoil: Class expression learning revisited, in: Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings, volume 11313 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 98–113.

[52] L. Iannone, I. Palmisano, N. Fanizzi, An algorithm based on counterfactuals for concept learning in the semantic web, Appl. Intell. 26 (2007) 139–159.

[53] B. ten Cate, M. Funk, J. C. Jung, C. Lutz, Sat-based PAC learning of description logic concepts, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, ijcai.org, 2023, pp. 3347–3355.

[54] W. M. P. van der Aalst, Process mining: Overview and opportunities, ACM Trans. Manag. Inf. Syst. 3 (2012) 7:1–7:17.

[55] W. M. P. van der Aalst, Process mining, Commun. ACM 55 (2012) 76–83.

[56] P. Wolper, Temporal logic can be more expressive, Inf. Control. 56 (1983) 72–99.

[57] Y. Fontenla-Seco, S. Winkler, A. Gianola, M. Montali, M. L. Penín, A. J. B. Diz, The Droid You're Looking For: C-4PM, a Conversational Agent for Declarative Process Mining, in: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM 2023, volume 3469 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 112–116.

[58] G. D. Giacomo, M. Y. Vardi, Synthesis for LTL and LDL on finite traces, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press, 2015, pp. 1558–1564.

[59] A. D. Stasio, Reasoning about LTL Synthesis over finite and infinite games, Ph.D. thesis, University of Naples Federico II, Italy, 2018.

[60] A. Camacho, J. A. Baier, C. J. Muise, S. A. McIlraith, Finite LTL synthesis as planning, in: Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, Delft, The Netherlands, June 24-29, 2018, AAAI Press, 2018, pp. 29–38.

[61] A. Artale, L. Geatti, N. Gigante, A. Mazzullo, A. Montanari, Complexity of safety and cosafety fragments of linear temporal logic, in: Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI, AAAI Press, 2023, pp. 6236–6244.