Sequent Calculi for Data-Aware Modal Logics

Carlos Areces
Universidad Nacional de Córdoba
and CONICET, Argentina
carlos.areces@unc.edu.ar

Valentin Cassano
Universidad Nacional de Río Cuarto
and CONICET, Argentina
valentin@dc.exa.unrc.edu.ar

Danae Dutto
Universidad Nacional de Córdoba
and CONICET, Argentina
ddutto@dc.exa.unrc.edu.ar

Raul Fervari
Universidad Nacional de Córdoba
and CONICET, Argentina
rfervari@unc.edu.ar

Data-aware modal logics offer a powerful formalism for reasoning about semi-structured queries in languages such as DataGL, XPath, and GQL. In brief, these logics can be viewed as modal systems capable of expressing both reachability statements and data-aware properties, such as value comparisons. One particularly expressive logic in this landscape is $\mathrm{HXPath}_{\mathrm{D}}$, a hybrid modal logic that captures not only the navigational core of XPath but also data comparisons, node labels (keys), and key-based navigation operators. While previous work on $\mathrm{HXPath}_{\mathrm{D}}$ has primarily focused on its model-theoretic properties, in this paper we approach $\mathrm{HXPath}_{\mathrm{D}}$ from a proof-theoretic perspective. Concretely, we present a sound and complete Gentzen-style sequent calculus for $\mathrm{HXPath}_{\mathrm{D}}$. Moreover, we show all rules in this calculus are invertible, and that it enjoys cut elimination. Our work contributes to the proof-theoretic foundations of data-aware modal logics, and enables a deeper logical analysis of query languages over graph-structured data. Moreover, our results lay the groundwork for extending proof-theoretic techniques to a broader class of modal systems.

1 Introduction

Semi-structured data organization [18] is reemerging as a flexible paradigm for representing and storing information, offering a contrast to the rigid rows-and-columns model of traditional relational databases. This approach is exemplified by *graph databases*, where information is represented and stored in a semi-structured form as *data graphs*, in which nodes and edges are labeled with values that capture both structure and data content [36]. Graph databases are widely adopted in domains such as the web, social networks, fraud detection, and recommendation engines, and are implemented in tools like Neo4j and Amazon Neptune. To support such range of applications, graph databases rely on expressive query languages designed to navigate and extract information from their underlying structures. However, unlike relational databases, where SQL serves as the *de facto* standard, query languages for graph databases are still evolving. Recent efforts, as discussed in [27, 28], are converging toward the Graph Query Language (GQL) Standard [30], which aims to unify existing industrial approaches. GQL builds on its academic predecessor G-CORE [7] and draws conceptual inspiration from Regular Path Queries [32] and the XML Path Language (XPath) [31], offering expressive capabilities for navigating graph structures and performing equality and inequality tests on data. This connection highlights the importance of logical foundations in understanding and advancing modern query languages for graph data.

In this article, we focus on XPath, approaching it from a logical perspective and building on prior work that formalizes fragments of this language as modal logics. Below, we outline a brief timeline highlighting key developments in this direction.

Submitted to:
LSFA 2025

Initial foundational results [15, 20], showed that the navigational fragment of XPath—known as Core-XPath [29]—is strictly less expressive than Propositional Dynamic Logic (PDL) over trees. This set the stage for subsequent work, ending with a complete axiomatization for Core-XPath in [20, 19]. As the study of XPath progressed, attention turned to features that go beyond pure navigation. Querying data graphs often requires the ability to reason about and compare data values—a limitation of purely navigational logics. How to properly represent and manipulate data within a modal setting gave rise to the development of *data-aware modal logics*: extensions of modal logic that incorporate operators for handling data alongside navigation. These logics are tailored to capturing the interplay between structure and data in graph-based systems, integrating mechanisms such as data comparison, binding, and quantification within the modal framework. This includes the development of expressive yet decidable languages, the design of automata-theoretic characterizations, and the identification of suitable semantics that balance expressive power with computational tractability (see e.g., [21]). The result is a family of expressive formalisms capable of modeling sophisticated queries over data-rich systems.

These efforts have opened up new avenues in both theoretical studies and practical applications, particularly in areas such as verification, knowledge representation, and query languages for graph databases. For instance, [16] introduces Core-Data-XPath as an extension of Core-XPath with equality and inequality comparisons. The satisfiability problem and complexity aspects of fragments of these logics are explored in [26, 22, 31, 23], while the model theory and bisimulation algorithms for this kind of logics are studied in [4, 24, 25, 1, 5, 2]. Axiomatic systems extending those from [19] are presented in [3, 6]. This theoretical groundwork led to concrete applications. For example, [29] discusses newly proposed algorithms with existing XPath processors, while the satisfiability methods in [23] can be used to check consistency in XML documents. Furthermore, the bisimulation algorithms in [25, 1] have applications in database minimization, while [13] examines how the complexity of fragments impacts real-world applications.

The work in [10] introduces a novel extension to data-aware modal logics by incorporating *keys*—or labels—for nodes, and *jump-to-key* operations. This results in the logic we refer to as $\mathrm{HXPath}_{\mathrm{D}}$, which extends Core-Data-XPath with expressive constructs from Hybrid Logic. In particular, nominals and the satisfiability operator @, well-studied in Hybrid Logics [9], provide the means to refer directly to specific nodes, and to "navigate" to specific nodes. This addition enables $\mathrm{HXPath}_{\mathrm{D}}$ to express properties and navigation patterns not previously captured in earlier formalisms. Furthermore, the hybrid features of $\mathrm{HXPath}_{\mathrm{D}}$ support a novel axiomatization, with completeness established using Henkin-style models.

In this article, we contribute to the logical study of data-aware modal logics from a *proof-theoretic* perspective by developing a sequent calculus for $\mathrm{HXPath}_{\mathrm{D}}$. We establish the completeness of our calculus by leveraging the completeness results in [10]. We also show that our calculus enjoys cut-elimination and invertibility of rules. Furthermore, we relate our work to research on labeled modal sequent calculi with equality [35] and the proof theory of hybrid logic [17]. Developing a proof theory for modal logics is notoriously difficult—see, e.g., the discussion in [34]. Several techniques, such as display calculi [38], hypersequents [12], deep inference [37], labeled systems [33], and hybridization [17], have been explored to establish a solid proof-theoretic foundation for various modal logics. Our results suggest that labeling and hybridization can also support a well-grounded proof theory for data-aware modal logics.

Outline. Sec. 2 introduces the logic $\mathrm{HXPath}_{\mathrm{D}}$. Sec. 3 presents the sequent calculus **G** for $\mathrm{HXPath}_{\mathrm{D}}$. Key properties of **G**—soundness, completeness, and invertibility of rules—are addressed in Secs. 3.1 to 3.3. Cut elimination is established in Sec. 4. Sec. 5 demonstrates a correspondence between **G** and the sequent calculus for Basic Hybrid Logic from [17]. Sec. 6 concludes with final remarks and directions for future work. Details and omitted proofs are available in [11].

2 Hybrid XPath with Data

We begin by presenting the formal syntax and semantics of $\mathrm{HXPath_D}$—further details are found in [10]. In our presentation, we assume Prop, Nom, Mod, and Cmp be pairwise disjoint sets of symbols for propositions, nominals, modalities, and data comparisons, respectively. Moreover, we assume that Mod and Cmp are finite, while Prop and Nom are countably infinite.

Definition 1. *The language of* $\mathrm{HXPath_D}$ *has* path *expressions (denoted α, β, ...) and* node *expressions (denoted φ, ψ, ...), mutually defined by the grammar:*

$$\alpha,\beta := \mathsf{a} \mid i{:} \mid \varphi? \mid \alpha\beta$$
$$\varphi,\psi := p \mid i \mid \bot \mid \varphi \to \psi \mid @_i\varphi \mid \langle \mathsf{a}\rangle\varphi \mid \langle \alpha =_{\mathsf{c}} \beta\rangle \mid \langle \alpha \neq_{\mathsf{c}} \beta\rangle,$$

where $p \in \mathsf{Prop}$, $i \in \mathsf{Nom}$, $\mathsf{a} \in \mathsf{Mod}$, and $\mathsf{c} \in \mathsf{Cmp}$.[1] *For path expressions, we use $\varepsilon := \top?$ to indicate the* empty *path. For node expressions, we use standard abbreviations: $\top := \bot \to \bot$, $\neg\varphi := \varphi \to \bot$, $\varphi \vee \psi := \neg\varphi \to \psi$, $\varphi \wedge \psi := \neg(\varphi \to \neg\psi)$, and $\varphi \leftrightarrow \psi := (\varphi \to \psi) \wedge (\psi \to \varphi)$. We also abbreviate: $\langle j{:}\rangle\varphi := @_j\varphi$, $\langle \psi?\rangle\varphi := \psi \wedge \varphi$, $\langle \alpha\beta\rangle\varphi := \langle\alpha\rangle\langle\beta\rangle\varphi$, and $[\alpha]\varphi := \neg\langle\alpha\rangle\neg\varphi$. These abbreviations complete all cases to allow arbitrary paths α inside a unary modality $\langle\alpha\rangle$. Finally, we abbreviate $[\alpha \blacktriangle \beta] := \neg\langle \alpha \blacktriangledown \beta\rangle$. In the last abbreviation, we use $\blacktriangle$ when there is no need to distinguish $=_{\mathsf{c}}$ and $\neq_{\mathsf{c}}$, and use $\blacktriangledown$ to indicate $\neq_{\mathsf{c}}$ if $\blacktriangle$ is $=_{\mathsf{c}}$, and to indicate $=_{\mathsf{c}}$ if $\blacktriangle$ is $\neq_{\mathsf{c}}$.*

Path and node expressions are interpreted over hybrid data models.

Definition 2. *A* (hybrid data) model *is a tuple $\mathfrak{M} = \langle \mathrm{N}, \{\mathrm{R_a}\}_{\mathsf{a}\in\mathsf{Mod}}, \{\approx_{\mathsf{c}}\}_{\mathsf{c}\in\mathsf{Cmp}}, g, \mathrm{V}\rangle$, where N is a non-empty set of nodes; each $\mathrm{R_a}$ is a (binary)* accessibility relation *on N; each $\approx_{\mathsf{c}}$ is an equivalence relation on N, called a* comparison*; $g : \mathsf{Nom} \to \mathrm{N}$ is a* nominal assignment*; and $\mathrm{V} : \mathsf{Prop} \to 2^{\mathrm{N}}$ is a* valuation.

The satisfiability relation for path and node expressions is as follows.

Definition 3. *Let $\mathfrak{M} = \langle \mathrm{N}, \{\mathrm{R_a}\}_{\mathsf{a}\in\mathsf{Mod}}, \{\approx_{\mathsf{c}}\}_{\mathsf{c}\in\mathsf{Cmp}}, g, \mathrm{V}\rangle$ be a model, and let $\{n,n'\} \subseteq \mathrm{N}$. The satisfiability relation $\Vdash$ is given by the following conditions:*

$$\begin{array}{ll}
\mathfrak{M},n,n' \Vdash \mathsf{a} & \textit{iff } n\mathrm{R_a}n' \\
\mathfrak{M},n,n' \Vdash i{:} & \textit{iff } g(i) = n' \\
\mathfrak{M},n,n' \Vdash \varphi? & \textit{iff } n = n' \textit{ and } \mathfrak{M},n \Vdash \varphi \\
\mathfrak{M},n,n' \Vdash \alpha\beta & \textit{iff exists } n'' \in \mathrm{N} \textit{ s.t. } \mathfrak{M},n,n'' \Vdash \alpha \textit{ and } \mathfrak{M},n'',n' \Vdash \beta \\
\mathfrak{M},n \Vdash p & \textit{iff } n \in \mathrm{V}(p) \\
\mathfrak{M},n \Vdash i & \textit{iff } g(i) = n \\
\mathfrak{M},n \Vdash \bot & \textit{never} \\
\mathfrak{M},n \Vdash \varphi \to \psi & \textit{iff } \mathfrak{M},n \Vdash \varphi \textit{ implies } \mathfrak{M},n \Vdash \psi \\
\mathfrak{M},n \Vdash @_i\varphi & \textit{iff } \mathfrak{M},g(i) \Vdash \varphi \\
\mathfrak{M},n \Vdash \langle\mathsf{a}\rangle\varphi & \textit{iff exists } n' \in \mathrm{N} \textit{ s.t. } \mathfrak{M},n,n' \Vdash \mathsf{a} \textit{ and } \mathfrak{M},n' \Vdash \varphi \\
\mathfrak{M},n \Vdash \langle \alpha =_{\mathsf{c}} \beta\rangle & \textit{iff exists } n',n'' \in \mathrm{N} \textit{ s.t. } \mathfrak{M},n,n' \Vdash \alpha,\ \mathfrak{M},n,n'' \Vdash \beta \textit{ and } n' \approx_{\mathsf{c}} n'' \\
\mathfrak{M},n \Vdash \langle \alpha \neq_{\mathsf{c}} \beta\rangle & \textit{iff exists } n',n'' \in \mathrm{N} \textit{ s.t. } \mathfrak{M},n,n' \Vdash \alpha,\ \mathfrak{M},n,n'' \Vdash \beta \textit{ and } n' \not\approx_{\mathsf{c}} n''.
\end{array}$$

Let Ψ be a set of node expressions, we use $\mathfrak{M},n \Vdash \Psi$ to indicate $\mathfrak{M},n \Vdash \psi$ for all $\psi \in \Psi$. We say Ψ is satisfiable *iff there exists $\mathfrak{M},n$ s.t. $\mathfrak{M},n \Vdash \Psi$. We call a node expression φ a* consequence *of Ψ, written $\Psi \vDash \varphi$, iff $\Psi \cup \{\neg\varphi\}$ is unsatisfiable. If Ψ is the empty set, we write $\vDash \varphi$ and call φ a* tautology.

Proposition 4, which follows directly from Def. 3, confirms that the abbreviations behave as intended.

[1] Unlike [10], we treat $@_i\varphi$ and $\langle\mathsf{a}\rangle\varphi$ as primitive. This choice simplifies the formulation of the sequent calculus in Sec. 3.

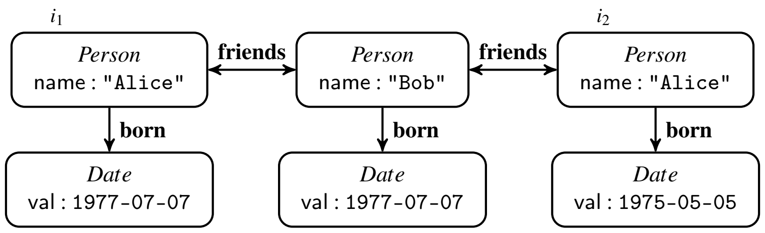

Figure 1: Example of a Data Graph

Proposition 4. *$\mathfrak{M},n \Vdash \langle\alpha\rangle\varphi$ iff exists $n' \in \mathrm{N}$ s.t., $\mathfrak{M},n,n' \Vdash \alpha$ and $\mathfrak{M},n' \Vdash \varphi$. Moreover, $\mathfrak{M},n \Vdash [\alpha =_{\mathsf{c}} \beta]$ iff for all $n',n'' \in \mathrm{N}$, $\mathfrak{M},n,n' \Vdash \alpha$ and $\mathfrak{M},n,n'' \Vdash \beta$ imply $n' \approx_{\mathsf{c}} n''$. Finally, $\mathfrak{M},n \Vdash [\alpha \neq_{\mathsf{c}} \beta]$ iff for all $n',n'' \in \mathrm{N}$, $\mathfrak{M},n,n' \Vdash \alpha$ and $\mathfrak{M},n,n'' \Vdash \beta$ imply $n' \not\approx_{\mathsf{c}} n''$.*

Some intuitive remarks are helpful here. Path expressions indicate reachability relations on nodes, i.e., nodes n and n' satisfy the path expression α if the pair (n,n') is in the relation defined by α; or, equivalently, if n' is reachable from n following an α path. The test expression φ? checks whether the node expression φ holds in a given point in a path. The jump-to-key expression i: resets the current path to start from the node whose key is i. Node expressions p, i, $\bot$, $\varphi \to \psi$, $@_i\varphi$, and $\langle \mathsf{a}\rangle\varphi$, have classical readings (see [9]). The novel data comparison operator $\langle\alpha =_{\mathsf{c}} \beta\rangle$ (resp. $\langle\alpha \neq_{\mathsf{c}} \beta\rangle$) requires the existence of paths α and β, from the current point of evaluation, whose end nodes are in the relation $\approx_{\mathsf{c}}$ (resp. $\not\approx_{\mathsf{c}}$) relation. This mirrors how XPath performs data comparisons over data graphs.

We conclude the presentation of HXPath$_{\mathrm{D}}$ by illustrating how the logic functions as a semi-structured query language over data graphs. The idea is that hybrid data models represent data graphs, while node expressions represent queries that specify both structural properties (e.g., reachability) and data-aware conditions (e.g., comparisons between node values). The following example clarifies this intuition.

Example 1. Fig. 1 shows a simple example of a data graph. We will not formally define data graphs here, see [10] for further details. Intuitively, we can see that the graph encodes information about people and their birth dates, and it also includes a friendship relation. Notice that the graph encodes information of different types: propositional information (certain nodes are defined as Persons, others are Dates), relational information (e.g., the **born** relation encodes the birthdate of a person), and concrete data encoded as pairs *attribute:data* (e.g., one of the persons in the data graph is named `"Alice"`). Moreover, the labels i_1 and i_2 are indexes, that allows direct access to certain nodes in the data graph.

The information in the data graph shown in Fig. 1 can be directly encoded as a hybrid data model. For the structure of the data graph this is immediate. The propositional content in the data graph is captured using proposition symbols, while node indices are managed via the nominal assignment. Crucially, concrete data values are abstracted and represented through data comparisons. Namely, for each attribute c, we introduce in the model a comparison relation $\approx_{\mathsf{c}}$ such that $n_1 \approx_{\mathsf{c}} n_2$ iff there is a data value d such that c : d (as shown in the figure) is both in n_1 and n_2—that is, n_1 and n_2 have the same value d in attribute c. The following hybrid data model corresponds to the data graph in Fig. 1.

$$\begin{array}{rll}
\mathrm{N} = \{n_1,n_2,n_3,n_4,n_5,n_6\} & \mathrm{V}(\mathit{Person}) = \{n_1,n_2,n_3\} & \approx_{\mathsf{name}} = \{(n_1,n_3)\}^{\mathsf{eq}} \\
\mathrm{R}_{\mathbf{friends}} = \{(n_1,n_2),(n_2,n_1),(n_2,n_3),(n_3,n_2)\} & \mathrm{V}(\mathit{Date}) = \{n_4,n_5,n_6\} & \approx_{\mathsf{val}} = \{(n_4,n_5)\}^{\mathsf{eq}} \\
\mathrm{R}_{\mathbf{born}} = \{(n_1,n_4),(n_2,n_5),(n_3,n_6)\} & g(i_1) = n_1 & g(i_2) = n_3
\end{array}$$

We use R^{eq} to denote the closure of R under reflexivity, symmetry and transitivity. The equivalence relations $\approx_{\mathsf{name}}$ and $\approx_{\mathsf{val}}$ are sufficient for the type of data operations permitted in HXPath$_{\mathrm{D}}$.

With the hybrid data model in place, we now turn to how node expressions act as queries. The table below presents a few example properties, along with their formalizations as node expressions:

PROPERTY	FORMALIZATION
The person with key i_1 is a friend of someone born on the same day.	$\langle i_1\colon \mathbf{born}\,(\mathit{Date}?) =_{\mathsf{val}} i_1\colon \mathbf{friends}\,\mathbf{born}\,(\mathit{Date}?)\rangle$
The person with key i_2 shares a birth date with none of her friends.	$[i_2\colon \mathbf{born}\,(\mathit{Date}?) \neq_{\mathsf{val}} i_2\colon \mathbf{friends}\,\mathbf{born}\,(\mathit{Date}?)]$
The persons with keys i_1 and i_2 have the same name but different birth dates.	$\langle i_1\colon (\mathit{Person}?) =_{\mathsf{name}} i_2\colon (\mathit{Person}?)\rangle \wedge \langle i_1\colon \mathbf{born}\,(\mathit{Date}?) \neq_{\mathsf{val}} i_2\colon \mathbf{born}\,(\mathit{Date}?)\rangle$

Each of the node expressions above evaluates to true at every point in the hybrid data model constructed from the example graph database. This showcases how $\mathrm{HXPath_D}$ combines structural navigation with data-aware comparisons to express rich, semantically meaningful queries.

To sum up, $\mathrm{HXPath_D}$ is of interest both in practice and in theory. On the practical side, it faithfully captures a fragment of XPath involving data-aware queries and key-based navigation. On the theoretical side, it poses new challenges for modal logic: its comparison modalities are complex as they combine navigation modalities with (in)equality reasoning. Despite these challenges, in the next section we show that an elegant sequent calculus for $\mathrm{HXPath_D}$ can indeed be defined.

3 A Sequent Calculus for $\mathrm{HXPath_D}$

In this section, we present our Gentzen-style sequent calculus **G** for $\mathrm{HXPath_D}$. The calculus draws inspiration from the modal system with equality developed in [35], but extends it to accommodate the distinctive features of $\mathrm{HXPath_D}$: nominals, satisfiability modalities, path expressions, and data comparisons. In brief, we build **G** on sequents $\Gamma \vdash \Delta$, where Γ, Δ is a set of node expressions in $\mathrm{HXPath_D}$. Sequents capture the idea that if all node expressions in Γ hold, then at least one node expression in Δ must also hold. The inference rules of **G** systematically decompose the expressions in sequents, closely mirroring the semantic behavior of each logical connective. Out aim is to construct a proof system that addresses the entailment problem for the logic, and to provide a rule-based decomposition of the meaning of its logical connectives. As we will see also, the calculus is particularly well-suited for structural analysis of proofs. With the basic ideas in place, let us begin by: defining sequents, introducing the inference rules, and outlining the construction of derivations in **G**.

Definition 5. *A* sequent *is a pair $\Gamma \vdash \Delta$, where Γ and Δ are finite, possibly empty sets of node expressions of the form $\langle i\colon \blacktriangle\ j\colon\rangle$ or $@_i\varphi$.*[2] *In a sequent $\Gamma \vdash \Delta$, the set Γ is called the* antecedent *and the set Δ is called the* consequent*. In turn, a* rule *is a pair $(\Gamma \vdash \Delta, \{\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n\})$, where $\Gamma \vdash \Delta$ is called the* conclusion*, and $\{\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n\}$ the set of* premisses *of the rule. If the set of premisses is empty, we say that the rule is an* axiom*. In each rule, certain node expressions in the conclusion sequent are designated as* principal*. These are the node expressions the rule acts upon. The remaining node expressions in the sequent are the* context *of the rule, and are carried unchanged across its application. This distinction enables the rule to isolate and analyze the logical structure of the principal formulas while treating the context uniformly. The rules of* **G** *are shown in Fig. 2, in a standard format—i.e., the conclusion is shown under a line, the premisses are above the line in no particular order, the principal node expressions and the context are identified immediately from the presentation.*

[2]Following standard notation, we will not use curly brackets when writing down sequents.

With sequents and rules defined, we proceed to formalize the notion of a derivation in **G**.

Definition 6. *A* derivation *in* **G** *is a sequent-labeled tree constructed according to the rules in Fig. 2. More precisely, each non-leaf node in the tree is a sequent that is the conclusion of a rule in* **G**, *its immediate successors in the tree (going upwards) are the premisses of that rule. The root of the tree is called the* end-sequent *of the derivation. A sequent is* derivable *if it is the end-sequent of some derivation, and it is* provable *if it has a derivation whose leaves are all instances of (Ax) or* $(\bot)$. *We use* $\Gamma \vdash_{\mathbf{G}} \Delta$ *to indicate that* $\Gamma \vdash \Delta$ *is provable. A rule is* derived *iff there is a derivation of the conclusion of the rule whose leaves are either axioms or belong to the premisses of the rule.*

PROPOSITIONAL RULES

$$\dfrac{}{\varphi,\Gamma\vdash\Delta,\varphi}\,(\text{Ax})^{\ddagger} \qquad \dfrac{}{@_i\bot,\Gamma\vdash\Delta}\,(\bot) \qquad \dfrac{\Gamma\vdash\Delta,@_i\varphi \quad @_i\psi,\Gamma\vdash\Delta}{@_i(\varphi\to\psi),\Gamma\vdash\Delta}\,(\to\text{L}) \qquad \dfrac{@_i\varphi,\Gamma\vdash\Delta,@_i\psi}{\Gamma\vdash\Delta,@_i(\varphi\to\psi)}\,(\to\text{R})$$

‡ φ is of the form $@_i p$, $@_i j$, or $\langle i{:} =_{\mathsf{c}} j{:}\rangle$

RULES FOR NOMINALS

$$\dfrac{@_i i,\Gamma\vdash\Delta}{\Gamma\vdash\Delta}\,(@\text{T}) \qquad \dfrac{@_j k, @_i j, @_i k,\Gamma\vdash\Delta}{@_i j, @_i k,\Gamma\vdash\Delta}\,(@5) \qquad \dfrac{@_i j,\Gamma\vdash\Delta}{\Gamma\vdash\Delta}\,(\text{Nom})^{\dagger}$$

$$\dfrac{@_j\varphi, @_i j, @_i\varphi,\Gamma\vdash\Delta}{@_i j, @_i\varphi,\Gamma\vdash\Delta}\,(\text{S}_1)^{\ddagger} \qquad \dfrac{@_i\langle\mathsf{a}\rangle k, @_j k, @_i\langle\mathsf{a}\rangle j,\Gamma\vdash\Delta}{@_j k, @_i\langle\mathsf{a}\rangle j,\Gamma\vdash\Delta}\,(\text{S}_2) \qquad \dfrac{\langle j{:} =_{\mathsf{c}} k{:}\rangle, @_i j, \langle i{:} =_{\mathsf{c}} k{:}\rangle,\Gamma\vdash\Delta}{@_i j, \langle i{:} =_{\mathsf{c}} k{:}\rangle,\Gamma\vdash\Delta}\,(\text{S}_3)$$

† j is not in the conclusion ‡ φ is of the form p, $\bot$, or $\langle\mathsf{a}\rangle k$

RULES FOR MODALITIES

$$\dfrac{@_i\varphi,\Gamma\vdash\Delta}{@_j@_i\varphi,\Gamma\vdash\Delta}\,(@\text{L}) \qquad \dfrac{\Gamma\vdash\Delta,@_i\varphi}{\Gamma\vdash\Delta,@_j@_i\varphi}\,(@\text{R}) \qquad \dfrac{@_i\langle\mathsf{a}\rangle j, @_j\varphi,\Gamma\vdash\Delta}{@_i\langle\mathsf{a}\rangle\varphi,\Gamma\vdash\Delta}\,(\langle\mathsf{a}\rangle\text{L})^{\dagger} \qquad \dfrac{@_i\langle\mathsf{a}\rangle j,\Gamma\vdash\Delta,@_i\langle\mathsf{a}\rangle\varphi,@_j\varphi}{@_i\langle\mathsf{a}\rangle j,\Gamma\vdash\Delta,@_i\langle\mathsf{a}\rangle\varphi}\,(\langle\mathsf{a}\rangle\text{R})$$

$$\dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j{:} \blacktriangle k{:}\rangle,\Gamma\vdash\Delta}{@_i\langle\alpha\blacktriangle\beta\rangle,\Gamma\vdash\Delta}\,(\langle\blacktriangle\rangle\text{L})^{\ddagger} \qquad \dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma\vdash\Delta, @_i\langle\alpha\blacktriangle\beta\rangle, \langle j{:} \blacktriangle k{:}\rangle}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma\vdash\Delta, @_i\langle\alpha\blacktriangle\beta\rangle}\,(\langle\blacktriangle\rangle\text{R})$$

† j is not in the conclusion ‡ j and k are different and not in the conclusion

RULES FOR DATA COMPARISON

$$\dfrac{\langle i{:} =_{\mathsf{c}} i{:}\rangle,\Gamma\vdash\Delta}{\Gamma\vdash\Delta}\,(\text{EqT}) \qquad \dfrac{\langle j{:} =_{\mathsf{c}} k{:}\rangle, \langle i{:} =_{\mathsf{c}} j{:}\rangle, \langle i{:} =_{\mathsf{c}} k{:}\rangle,\Gamma\vdash\Delta}{\langle i{:} =_{\mathsf{c}} j{:}\rangle, \langle i{:} =_{\mathsf{c}} k{:}\rangle,\Gamma\vdash\Delta}\,(\text{Eq5}) \qquad \dfrac{\Gamma\vdash\Delta,\langle i{:} =_{\mathsf{c}} j{:}\rangle}{\langle i{:} \neq_{\mathsf{c}} j{:}\rangle,\Gamma\vdash\Delta}\,(\text{NEqL}) \qquad \dfrac{\langle i{:} =_{\mathsf{c}} j{:}\rangle,\Gamma\vdash\Delta}{\Gamma\vdash\Delta,\langle i{:} \neq_{\mathsf{c}} j{:}\rangle}\,(\text{NEqR})$$

STRUCTURAL RULES

$$\dfrac{\Gamma\vdash\Delta,\varphi \quad \varphi,\Gamma'\vdash\Delta'}{\Gamma,\Gamma'\vdash\Delta,\Delta'}\,(\text{Cut}) \qquad \dfrac{\Gamma\vdash\Delta}{\varphi,\Gamma\vdash\Delta}\,(\text{WL}) \qquad \dfrac{\Gamma\vdash\Delta}{\Gamma\vdash\Delta,\varphi}\,(\text{WR})$$

Figure 2: Sequent Calculus **G** for HXPath$_{\text{D}}$.

The inference rules of **G** are organized into five groups, each corresponding to a different fragment of the language of HXPath$_{\text{D}}$. We briefly explain the rationale behind the rules in each group, highlighting the logical principles and the design choices they reflect.

PROPOSITIONAL RULES. The propositional rules govern implication and falsity, and follow familiar patterns from classical sequent calculi. Note that the node expression φ in the axiom rule (Ax) is restricted in form. This restriction suffices for completeness. The generalized version of the rule where φ is of the form $@_i\psi$ for ψ an arbitrary node expression can be obtained as a derived rule.

Rules for Nominals. The rules for nominals handle node expressions involving named points in the model, ensuring they behave as references to specific nodes. (@T) and (@5) characterize reflexivity and Euclideanness. The (Nom) rule allows a named node to be given a fresh alias. This plays a technical role in the completeness proof. Finally, the substitution rules (S_m) reflect that when $@_i j$ holds, then i can be substituted by j in different contexts. These rules are inspired by the treatment of equality in the labeled sequent calculus with equality from [35]. Again, we note that all (S_i) are presented in a somewhat restricted form. Specifically, a more general version of (S_1) can be formulated without the restriction on φ. In turn, (S_2) can be generalized to handle arbitrary paths α, not just atomic modalities a. Finally, (S_3) can be generalized to handle arbitrary comparisons $\blacktriangle$, not just data equality. These generalized rules are derivable within the system. Presenting them in their restricted form helps to simplify the system by limiting the scope of interactions between rules, which is particularly beneficial for proving cut elimination.

Rules for Modalities. The modal rules capture the navigational fragment of the language. (@L) and (@R) reflect the global nature of the satisfiability operator: when the operator is nested, only the inner occurrence is relevant. The diamond rules ($\langle \mathsf{a} \rangle$L) and ($\langle \mathsf{a} \rangle$R) follow standard patterns from modal sequent calculi. In particular, ($\langle \mathsf{a} \rangle$L) introduces fresh nominals as witnesses for the $\langle \mathsf{a} \rangle$ modality. Once again, one can consider generalized versions of the diamond rules in which the atomic modality a is replaced by an arbitrary path expression α. These generalized forms are available as derived rules within the system. Finally, the rules ($\langle \blacktriangle \rangle$L) and ($\langle \blacktriangle \rangle$R) decompose a node expression $@_i\langle \alpha \blacktriangle \beta \rangle$ into its essential components: node expressions $@_i\langle \alpha \rangle j$ and $@_i\langle \beta \rangle k$, capturing the navigation paths, and the atomic comparison $\langle j{:} \blacktriangle k{:} \rangle$ at the endpoints of the paths. Notice there is a direct analogy between the behavior of these last two rules and the diamond rules.

Rules for Data Comparison. Data comparison rules handle equality and inequality between data values at the endpoints of paths. (EqT) and (Eq5) express that data equality is an equivalence relation, while (NeqL) and (NeqR) allow inequality to be reasoned about in terms of equality.

Structural Rules. Finally, the structural rules govern manipulation of the sequent structure itself. These rules play a central role in the meta-theoretical analysis of the system. We will show in Sec. 4, however, that these rules can be eliminated.

3.1 Soundness

We now turn to the proof of soundness for **G**. We define a formal notion of sequent validity in the context of hybrid data models and then verify, via a case-by-case analysis, that each inference rule of **G** preserves this notion of validity. This strategy ensures that all provable sequents are valid. We begin by formally defining the semantics of sequents in terms of the satisfaction relation introduced earlier.

Definition 7. *A sequent $\Gamma \vdash \Delta$ is* valid *iff for all $\mathfrak{M}$, it follows that $\mathfrak{M} \Vdash \Gamma$ implies $\mathfrak{M} \Vdash \psi$ for some $\psi \in \Delta$. A rule preserves validity iff the validity of the premisses of the rule implies the validity of the conclusion of the rule.*

Lemma 8 (Soundness)**.** *Every rule in* **G** *preserves validity.*

Proof. We present a selection of representative cases below. The remaining cases use a similar argument and can be verified by routine inspection. In all cases below we reason by contradiction.

(Nom) Let $\mathfrak{A}$ be a model s.t. $\mathfrak{A} \Vdash \Gamma$ and $\mathfrak{A} \nVdash \psi$ for all $\psi \in \Delta$. Then, introduce a new nominal j that is not in Γ, Δ and build a model $\mathfrak{B}$ that is just like $\mathfrak{A}$ with the exception that $\mathfrak{B} \Vdash @_i j$. We have $\mathfrak{B} \Vdash @_i j, @_i\varphi, \Gamma$ and $\mathfrak{B} \nVdash \psi$ for all $\psi \in \Delta$. This contradicts the validity of the premiss of the rule.

($\langle\blacktriangle\rangle$L) We have: (1) $\blacktriangle$ is $=_{\mathsf{c}}$, or (2) $\blacktriangle$ is $\neq_{\mathsf{c}}$. For (1), take any model $\mathfrak{A}$ s.t.: $\mathfrak{A}\Vdash @_i\langle\alpha =_{\mathsf{c}} \beta\rangle,\Gamma$, and $\mathfrak{A}\nVdash \psi$ for all $\psi\in\Delta$. The semantics of $@_i\langle\alpha =_{\mathsf{c}} \beta\rangle$ tells us there are n and n' in $\mathfrak{A}$ s.t.: $\mathfrak{A},g(i),n\Vdash\alpha$, $\mathfrak{A},g(i),n'\Vdash\beta$, and $(n,n')\in\approx_{\mathsf{c}}$. Then, choose nominals j and k that do not appear in $\Gamma,\Delta,\alpha,\beta$ and build a model $\mathfrak{B}$ that is identical to $\mathfrak{A}$ with the exception that $\mathfrak{B},n\Vdash j$ and $\mathfrak{B},n'\Vdash k$. It is clear that $\mathfrak{B}\Vdash @_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j\!: =_{\mathsf{c}} k\!:\rangle,\Gamma$ and $\mathfrak{B}\nVdash\psi$ for all $\psi\in\Delta$. This contradicts the validity of the premiss of the rule. The case for (2) is similar.

($\langle\blacktriangle\rangle$R) We have: (1) $\blacktriangle$ is $=_{\mathsf{c}}$, or (2) $\blacktriangle$ is $\neq_{\mathsf{c}}$. For (1), take any model $\mathfrak{A}$ s.t.: $\mathfrak{A}\Vdash @_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma$ and $\mathfrak{B}\nVdash\psi$ for all $\psi\in\Delta, @_i\langle\alpha =_{\blacktriangle}\beta\rangle$. In particular, $\mathfrak{A}\nVdash @_i\langle\alpha =_{\mathsf{c}}\beta\rangle$. This means that for all n and n' in $\mathfrak{A}$, it follows that $\mathfrak{A},g(i),n\Vdash\alpha$, $\mathfrak{A},g(i),n'\Vdash\beta$, and $(n,n')\notin\approx_{\mathsf{c}}$. This contradicts the validity of premiss of the rule. The case for (2) is similar. □

Soundness of **G** follows from Lemma 8 by induction on the structure of a derivation of a sequent.

Theorem 9 (Soundness)**.** *Every provable sequent in* **G** *is valid, i.e.,* $\Gamma\vdash_{\mathbf{G}}\Delta$ *implies* $\Gamma\vDash\Delta$.

3.2 Invertibility of Rules

Having established the soundness of the calculus **G**, we prove one of its key properties: all inference rules are invertible. Invertibility plays a central role in the proof-theoretic analysis of a sequent calculus. It allows backward reasoning—from a conclusion to its premisses—without loss of validity. Invertible rules are also particularly well-suited for proof search procedures (see, e.g., [35]). In our case, invertibility also plays a role in our proof of completeness of **G**, where we rely on the ability to apply certain rules in reverse to construct derivations..

Definition 10. *A rule* (P) *is* invertible *iff there is a derivation of each premiss of the rule whose leaves are either axioms or the conclusion of the rule. Any such derivation is called an* inverse *of the rule and is denoted by* (P^{-1}).

Theorem 11. *Every rule in* **G** *is invertible.*

Proof. Invertibility of propositional rules is standard. For (@T), (@5), (Nom), (S$_1$), (S$_2$), (S$_3$), ($\langle\mathsf{a}\rangle$R), ($\langle\blacktriangle\rangle$R), (EqT), and (Eq5) invertibility follows by weakening. We show (@L^{-1}), ($\langle\mathsf{a}\rangle$L^{-1}), and ($\langle\blacktriangle\rangle$L^{-1}).[3]

Case (@L^{-1}): Given a derivation of $@_i\varphi,\Gamma\vdash\Delta$, we build a derivation of $@_i\varphi$ as:

$$\dfrac{\dfrac{\dfrac{}{@_i\varphi,\Gamma\vdash\Delta,@_i\varphi}\,(\mathrm{Ax})}{@_i\varphi,\Gamma\vdash\Delta,@_j@_i\varphi}\,(@\mathrm{R}) \qquad @_j@_i\varphi,\Gamma\vdash\Delta}{@_i\varphi,\Gamma\vdash\Delta}\,(\mathrm{Cut})$$

Case ($\langle\mathsf{a}\rangle$L^{-1}): Given a derivation of $@_i\langle\mathsf{a}\rangle\varphi,\Gamma\vdash\Delta$, we build a derivation of $@_i\langle\mathsf{a}\rangle j, @_j\varphi,\Gamma\vdash\Delta$ as:

$$\dfrac{\dfrac{\dfrac{}{@_i\langle\mathsf{a}\rangle j, @_j\varphi,\Gamma\vdash\Delta, @_i\langle\mathsf{a}\rangle\varphi, @_j\varphi}\,(\mathrm{Ax})}{@_i\langle\mathsf{a}\rangle j, @_j\varphi,\Gamma\vdash\Delta, @_i\langle\mathsf{a}\rangle\varphi}\,(\langle\mathsf{a}\rangle\mathrm{R}) \qquad @_i\langle\mathsf{a}\rangle\varphi,\Gamma\vdash\Delta}{@_i\langle\mathsf{a}\rangle j, @_j\varphi,\Gamma\vdash\Delta}\,(\mathrm{Cut})$$

Case ($\langle\blacktriangle\rangle$L): Given a derivation of $@_i\langle\alpha\blacktriangle\beta\rangle,\Gamma\vdash\Delta$, we derive $@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j\!:\blacktriangle k\!:\rangle,\Gamma\vdash\Delta$ as:

$$\dfrac{\dfrac{\dfrac{}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j\!:\blacktriangle k\!:\rangle,\Gamma\vdash\Delta, @_i\langle\alpha\blacktriangle\beta\rangle, \langle j\!:\blacktriangle k\!:\rangle}\,(\mathrm{Ax})}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j\!:\blacktriangle k\!:\rangle,\Gamma\vdash\Delta, @_i\langle\alpha\blacktriangle\beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{R}) \qquad @_i\langle\alpha\blacktriangle\beta\rangle,\Gamma\vdash\Delta}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j\!:\blacktriangle k\!:\rangle,\Gamma\vdash\Delta}\,(\mathrm{Cut})$$

□

[3]We use colors to guide derivations: cyan for cut expressions, pink for principal, and magenta for other relevant elements.

3.3 Completeness

In this section, we establish the completeness of **G**. Rather than constructing canonical models, we prove completeness by way of a Hilbert-style axiomatization for HXPath$_{\mathrm{D}}$, which we refer to as **H**. The axiomatization, introduced in [10], has been shown to be both sound and complete. Concretely, we demonstrate that every theorem of **H** corresponds to a provable sequent in **G**. This ensures that all semantically valid sequents are derivable in the sequent calculus. Our strategy underscores the expressiveness of **G**, and leverages existing results. As a further benefit, we reveal a close correspondence between axiomatic and sequent-based reasoning in HXPath$_{\mathrm{D}}$.

Lemma 12. *Let $\vdash^n_{\mathbf{H}} \varphi$ indicate that φ is a theorem in* **H***, with a derivation of length n. In addition, let i be a nominal not in φ. It follows that, for any n, $\vdash^n_{\mathbf{H}} \varphi$ implies $\vdash_{\mathbf{G}} @_i\varphi$ (i.e, $\vdash @_i\varphi$ is provable in* **G***).*

Proof. The proof proceeds by (strong) induction on the length of a derivation in **H**. The base case requires that each axiom φ in **H** has a corresponding provable sequent $\vdash_{\mathbf{G}} @_i\varphi$. To simplify these proofs we will make use of the following derived rules in **G**.

Derived Rules. We introduce derived rules for treating other propositional connectives as if they were primitive. Notably,

$$\frac{@_i\top,\Gamma\vdash\Delta}{\Gamma\vdash\Delta}(\top\mathrm{L}) \quad \frac{@_i\varphi, @_i\psi,\Gamma\vdash\Delta}{@_i(\varphi\wedge\psi),\Gamma\vdash\Delta}(\wedge\mathrm{L}) \quad \frac{\Gamma\vdash\Delta, @_i\psi \quad \Gamma\vdash\Delta, @_i\varphi}{\Gamma\vdash\Delta, @_i(\varphi\wedge\psi)}(\wedge\mathrm{R}) \quad \frac{@_i\varphi,\Gamma\vdash\Delta, @_i\psi \quad @_i\psi,\Gamma\vdash\Delta, @_i\varphi}{\Gamma\vdash\Delta, @_i(\varphi\leftrightarrow\psi)}(\leftrightarrow\mathrm{R})$$

As mentioned earlier, we introduce a generalized form of (Ax) as a derived rule.

$$\frac{}{@_i\varphi,\Gamma\vdash\Delta, @_i\varphi}(\mathrm{AxG})$$

In (AxG) φ is an arbitrary node expression. When no confusion arises, we will also refer to this rule simply as (Ax). Finally, we introduce derived rules characterizing the symmetry of $\blacktriangle$ explicitly. Precisely,

$$\frac{\langle j{:}\blacktriangle i{:}\rangle,\Gamma\vdash\Delta}{\langle i{:}\blacktriangle j{:}\rangle,\Gamma\vdash\Delta}(\langle\blacktriangle\rangle\mathrm{B}).$$

Base Case. We are now ready to present derivations for selected axioms of **H**. Beyond their technical role, these derivations also serve an explanatory purpose: they illustrate how the rules of **G** reveal the semantic behavior of logical connectives, offering insight into their proof-theoretic interpretation and clarifying the structure of the corresponding axioms. We focus on the axioms (equal), ($\blacktriangle$-comm), and (ε-trans) expressing the equivalence properties of data equality: reflexivity, symmetry, and transitivity. These axioms are listed below for reference.

$$\begin{aligned} \text{(equal)}\ & \langle\varepsilon =_{\mathsf{c}} \varepsilon\rangle \\ (\blacktriangle\text{-comm})\ & \langle\alpha\blacktriangle\beta\rangle \leftrightarrow \langle\beta\blacktriangle\alpha\rangle \\ (\varepsilon\text{-trans})\ & \langle\alpha =_{\mathsf{c}} \varepsilon\rangle \wedge \langle\varepsilon =_{\mathsf{c}} \beta\rangle \to \langle\alpha =_{\mathsf{c}} \beta\rangle. \end{aligned}$$

For (equal), we must show $\vdash_{\mathbf{G}} @_i\langle\varepsilon =_{\mathsf{c}} \varepsilon\rangle$. For ($\blacktriangle$-comm), we must show $\vdash_{\mathbf{G}} @_i(\langle\alpha\blacktriangle\beta\rangle \leftrightarrow \langle\beta\blacktriangle\alpha\rangle)$. Finally, for ($\varepsilon$-trans), we must show $\vdash_{\mathbf{G}} @_i(\langle\alpha =_{\mathsf{c}} \varepsilon\rangle \wedge \langle\varepsilon =_{\mathsf{c}} \beta\rangle \to \langle\alpha =_{\mathsf{c}} \beta\rangle)$. We deal with each of these cases individually.

Reflexivity. The derivation below establishes $\vdash_{\mathbf{G}} @_i\langle\varepsilon =_{\mathsf{c}} \varepsilon\rangle$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\langle i{:} =_{\mathrm{c}} i{:}\rangle, @_i\langle\varepsilon\rangle i \vdash @_i\langle\varepsilon =_{\mathrm{c}} \varepsilon\rangle, \langle i{:} =_{\mathrm{c}} i{:}\rangle}\,(\mathrm{Ax})}{@_i\langle\varepsilon\rangle i \vdash @_i\langle\varepsilon =_{\mathrm{c}}\varepsilon\rangle, \langle i{:} =_{\mathrm{c}} i{:}\rangle}\,(\mathrm{EqT})}{@_i\langle\varepsilon\rangle i \vdash @_i\langle\varepsilon =_{\mathrm{c}}\varepsilon\rangle}\,(\langle\blacktriangle\rangle\mathrm{R})}{@_i\top, @_i i \vdash @_i\langle\varepsilon =_{\mathrm{c}}\varepsilon\rangle}\,(\wedge\mathrm{L}^{-1})}{@_i i \vdash @_i\langle\varepsilon =_{\mathrm{c}}\varepsilon\rangle}\,(\top\mathrm{L})}{\vdash @_i\langle\varepsilon =_{\mathrm{c}}\varepsilon\rangle}\,(@\mathrm{T})$$

When reading the derivation bottom-up, most of the effort goes into constructing an empty path on the antecedent side of the sequent. Intuitively, the empty path encodes the idea that we remain at the current node. We then use ($\langle\blacktriangle\rangle$R) to compare the data at the node and itself. The derivation concludes with the rule (EqT) just before (Ax), which captures the reflexivity of data equality in **G**.

Symmetry. The next derivation establishes $\vdash_{\mathbf{G}} @_i(\langle\alpha \blacktriangle \beta\rangle \leftrightarrow \langle\beta \blacktriangle \alpha\rangle)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle k{:} \blacktriangle j{:}\rangle \vdash @_i\langle\beta \blacktriangle \alpha\rangle, \langle k{:} \blacktriangle j{:}\rangle}\,(\mathrm{Ax})}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j{:} \blacktriangle k{:}\rangle \vdash @_i\langle\beta \blacktriangle \alpha\rangle, \langle k{:} \blacktriangle j{:}\rangle}\,(\langle\blacktriangle\rangle\mathrm{B})}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j{:} \blacktriangle k{:}\rangle \vdash @_i\langle\beta \blacktriangle \alpha\rangle}\,(\langle\blacktriangle\rangle\mathrm{R})}{@_i\langle\alpha \blacktriangle \beta\rangle \vdash @_i\langle\beta \blacktriangle \alpha\rangle}\,(\langle\blacktriangle\rangle\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{@_i\langle\beta\rangle k, @_i\langle\alpha\rangle j, \langle j{:} \blacktriangle k{:}\rangle \vdash @_i\langle\alpha \blacktriangle \beta\rangle, \langle j{:} \blacktriangle k{:}\rangle}\,(\mathrm{Ax})}{@_i\langle\beta\rangle k, @_i\langle\alpha\rangle j, \langle k{:} \blacktriangle j{:}\rangle \vdash @_i\langle\alpha \blacktriangle \beta\rangle, \langle j{:} \blacktriangle k{:}\rangle}\,(\langle\blacktriangle\rangle\mathrm{B})}{@_i\langle\beta\rangle k, @_i\langle\alpha\rangle j, \langle k{:} \blacktriangle j{:}\rangle \vdash @_i\langle\alpha \blacktriangle \beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{R})}{@_i\langle\beta \blacktriangle \alpha\rangle \vdash @_i\langle\alpha \blacktriangle \beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{L})}{\vdash @_i(\langle\alpha \blacktriangle \beta\rangle \leftrightarrow \langle\beta \blacktriangle \alpha\rangle)}\,(\leftrightarrow\mathrm{R})$$

As before, reading the derivation bottom-up, we observe that we first construct the appropriate paths in the antecedent. These paths serve to navigate to the nodes whose data values we aim to compare. As mentioned, in **G** the derived rule ($\langle\blacktriangle\rangle$B) captures symmetry. The derivation concludes with an application of this rule before (Ax).

Transitivity. The final derivation establishes $\vdash_{\mathbf{G}} @_i(\langle\alpha =_{\mathrm{c}} \varepsilon\rangle \wedge \langle\varepsilon =_{\mathrm{c}} \beta\rangle \to \langle\alpha =_{\mathrm{c}} \beta\rangle)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\langle a{:} =_{\mathrm{c}} b{:}\rangle, \langle c{:} =_{\mathrm{c}} a{:}\rangle, \langle c{:} =_{\mathrm{c}} b{:}\rangle \vdash \langle a{:} =_{\mathrm{c}} b{:}\rangle}\,(\mathrm{Ax})}{\langle c{:} =_{\mathrm{c}} a{:}\rangle, \langle c{:} =_{\mathrm{c}} b{:}\rangle \vdash \langle a{:} =_{\mathrm{c}} b{:}\rangle}\,(\mathrm{Eq5})}{\langle a{:} =_{\mathrm{c}} c{:}\rangle, \langle c{:} =_{\mathrm{c}} b{:}\rangle \vdash \langle a{:} =_{\mathrm{c}} b{:}\rangle}\,(\langle\blacktriangle\rangle\mathrm{B})}{@_d c, \langle a{:} =_{\mathrm{c}} c{:}\rangle, \langle d{:} =_{\mathrm{c}} b{:}\rangle \vdash \langle a{:} =_{\mathrm{c}} b{:}\rangle}\,(\mathrm{S}_3,\mathrm{WL})}{@_i c, @_i d, @_i\top, \langle a{:} =_{\mathrm{c}} c{:}\rangle, @_i\langle\alpha\rangle a, @_i\langle\beta\rangle b, \langle d{:} =_{\mathrm{c}} b{:}\rangle \vdash @_i\langle\alpha =_{\mathrm{c}} \beta\rangle, \langle a{:} =_{\mathrm{c}} b{:}\rangle}\,(@5,\mathrm{WL})}{@_i\langle\alpha\rangle a, @_i\langle\varepsilon\rangle c, \langle a{:} =_{\mathrm{c}} c{:}\rangle, @_i\langle\varepsilon\rangle d, @_i\langle\beta\rangle b, \langle d{:} =_{\mathrm{c}} b{:}\rangle \vdash @_i\langle\alpha =_{\mathrm{c}} \beta\rangle}\,(\wedge\mathrm{L},\langle\blacktriangle\rangle\mathrm{R})}{@_i\langle\alpha =_{\mathrm{c}} \varepsilon\rangle, @_i\langle\varepsilon =_{\mathrm{c}} \beta\rangle \vdash @_i\langle\alpha =_{\mathrm{c}} \beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{L})}{\vdash @_i(\langle\alpha =_{\mathrm{c}} \varepsilon\rangle \wedge \langle\varepsilon =_{\mathrm{c}} \beta\rangle \to \langle\alpha =_{\mathrm{c}} \beta\rangle)}\,(\to\mathrm{R},\wedge\mathrm{L})$$

The key idea is to chain two data equalities together. The derivation (going again bottom-up) begins by using empty paths ε to refer to the intermediate node that links α and β. Because of the side conditions on the rules being used, this intermediate node ends up having two names c and d. The rule (@5) indicates c and d are alias. Given this information, the rule (S_3) replaces d by c in $\langle d{:} =_{\mathrm{c}} b{:}\rangle$, yielding $\langle c{:} =_{\mathrm{c}} b{:}\rangle$. The rule ($\langle\blacktriangle\rangle$B) prepares the sequent for the application of (Eq5), which captures transitivity in **G**, before ending with (Ax).

To sum up, (equal), ($\blacktriangle$-comm), and (ε-trans) are introduced in [10] to express the equivalence properties of data equality. Arguably, rules (EqT) and (Eq5) make these properties more transparent. The derivation in **G** of all axioms in **H** can be found in [11].

Inductive Case. We must show that $\vdash^m_{\mathbf{H}} \psi$ implies $\vdash_{\mathbf{G}} @_i\psi$. The inductive hypothesis (IH) is: for all $1 \le n < m$, if $\vdash^n_{\mathbf{H}} \varphi$, then $\vdash_{\mathbf{G}} @_i\varphi$, for some nominal i not occurring in φ. We proceed by cases, depending on whether $\vdash^m_{\mathbf{H}} \psi$ is obtained using one of the four inference rules in **H**: (MP), (Nec), (Name), or (Paste). The first three cases follow directly; details can be found in [11]. Let us discuss the (Paste) rule:

$$\dfrac{\vdash @_j\langle \mathsf{a}\rangle k \wedge \langle k{:}\alpha \blacktriangle \beta\rangle \to \chi}{\vdash \langle j{:}\mathsf{a}\alpha \blacktriangle \beta\rangle \to \chi}\ \text{where } j \text{ and } k \text{ are different and not in } \chi,\ \alpha,\ \beta.$$

An application of this rule assumes that for $n < m \vdash^n_{\mathbf{H}} (@_j\langle \mathsf{a}\rangle k \wedge \langle k{:}\alpha \blacktriangle \beta\rangle) \to \chi$. From the IH, $\vdash_{\mathbf{G}} @_i((@_j\langle \mathsf{a}\rangle k \wedge \langle k{:}\alpha \blacktriangle \beta\rangle) \to \chi)$. The next derivation establishes $\vdash_{\mathbf{G}} @_i(\langle j{:}\mathsf{a}\alpha \blacktriangle \beta\rangle \to \chi)$, as required.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{@_j\langle \mathsf{a}\rangle k \vdash @_j\langle \mathsf{a}\rangle k}\,(\text{Ax})}{@_j\langle \mathsf{a}\rangle k \vdash @_i @_j\langle \mathsf{a}\rangle k}\,(@\text{R}) \quad \dfrac{\dfrac{\dfrac{}{\langle a{:}\blacktriangle b{:}\rangle \vdash \langle a{:}\blacktriangle b{:}\rangle}\,(\text{Ax})}{@_i\langle k{:}\alpha\rangle a, @_i\langle\beta\rangle b, \langle a{:}\blacktriangle b{:}\rangle \vdash @_i\langle k{:}\alpha \blacktriangle \beta\rangle}\,(\langle\blacktriangle\rangle\text{R,W}*)}{@_k\langle\alpha\rangle a, @_i\langle\beta\rangle b, \langle a{:}\blacktriangle b{:}\rangle \vdash @_i\langle k{:}\alpha \blacktriangle \beta\rangle}\,(@\text{L}^{-1})}{@_j\langle \mathsf{a}\rangle k, @_k\langle\alpha\rangle a, @_i\langle\beta\rangle b, \langle a{:}\blacktriangle b{:}\rangle \vdash @_i(@_j\langle \mathsf{a}\rangle k \wedge \langle k{:}\alpha \blacktriangle \beta\rangle)}\,(\wedge\text{R,W}*) \quad \dfrac{\vdash_{\mathbf{G}} @_i(@_j\langle \mathsf{a}\rangle k \wedge \langle k{:}\alpha \blacktriangle \beta\rangle \to \chi)}{@_i(@_j\langle \mathsf{a}\rangle k \wedge \langle k{:}\alpha \blacktriangle \beta\rangle) \vdash @_i\chi}\,(\to\text{R}^{-1})}{@_j\langle \mathsf{a}\rangle k, @_k\langle\alpha\rangle a, @_i\langle\beta\rangle b, \langle a{:}\blacktriangle b{:}\rangle \vdash @_i\chi}\,(\text{Cut})}{@_j\langle \mathsf{a}\alpha\rangle a, @_i\langle\beta\rangle b, \langle a{:}\blacktriangle b{:}\rangle \vdash @_i\chi}\,(\langle\mathsf{a}\rangle\text{L})}{@_i\langle j{:}\mathsf{a}\alpha\rangle a, @_i\langle\beta\rangle b, \langle a{:}\blacktriangle b{:}\rangle \vdash @_i\chi}\,(@\text{L})}{@_i\langle j{:}\mathsf{a}\alpha \blacktriangle \beta\rangle \vdash @_i\chi}\,(\langle\blacktriangle\rangle\text{L})}{\vdash_{\mathbf{G}} @_i(\langle j{:}\mathsf{a}\alpha \blacktriangle \beta\rangle \to \chi)}\,(\to\text{R})$$

In the derivation above, (W$*$) indicates the simultaneous application of (WL) and (WR). $\square$

Theorem 13 (Completeness)**.** *Every valid sequent is provable.*

Proof. Suppose $\gamma_1, \ldots, \gamma_n \vdash \delta_1, \ldots, \delta_m$ is valid. From the completeness result in [10], we know there is k such that $\vdash^k_{\mathbf{H}} \bigwedge_{1 \le i \le n} \gamma_i \to \bigvee_{1 \le j \le m} \delta_j$. From Lemma 12, we get $\vdash_{\mathbf{G}} @_i(\bigwedge_{1 \le i \le n} \gamma_i \to \bigvee_{1 \le j \le m} \delta_j)$. This implies $@_i\gamma_1, \ldots, @_i\gamma_n \vdash_{\mathbf{G}} @_i\delta_1, \ldots, @_i\delta_m$, and so $\gamma_1, \ldots, \gamma_n \vdash_{\mathbf{G}} \delta_1, \ldots, \delta_m$ as required. $\square$

4 Cut Elimination

The rule (Cut) plays a crucial role in our proof of the completeness of **G**. Specifically, it allows us to compose translations of Hilbert-style derivations within **G**. Despite its utility, however, (Cut) can be eliminated. The general strategy, following [35], is to push applications of (Cut) "upwards" in the derivation tree until they disappear at the level of leaves. To this end, we perform an induction on the *size* of the *active cut expression*, with a sub-induction on the *(Cut) height*. We introduce these notions and related results below.

Definition 14. *The* size *of a node expression is defined by mutual recursion as:*

$$\begin{array}{rrr}
\text{size}(p) = 1 & \text{size}(\varphi \to \psi) = 1 + \text{size}(\varphi) + \text{size}(\psi) & \text{size}(\mathsf{a}) = 1 \\
\text{size}(i) = 1 & \text{size}(@_i\varphi) = 1 + \text{size}(\varphi) & \text{size}(i{:}) = 1 \\
\text{size}(\bot) = 1 & \text{size}(\langle \mathsf{a}\rangle\varphi) = 1 + \text{size}(\varphi) & \text{size}(\varphi?) = 1 + \text{size}(\varphi) \\
 & \text{size}(\langle \alpha \blacktriangle \beta\rangle) = 1 + \text{size}(\alpha) + \text{size}(\beta) & \text{size}(\alpha\beta) = \text{size}(\alpha) + \text{size}(\beta)
\end{array}$$

The function size *induces a well-founded partial order over the set of node expressions.*

Definition 15. *The* height *of a derivation is the length of its longest branch (e.g., a derivation consisting of only an application of (Ax) has height 1). If $\Gamma \vdash \Delta$ is the end-sequent in a derivation, we will use $\Gamma \vdash^n \Delta$ to indicate that the derivation has height n. The* cut height *of an application of (Cut) in a given derivation is the sum of the heights of the derivations of the premisses of the rule; i.e., if we have derivations $\Gamma \vdash^n \Delta, \varphi$, and $\varphi, \Gamma' \vdash^m \Delta'$, using (Cut), we obtain a derivation $\Gamma, \Gamma' \vdash^{(\max(n,m)+1)} \Delta, \Delta'$. In this case, the cut height is $n + m$. In such an application of (Cut), we call φ the* active cut *expression.*

Intuitively, the *cut height* measures how close to the leaves of a derivation a particular application of (Cut) occurs, taking into consideration the derivations of *both* premisses of the rule. This will be important in our proof of cut elimination. We can now state and prove the main result of this section.

Theorem 16. *Every use of (Cut) in the derivation of a provable sequent can be eliminated.*

Proof. The proof is by induction on two measures: the size of the active cut expression, and the cut height. More precisely, in the proof we associate with each application of (Cut) in a derivation a pair (k,h), called *cut complexity*, where: k corresponds to the size of the active cut expression, and h corresponds to the cut height. The induction is on the lexicographic order of the pairs (k,h).

Base Case. The base cases of the induction are relatively direct. They involve derivations in which (Cut) is applied only once, with axiom rules applied to its premisses, i.e., the premisses of (Cut) must be instances of (Ax) or ($\bot$). Eliminating (Cut) in such configurations is unproblematic. We illustrate one representative case below. Suppose $\mathscr{C}$ and $\mathscr{A}$ are derivations with the following structure:

$$\mathscr{C} = \dfrac{\dfrac{}{\Gamma \vdash \Delta, \varphi}\,(\mathrm{Ax}) \qquad \dfrac{}{\varphi, \Gamma' \vdash \Delta'}\,(\bot)}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\mathrm{Cut}) \qquad\qquad \mathscr{A} = \dfrac{}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\mathrm{P})$$

The derivation $\mathscr{C}$ represents a case in which one of the premisses of (Cut) is (Ax) and the other is ($\bot$). This derivation can be transformed into the cut-free derivation $\mathscr{A}$ in which: (P) is (Ax) if $\varphi \notin \Gamma$; and it is ($\bot$) otherwise. The remaining cases use a similar argument.

Inductive Step. The key idea is to identify an application of (Cut) that is minimal according to the cut height, and eliminate it. More precisely, we proceed by considering in a derivation a sub-derivation ending in an application of (Cut) with complexity (k,h) where the value for h is minimal. This guarantees that no other instance of (Cut) appears above it in the sub-derivation. We show that this sub-derivation can be replaced by an alternative one which uses only (Cut) instances whose associated pairs (k',h') are strictly smaller than (k,h). We can then invoke a (strong) inductive hypothesis to claim that these (Cut) instances can also be eliminated.

We proceed by a case analysis based on the syntactic form of the active cut, and on whether the active cut is principal in either premiss of the application of (Cut). This analysis guides a systematic transformation of the derivation where: we push the (Cut) upwards, or replace it with applications of (Cut) involving smaller active cut expressions. In either case, the process eventually leads to a cut-free derivation. We illustrate how this transformation unfolds in some representative scenarios.

Non-principal Cases. First, let us cover the case where the active cut is not principal in the right premiss of (Cut). In this case, the application of (Cut) is permuted up. To illustrate this process, consider, e.g., the derivation:

$$\dfrac{\Gamma \vdash^{n} \Delta, \varphi \qquad \dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j{:} \blacktriangle k{:}\rangle, \varphi, \Gamma' \vdash^{m} \Delta}{\varphi, @_i\langle\alpha \blacktriangle \beta\rangle, \Gamma' \vdash \Delta}\,(\langle\blacktriangle\rangle\mathrm{L})}{@_i\langle\alpha \blacktriangle \beta\rangle, \Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\mathrm{Cut})$$

Suppose that in this derivation, the use of (Cut) has an associated complexity $(\mathrm{size}(\varphi), n + (m+1))$, where $n + (m+1)$ is minimal. W.l.o.g., assume that j and k do not appear in $\Gamma \vdash \Delta$.[4] We transform this

[4] If j or k do appear in $\Gamma \vdash \Delta$, we can simply choose different nominals, and rewrite the derivation of $\varphi, @_i\langle\alpha \blacktriangle \beta\rangle, \Gamma' \vdash \Delta$ using the new selection of nominals.

derivation into:

$$\dfrac{\dfrac{\Gamma \vdash^n \Delta, \varphi \qquad \varphi, @_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j{:} \blacktriangle k{:}\rangle, \Gamma' \vdash^m \Delta}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k, \langle j{:} \blacktriangle k{:}\rangle, \Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\text{Cut})}{@_i\langle\alpha \blacktriangle \beta\rangle, \Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\langle\blacktriangle\rangle\text{L})$$

The application of (Cut) in the transformed derivation has complexity $(\text{size}(\varphi), n+m)$, so it can be eliminated by the inductive hypothesis.

The remaining cases where the active cut is not principal in the right premiss follow the same strategy. To see why, note that all such derivations have the following general structure:

$$\dfrac{\Gamma \vdash^n \Delta, \varphi \qquad \dfrac{\varphi, \Phi', \Gamma' \vdash^m \Delta', \Sigma'}{\varphi, \Phi, \Gamma' \vdash \Delta', \Sigma}\,(\text{P})}{\Phi, \Gamma, \Gamma' \vdash \Delta, \Delta', \Sigma}\,(\text{Cut})$$

In this derivation, we are considering (P) is a single premiss rule of **G**, the set $\varphi, \Gamma', \Delta'$ is the context for the rule, and the set $\Phi', \Phi, \Sigma', \Sigma$ are the node expressions the rule acts upon. Again, we assume that (Cut) has complexity $(\text{size}(\varphi), n+(m+1))$ where $n+(m+1)$ is minimal; i.e., where (Cut) is not used in $\Gamma \vdash^n \Delta, \varphi$, nor in $\varphi, \Phi', \Gamma' \vdash^m \Delta', \Sigma'$. Modulo a possible renaming of nominals, we transform this derivation into:

$$\dfrac{\dfrac{\Gamma \vdash^n \Delta, \varphi \qquad \varphi, \Phi', \Gamma' \vdash^m \Delta', \Sigma'}{\Phi', \Gamma', \Gamma \vdash \Delta, \Delta', \Sigma'}\,(\text{Cut})}{\Phi, \Gamma', \Gamma \vdash \Delta, \Delta', \Sigma}\,(\text{P})$$

The use of (Cut) in the transformed derivation has complexity $(\text{size}(\varphi), n+m)$, using the inductive hypothesis, we obtain a cut-free derivation of $\Phi, \Gamma', \Gamma \vdash \Delta, \Delta', \Sigma$.

The cases where the active cut is not principal in the left premiss is symmetric. The ($\rightarrow$L) case—the only two-premise rule in **G**—is handled similarly, and is well known in the literature.

Principal Cases. Let us now turn our attention to derivations where the active cut is principal in both premisses. Such cases are central to the proof and require careful handling to ensure that the application of (Cut) can still be pushed upwards, and ultimately eliminated. We must examine all combinations of rules with a possibly matching active cut. We illustrate a few representative cases below.

Let us consider first the interaction between ($\langle\mathsf{a}\rangle$R) and ($\langle\mathsf{a}\rangle$L). We adapt the strategy presented in [35]. Suppose that in a derivation we encounter an application of (Cut) of minimal height of the form:

$$\dfrac{\dfrac{@_i\langle\mathsf{a}\rangle j, \Gamma \vdash^n \Delta, @_i\langle\mathsf{a}\rangle\varphi, @_j\varphi}{@_i\langle\mathsf{a}\rangle j, \Gamma \vdash \Delta, @_i\langle\mathsf{a}\rangle\varphi}\,(\langle\mathsf{a}\rangle\text{R}) \qquad \dfrac{@_i\langle\mathsf{a}\rangle j, @_j\varphi, \Gamma' \vdash^m \Delta'}{@_i\langle\mathsf{a}\rangle\varphi, \Gamma' \vdash \Delta'}\,(\langle\mathsf{a}\rangle\text{L})}{@_i\langle\mathsf{a}\rangle j, \Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\text{Cut})$$

The exhibited (Cut) has complexity $(\text{size}(@_i\langle\mathsf{a}\rangle\varphi), (n+1)+(m+1))$. We proceed to transform the derivation into a new one that reduces the complexity and moves us closer to a cut-free derivation.

$$\dfrac{\dfrac{@_i\langle\mathsf{a}\rangle j, \Gamma, \vdash^n \Delta, @_j\varphi, @_i\langle\mathsf{a}\rangle\varphi \qquad \dfrac{@_i\langle\mathsf{a}\rangle j, @_j\varphi, \Gamma' \vdash^m \Delta'}{@_i\langle\mathsf{a}\rangle\varphi, \Gamma' \vdash \Delta'}\,(\langle\mathsf{a}\rangle\text{L})}{@_i\langle\mathsf{a}\rangle j, \Gamma, \Gamma' \vdash \Delta, \Delta', @_j\varphi}\,(\text{Cut}_1) \qquad @_j\varphi, @_i\langle\mathsf{a}\rangle j, \Gamma' \vdash^m \Delta'}{@_i\langle\mathsf{a}\rangle j, \Gamma, \Gamma' \vdash \Delta, \Delta'}\,(\text{Cut}_2)$$

In the transformed derivation, there are two applications of (Cut), labeled (Cut$_1$) and (Cut$_2$). We reason as follows, (Cut$_1$), is the only (Cut) in its subderivation and has complexity $(\text{size}(@_i\langle\mathsf{a}\rangle\varphi), n+(m+1))$. Therefore, by the inductive hypothesis, (Cut$_1$) can be eliminated resulting in a cut-free derivation of $@_i\langle\mathsf{a}\rangle j, \Gamma, \Gamma' \vdash^h \Delta, \Delta', @_j\varphi$, for some unknown h. We use this cut-free derivation as a building block to construct the derivation:

$$\dfrac{@_i\langle \mathsf{a}\rangle j,\Gamma,\Gamma' \vdash^h \Delta,\Delta', @_j\varphi \qquad @_j\varphi, @_i\langle\mathsf{a}\rangle j,\Gamma' \vdash^m \Delta'}{@_i\langle\mathsf{a}\rangle j,\Gamma,\Gamma'\vdash\Delta,\Delta'}\,(\mathrm{Cut}_2)$$

We can now see that (Cut_2) can also be eliminated. It has complexity $(\mathrm{size}(@_j\varphi), h+m)$ and it is the only (Cut) application in the derivation. Since $\mathrm{size}(@_j\varphi) < \mathrm{size}(@_i\langle\mathsf{a}\rangle\varphi)$, we can apply the inductive hypothesis, and obtain a fully cut-free derivation of $@_i\langle\mathsf{a}\rangle j,\Gamma,\Gamma'\vdash\Delta,\Delta'$.

Let us now consider a case involving data comparisons: the interaction between $(\langle\blacktriangle\rangle\mathrm{R})$ and $(\langle\blacktriangle\rangle\mathrm{L})$. Namely, suppose that in a derivation we encounter an application of (Cut) of minimal height of the form:

$$\dfrac{\dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma\vdash^n\Delta, @_i\langle\alpha\blacktriangle\beta\rangle,\langle j\colon\blacktriangle k\colon\rangle}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma\vdash\Delta, @_i\langle\alpha\blacktriangle\beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{R}) \qquad \dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\langle j\colon\blacktriangle k\colon\rangle,\Gamma'\vdash^m\Delta'}{@_i\langle\alpha\blacktriangle\beta\rangle,\Gamma'\vdash\Delta'}\,(\langle\blacktriangle\rangle\mathrm{L})}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma,\Gamma'\vdash\Delta,\Delta'}\,(\mathrm{Cut})$$

We transform this derivation into:

$$\dfrac{\dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma\vdash^n\Delta,\langle j\colon\blacktriangle k\colon\rangle, @_i\langle\alpha\blacktriangle\beta\rangle \qquad \dfrac{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\langle j\colon\blacktriangle k\colon\rangle,\Gamma'\vdash^m\Delta'}{@_i\langle\alpha\blacktriangle\beta\rangle,\Gamma'\vdash\Delta'}\,(\langle\blacktriangle\rangle\mathrm{L})}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma,\Gamma'\vdash\Delta,\Delta',\langle j\colon\blacktriangle k\colon\rangle}\,(\mathrm{Cut}_1) \qquad \langle j\colon\blacktriangle k\colon\rangle, @_i\langle\alpha\rangle j, @_i\langle\beta\rangle j,\Gamma'\vdash^m\Delta'}{@_i\langle\alpha\rangle j, @_i\langle\beta\rangle k,\Gamma,\Gamma'\vdash\Delta,\Delta'}\,(\mathrm{Cut}_2)$$

The argument is similar, the original use of (Cut) has complexity $(\mathrm{size}(@_i\langle\alpha\blacktriangle\beta\rangle),(n+1)+(m+1))$. When we transform the derivation, we introduce two new applications of (Cut), labeled (Cut_1) and (Cut_2). (Cut_1) has complexity $(\mathrm{size}(@_i\langle\alpha\blacktriangle\beta\rangle), n+(m+1))$ and can be eliminated. Then, (Cut_2), can be eliminated from the resulting derivation as it involves a simpler active cut expression.

For the particular to **G** case, let us consider the interaction between the rules $(\langle\mathsf{a}\rangle\mathrm{R})$ and $(\langle\blacktriangle\rangle\mathrm{R})$. Namely, suppose that in a derivation we encounter an application of (Cut) of minimal height of the form:

$$\dfrac{\dfrac{@_i\langle\mathsf{a}\rangle j,\Gamma\vdash^n\Delta, @_i\langle\mathsf{a}\rangle a, @_j a}{@_i\langle\mathsf{a}\rangle j,\Gamma\vdash\Delta, @_i\langle\mathsf{a}\rangle a}\,(\langle\mathsf{a}\rangle\mathrm{R}) \qquad \dfrac{@_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash^m\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle,\langle a\colon\blacktriangle b\colon\rangle}{@_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{R})}{@_i\langle\mathsf{a}\rangle j, @_i\langle\beta\rangle b,\Gamma,\Gamma'\vdash\Delta,\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\mathrm{Cut})$$

Unlike the other cut-elimination cases we have examined, this particular case involves two right rules applied in the premisses of (Cut), rather than a mix of left and right rules. This synchronization of right rules is unusual, but it does not pose a problem. We can still transform the derivation to eliminate (Cut).

$$\mathscr{D} = @_i\langle\mathsf{a}\rangle j,\Gamma\vdash^n\Delta, @_j a, @_i\langle\mathsf{a}\rangle a$$

$$\dfrac{\dfrac{\mathscr{D} \qquad \dfrac{@_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash^m\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle,\langle a\colon\blacktriangle b\colon\rangle}{@_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{R})}{@_i\langle\mathsf{a}\rangle j, @_i\langle\beta\rangle b,\Gamma,\Gamma'\vdash\Delta,\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle, @_j a}\,(\mathrm{Cut}_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{@_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash^m\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle,\langle a\colon\blacktriangle b\colon\rangle}{@_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\langle\blacktriangle\rangle\mathrm{R})}{@_i\langle\mathsf{a}\rangle j, @_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\mathrm{WL})}{@_j a, @_i\langle\mathsf{a}\rangle j, @_i\langle\mathsf{a}\rangle a, @_i\langle\beta\rangle b,\Gamma'\vdash\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\mathrm{WL})}{@_j a, @_i\langle\mathsf{a}\rangle j, @_i\langle\beta\rangle b,\Gamma'\vdash\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\mathrm{S}_2)}{@_i\langle\mathsf{a}\rangle j, @_i\langle\beta\rangle b,\Gamma,\Gamma'\vdash\Delta,\Delta', @_i\langle\mathsf{a}\blacktriangle\beta\rangle}\,(\mathrm{Cut}_2)$$

Just as in earlier cases, the original (Cut) is replaced by two new applications: (Cut_1), which has strictly smaller cut height, and (Cut_2), which involves a simpler active cut. Both of these are less complex than the original in the lexicographic sense, so we can apply the inductive hypothesis to eliminate them. This shows that even when both rules in the (Cut) are right rules, the same general strategy still works. The remaining cases of interaction of rules in cut-elimination can be found in [11]. □

5 A Sequent System for Hybrid Logic

It is helpful to view HXPath$_{\mathrm{D}}$ as a modular extension of the Basic Hybrid Logic $\mathscr{H}(@)$ with additional constructs for data comparison. This naturally raises the question: can we recover a sequent calculus $\mathbf{G}'$ for $\mathscr{H}(@)$ by simply removing the rules related to data comparisons from $\mathbf{G}$? In this section, we make this correspondence precise by showing that the resulting fragment of $\mathbf{G}$ faithfully simulates the sequent calculus for $\mathscr{H}(@)$ presented in [17].

Definition 17. *A formula of $\mathscr{H}(@)$ is a node expression without data comparisons, and where diamonds are only of the form $\langle \mathsf{a}\rangle\varphi$. A hybrid model for $\mathscr{H}(@)$ is obtained by dropping the $\{\approx_{\mathsf{c}}\}_{\mathsf{c}\in\mathsf{Cmp}}$ component from the hybrid data models in Def. 2. The semantics of formulas on hybrid models is as in Def. 3. Let $\mathbf{G}'$ be obtained from $\mathbf{G}$ by restricting sequents to formulas in $\mathscr{H}(@)$ of the form $@_i\varphi$, and dropping all rules involving data comparisons.*

The calculus $\mathbf{G}'$ simulates the sequent calculus for $\mathscr{H}(@)$ in [17], which we refer to as $\mathbf{G}_{\mathscr{H}(@)}$

Lemma 18. *Every provable sequent in $\mathbf{G}_{\mathscr{H}(@)}$ is also provable in $\mathbf{G}'$.*

Proof. We establish this result by showing that every rule in $\mathbf{G}_{\mathscr{H}(@)}$ is a derived rule in $\mathbf{G}'$. The axioms in $\mathbf{G}_{\mathscr{H}(@)}$ align exactly with those in $\mathbf{G}'$, establishing a one-to-one correspondence. Additionally, ($\to$L), ($\to$R), (@L), and (@R) are present in both calculi. The rule (Ref) from $\mathbf{G}_{\mathscr{H}(@)}$ is also present in $\mathbf{G}'$ under the name (@T). The rules ($\wedge$L) and ($\wedge$R) are included as basic rules in $\mathbf{G}_{\mathscr{H}(@)}$, in contrast, they are derived in $\mathbf{G}'$. Similarly, the rules ([a]L) and ([a]R), and (Nom$_1$) and (Nom$_2$), listed below, which are primitive in $\mathbf{G}_{\mathscr{H}(@)}$, are also derivable in $\mathbf{G}'$. This shows that every derivation in $\mathbf{G}_{\mathscr{H}(@)}$ can be simulated in $\mathbf{G}'$, completing the correspondence between the two systems. Details can be found in [11].

$$\frac{\Gamma\vdash\Delta,@_i j\quad \Gamma\vdash\Delta,@_i\varphi}{\Gamma\vdash\Delta,@_j\varphi}\,(\mathrm{Nom}_1)\qquad \frac{\Gamma\vdash\Delta,@_i j\quad \Gamma\vdash\Delta,@_i\langle\mathsf{a}\rangle k\quad @_j\langle\mathsf{a}\rangle k,\Gamma\vdash\Delta}{\Gamma\vdash\Delta}\,(\mathrm{Nom}_2)$$

$$\frac{\Gamma\vdash\Delta,@_i\langle\mathsf{a}\rangle j\quad @_j\varphi,\Gamma\vdash\Delta}{@_i[\mathsf{a}]\varphi,\Gamma\vdash\Delta}\,([\mathsf{a}]\mathrm{L}_1)\qquad \frac{@_i\langle\mathsf{a}\rangle j,\Gamma\vdash\Delta,@_j\varphi}{\Gamma\vdash\Delta,@_i[\mathsf{a}]\varphi}\,([\mathsf{a}]\mathrm{R})\ j\text{ is new.}$$

As a concrete illustration of the correspondence between $\mathbf{G}_{\mathscr{H}(@)}$ and $\mathbf{G}$, we show how the rule Nom$_2$ from $\mathbf{G}_{\mathscr{H}(@)}$ can be obtained as a derived rule within $\mathbf{G}$.

$$\dfrac{\Gamma\vdash\Delta,@_i j\qquad \dfrac{\dfrac{\Gamma\vdash\Delta,@_i\langle\mathsf{a}\rangle k}{@_i j,\Gamma\vdash\Delta,@_i\langle\mathsf{a}\rangle k}\,(\mathrm{WL})\qquad \dfrac{\dfrac{@_j\langle\mathsf{a}\rangle k,\Gamma\vdash\Delta}{@_j\langle\mathsf{a}\rangle k,@_i j,@_i\langle\mathsf{a}\rangle k,\Gamma\vdash\Delta}\,(\mathrm{WL})}{@_i\langle\mathsf{a}\rangle k,@_i j,\Gamma\vdash\Delta}\,(\mathrm{S}_1)}{@_i j,\Gamma\vdash\Delta}\,(\mathrm{Cut})}{\Gamma\vdash\Delta}\,(\mathrm{Cut})$$

We can see from the derivation how (Nom$_1$) is directly related to (S$_1$) in $\mathbf{G}$. □

Theorem 19 (Soundness and Completeness)**.** *A sequent in $\mathscr{H}(@)$ is valid in all hybrid models iff it is provable in $\mathbf{G}'$.*

In conclusion, the system $\mathbf{G}_{\mathscr{H}(@)}$ in [17] is derived from a natural deduction system for $\mathscr{H}(@)$ presented in the same work. To reflect the structure and constraints of such a natural deduction system, $\mathbf{G}_{\mathscr{H}(@)}$ does not include weakening rules and, more importantly, is designed to be cut-free. This comes with a trade-off: the rules (Nom$_1$), (Nom$_2$), and ([a]L) in $\mathbf{G}_{\mathscr{H}(@)}$ are not invertible in $\mathbf{G}_{\mathscr{H}(@)}$. In contrast, as a subsystem of $\mathbf{G}$, $\mathbf{G}'$ is fully invertible and enjoys cut elimination. We bring attention also to the fact that, while (Cut) is admissible in $\mathbf{G}_{\mathscr{H}(@)}$, its use in a derivation cannot be systematically removed. Thus, by supporting both admissibility and effective elimination of (Cut), $\mathbf{G}'$ offers a more robust and analytically tractable framework for $\mathscr{H}(@)$.

6 Final Comments

In this article, we introduce the Gentzen-style sequent calculus **G** for the logic Hybrid XPath with Data ($\mathrm{HXPath_D}$). $\mathrm{HXPath_D}$ is a hybrid modal logic that captures not only the navigational core of XPath but also data comparisons, node labels (keys), and key-based navigation operators. In **G**, we provide rules for handling complex path expressions (composition and tests), as well as mechanisms for data comparisons (equality and inequality), nominals, and satisfiability operators.

We establish the completeness of **G** by leveraging the completeness of the Hilbert-style calculus **H** for $\mathrm{HXPath_D}$ introduced in [10]. We also show that every rule in **G** is invertible, a crucial property in proof search. The system **G** includes weakening rules (WL) and (WR), and the (Cut) rule. (WL) and (WR) are never needed in derivations (they are only used to simplify sequents and eliminate irrelevant formulas at a given point in a proof). Moreover, we showed that (Cut) is also irrelevant for completeness, and we provide a cut elimination algorithm. Finally, we discuss how to define a subcalculus of **G**, which is a sound and complete sequent calculus for the Basic Hybrid Logic $\mathscr{H}(@)$, inheriting rule invertibility and cut elimination. This system improves on previously known sequent calculi for $\mathscr{H}(@)$ (see [17]).

Similarly to the approach in [17], we can extend **G** by incorporating pure axioms and existential saturation rules. A general result can be proved, showing that these extended calculi are sound and complete with respect to a large family of frame classes (see [11] for further details). There is a cost though: rule invertibility and cut-elimination are not guaranteed in these extensions. Future work will explore whether these properties can be preserved under certain reasonable conditions.

Many aspects of XPath have been extensively investigated from a logical perspective, but their proof-theoretical properties have remained largely unexplored. To our knowledge, the only exception is the sequent calculus for DataGL from [14]. In brief, DataGL can be seen as a formalization of a highly restricted fragment of XPath in which data comparisons are allowed only between the evaluation point and its strict descendants in a data tree. In comparison, **G** provides much more expressivity.

The proof theory of modal logics has always been challenging. One reason for this is that modal languages combine a simple propositional syntax with a rich relational semantics. As a result, proof-theoretical approaches that rely solely on the *form* of logical expressions struggle to capture their full expressive power. Various proof-theoretical techniques——such as display calculi, hypersequents, deep inference, labeled systems, and hybridization——have been explored to establish a well-rounded proof theory for modal logics. Data-aware modal logics extend the expressive power of traditional modal languages even further, incorporating mechanisms to handle pairs of complex paths in graphs and data comparisons. Our results indicate that hybridization techniques (i.e., internalizing the use of labels via nominals and satisfiability operators) can provide the basis for a solid proof theory for modal logics with data-comparison operators.

There are several open lines for future research. In particular, we would like to prove the termination of our calculus and define an optimal proof search strategy. Moreover, we wish to extend the calculus to more expressive fragments—such as those involving sibling relations, transitive closure navigation, or novel forms of data comparison, and to study proof theoretical aspects of an intuitionistic variant of $\mathrm{HXPath_D}$ (see [8]).

References

[1] S. Abriola, P. Barceló, D. Figueira & S. Figueira (2016): *Bisimulations on Data Graphs*. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*, pp. 309–318.

[2] S. Abriola, P. Barceló, D. Figueira & S. Figueira (2018): *Bisimulations on Data Graphs*. *Journal of Artificial Intelligence Research* 61, pp. 171–213, doi:10.1613/jair.5637.

[3] S. Abriola, M. Descotte, R. Fervari & S. Figueira (2017): *Axiomatizations for downward XPath on data trees*. *Journal of Computer and System Sciences* 89, pp. 209–245, doi:10.1016/j.jcss.2017.05.008.

[4] S. Abriola, M. Descotte & S. Figueira (2014): *Definability for Downward and Vertical XPath on Data Trees*. In: *21th Workshop on Logic, Language, Information and Computation*, *LNCS* 6642, pp. 20–34, doi:10.1007/978-3-662-44145-9_2.

[5] S. Abriola, M. Descotte & S. Figueira (2017): *Model theory of XPath on data trees. Part II: Binary bisimulation and definability*. *Information and Computation* 255, pp. 195–223, doi:10.1016/J.IC.2017.01.002.

[6] S. Abriola, S. Figueira & N. González (2024): *Axiomatization of XPath with general data comparison*. *Journal of Applied Non-Classical Logics*, pp. 1–20.

[7] R. Angles, M. Arenas, P. Barceló, P. A. Boncz, G. H. L. Fletcher, C. Gutierrez, T. Lindaaker, M. Paradies, S. Plantikow, J. F. Sequeda, O. van Rest & H. Voigt (2018): *G-CORE: A Core for Future Graph Query Languages*. In: *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018*, ACM, pp. 1421–1432, doi:10.1145/3183713.3190654.

[8] C. Areces, V. Cassano, D. Dutto & R. Fervari (2023): *Data Graphs with Incomplete Information (and a Way to Complete Them)*. In: *18th European Conference on Logics in Artificial Intelligence (JELIA 2023)*, *LNCS* 14281, Springer, pp. 729–744, doi:10.1007/978-3-031-43619-2_49.

[9] C. Areces & B. ten Cate (2006): *Hybrid Logics*. In: *Handbook of Modal Logic*, Elsevier, pp. 821–868.

[10] C. Areces & R. Fervari (2021): *Axiomatizing Hybrid XPath with Data*. *Logical Methods in Computer Science* 17(3), doi:10.46298/LMCS-17(3:5)2021.

[11] Carlos Areces, Valentin Cassano, Danae Dutto & Raul Fervari (2025): *Sequent Calculi for Data-Aware Modal Logics: Compendium of Proofs*. Available at `https://arxiv.org/abs/2505.17240`.

[12] A. Avron (1996): *The method of hypersequents in the proof theory of propositional nonclassical logics*. In W. Hodges, M. Hyland, C. Steinhorn & J. Truss, editors: *Logic: from foundations to applications*, Oxford Science Publications, pp. 1–32.

[13] D. Baelde, A. Lick & S. Schmitz (2019): *Decidable XPath Fragments in the Real World*. In: *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019*, ACM, pp. 285–302, doi:10.1145/3294052.3319685.

[14] D. Baelde, S. Lunel & S. Schmitz (2016): *A Sequent Calculus for a Modal Logic on Finite Data Trees*. In: *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, *LIPIcs* 62, Schloss Dagstuhl, pp. 32:1–32:16, doi:10.4230/LIPICS.CSL.2016.32.

[15] M. Benedikt & C. Koch (2008): *XPath leashed*. *ACM Computing Surveys* 41(1), doi:10.1145/1456650.1456653.

[16] M. Bojańczyk, A. Muscholl, T. Schwentick & L. Segoufin (2009): *Two-variable logic on data trees and XML reasoning*. *Journal of the ACM* 56(3), doi:10.1145/1516512.1516515.

[17] T. Braüner (2011): *Hybrid Logic and its Proof-Theory*. *Applied Logics Series* 37, Springer.

[18] P. Buneman (1997): *Semistructured data*. In: *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS ’97, Association for Computing Machinery, New York, NY, USA, p. 117–121, doi:10.1145/263661.263675.

[19] B. ten Cate, T. Litak & M. Marx (2010): *Complete axiomatizations for XPath fragments*. *Journal of Applied Logic* 8(2), pp. 153–172, doi:10.1016/j.jal.2009.09.002.

[20] B. ten Cate & M. Marx (2009): *Axiomatizing the Logical Core of XPath 2.0*. *Theory of Computing Systems* 44(4), pp. 561–589, doi:10.1007/11965893_10.

[21] D. Figueira (2010): *Reasoning on Words and Trees with Data*. PhD thesis, Laboratoire Spécification et Vérification, ENS Cachan, France.

[22] D. Figueira (2012): *Decidability of Downward XPath*. *ACM Transactions on Computational Logic* 13(4), p. 34, doi:10.1145/2362355.2362362.

[23] D. Figueira (2018): *Satisfiability of XPath on data trees*. *ACM SIGLOG News* 5(2), pp. 4–16, doi:10.1145/3212019.3212021.

[24] D. Figueira, S. Figueira & C. Areces (2014): *Basic Model Theory of XPath on Data Trees*. In: *International Conference on Database Theory*, pp. 50–60, doi:10.5441/002/ICDT.2014.09.

[25] D. Figueira, S. Figueira & C. Areces (2015): *Model Theory of XPath on Data Trees. Part I: Bisimulation and Characterization*. *Journal of Artificial Intelligence Research* 53, pp. 271–314, doi:10.1613/JAIR.4658.

[26] D. Figueira & L. Segoufin (2011): *Bottom-up automata on data trees and vertical XPath*. In: *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, pp. 93–104, doi:10.4230/LIPICS.STACS.2011.93.

[27] N. Francis, A. Gheerbrant, P. Guagliardo, L. Libkin, V. Marsault, W. Martens, F. Murlak, L. Peterfreund, A. Rogova & D. Vrgoc (2023): *GPC: A Pattern Calculus for Property Graphs*. In: *42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, (PODS'23)*, ACM, pp. 241–250, doi:10.1145/3584372.3588662.

[28] N. Francis, A. Gheerbrant, P. Guagliardo, L. Libkin, V. Marsault, W. Martens, F. Murlak, L. Peterfreund, A. Rogova & D. Vrgoc (2023): *A Researcher's Digest of GQL*. In: *26th International Conference on Database Theory*, *LIPIcs* 255, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 1:1–1:22, doi:10.4230/LIPICS.ICDT.2023.1.

[29] G. Gottlob, C. Koch & R. Pichler (2005): *Efficient algorithms for processing XPath queries*. *ACM Transactions on Database Systems* 30(2), pp. 444–491, doi:10.1145/1071610.1071614.

[30] (2024): *ISO/IEC 39075. 2024. Information technology - Database languages - GQL*. Standard, International Organization for Standardization, Geneva, CH.

[31] L. Libkin, W. Martens & D. Vrgoč (2016): *Querying Graphs with Data*. *Journal of the ACM* 63(2), pp. 14:1–14:53, doi:10.1145/2850413.

[32] L. Libkin & D. Vrgoč (2012): *Regular path queries on graphs with data*. In: *International Conference on Database Theory*, ACM, pp. 74–85, doi:10.1145/2274576.2274585.

[33] S. Negri (2005): *Proof analysis in modal logic*. *Journal of Philosophical Logic* 34, pp. 507–544.

[34] S. Negri (2011): *Proof Theory for Modal Logic*. *Philosophy Compass* 6(8), pp. 523–538.

[35] S. Negri & J. von Plato (2014): *Proof Analysis: A Contribution to Hilbert's Last Problem*. Cambridge University Press.

[36] I. Robinson, J. Webber & E. Eifrem (2013): *Graph Databases*. O'Reilly Media, Inc.

[37] C. Stewart & P. Stouppa (2004): *A Systematic Proof Theory for Several Modal Logics*. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds & H. Wansing, editors: *Advances in Modal Logic 5, 2004*, King's College Publications, pp. 309–333.

[38] H. Wansing (2002): *Sequent systems for modal logics*. In Dov Gabbay & Franz Guenthner, editors: *Handbook of Philosophical Logic*, 2 edition, 8, Kluwer, pp. 61–145.