"Please, Vadalog, tell me why": Interactive Explanation of Datalog-based Reasoning

Teodoro Baldazzi
Università Roma Tre
Rome, Italy
teodoro.baldazzi@uniroma3.it

Luigi Bellomarini
Bank of Italy
Rome, Italy
luigi.bellomarini@bancaditalia.it

Stefano Ceri
Politecnico di Milano
Milan, Italy
stefano.ceri@polimi.it

Andrea Colombo
Politecnico di Milano
Milan, Italy
andrea1.colombo@polimi.it

Andrea Gentili
Bank of Italy
Rome, Italy
andrea.gentili@bancaditalia.it

Emanuel Sallinger
TU Wien and University of Oxford
Vienna, Austria
sallinger@dbai.tuwien.ac.at

ABSTRACT

Integrating Large Language Models (LLMs) with logic-based Enterprise Knowledge Graphs (EKGs) and more generally with Knowledge Representation and Reasoning (KRR) approaches is currently at the forefront of research in many data-intensive areas, as language models may complement EKGs and ontological reasoning with flexibility and human orientation. Conversely, EKGs provide transparency and explainability on the conclusions drawn, a typical weak point of LLMs, which operate opaquely.

In this demo, we integrate Llama 2 with our reasoning system Vadalog and use it to turn a chase graph, i.e., the trace of an ontological reasoning process, into a human-readable business report. In other words, we show the amazing capabilities of state-of-the-art LLMs in combination with a principled exploitation of the theoretical underpinnings of logic-based reasoning.

We walk the audience through a visual environment, unfolding real-world reasoning settings from the Central Bank of Italy.

1 INTRODUCTION

In today's data-driven industrial landscape, the principles of Fairness, Accountability, Transparency, and Ethics (FATE) have become vital for AI applications [13]. This is particularly relevant in high-stakes domains such as finance and bio-medicine, where the characteristic typically shared as *explainability* is of the essence for decision-making, users' trust and adherence to ethical standards [11].

With the breakthrough of generative AI tools and Large Language Models (LLMs) [14], we are experiencing a paradigm shift in the access to data and knowledge, which is more and more based on Natural Language Processing (NLP) tasks and therefore natural, friendly and at a high level. Yet, LLMs are often criticized for their lack of factual knowledge [3] and, more importantly, very limited explainability [19].

On the other hand, traditional *Knowledge Representation and Reasoning* (KRR) approaches are inherently *explainable* [8]. For instance, logic reasoning in query answering, often dubbed as *ontological reasoning* [7], is designed to provide factual answers based on logically consequential steps. Yet, the interaction is query-based, often at a low level, and unfriendly to nonspecialists.

The aim of this demo is to show our results in an attempt to augment ontological reasoning with natural language explanations about the generated factual information. In this sense, it is in the context of *neuro-symbolic* methodologies, whose goal is to combine the intrinsic explainability and transparency of deductive systems with the power of LLMs in understanding and generating fluent and interpretable text [9, 18].

In particular, this work relies on a recently published integration of the Vadalog system [3], a state-of-the-art ontological reasoner [6], and LLMs to activate natural language explanations on the well-known chase [16] procedure of databases. We implemented the demo within *KG-Roar* [5], a framework we created to showcase ontological reasoning with Vadalog on real-world cases that can be suitably modeled with a *Knowledge Graph* (KG). The demo is extensible as it considers real-world applications in the financial realm, deriving from our projects with the Central Bank of Italy (e.g., [2, 4]).

2 SCOPE OF THE DEMO

The demonstration will showcase two relevant scenarios from the financial sector, letting the audience play the role of a business analyst who wants to discover:

- who can take decisions on, i.e., *controls*, a given company within an ownership knowledge graph;
- whether two companies have a special relationship that makes them too close, namely, they are a *close link*, to act one as the guarantor of the other in credit operations such as the issuance of asset-back securities.

Both problems are relevant to financial authorities and have been studied at length, achieving a rule-based formalization in the Vadalog language. Vadalog extends the well-known language *Datalog* [1] of databases with features of practical utility such as aggregations and algebra. In recent papers, we have shown how Vadalog can provide efficient and exhaustive answers to control and close link questions. Still, such answers often involve a complex and long reasoning inference process, which renders them hard to explain, understand, and trust for the user.

In simple terms: *how can we answer "why this conclusion" questions in natural language and posed by the user in natural language?* Our idea consists in using Llama 2-70B to unfold the intrinsic explanation of the chase graph, generated as a byproduct of ontological reasoning. In the tool, we: (a) Show the domain data as a graph. (b) Augment the graph by executing intensional business knowledge defined as Vadalog rules and encoding the two problems at hand; we render the reasoning results as new edges on the graph. (c) Allow the user to obtain textual explanation for each of the edges by either clicking on them or posing the question in natural language. The explanations given are active and nouns corresponding to entities can be selected in the text to visually contextualize them in the rendered graph.

The audience will experience how LLAMA 2 can effectively synergize with ontological reasoning to enhance the productivity of users and the transparency of data analysis tasks. We will show a novel visual metaphor based on gradually increasing the original extensional knowledge with newly generated facts, for which a natural language explanation can be easily obtained. We will demonstrate how our system can assist users in understanding the reasoning process on-the-fly.

Resources. A brief accompanying video is available.[1] Moreover, we provide the used knowledge graph,[2] that is artificially generated mimicking the distribution of the real one managed by the Central Bank of Italy.

Overview. We lay out the technical background in Section 3. The system workflow is presented in Section 4. In Section 5, we illustrate the demo plan.

3 OUR REASONING TASKS

We first introduce some technical background of ontological reasoning and then the use cases dealt with in our demo.

Preliminaries. KRR approaches model a domain of interest as the combination of an *extensional component*, essentially a database with business data, and an *intensional component*, which formally describes the business knowledge.

Let **C** and **V** be disjoint countably infinite sets of *constants* and *variables*, respectively. A (*relational*) *schema* 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 S is an expression of the form $R(\bar{v})$, where $R \in S$ is of arity $n > 0$ and $\bar{v}$ is an n-tuple of terms. A *database* (*instance*) over S associates to each symbol in S a relation of the respective arity over the domain of constants. The members of the relations are called *facts* or *tuples*.

We model the intensional component as a VADALOG program Σ. The VADALOG language is an extension of Datalog with features of practical use in reasoning settings [6]. In particular, in this demo, we will use *aggregations*, defined according to the usual stratified semantics [17], *built-in comparison predicates* (e.g., $\geq, >, \leq, <, \neq$), and *algebraic expressions*. A VADALOG program is a set of rules 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, $\bar{x},\bar{y}$ are vectors of universally quantified variables and constants, and $\bar{z}$ is a vector of existentially quantified variables. In our settings, values for existentially quantified variables are computed with aggregations or algebraic expressions and the $\exists$ symbol is therefore omitted.

An (ontological) *reasoning task* consists in answering a query over a database $\mathcal{D}$ extended with all the possible facts obtained by applying the rules of Σ until fixpoint, namely, $\Sigma(\mathcal{D})$. While VADALOG guarantees that such fixpoint exists when only the core features are used [6], the joint presence of algebraic operations and recursion must be carefully handled, as even simple Datalog programs can be in general non-terminating [1].

The CHASE is a typical procedure used in database settings to compute $\Sigma(\mathcal{D})$ [16] (which is sometimes called the CHASE itself). It activates all rules until new facts can be generated. Each rule activation is called *chase step*. $\Sigma(\mathcal{D})$ can be represented as a directed acyclic graph, namely, the CHASE *graph*, where there is a node for each fact in $\Sigma(\mathcal{D})$ and an edge from n to m if the fact associated with m derives from that of n via a CHASE step.

[1]Link to the video: http://bit.ly/edbt-2024-kglm-video
[2]Link to the data generator and the knowledge graph: https://bitly.ws/33Sgh

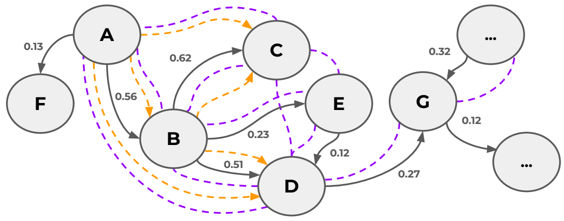

Figure 1: Portion of ownership KG. Nodes are companies. Solid edges are *owns* relationships, with their shares. Dashed edges respectively denote *control* relationships (orange, directed) and *close links* (purple, undirected).

Reasoning Use Cases. Let us focus on the demo scenarios. We refer to an *ownership knowledge graph*, where the extensional component consists of *Owns* relationships; the intensional component is formalized as VADALOG rules, which augment the graph with *Control* and *CloseLink* edges. An example is in Figure 1.

Example 3.1. (Company Control) This reasoning task allows analysts to understand who has decision power in companies, based on who controls the majority of votes, in a "one-share one-vote" assumption. To this end, the task augments the ownership graph with "*control*" edges, as follows [12]: *A company x directly owning s shares of a company y, controls such shares via y itself* (rule 1). *If x controls z and z owns s shares of y, then x controls s shares of y via z* (rule 2). *Finally, if x controls the majority of the shares of y, directly or indirectly, then x controls y* (rule 3).

$$Owns(x,y,s) \rightarrow ControlledShares(x,y,y,s) \quad (1)$$
$$Control(x,z), Owns(z,y,s) \rightarrow ControlledShares(x,z,y,s) \quad (2)$$
$$ControlledShares(x,\_,y,s),\ ts = sum(s),$$
$$ts > 0.5 \rightarrow Control(x,y) \quad (3)$$

Example 3.2. (Close Links) To describe this reasoning task, we first introduce the concept of *integrated ownership I* of a company x on y, which accounts for all the possible direct or indirect shares that x owns of y throughout the graph, with finite or infinite paths [15]. The value of I is calculated as $lim_{\epsilon\rightarrow 0}\sum_{P_i\in B_\epsilon} w_\epsilon(P_i)$, where B_ϵ is the set of all paths $P = [x, p_1, \ldots, p_k, y]$ in the ownership graph such that $x \neq p_i$ for $i = 1, \ldots, k$, and where $w_\epsilon(P) = \Pi_{(p_i,p_j)\in P} w(p_i,p_j) > \epsilon$, with $w(p_i,p_j)$ representing the direct ownership of p_i on p_j, $\epsilon \in \mathbb{R}^+$, and $0 < \epsilon \leq 1$. Note that integrated ownership is different from simple ownership used in Example 3.1. We consider integrated ownership between companies as an input to our scenario and denote it with the *IntOwns* predicate (omitted in Figure 1 to avoid clutter).

Applying this formulation to the regulation of the European Central Bank [10], we can say that x is in close link with y if: *(i) the integrated ownership of x on y is at least* 20% (rule 1); *(ii) y is in close link with x* (rule 2); *(iii) there is a third company z, whose integrated ownership on x and y is at least* 20% (rule 3).

$$IntOwns(x,y,s), s \geq 0.2 \rightarrow CloseLink(x,y) \quad (1)$$
$$CloseLink(y,x) \rightarrow CloseLink(x,y) \quad (2)$$
$$IntOwns(z,x,s), IntOwns(z,y,t), s \geq 0.2, t \geq 0.2,$$
$$x \neq y, x \neq z, y \neq z \rightarrow CloseLink(x,y) \quad (3)$$

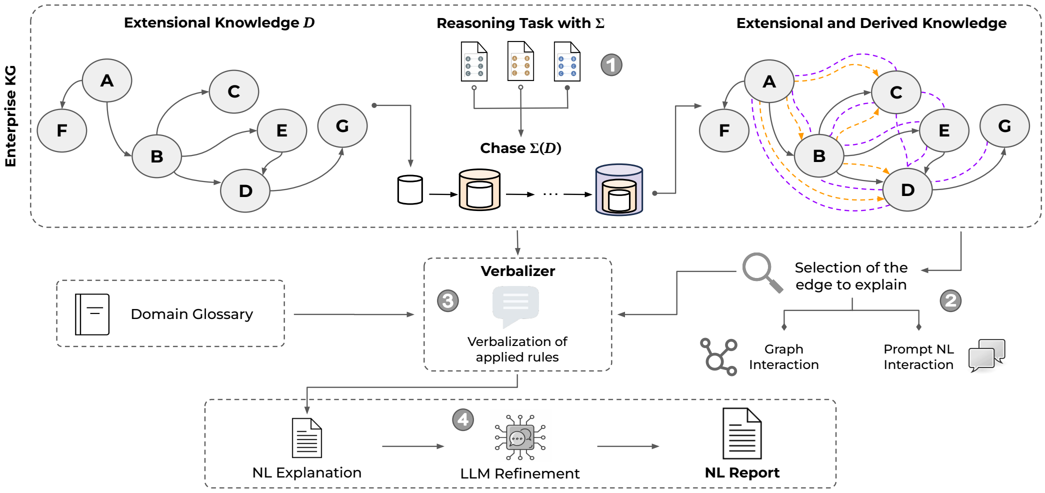

Figure 2: Workflow of our framework. Given the *extensional knowledge* in $\mathcal{D}$, the user chooses a reasoning task to infer *derived knowledge* via the CHASE $\Sigma(\mathcal{D})$ and produce an enriched KG. The user then selects an edge of the graph, by either visual or natural language interaction. The verbalizer cooperates with the LLM to generate the explanation.

4 OVERVIEW OF THE SYSTEM

KG-Roar offers an interactive productivity environment for KGs, performing reasoning tasks via the VADALOG system in the back-end and rendering the resulting graph, augmented with the newly inferred edges, in a dynamic and navigable fashion. It goes beyond the visual representation of the derived edges, often unsatisfactory as missing any motivation for their existence, and features a *verbalizer* module that generates natural language explanations from the VADALOG reasoning process.

Main Workflow. Figure 2 illustrates the main steps of the high-level workflow implemented by KG-Roar.

(1) Reasoning Task. In KG-Roar, the user interaction begins with the selection of a reasoning task. Tasks are rendered as *widgets*, embedded micro-IDE environments containing the textual and VADALOG specification of a domain of interest. Widgets also contain the mapping of predicates into nodes and edges. The corresponding task is executed by the VADALOG system in the back-end. The derived facts are rendered to augment the visualized KG, while the CHASE graph produced is stored.

(2) Fact Selection. The user asks the system to explain a generated fact f. This can be done either by selecting the corresponding edge of the visualized KG and clicking "*Explain Relationship*", or in NL with a prompt-based interaction. In the latter case, an underlying LLAMA 2-70B LLM maps the prompt into the corresponding VADALOG fact in the domain of interest.

(3) Fact Explanation. The selected fact is passed to the verbalizer module in the back-end, which—in case of selection via NL prompt—verifies that f is in $\Sigma(\mathcal{D})$. If that is the case, the actual generation of the textual explanation takes place. First, the verbalizer extracts from $\mathcal{G}$ a subgraph $\mathcal{G}'$ composed of all the paths from a fact of $\mathcal{D}$ to f, i.e., the complete structure of CHASE steps that produced f from the extensional component. A breadth-first traversal of $\mathcal{G}'$ is then performed and, in the process, each edge e of $\mathcal{G}'$ is translated into a natural language sentence of the form *"Since {body}, then {head}"*. The tokens *{body}* and *{head}* are obtained by verbalizing the CHASE step corresponding to e, i.e., the bindings for the body and head of the respective rule. To this end, a *domain glossary*, a map of the predicates of our domain schema S into their NL equivalent, is used.

All VADALOG syntactic elements are converted into their textual counterparts, for example: conjunctions are rendered as *"and"* tokens; built-in operators are rendered with specific keywords, e.g., > becomes *is higher than*, and so on. Note that the final order of the verbalized CHASE steps reflects the breadth-first traversal of $\mathcal{G}'$ and thus respects logical dependency.

(4) LLM Refinement. The explanation produced in step (3) of the workflow can be long, complex, and hard to read, especially in the presence of complex reasoning paths. To make it more understandable, we leverage the text manipulation capabilities of our LLAMA 2-70B to improve the fluency and clarity of the result. By prompting the LLM with the following request: *"Please produce a more readable version of the explanation: ..."*, we achieve a refined report that is highly accurate in content and comprehensible.

Implementation. We implemented the verbalizer in Python and made it available.[3] The LLAMA 2-70B model is deployed on enterprise servers for confidentiality reasons.

Example 4.1. To conclude, let us consider again the company control scenario introduced in Example 3.1. The following domain glossary captures the definitions for the predicates involved.

Predicate	Description
$Owns(x, y, s)$	<x> owns <s> shares of <y>
$ControlledShares(x, z, y, s)$	<x> controls <s> of <y> via <z>
$Control(x, y)$	<x> controls <y>

Let us also consider the portion of ownership KG in Figure 1, augmented with *control* edges derived from the reasoning task. Now, we may wonder *"Why does company A exert control over company C?"* and submit such a request to KG-Roar as an NL

[3]Link to the module: https://bitly.ws/33VYQ

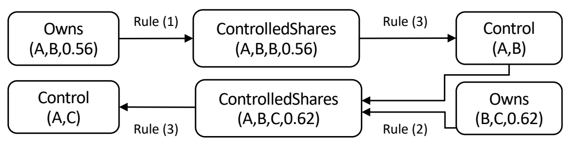

Figure 3: An explanation subgraph.

prompt-based interaction. The system uses Llama 2 to single out the fact "*Control*(A, C)", of our interest. In the back-end, the verbalizer verifies that such a fact is in $\Sigma(\mathcal{D})$ and extracts from the chase graph the explanation subgraph $\mathcal{G}'$, in Figure 3. The explanation produced by KG-Roar in step (3) of the workflow is: *"Since A owns* 0.56 *shares of B, then A controls* 0.56 *of B via B. Since A controls* 0.56 *of B via B and* 0.56 *is higher than* 0.5*, then A controls B. Since A controls B and B owns* 0.62 *shares of C, then A controls* 0.62 *of C via B. Since A controls* 0.62 *of C via B and* 0.62 *is higher than* 0.5 *then A controls C"*. Finally, the LLM refines the text and the following explanation is provided as a response to the user: *"A's direct ownership of* 0.56 *of B translates to its control over B. With B's ownership of* 0.62 *of C, A, through B, also controls* 0.62 *of C. As the percentage exceeds* 0.5*, A effectively controls C"*.

5 DEMONSTRATION PLAN

The demonstration shows the effectiveness of our framework in supporting data analysis over enterprise KGs via reasoning and the interactive generation of natural language explanations. After a brief overview of the interface of KG-Roar, the presentation will start by letting our visitor play the role of an analyst who investigates a financial entity under scrutiny.

Financial Analysis. First, we will use simple widgets to load and render the ownership KG required in our analysis. Then, we will explore the control use case presented in Example 3.1. After selecting the corresponding widget to perform the reasoning task and augment the KG with the inferred *control* edges, we will focus on a company of interest, investigating the chains of ownership that lead to its control by other entities. To achieve this, we will select target edges and trigger their explanations via a right click "*Explain Relationship*" option. In KG-Roar, the generated reports will be presented in a dedicated box that can be interactively explored, by highlighting tagged entities (Figure 4).

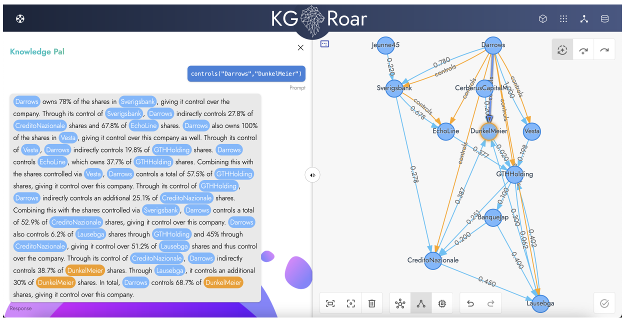

Figure 4: Explanation panel for *control* use case.

Then, we will continue the analysis by shifting the attention to our second use case, *close link* detection. After launching the new reasoning task, we will observe how a visual exploration of the enriched KG becomes less intuitive due to its highly interconnected nature. To address this, we will showcase how we can swiftly detect and explain specific edges by leveraging the NL prompt and the underlying Llama 2 model (Figure 5). Moreover, in case a close link exists KG-Roar will zoom in on the portion of the graph, thus providing both a visual and textual explanation.

Exploratory Analysis. In the last part of the demonstration, the participants will be allowed freely to browse the KG, trigger reasoning widgets, and prompt KG-Roar for NL explanation.

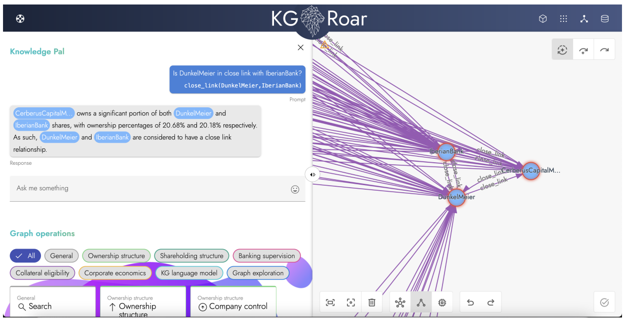

Figure 5: Explanation panel for *close link* use case.

ACKNOWLEDGMENTS

S.C. is supported by the PNRR-PE-AI FAIR project, funded by the NextGenerationEU program. A.C. kindly acknowledges INPS for the funding of his Ph.D. program. The work was partially supported by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT2201, 10.47379/VRG18013, 10.47379/NXT22018]; and the Christian Doppler Research Association (CDG) JRC LIVE.

REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of databases*. Vol. 8. Addison-Wesley Reading.
[2] Paolo Atzeni, Luigi Bellomarini, Michela Iezzi, Emanuel Sallinger, and Adriano Vlad. 2020. Weaving Enterprise Knowledge Graphs: The Case of Company Ownership Graphs.. In *EDBT*. 555–566.
[3] Teodoro Baldazzi, Luigi Bellomarini, Stefano Ceri, Andrea Colombo, Andrea Gentili, and Emanuel Sallinger. 2023. Fine-tuning Large Enterprise Language Models via Ontological Reasoning. *arXiv preprint arXiv:2306.10723* (2023).
[4] Luigi Bellomarini, Marco Benedetti, Andrea Gentili, Rosario Laurendi, Davide Magnanimi, Antonio Muci, and Emanuel Sallinger. 2020. COVID-19 and Company Knowledge Graphs: Assessing Golden Powers and Economic Impact of Selective Lockdown via AI Reasoning. *CoRR* abs/2004.10119 (2020).
[5] Luigi Bellomarini, Marco Benedetti, Andrea Gentili, Davide Magnanimi, and Emanuel Sallinger. 2023. KG-Roar: Interactive Datalog-Based Reasoning on Virtual Knowledge Graphs. *Proc. VLDB Endow.* 16, 12 (aug 2023), 4014–4017.
[6] Luigi Bellomarini, Davide Benedetto, Georg Gottlob, and Emanuel Sallinger. 2022. Vadalog: A modern architecture for automated reasoning with large knowledge graphs. *Information Systems* 105 (2022), 101528.
[7] Andrea Calì, Georg Gottlob, and Andreas Pieris. 2012. Towards more expressive ontology languages: The query answering problem. *Artificial Intelligence* 193 (2012), 87–128.
[8] Luciano Caroprese, Eugenio Vocaturo, and Ester Zumpano. 2022. Argumentation approaches for explanaible AI in medical informatics. *Intelligent Systems with Applications* 16 (2022), 200109.
[9] Xin Luna Dong. 2023. Generations of Knowledge Graphs: The Crazy Ideas and the Business Impact. *arXiv preprint arXiv:2308.14217* (2023).
[10] European Central Bank. 2011. GUIDELINE (EU) 2011/14 of the ECB Guideline.
[11] Jorge Fandinno and Claudia Schulz. 2019. Answering the "why" in answer set programming–A survey of explanation approaches. *Theory and Practice of Logic Programming* 19, 2 (2019), 114–203.
[12] A. Gulino, S. Ceri, G. Gottlob, E. Sallinger, and L. Bellomarini. 2021. Distributed Company Control in Company Shareholding Graphs. In *IEEE 37th International Conference on Data Engineering (ICDE)*. Los Alamitos, CA, USA, 2637–2648.
[13] Jason I. Hong. 2023. Teaching the FATE Community about Privacy. *Commun. ACM* 66, 8 (jul 2023), 10–11.
[14] Dominik K Kanbach, Louisa Heiduk, Georg Blueher, Maximilian Schreiter, and Alexander Lahmann. 2023. The GenAI is out of the bottle: generative artificial intelligence from a business model innovation perspective. *Review of Managerial Science* (2023), 1–32.
[15] Davide Magnanimi and Michela Iezzi. 2022. Ownership Graphs and Reasoning in Corporate Economics. In *EDBT/ICDT Workshops (CEUR Workshop Proceedings)*, Vol. 3135. CEUR-WS.org.
[16] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. 1979. Testing Implications of Data Dependencies. *ACM TODS* 4, 4 (1979), 455–469.
[17] Inderpal Singh Mumick, Hamid Pirahesh, and Raghu Ramakrishnan. 1990. The magic of duplicates and aggregates. In *VLDB*.
[18] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv preprint arXiv:2306.08302* (2023).
[19] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *arXiv preprint arXiv:2309.01029* (2023).