Reasoning in the Financial Space with the Vadalog System

Teodoro Baldazzi[1,3], Luigi Bellomarini[2] and Emanuel Sallinger[3,4]

[1]Università Roma Tre, Department of Computer Science and Engineering, Rome, Italy
[2]Banca d'Italia, Italy
[3]TU Wien, Faculty of Informatics, Vienna, Austria
[4]University of Oxford, Department of Computer Science, Oxford, UK

Abstract

The financial and economic sector has witnessed growing interest in intelligent systems that model complex domains using expressive languages and reason over enterprise data to infer actionable knowledge. Responding to this trend and to the concrete needs of financial institutions, in this short paper, we illustrate our experience with Vadalog, a Datalog-based reasoner originating from a collaboration among the Central Bank of Italy, the University of Oxford, and TU Wien. Vadalog is tailored for high-stakes financial applications and operates over Enterprise Knowledge Graphs. We discuss its core features through real-world examples, offering practical insights into modeling and engineering Knowledge Representation and Reasoning systems. We also reflect on the technical challenges that limit a broader adoption of reasoners in finance and briefly touch on the potential of neurosymbolic approaches involving Large Language Models to help lower these barriers.

1. Introduction

The financial and economic sector is showing growing interest towards intelligent systems that model complex business domains using expressive, logic-based *Knowledge Representation and Reasoning* (KRR) languages. These systems reason over structured enterprise data – often in the form of *Enterprise Knowledge Graphs* (EKGs) – to derive actionable insights while offering transparency and explainability. These characteristics are crucial in such high-stakes domains, where traceable, logical decisions are essential for meeting legal, ethical, and compliance standards [1]. As a result, key financial stakeholders such as central banks, supervisory authorities, national statistical offices, and financial intelligence units are increasingly looking at KRR systems to support important use cases, including *banking supervision*, *credit-worthiness evaluation*, *anti-money laundering*, and *insurance fraud detection*.

In the database and AI communities, logic-based KRR languages – particularly Datalog and its extensions such as the Datalog$^{\pm}$ family [2, 3] – have matured considerably, offering a balance between expressive features like *recursion* and *existential quantification* and scalability. Over the past eight years, we have contributed to spurring and structuring the adoption of such approaches from the AI and database communities in financial institutions. In particular, within a collaboration among the Central Bank of Italy, the University of Oxford, and TU Wien, namely, the *KG Labs*, we developed VADALOG [4], a Datalog-based reasoner. Successfully applied in the economic and financial domain, the system has proved state-of-the-art among reasoners [5] and is now adopted in a wide range of high-stakes domains [6, 7].

While applying automated reasoning and knowledge graphs in the financial setting, we understood that the technical complexity of deploying and managing KRR-based reasoning systems remains a major barrier for analysts and decision-makers. Not only does the challenge lie in understanding language features such as *recursion* or *existential quantification*, but also in grasping their interaction and impact on system behavior, including scalability and decidability [8]. These capabilities are critical for expressing sophisticated financial rules and navigating knowledge graphs, but, more importantly, they require careful configuration and domain-specific tuning to ensure correctness and tractability.

The current technological scenario is even richer. Recent work has turned to *neurosymbolic* approaches, which combine the formal rigor of logic-based reasoning with the usability of *Large Language*

Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy
teodoro.baldazzi@uniroma3.it (T. Baldazzi); luigi.bellomarini@bancaditalia.it (L. Bellomarini); sallinger@dbai.tuwien.ac.at (E. Sallinger)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-4121/Ital-IA_2025_paper_37.pdf
CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-07

Models (LLMs) [9]. In this setting, reasoning engines can contribute guarantees of consistency and explainability [10], while LLMs act as a natural language interface, mediating user interaction with the underlying logic [11], or as a means to add human-like reasoning flexibility [12].

Contribution. Motivated by the drive to reduce the technical barriers to adopting KRR in financial domains, this paper – a short but updated version of a recent one [5] – illustrates the core features of VADALOG through its application to the specific context of the Central Bank of Italy. The examples also offer insights on how to model and engineer real-world use cases using KRR and related areas of AI, with a final glimpse on the prospective role of LLMs in hybrid systems.

2. Preliminaries

To enable the full potential of ontological reasoning, existential quantification is an essential requirement. This drove the development of the *Datalog*$^{\exists}$ family of logic languages, a natural extension of Datalog that allows existentially-quantified variables in rule heads. We now outline key foundational notions.

Relational Foundations. A (*relational*) *schema* $\mathbf{S}$ is a finite set of relation symbols (or *predicates*) with associated arity. A *term* is either a constant or a variable. An *atom* over $\mathbf{S}$ is an expression of the form $R(\bar{v})$, where $R \in \mathbf{S}$ is of arity $n > 0$ and $\bar{v}$ is an n-tuple of terms. A *database* (*instance*) over $\mathbf{S}$ associates to each symbol in $\mathbf{S}$ a relation of the respective arity over the domain of constants and nulls. A homomorphism is a constant-preserving mapping h from atom a_1 to atom a_2 if $h(a_1) = a_2$.

Syntax. A Datalog$^{\exists}$ program consists of a set of facts and *existential rules*, or *tuple-generating dependencies* (TGDs), function-free Horn clauses of the form $\forall\bar{x}\forall\bar{y}(\varphi(\bar{x},\bar{y})\rightarrow\exists\bar{z}\ \psi(\bar{x},\bar{z}))$, where $\varphi(\bar{x},\bar{y})$ (the *body*) and $\psi(\bar{x},\bar{z})$ (the *head*) are conjunctions of atoms over the respective predicates and the arguments are vectors of variables and constants. It may also feature *equality-generating dependencies* (EGDs), first-order implications of the form $\forall\bar{x}(\varphi(\bar{x})\rightarrow x_i = x_j)$, where $\varphi(\bar{x})$ is a conjunction of atoms and $x_i, x_j \in \bar{x}$. Moreover, real-world applications require reasoning systems to support multiple features that extend the declarative language. Among them, *aggregate functions*, namely *sum*, *prod*, *min*, *max* and *count*, as well as SQL-like grouping constructs, are particularly relevant. Important extensions also include *negations*, *negative constraints*, and *expressions* in rule bodies, modelled with *comparison* ($>, <, \geq, \leq, \neq$) and *algebraic* ($+, -, *, /$, etc.) operators.

Reasoning and Query Answering. Intuitively speaking, an *ontological reasoning* task consists in answering a *conjunctive query* (CQ) Q over a database D, augmented with a set Σ of logical rules. A conjunctive query (CQ) Q over a schema $\mathbf{S}$ is an implication $q(\mathbf{x}) \leftarrow \boldsymbol{\phi}(\mathbf{x},\mathbf{y})$, where $\boldsymbol{\phi}(\mathbf{x},\mathbf{y})$ is a conjunction of atoms, and $q(\mathbf{x})$ is an n-ary predicate not in $\mathbf{S}$. A CQ Q is satisfied in D if there exists a homomorphism h from the atoms in $\boldsymbol{\phi}(\mathbf{x},\mathbf{y})$ to the facts in D. The semantics of a Datalog$^{\pm}$ program is usually defined in an operational way with an algorithmic tool known as the *chase procedure* [13], which enforces the satisfaction of a set of dependencies Σ over a database D, incrementally expanding D with facts entailed via the application of the rules over D, until all of them are *satisfied*. Such facts possibly contain fresh new symbols ν (technically, *labelled nulls*) to satisfy existential quantification.

3. The Vadalog System through Financial Scenarios

Translating the foundations of Datalog$^{\exists}$-based reasoning into features of practical utility, and employing them to reason over complex real-world settings is by no means trivial. In this section, we outline the main features of the VADALOG system by discussing relevant financial scenarios applied to the *ownership knowledge graph* of the Bank of Italy, connecting companies, banks, and individuals [6].

Scenario 1. (**Company Control**) *This scenario captures who has decision power in a company network, that is, who controls the majority of votes for each company, under a "one-share one-vote" assumption. A company c_1 controls a company c_2, if: (i) c_1 directly owns more than 50% of c_2; or, (ii) c_1 controls a set of companies that jointly, and possibly together with c_1 itself, own more than 50% of c_2 [6].*

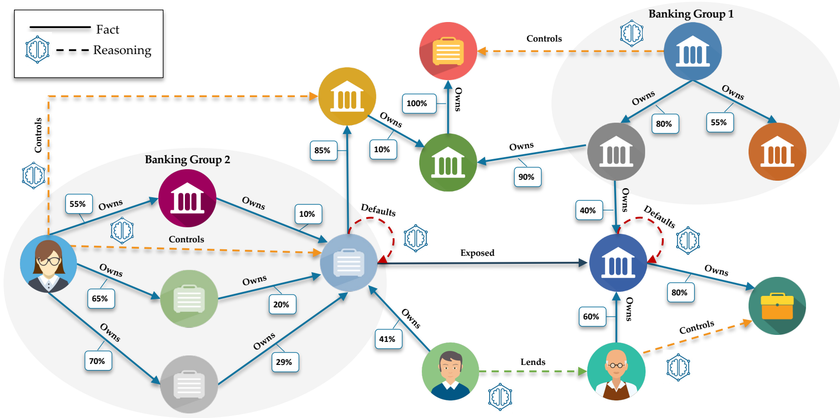

Figure 1: Excerpt of ownership knowledge graph © Central Bank of Italy.

$$\mathit{Owns}(c_1, c_2, s) \to \mathit{ControlledShares}(c_1, c_2, c_2, s) \quad (\sigma_1)$$

$$\mathit{Controls}(c_1, c_2), \mathit{Owns}(c_2, c_3, s), c_1 \neq c_3 \to \mathit{ControlledShares}(c_1, c_2, c_3, s) \quad (\sigma_2)$$

$$\mathit{ControlledShares}(c_1, \_, c_2, s), \mathit{ts} = \mathit{msum}(s), \mathit{ts} > 0.5 \to \mathit{Controls}(c_1, c_2) \quad (\sigma_3)$$

A company c_1 directly owning s shares of a company c_2 controls such shares via c_2 itself (rule σ_1). If c_1 controls c_2 and c_2 owns s shares of another company c_3, then c_1 controls s shares of c_3 via c_2 (rule σ_2). Finally, if c_1 controls a total amount of shares of c_2, directly or indirectly (i.e., through other controlled companies), that is greater than 0.5 (i.e., is the majority), then c_1 controls c_2 (rule σ_3).

This simple scenario – an *ontological reasoning* setting – highlights the need to combine multiple reasoning features to support real-world applications in the financial realm. At its core, identifying control relationships involves traversing chains of ownership of arbitrary length, potentially across multiple intermediaries. This requires support for recursion, which propagates shareholding through controlled entities. Yet, control is determined not by reachability alone, but by aggregating the total fraction of voting shares held, directly or indirectly. Thus, the reasoning system must also support aggregations, such as the sum computed in rule σ_3. Still, *standard* aggregate functions, known to be affected by multiple limitations, and are typically disallowed in recursive settings. This is addressed by the class of *monotonic* aggregates [14], which guarantee well-defined behavior under recursion by ensuring that aggregate values evolve consistently, only increasing or only decreasing throughout the evaluation. In VADALOG, we further extended this class to operate seamlessly in the presence of existential quantification [4]. Finally, determining whether a company has majority control requires the ability to evaluate numeric comparisons, as expressed by the threshold $\mathit{ts} > 0.5$ in rule σ_3. These kinds of expressions fall outside of the expressive power of Datalog and needed to be explicitly supported.

Scenario 2. (**Financial Shock Propagation**) *This scenario models how the default of a financial intermediary affects other intermediaries that are financially exposed with it.*

$$\mathit{FinInt}(x), \mathit{Own}(p, x, w), w > 0.3 \to \mathit{KP}(p, x) \quad (\sigma_1)$$

$$\mathit{FinInt}(x), \mathit{NPL}(x) \to \exists f\ \mathit{Default}(x, f, f) \quad (\sigma_2)$$

$$\mathit{Default}(x_1, f_x, f_1), \mathit{Exp}(x_1, x_2) \to \exists f_2\ \mathit{Default}(x_2, f_1, f_2) \quad (\sigma_3)$$

$$\mathit{Default}(x, f_1, f_2), \mathit{KP}(p, x) \to \exists i\ \mathit{Inv}(p, x, i) \quad (\sigma_4)$$

$$\mathit{KP}(p_1, x), \mathit{KP}(p_2, x), \mathit{Inv}(p_1, x, i_1), \mathit{Inv}(p_2, x, i_2) \to i_1 = i_2 \quad (\sigma_5)$$

$$\mathit{Inv}(p_1, x_1, i_1), \mathit{Inv}(p_2, x_2, i_2), \mathit{Exp}(x_1, x_2) \to i_1 = i_2 \quad (\sigma_6)$$

An individual p is a key person (KP) of a financial intermediary (FinInt) x if p owns more than 30% of the shares (w) of x (rule σ_1). If a FinInt x is involved in non-performing loans (NPL), then it will default on its debts, initiating a failure event f (rule σ_2). If a FinInt x_2 is financially exposed (Exp) with another FinInt x_1 which undergoes a failure event f_1, caused by another failure f_x, then x_2 will be in turn involved in a failure f_2 caused by f_1 (rule σ_3). A financial investigation i is launched for each KP of a defaulted FinInt x

(rule σ_4). If p_1 and p_2 are KPs of the same defaulted FinInt x (rule σ_5), or x_2 is exposed with x_1 (rule σ_6), then they are involved in the same investigation.

In this scenario, a defining aspect is the joint presence of existential quantification and recursion. It involves the generation of fresh existential witnesses — such as failure events and investigations — which must be recursively propagated through chains of financial exposure. However, ontological reasoning under Datalog$^\exists$ is known to be undecidable due to the interplay between these two features, which may cause the chase to generate infinitely many labelled nulls and result in non-termination [8]. This led to the proposal in the literature of powerful Datalog$^\pm$ languages which apply specific syntactic restrictions to achieve a good trade-off between expressive power and computational complexity of reasoning [3]. In VADALOG, we address this challenge by adopting the *Warded Datalog*$^\pm$ fragment [8] as the logic core of the formalism, applying sophisticated optimizations to the chase procedure to maintain soundness while ensuring tractability even in these settings. Additionally, this scenario includes equalities over variables affected by existential quantification, modeling the merging of investigations across overlapping networks of key individuals. Since the interplay between equalities and existentials may lead to undecidability, we extended VADALOG with the support for a new class of EGDs, namely, *harmless EGDs* [15]. Unlike classical equality dependencies, harmless EGDs are guaranteed not to interfere with rule triggering, thus retaining both expressiveness and decidability.

Scenario 3. (**Collateral Eligibility under Uncertainty**) *This scenario models how lenders of distinct types are subject to restrictions by supervision authorities requiring loans to be covered by a collateral.*

$$0.9 :: \mathit{LenderT}(x, y), \mathit{RR}(y, z) \rightarrow \exists v\ \mathit{Guarantee}(x, z, v) \qquad (\sigma_1)$$

$$0.8 :: \mathit{LenderT}(x, y), \mathit{LenderC}(y, z) \rightarrow \mathit{LenderT}(x, z) \qquad (\sigma_2)$$

$$0.7 :: \mathit{Contract}(x, y, z), \mathit{Exp}(y, w) \rightarrow \mathit{Contract}(z, w, x) \qquad (\sigma_3)$$

$$\mathit{Contract}(x, y, z), \mathit{RR}(w, y) \rightarrow \mathit{LenderT}(x, w) \qquad (\sigma_4)$$

Ignoring what precedes the :: symbols, if a lender x is of type y (e.g., a bank, a small company, etc.) and lenders of type y are subject to regulatory restrictions (RR) of type z (e.g., securities, real estate properties, cash, etc.), then there exists a collateral of type z for x issued by a guarantor v (an individual, a bank, or a financial intermediary) (rule σ_1). If the type y is a subclass of the lender class z (e.g., credit unions is a subclass of retail lender), then the lender x is of type z (rule σ_2). If the loan from a lender x to a borrower z has been formalized by a contract, then there exists another contract for the financial exposure of type w (corresponding to loan type y) from z to x (rule σ_3). If a contract from x to z is in place to satisfy a RR that requires a guarantee with contract type y, based on a lender type w, then x is also of type w (rule σ_4).

Traditional KRR languages and approaches are often insufficient to properly address complex real-world scenarios. For instance, how should one model events that are not always definitive, but rather may only occur with a given probability or in specific time frames? For these reasons, ontological reasoning has been complemented with other forms of automated reasoning, such as *probabilistic*, *temporal* reasoning, etc. Specifically, the scenario we consider here features some notion of uncertainty related to our domain. Indeed, depending on how each financial intermediary implements the regulations, rules σ_1-σ_3 may apply or not. Intuitively, probabilistic reasoning requires computing the probabilistic answer to a query as a set $\{\langle \bar{t}, P(\bar{t})\rangle\}$, where $\bar{t}$ is a fact and $P(\bar{t})$ is its *marginal probability*, i.e., the probability for $\bar{t}$ to be entailed. To address this task we extended VADALOG for probabilistic reasoning. Since computing exact marginal probabilities is intractable, we introduced the *MCMC-chase*, a chase variant that approximates them, performing logical and probabilistic inference while preserving tractability [16].

Scenario 4. (**Company Supervision through Time**) *This scenario models how a governmental institution supervises the changes in the corporate structure of companies with strategic relevance, as well as the actions of those shareholders who are buying into the companies later in the game.*

$$\diamondplus_{[0,1]}\mathit{SignificantShare}(x, y), \neg \diamondminus_{[0,1]}\ \mathit{SignificantShare}(x, y) \rightarrow \mathit{SignificantOwner}(x, y) \qquad (\sigma_1)$$

$$\mathit{WatchCompany}(y), \mathit{SignificantOwner}(x, y), \mathit{Connected}(x, z) \rightarrow \mathit{WatchCompany}(z) \qquad (\sigma_2)$$

If in an interval in the past (denoted by $\diamondminus_{[0,1]}$) a shareholder x does not own a significant amount of shares of a company y, while that is the case at some point in a future interval (denoted by $\diamondplus_{[0,1]}$), then x is a significant owner of y (rule σ_1). If x is a significant owner of a company y in a watchlist, then all the other companies z that are connected to x are also added to the watchlist (rule σ_2).

This last scenario represents a temporal reasoning task. To support this form of reasoning in Datalog, the AI community introduced *DatalogMTL*, which extends the language with forms of *time awareness* and encodes operators that link standard atemporal assertions to references and streams of data valid within specific *time windows*. Yet, the development of temporal reasoners based on DatalogMTL was still in its infancy. We extended VADALOG with temporal reasoning capabilities, providing for the first time, to the best of our knowledge, a production-ready temporal reasoner that fully supports recursion and allows addressing real-world temporal tasks with good results in performance and scalability [17]. Finally, this scenario also features negation, which VADALOG supports in two forms: *grounded*, applied to extensional (i.e., database) predicates, and *stratified*, applied to non-recursive head predicates.

Neurosymbolic Synergies with LLMs. The recent advances in LLMs have opened promising directions to bridge knowledge graph reasoning and natural language interfaces. In this context, we combined the VADALOG reasoning engine with LLM-based interfaces to create KG-ROAR, a hybrid neurosymbolic platform that integrates logical inference with intuitive, conversational access [10]. As illustrated in Figure 2, this architecture enables users to trigger reasoning tasks through graphical or verbal input, and to query the resulting knowledge graph using natural language. A key component is the use of *chase-based Retrieval-Augmented Generation* mechanisms [18], which expose both inferred facts and their derivation paths to the LLM to support transparent, logic-based retrieval of facts and explanations. This synergy allows the LLM to act as a knowledgeable assistant grounded in the underlying KG, combining accessibility with transparency and trustworthiness. In practice, financial analysts can use this setup to navigate complex regulatory logic, ask explanatory questions, and receive faithful answers with grounded justifications, without requiring expertise of the underlying formalism.

4. Conclusion

The growing interest in knowledge graph reasoning within the economic and financial sectors is often tempered by the technical complexity and unfamiliarity surrounding the underlying languages and systems. In this paper, we aimed to show how this gap can be bridged by providing a practical, use case-driven overview of the VADALOG reasoner, illustrating how its expressive features address real-world challenges in finance. Additionally, we touched on how combining logic-based reasoning with natural language interfaces powered by LLMs can ease adoption and improve accessibility, paving the way for more transparent and explainable reasoning workflows in high-stakes domains.

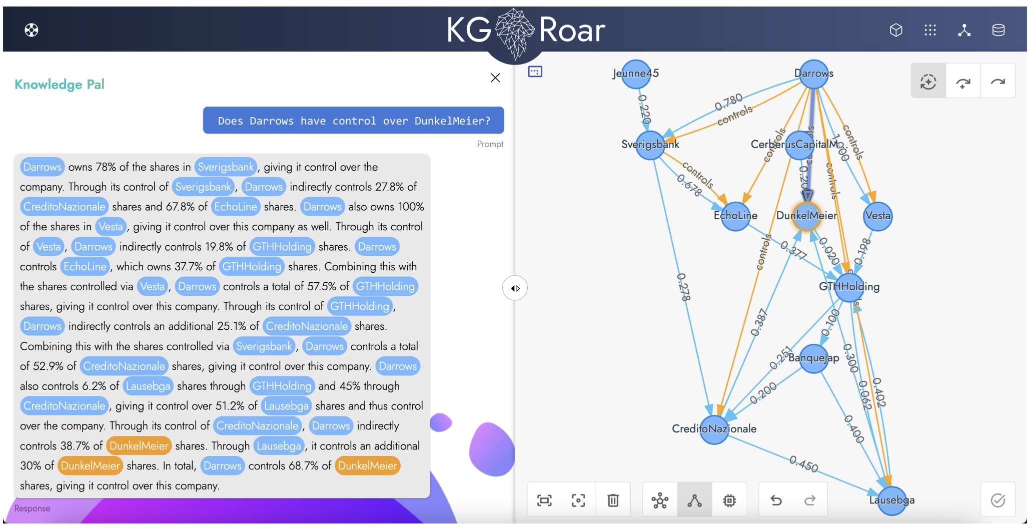

Figure 2: Instance of natural language-based interaction over Scenario 1 in KG-ROAR.

Acknowledgments

This work has been supported by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18013, 10.47379/NXT22018, 10.47379/ICT2201].

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o for grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. I. Hong, Teaching the fate community about privacy, Commun. ACM 66 (2023) 10–11.
[2] S. Abiteboul, R. Hull, V. Vianu, Foundations of databases, volume 8, 1995.
[3] A. Calì, G. Gottlob, T. Lukasiewicz, B. Marnette, A. Pieris, Datalog+/-: A family of logical knowledge representation and query languages for new applications, in: IEEE, 2010, pp. 228–242.
[4] L. Bellomarini, D. Benedetto, G. Gottlob, E. Sallinger, Vadalog: A modern architecture for automated reasoning with large knowledge graphs, Inf. Syst. 105 (2022) 101528.
[5] T. Baldazzi, L. Bellomarini, E. Sallinger, Reasoning over financial scenarios with the vadalog system, in: EDBT, OpenProceedings.org, 2023, pp. 782–791. doi:10.48786/edbt.2023.66.
[6] P. Atzeni, L. Bellomarini, M. Iezzi, E. Sallinger, A. Vlad, Weaving enterprise knowledge graphs: The case of company ownership graphs, in: EDBT, OpenProceedings.org, 2020, pp. 555–566.
[7] L. Bellomarini, L. Bencivelli, C. Biancotti, L. Blasi, F. P. Conteduca, A. Gentili, R. Laurendi, D. Magnanimi, M. S. Zangrandi, F. Tonelli, S. Ceri, D. Benedetto, M. Nissl, E. Sallinger, Reasoning on company takeovers: From tactic to strategy, Data Knowl. Eng. 141 (2022) 102073.
[8] G. Gottlob, A. Pieris, Beyond SPARQL under OWL 2 QL entailment regime: Rules to the rescue, in: IJCAI, AAAI Press, 2015, pp. 2999–3007.
[9] A. d. Garcez, L. C. Lamb, Neurosymbolic ai: The 3 rd wave, Artificial Intelligence Review 56 (2023).
[10] T. Baldazzi, L. Bellomarini, S. Ceri, A. Colombo, A. Gentili, E. Sallinger, "please, vadalog, tell me why": Interactive explanation of datalog-based reasoning, in: EDBT, OpenProceedings.org, 2024, pp. 834–837. doi:10.48786/EDBT.2024.82.
[11] T. Baldazzi, L. Bellomarini, S. Ceri, A. Colombo, A. Gentili, E. Sallinger, P. Atzeni, Explaining enterprise knowledge graphs with large language models and ontological reasoning, in: Tannen's Festschrift, volume 119 of *OASIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 1:1–1:20.
[12] T. Baldazzi, D. Benedetto, L. Bellomarini, E. Sallinger, A. Vlad, Softening ontological reasoning with large language models, in: RuleML+RR (Companion), volume 3816 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024.
[13] D. Maier, A. O. Mendelzon, Y. Sagiv, Testing implications of data dependencies, ACM TODS 4 (1979) 455–469.
[14] A. Shkapsky, M. Yang, C. Zaniolo, Optimizing recursive queries with monotonic aggregates in deals, in: ICDE, IEEE Computer Society, 2015, pp. 867–878.
[15] L. Bellomarini, D. Benedetto, M. Brandetti, E. Sallinger, Exploiting the power of equality-generating dependencies in ontological reasoning, Proc. VLDB Endow. 15 (2022) 3976 – 3988.
[16] L. Bellomarini, E. Laurenza, E. Sallinger, E. Sherkhonov, Swift markov logic for probabilistic reasoning on knowledge graphs, CoRR abs/2210.00283 (2022).
[17] L. Bellomarini, L. Blasi, M. Nissl, E. Sallinger, The temporal vadalog system: Temporal datalog-based reasoning, arXiv preprint arXiv:2412.13019 (2024).
[18] T. Baldazzi, L. Bellomarini, E. Sallinger, Knowledge graph-based reasoning in large language models, in: Handbook on Neurosymbolic AI and Knowledge Graphs, IOS Press, 2025, pp. 441–465.