A local perspective on inconsistency in data-graphs

N. Pardal[1,*,†], M. V. Martinez[2,†]

[1]LFCS, School of Informatics, University of Edinburgh, UK
[1]Instituto de Investigación en Inteligencia Artificial (IIIA-CSIC), Barcelona, Spain

Abstract
We propose a family of local inconsistency measures for graph databases subject to Regular Path Constraints, grounded in an origin-based semantics that scopes constraint evaluation to a designated subset of nodes. We formalize this notion, study the properties of the proposed measures with respect to a set of postulates, and illustrate their behavior on a running example. Our framework enables fine-grained inconsistency assessment, surfacing localized violations that global measures may fail to detect.

Keywords
Graph Databases, Inconsistency Measures, Locality

1. Introduction

Measuring inconsistency in knowledge bases is an active area of research in knowledge representation and reasoning [1, 2]. In the context of graph databases, [3] proposed a family of inconsistency measures for graphs subject to Regular Path Constraints (RPCs). These measures, however, are inherently global: they assess inconsistency over the entire graph. The topological structure of graphs suggests that inconsistency may be highly localized, potentially concentrated in small but structurally significant regions, and thus not adequately captured or sometimes diluted by global measures.

Consider a graph database representing relationships between government officials, regulatory bodies, and private financial institutions, where edges capture relationships such as *regulated*, *employed-by*, or *advised*, and integrity constraints encode conflict-of-interest rules. For example, a civil servant who directly regulated a bank and immediately after joined its board of directors constitutes a clear anomaly. Rather than measuring inconsistency across the entire graph, it is natural to restrict attention to nodes representing individuals who held public office with regulatory responsibilities and locally measure the degree to which constraints are violated with respect to them. Aggregating such violations across the selected nodes can surface systemic revolving-door patterns that a global measure might fail to detect.

Motivated by this, we propose a family of *local* inconsistency measures for graph databases under RPCs, grounded in an origin-based semantics [4, 5]. Rather than evaluating constraint violations globally, we scope evaluation to a designated set of origins—a subset of nodes that acts as the local perspective from which inconsistency is assessed. We formalize this notion, define several inconsistency measures under this semantics, and we start to study their properties through a set of postulates adapted from the literature. Our work builds upon and extends the proposal of [3], contributing to a more fine-grained framework for measuring inconsistency in graph-structured data.

2. Preliminaries

Let $\mathbb{P}$ and $\mathbb{A}$ be countably infinite disjoint sets of atomic *propositions* and atomic *programs*, respectively, and let $\Sigma_n \subsetneq \mathbb{P}$ and $\Sigma_e \subsetneq \mathbb{A}$ be finite sets. A *graph database or data-graph* G over Σ_n, Σ_e is a tuple

LINDA Workshop
*Corresponding author.
†These authors contributed equally.
npardal@ed.ac.uk (N. Pardal); vmartinez@iiia-csic.es (M. V. Martinez)
https://homepages.inf.ed.ac.uk/npardal (N. Pardal)
0000-0002-5150-6947 (N. Pardal); 0000-0003-2819-4735 (M. V. Martinez)

$G = (V, E, \{X_p \mid p \in \Sigma_n\})$, where V is a finite set, $E \subseteq V \times \Sigma_e \times V$, and $X_p \subseteq V$ for every $p \in \Sigma_n$. Notice that the pair (V, E) is a traditional edge-labeled graph and the interpretation of the sets X_p is the sets of nodes of G with label p. To ease notation, we will refer to a graph database simply as (V, E, X), and sometimes we will not mention the vertex set if it is clear from the context.

We define two formal languages: one for expressing *locality*, i.e., properties that identify where inconsistency should be evaluated within a graph database, and one for specifying integrity constraints.

The Locality Language. The purpose of the locality language is to express formal sentences that identify nodes in which we wish to focus the study of inconsistency. Here we chose Propositional dynamic logic [6], however note that any language with the power to express node constraints or properties of the nodes is interchangeable. This choice does not affect the results in this paper.

Propositional dynamic logic (PDL) with *converse* (CPDL) is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\pi\rangle \quad \pi ::= \varepsilon \mid a \mid a^- \mid \pi \cup \pi \mid \pi \circ \pi \mid \pi^* \mid \varphi? \tag{1}$$

Although PDL was originally conceived as a logic for reasoning about programs [6], it can be related to several languages used to query or constraint database-like structures. For instance, with PDL formulas, several types of expressions used in description logic can be captured: concept inclusions, functional restrictions, memberships, etc [7]. PDL is equivalent to the DL $\mathscr{ALC}_{reg}$[1].

PDL expressions are usually interpreted on *Kripke structures*, which we can relate to knowledge bases or graphs with node and edge labels. Since we want to analyze properties on graph databases, we define the semantics over finite graphs with node and edge labels. The semantics can be found in the Appendix. CPDL is also quite expressive, since it generalizes modal logic and also basic regular expressions. We use CPDL as a basic language to query a graph database and restrict the semantics over a set of structures that we call *bases*. Thus, node labels will be used only in the process of querying a graph database, since they seem valuable for the recognition and classification of local behavior. When node labels are irrelevant (as it will be when we talk about the constraint language) we will refer to a graph database simply as a pair (V, E).

The Constraint Language. We represent integrity constraints through RPCs [4, 8] over data-graphs. A path from nodes v_1 to v_n in a graph is a sequence $\pi = v_1 a_1 v_2 a_2 v_3 \dots v_{n-1} a_{n-1} v_n$ such that each (v_i, a_i, v_{i+1}), for $i < n$, is an edge in E. We denote by $\lambda(\pi)$ the label of path π, i.e., the word $a_1 \dots a_{n-1} \in \Sigma_e^*$. An RPQ is an expression of the form $x \xrightarrow{L} y$, where L is a regular language over Σ_e (typically represented by a regular expression). Given a data-graph G, the answer to an RPQ Q is the set of pairs of nodes (x, y), denoted by $[\![Q]\!]_G$, such that there is a path π from x to y with $\lambda(\pi) \in L$. We denote by *Paths*(Q, G) the set of all such paths. According to this semantics, RPQ can be seen as the fragment of PDL programs without the testing operator $\varphi?$ as a subexpression. A *regular path constraint* (RPC) is of the form $Q_1 \subseteq Q_2$, where Q_1 and Q_2 are RPQs. A data-graph G satisfies $Q_1 \subseteq Q_2$ if $[\![Q_1]\!]_G \subseteq [\![Q_2]\!]_G$.

Two different ways have been studied to evaluate consistency of a data-graph with respect to a set of RPCs. The so-called *global semantics* [9, 10] takes a global perspective, requiring that every pair of nodes in the data-graph must satisfy the set of constraints. The second one, the *origin semantics* [5, 11], instead asks for satisfaction of the constraints from a specific node called the *origin*; we focus on this notion of consistency in this work and we extend this to simultaneously consider a set of origins $\mathscr{O} \subseteq V$.

Definition 1 (Consistency). *Let G be a data-graph, $\mathscr{R}$ a finite set of RPCs, and $\mathscr{O} \subseteq V$ a set of nodes that we call origins. We say that G is* $\mathscr{O}$-consistent *w.r.t. $\mathscr{R}$, noted as $G \models_{\mathscr{O}} \mathscr{R}$, if $\mathscr{O} \times V_G \subseteq [\![\alpha]\!]_G$ for all $\alpha \in \mathscr{R}$.*

3. Measuring inconsistency locally

The work of [3] defines a set of inconsistency measures and corresponding postulates for graph databases (without node labels) and RPCs, focusing on the notion of global consistency, that is, the measure applies

[1]https://lat.inf.tu-dresden.de/research/papers/2006/BaLu-ML-Handbook-06.pdf

to the graph as a whole. In this section we develop the theoretical machinery to address situations such as the one described in the introduction, where a local treatment of inconsistency measuring seems more appropriate. For this, we assume $\mathscr{C}$ is a set of expressions that allow us to define the origins from where we are interested in measuring the inconsistency.

Contexts and Bases. Given Σ_e and Σ_n, let $G = (V, E, X)$ be a graph database, and $\mathscr{C}$ be a (finite) set of *contextual constraint* formulas in CPDL, both over Σ_n, Σ_e.

Definition 2. *The* $\mathscr{C}$-context subgraph of G *is the maximal subgraph* $W_{\mathscr{C}}$ *of* G *whose nodes satisfy all node expressions in* $\mathscr{C}$ *over* G*. That is,* $x \in V(W_{\mathscr{C}})$ *iff* $x \in [\![\varphi]\!]_G$ *for every* $\varphi \in \mathscr{C}$*; in other words,* $V(W_{\mathscr{C}}) = [\![\bigwedge_{\varphi\in\mathscr{C}} \varphi]\!]_G$*. We call* $V(W_{\mathscr{C}})$ *the* $\mathscr{C}$*-base (or simply the base).*

Intuitively, $\mathscr{C}$ helps identifying a portion of interest of the graph, in which the inconsistency may accumulate or where we are more likely to find issues, e.g., $\mathscr{C}$ pinpoints nodes representing individuals who have held public office with regulatory responsibilities. Assuming only nodes expressions, note that the $\mathscr{C}$-context subgraph could also be merely a subset of V. Moreover, the $\mathscr{C}$-context subgraph does not necessarily witness these constraints when evaluated over itself, i.e., $W_{\mathscr{C}}$ may fail to certify that its nodes belong to $[\![\varphi]\!]$ for $\varphi \in \mathscr{C}$, since restricting the graph may remove edges required to satisfy the formulas. However, we are only interested in the nodes in the context subgraph—that is, the base—and not in its edges, since we will use them as origins to test inconsistency locally.

Example 1. Consider a data-graph representing authors, papers, affiliations, and conferences. The base is the set of *all authors that have a paper in at least one conference.* Consistency is enforced by the rule: *"a published author cannot have a paper in a conference organized by their affiliation".*

Witnesses of inconsistency. We build from the definitions in [3] and redefine concepts for origins. Given G and a constraint α of the form $Q_1 \subseteq Q_2$, the problematic pairs in G w.r.t. α and $\mathscr{O}$, denoted by *ProblematicPairs*$(G, \alpha, \mathscr{O})$, is the set of pairs $(x, y) \in [\![Q_1]\!] \setminus [\![Q_2]\!]$ such that $x \in \mathscr{O}$. Let *ProblematicPairs*$(G, \mathscr{O})$ be the set of problematic pairs of nodes for some $\alpha \in \mathscr{R}$, that is, the union of *ProblematicPairs*$(G, \alpha, \mathscr{O})$ over all $\alpha \in \mathscr{R}$. We also say that $\pi \in$ *ProblematicPaths*$(G, \mathscr{O})$ iff there exists $(x, y) \in$ *ProblematicPairs*$(G, Q_1 \subseteq Q_2, \mathscr{O})$ and $\pi \in$ *Paths*(Q_1, G) for some $Q_1 \subseteq Q_2 \in \mathscr{R}$. A path ϕ is *free* with respect to a set of origins $\mathscr{O}$ iff $\phi \notin$ *ProblematicPaths*$(G, \mathscr{O})$; a pair of nodes (x, y) is *free* iff $(x, y) \notin$ *ProblematicPairs*$(G, \mathscr{O})$; this extends to individual nodes.

Given a graph G, a set of rules $\mathscr{R}$ and a set of origins $\mathscr{O}$, we say N *is a monotonic subgraph of* G w.r.t. $\mathscr{O}$, denoted by $N \subseteq_m^{\mathscr{O}} G$, if *ProblematicPaths*$(N, \mathscr{O}) \subseteq$ *ProblematicPaths*$(G, \mathscr{O})$. We may say that N is monotonic when G, $\mathscr{R}$, and $\mathscr{O}$ are clear from the context.

Now, given fixed set $\mathscr{R}$ of RPCs over Σ_n, we define $\mathscr{O}$-MIMS, a generalization of MIMS from [3], here we interpret them as "(minimal) witnesses of inconsistency" of $\mathscr{R}$ in G given a set of origins $\mathscr{O}$.

Definition 3. *Given a set of origins* $\mathscr{O}$ *and a set of constraints* $\mathscr{R}$*, we say that a subgraph* N *of* G *is a* $\mathscr{O}$-MIMS of G *if: (1)* $N \subseteq_m^{\mathscr{O}} G$*, i.e.,* N *is a monotonic subgraph of* G*, (2)* $N \nvDash_o r$ *for some* $r \in \mathscr{R}$ *and some* $o \in \mathscr{O}$*, and (3) there is no inconsistent (w.r.t.* $\mathscr{R}$ *and all origins), monotonic subgraph* $N' \subset N$ *of* G*.*

Example 2. Let $G = \{a \xrightarrow{p} b,\ b \xrightarrow{p} c,\ c \xrightarrow{p} d,\ d \xrightarrow{p} f,\ a \xrightarrow{q} c\}$, consider the set of origins $\mathscr{O} = \{a, c\}$ and the RPC $\phi : p.p \subseteq q$. Note that $G \vDash_a \phi$ but $G \nvDash_c \phi$. Let $N = \{a \xrightarrow{p} b,\ b \xrightarrow{p} c,\ c \xrightarrow{p} d,\ d \xrightarrow{p} f,\ a \xrightarrow{q} c\}$. Then N is monotonic and $N \nvDash_c \phi$. However, N is not minimal: the subgraph $M = \{c \xrightarrow{p} d,\ d \xrightarrow{p} f\}$ is still monotonic and violates ϕ at c.

Now, consider $N' = \{a \xrightarrow{p} b,\ b \xrightarrow{p} c,\ c \xrightarrow{p} d,\ d \xrightarrow{p} f\}$, which violates ϕ at a, though $G \vDash_a \phi$. This violation is spurious: it arises because removing edges may destroy witnesses of the satisfaction of the right part of the constraint, thereby creating violations not present in G. A witness of inconsistency in this case should be focusing only on the violations related to the origin e, and allow us to precisely identify these in the graph. For this reason, we do not consider subgraphs such as N' to be informative for witnessing the inconsistency of G w.r.t. $\mathscr{O}$: requiring monotonicity in (1) and checking minimality in

(3) w.r.t. *all* origins ensures that no new *pertinent* problematic paths (or pairs or edges) are introduced, and the information contained in the inconsistency witness is “tight”.

4. Inconsistency Measures for Data-graphs

In this section we extend the notion of inconsistency measures for data-graphs based on the origins semantics for consistency.

Let $\mathscr{G}$ be the set of all finite data-graphs given a fixed set of edge labels Σ_e and data labels Σ_n. We also assume a fixed set $\mathscr{R}$ of RPCs over Σ_n. We start by extending the basic definition of inconsistency measure from [3], to account for a set of origins from which inconsistency is measured. Then, we redefine some basic notions of inconsistency measures for data-graphs from [3]. Note that in the original definition, an inconsistency measure maps graph databases to $\mathbb{R}^{\infty}_{\geq 0}$.

Definition 4. *A function* $I : \mathscr{G} \times 2^V \to \mathbb{R}^{\infty}_{\geq 0}$ *is a* local inconsistency measure *iff for all* $G \in \mathscr{G} : I(G, \mathscr{O}) = 0$ *iff* G *is* $\mathscr{O}$-consistent.

The above definition works for an arbitrary set of nodes $\mathscr{O} \subseteq V$; in order to analyze inconsistency from a local point of view given $\mathscr{C}$, we consider $\mathscr{O} = V(W_{\mathscr{C}})$, that is, the base obtained from $\mathscr{C}$ (Definition 2).

In the following we provide some examples of interesting inconsistency measures that can exploit the origins semantics. The first and second to last are direct reinterpretations of inconsistency measures developed in [3], while the rest are defined considering the locality provided by $\mathscr{O}$-consistency.

Definition 5. *Let* $\mathscr{O} = V(W_{\mathscr{C}})$*, and consider* $\mathscr{O}$*-consistency evaluated w.r.t. a set of ICs* $\mathscr{R}$*. The following define a range of local inconsistency measures:*

- $I_B(G, \mathscr{O}) = 1$ *iff* G *is not* $\mathscr{O}$-consistent *(0 otherwise).*
- $I_{B'}(G, \mathscr{O}) = \frac{|\mathscr{O}'|}{|\mathscr{O}|}$*, where* $\mathscr{O}' = \{o \in \mathscr{O} \mid G \not\models_o \mathscr{R}\}$.
- *Let* $C_o = \{\phi \in \mathscr{R} : G \not\models_o \phi\}$*. Then, we have:*
  - $I_C(G, \mathscr{O}) = \max_{o \in \mathscr{O}} |C_o|$*, maximum number of constraints in* $\mathscr{R}$ *that are violated in* G *by a single origin in* $\mathscr{O}$.
  - $I_{C'}(G, \mathscr{O}) = \left|\bigcap_{o \in \mathscr{O}, C_o \neq \varnothing} C_o\right|$*, that is, the number of constraints in* $\mathscr{R}$ *that are violated in* G *from all origins in* $\mathscr{O}$.
  - $I_{C''}(G, \mathscr{O}) = \left|\bigcup_{o \in \mathscr{O}} C_o\right|$*, that is, the number of constraints in* $\mathscr{R}$ *that are violated in* G *for at least one origin in* $\mathscr{O}$.
- $I_M(G, \mathscr{O})$ *is the number of* $\mathscr{O}$*-MIMS in* G *w.r.t.* $\mathscr{O}$.
- $I_{M'}(G, \mathscr{O}) = \max_{o \in \mathscr{O}} I_M(G, \{o\}) = \max_{o \in \mathscr{O}} |o\text{-}MIMS(G)|$.

Intuitively, I_B is a binary measure that simply differentiates between consistent and inconsistent graph databases.. The second, $I_{B'}$, refines this by computing the proportion of origins that witness an inconsistency. The third is a family of measures that are based on the per-origin violated constraint set C_o, yielding three variants: I_C, which measures the maximum number of constraints violated by any single origin, $I_{C'}$, which counts constraints violated from *every* origin, and $I_{C''}$, which counts constraints violated by *at least one* origin. Measure $I_M(G, \mathscr{O})$ counts the number of $\mathscr{O}$-MIMS in G, capturing inconsistency through the structure of minimally $\mathscr{O}$-inconsistent subgraphs, this is a direct extension of the measure $I_M(G)$ proposed in [3]. Finally, $I_{M'}(G, \mathscr{O})$ intends to be more granular by looking at the number of minimal inconsistent monotonic subgraphs for each node separately and aggregating them. Note that looking at $\mathscr{O}$-MIMS for the whole set $\mathscr{O}$ could dilute some of the inconsistencies due to the definition of minimality of $\mathscr{O}$-MIMS across the whole set $\mathscr{O}$. The remaining inconsistency measures from [3] can be expressed straightforwardly in terms of $\mathscr{O}$-based inconsistency.

Example 3. Consider a graph database $G = \{a \xrightarrow{p} b,\ a \xrightarrow{q} b,\ b \xrightarrow{p} c,\ c \xrightarrow{q} d,\ c \xrightarrow{p} d\}$. Let $\mathscr{C} = \{\langle p \cup q \rangle\}$, stating that we want to focus on those vertices with outgoing edges labeled with either p or q. Then,

$\mathscr{C}$-base of G yields origins $\mathcal{O} = \{a, b, c\}$. Let $\mathscr{R} = \{\phi_1 : p \subseteq q, \phi_2 : p \cdot p \subseteq p, \phi_3 : q \cdot p \subseteq q\}$ be the set of integrity constraints. The per-origin violated constraint sets are $C_a = \{\phi_2, \phi_3\}$, $C_b = \{\phi_1, \phi_2\}$, and $C_c = \emptyset$. We can now compute each inconsistency measure:

- $I_B(G, \mathcal{O}) = 1$, since G is not $\mathcal{O}$-consistent.
- $I_{B'}(G, \mathcal{O}) = |\mathcal{O}'|/|\mathcal{O}| = 2/3$, since two origins ($a$ and b) witness a violation.
- $I_C(G, \mathcal{O}) = \max_{o \in \mathcal{O}} |C_o| = |C_a| = 2$, as a violates the most constraints.
- $I_{C'}(G, \mathcal{O}) = |\bigcap_{o \in \mathcal{O}} C_o| = |\{\phi_2\}| = 1$, since only ϕ_2 is in every non-empty C_o.
- $I_{C''}(G, \mathcal{O}) = |\bigcup_{o \in \mathcal{O}} C_o| = |\{\phi_1, \phi_2, \phi_3\}| = 3$, since all constraints are violated by at least one origin.
- $I_M(G, \mathcal{O}) = |\{\{ b \xrightarrow{p} c\}\}| = 1$ counts the number of $\mathcal{O}$-MIMS in G.
- $I_{M'}(G, \mathcal{O}) = \max_{o \in \mathcal{O}} |o\text{-}MIMS(G)| = \max |\{\{\{ a \xrightarrow{p} b,\ b \xrightarrow{p} c\}, \{ a \xrightarrow{q} b,\ b \xrightarrow{p} c\}\}, \{\{ b \xrightarrow{p} c\}\}\}| = 2$ returns the maximum count of o-MIMS in G among all $o \in \mathcal{O}$.

5. Rationality postulates for local inconsistency measures

The following postulates are taken from [3] and reinterpreted considering a set of origins.

Monotony: If $G' \subseteq_m^{\mathcal{O}} G$, then $I(G', \mathcal{O}) \leq I(G, \mathcal{O})$.

Edge Independency: Let $G' = G - \{e\}$, if $G' \subseteq_m^{\mathcal{O}} G$, and e is free w.r.t. $\mathcal{O}$, then $I(G', \mathcal{O}) = I(G, \mathcal{O})$.

Vertex Independency: Let $G' = G - \{v\}$, if $G' \subseteq_m^{\mathcal{O}} G$, and v is free w.r.t. $\mathcal{O}$, then $I(G', \mathcal{O}) = I(G, \mathcal{O})$.

Edge Super-Additivity: If $G_1 = (V_1, E_1, X_1)$, $G_2 = (V_2, E_2, X_2)$, $G_i \subseteq_m^{\mathcal{O}_i} G_1 \cup G_2$ for origin sets $\mathcal{O}_i$, $i = 1, 2$, and $E_1 \cap E_2 = \emptyset$, then $I(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) \geq I(G_1, \mathcal{O}_1) + I(G_2, \mathcal{O}_2)$.

Vertex Super-Additivity: If $G_1 = (V_1, E_1, X_1)$, $G_2 = (V_2, E_2, X_2)$, and $V_1 \cap V_2 = \emptyset$, then $I(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) \geq I(G_1, \mathcal{O}_1) + I(G_2, \mathcal{O}_2)$.

Remark: Note that if $G_1 \models_{\mathcal{O}_1} \mathscr{R}$, then $G_1 \cup G_2$ may not be consistent for $\mathcal{O}_1$, in particular for $\mathcal{O}_1 \cup \mathcal{O}_2$. On the other hand, if $G_1 \not\models_{\mathcal{O}_1} \mathscr{R}$, there is a constraint $r \in \mathscr{R}$ and an origin $o \in \mathcal{O}$ such that $G_1 \not\models_o r$. There is no edge (or vertex) in G_2 that can fix this inconsistency w.r.t. $\mathcal{O}_1$, since we are enforcing monotony $G_1 \subseteq_m^{\mathcal{O}} G_1 \cup G_2$, and thus there cannot be inconsistencies in G_1 that are not in the union.

$\mathcal{O}$-MIMS separability: If $\mathcal{O}_1\text{-}MIMS(G_1) \cap \mathcal{O}_2\text{-}MIMS(G_2)$ and $\mathcal{O}_1 \cup \mathcal{O}_2\text{-}MIMS(G_1 \cup G_2) = \mathcal{O}_1\text{-}MIMS(G_1) \cup \mathcal{O}_2\text{-}MIMS(G_2)$, then $I(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) = I(G_1, \mathcal{O}_1) + I(G_2, \mathcal{O}_2)$.

This postulate captures the conditions under which an inconsistency measure can be computed in a modular fashion. However, when origins-based inconsistency is considered, these conditions become quite demanding, in particular requiring $\mathcal{O}_1 \cup \mathcal{O}_2\text{-}MIMS(G_1 \cup G_2) = \mathcal{O}_1\text{-}MIMS(G_1) \cup \mathcal{O}_2\text{-}MIMS(G_2)$.

$\mathcal{O}$-MIMS Super-Additivity: If $\mathcal{O}_1\text{-}MIMS(G_1) \cap \mathcal{O}_2\text{-}MIMS(G_2) = \emptyset$, $\mathcal{O}_i\text{-}MIMS(G_i) \subseteq \mathcal{O}_1 \cup \mathcal{O}_2\text{-}MIMS(G_1 \cup G_2)$ for $i = 1, 2$, then $I(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) \geq I(G_1, \mathcal{O}_1) + I(G_2, \mathcal{O}_2)$.

$\mathcal{O}$-MIMS Normalization: If $\mathcal{O}\text{-}MIMS(G) = \{G\}$, then $I(G) = 1$.

$I_M(G, \mathcal{O})$ **satisfies all postulates** (Lemma 5 in Appendix). **Monotony**: if $G' \subseteq_m^{\mathcal{O}} G$ then every $\mathcal{O}$-MIMS of G' is also a $\mathcal{O}$-MIMS of G, so $I_M(G', \mathcal{O}) \leq I_M(G, \mathcal{O})$; intuitively, adding edges can only introduce new minimal inconsistent subgraphs. **Edge** and **Vertex Independency:** any element free w.r.t. $\mathcal{O}$ participates in no $\mathcal{O}$-MIM, so removing it leaves the count unchanged. **Edge** and **Vertex Super-Additivity:** when the respective edge or vertex sets are disjoint, no MIMS from one part shares structure with any MIMS from the other, so every MIMS from each part survives in the union and the count of the union is at least the sum of the parts. **$\mathcal{O}$-MIMS Separability:** when the MIMS of G_1 and G_2 are disjoint and their union accounts for all MIMS of $G_1 \cup G_2$ exactly, counting gives $I_M(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) = I_M(G_1, \mathcal{O}_1) + I_M(G_2, \mathcal{O}_2)$. **$\mathcal{O}$-MIMS Super-Additivity:** when the MIMS of each part are disjoint and each is contained in those of the union, the combined origin set $\mathcal{O}_1 \cup \mathcal{O}_2$ may reveal additional inconsistencies not visible from either set individually, yielding a strict inequality. Finally, **$\mathcal{O}$-MIMS Normalization** holds trivially.

Future work. We plan to extend the study of the proposed measures and their relation to postulates, investigate whether the origin-based setting calls for measures and postulates that are intrinsically local—that is, ones that cannot be naturally obtained by adapting existing proposals—and study the computational complexity of computing the proposed measures.

Declaration on Generative AI

The authors used generative AI tools solely for grammar and language revision purposes.

References

[1] J. Grant, Classifications for inconsistent theories, Notre Dame Journal of Formal Logic 19 (1978) 435–444. doi:10.1305/ndjfl/1093888404.

[2] J. Grant, M. V. Martinez, Measuring Inconsistency in Information, volume 73 of *Studies in Logic*, College Publications, London, 2018.

[3] J. Grant, F. Parisi, On measuring inconsistency in graph databases with regular path constraints, Artificial Intelligence 335 (2024) 104197. doi:https://doi.org/10.1016/j.artint.2024.104197.

[4] S. Abiteboul, V. Vianu, Regular path queries with constraints, Journal of Computer and System Sciences 58 (1999) 428–452.

[5] P. Barceló, G. Fontaine, On the data complexity of consistent query answering over graph databases, Journal of Computer and System Sciences 88 (2017) 164–194.

[6] M. J. Fischer, R. E. Ladner, Propositional modal logic of programs, STOC '77, Association for Computing Machinery, New York, NY, USA, 1977, p. 286–294. URL: https://doi.org/10.1145/800105.803418. doi:10.1145/800105.803418.

[7] G. D. Giacomo, M. Lenzerini, Boosting the correspondence between description logics and propositional dynamic logics, in: Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence, AAAI'94, AAAI Press, 1994, p. 205–212.

[8] P. Barceló Baeza, Querying graph databases, in: Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, ACM, 2013, p. 175–188.

[9] A. Deutsch, V. Tannen, Optimization properties for classes of conjunctive regular path queries, in: International Workshop on Database Programming Languages, Springer, 2001, pp. 21–39.

[10] G. Grahne, A. Thomo, Query containment and rewriting using views for regular path queries under constraints, in: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2003, pp. 111–122.

[11] D. Calvanese, M. Ortiz, M. Šimkus, Verification of evolving graph-structured data under expressive path constraints, in: 19th International Conference on Database Theory (ICDT 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

Appendix

PDL Semantics. The semantics for (1) is given by:

$$\begin{array}{rlrl}
[\![p]\!]_G \overset{\text{def}}{=} X_p \text{ for } p \in \Sigma_n, & & [\![a]\!]_G \overset{\text{def}}{=} \{(u,v) \in V^2 \mid (u,a,v) \in E\} \\
[\![\neg\varphi]\!]_G \overset{\text{def}}{=} V \setminus [\![\varphi]\!]_G, & & [\![\bar{a}]\!]_G \overset{\text{def}}{=} \{(v,u) \in V^2 \mid (u,v) \in [\![a]\!]_G\} \\
[\![\varphi_1 \wedge \varphi_2]\!]_G \overset{\text{def}}{=} [\![\varphi_1]\!]_G \cap [\![\varphi_2]\!]_G & & [\![\pi_1 \star \pi_2]\!]_G \overset{\text{def}}{=} [\![\pi_1]\!]_G \star [\![\pi_2]\!]_G \text{ for } \star \in \{\cup, \circ\} \\
[\![\langle\pi\rangle]\!]_G \overset{\text{def}}{=} \{u \in V \mid \exists v \in V.(u,v) \in [\![\pi]\!]_G\} & & [\![\pi^*]\!]_G \overset{\text{def}}{=} \text{reflexive transitive closure of } [\![\pi]\!]_G \\
[\![\varepsilon]\!]_G \overset{\text{def}}{=} \{(u,u) \mid u \in V\} & & [\![\varphi?]\!]_G \overset{\text{def}}{=} \{(u,u) \mid u \in V. u \in [\![\varphi]\!]_G\}
\end{array}$$

About the postulates. The following example illustrates a case in which one of the conditions in $\mathcal{O}$-MIMS separability fails, yet we can still relate the union's inconsistency measure to the sum of the individual measures.

Example 4. Consider a graph database G_1 with nodes $\{a, b, c, d, e\}$ and edges:

$$G_1 = \{a \xrightarrow{p} b,\ e \xrightarrow{p} b,\ b \xrightarrow{p} d,\ c \xrightarrow{q} d\}$$

and G_2 with nodes $\{a, b, c, d, e\}$ and edges:

$$G_2 = \{a \xrightarrow{r} e,\ e \xrightarrow{p} b,\ b \xrightarrow{p} d\}$$

Let $\mathscr{R} = \{\phi\}$ where ϕ is the following RPC:

$$\phi : p \cdot p \subseteq q \cdot p$$

If we take $\mathcal{O}_1 = \{a\}$ then, it must be the case that for every path from a (resp. if we take $\mathcal{O}_2 = \{e\}$, then it must be the case that for every path from e) going through two consecutive edges labeled with p to a node v, there must exist another path from a (resp. from e) that goes to v through an edge labeled with q followed by and edge labeled with p.

We evaluate ϕ from each origin. In G_1, the path $a \xrightarrow{p} b \xrightarrow{p} d$ witnesses a violation from a, since no q-edge leads into b from a's perspective, so $G_1 \nvDash_a \phi$. Similarly, in G_2 the path $e \xrightarrow{p} b \xrightarrow{p} d$ witnesses a violation from e, so $G_2 \nvDash_e \phi$.

Moreover:

$$a\text{-}MIMS(G_1) = \{\{a \xrightarrow{p} b,\ b \xrightarrow{p} d\}\}$$

$$e\text{-}MIMS(G_2) = \{\{e \xrightarrow{p} b,\ b \xrightarrow{p} d\}\}$$

and therefore $a\text{-}MIMS(G_1) \cap e\text{-}MIMS(G_2) = \emptyset$. Letting $G = G_1 \cup G_2$, we also have $G \nvDash_{\{a,e\}} \phi$, and:

$$(\mathcal{O}_1 \cup \mathcal{O}_2)\text{-}MIMS(G) = a\text{-}MIMS(G_1) \cup e\text{-}MIMS(G_2)$$

So the separability condition is satisfied, and accordingly, in this case, it makes sense that

$$I(G_1 \cup G_2,\ \mathcal{O}_1 \cup \mathcal{O}_2) = I(G_1, \mathcal{O}_1) + I(G_2, \mathcal{O}_2)$$

Now consider a variant where:

$$G_2 = \{a \xrightarrow{p} e,\ f \xrightarrow{p} b,\ e \xrightarrow{p} b,\ b \xrightarrow{p} d\}$$

over nodes $\{a, b, c, d, e, f\}$. Here, $G_2 \nvDash_e \phi$ still holds. However, it contains an additional path $a \xrightarrow{p} e \xrightarrow{p} b$ that also violates ϕ from the origin a. Therefore, when we consider $G_1 \cup G_2$, we have that $\mathcal{O}_1\text{-}MIMS(G_1) \cup \mathcal{O}_2\text{-}MIMS(G_2)$ no longer coincides with $(\mathcal{O}_1 \cup \mathcal{O}_2)\text{-}MIMS(G_1 \cup G_2)$: the path $a \xrightarrow{p} e \xrightarrow{p} b$ is not captured by $e\text{-}MIMS(G_2)$, as $e\text{-}MIMS(G_2)$ only tracks violations from the origin e, but it is captured in the union. In this case, the separability condition fails, yet it is natural to require that the inconsistency of the union is at least as large as the sum of its parts:

$$I(G_1 \cup G_2,\ \mathcal{O}_1 \cup \mathcal{O}_2) \geq I(G_1, \mathcal{O}_1) + I(G_2, \mathcal{O}_2)$$

Lemma 6. *$I_M(G, \mathcal{O})$, defined as the number of $\mathcal{O}$-MIMS in G, satisfies all postulates.*

Proof Sketch: **Monotony.** If $G' \subseteq_m^{\mathcal{O}} G$, then by the definition of monotonic subgraph and $\mathcal{O}$-MIMS we have that every $\mathcal{O}$-MIMS of G' is also an $\mathcal{O}$-MIMS of G, so $I_M(G', \mathcal{O}) \leq I_M(G, \mathcal{O})$. Intuitively, being monotonic means that adding edges can only introduce new minimal inconsistent subgraphs, never remove existing ones.

Edge Independency. If e is free w.r.t. $\mathcal{O}$, then e does not participate in any $\mathcal{O}$-MIMS of G: it neither witnesses nor enables any violation from any origin in $\mathcal{O}$. Removing it therefore leaves all $\mathcal{O}$-MIMS intact, so $I_M(G - \{e\}, \mathcal{O}) = I_M(G, \mathcal{O})$.

Vertex Independency. Analogous to edge independency: if v is free w.r.t. $\mathcal{O}$, it does not appear in any $\mathcal{O}$-MIM, and removing it does not affect the count.

Edge Super-Additivity. If $E_1 \cap E_2 = \emptyset$, then no $\mathcal{O}_1$-MIMS of G_1 shares an edge with any $\mathcal{O}_2$-MIMS of G_2. Moreover, even though G_1 and G_2 could share vertices, $G_i \subseteq_m^{\mathcal{O}_i} G_1 \cup G_2$ means that there cannot be inconsistencies in G_1 w.r.t. $\mathcal{O}_1$ (resp. G_2 w.r.t. $\mathcal{O}_2$) that are not in the union, and this implies that no edge of G_2 (resp. G_1) will fix an $\mathcal{O}_1$-MIMS of G_1 (resp. $\mathcal{O}_2$-MIMS of G_2). Thus, every MIMS from each part survives in the union, and the union has at least as many MIMS as the sum of the parts: $I_M(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) \geq I_M(G_1, \mathcal{O}_1) + I_M(G_2, \mathcal{O}_2)$.

Vertex Super-Additivity. If $V_1 \cap V_2 = \emptyset$, the two subgraphs are completely disjoint, so no new inconsistencies can arise across them. Each MIMS from G_1 and G_2 remains a MIMS in the union, giving the same inequality.

$\mathcal{O}$**-MIMS Separability.** If $\mathcal{O}_1\text{-}MIMS(G_1) \cap \mathcal{O}_2\text{-}MIMS(G_2) = \emptyset$ and $(\mathcal{O}_1 \cup \mathcal{O}_2)\text{-}MIMS(G_1 \cup G_2) = \mathcal{O}_1\text{-}MIMS(G_1) \cup \mathcal{O}_2\text{-}MIMS(G_2)$, then the MIMS of the union are exactly those of the parts, with no overlap and no new ones introduced. Counting them gives $I_M(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) = I_M(G_1, \mathcal{O}_1) + I_M(G_2, \mathcal{O}_2)$ directly.

$\mathcal{O}$**-MIMS Super-Additivity.** If the MIMS of G_1 and G_2 are disjoint and each is contained in the MIMS of the union, then the union has at least the same MIMS as either part alone—the combination of origins $\mathcal{O}_1 \cup \mathcal{O}_2$ may, however, reveal additional inconsistencies not visible from either origin set individually, so $I_M(G_1 \cup G_2, \mathcal{O}_1 \cup \mathcal{O}_2) \geq I_M(G_1, \mathcal{O}_1) + I_M(G_2, \mathcal{O}_2)$.

$\mathcal{O}$**-MIMS Normalization.** If $\mathcal{O}\text{-}MIMS(G) = \{G\}$, then G itself is the unique maximally inconsistent subgraph, so $I_M(G, \mathcal{O}) = 1$ by definition.