A Hypersequent Calculus with Clusters for Data Logic over Ordinals

Anthony Lick

A Hypersequent Calculus with Clusters for Data Logic over Ordinals *

Anthony Lick* (junior researcher)

LSV, CNRS & ENS Paris-Saclay, Université Paris-Saclay, France

Abstract. We study freeze tense logic over well-founded data streams. The logic features past- and future-navigating modalities along with freeze quantifiers, which store the datum of the current position and test data (in)equality later in the formula. We introduce a decidable fragment of that logic, and present a proof system that is sound for the whole logic, and complete for this fragment. Technically, this is a hypersequent system enriched with an ordering, clusters, and annotations. The proof system is tailored for proof search, and yields an optimal coNP complexity for validity and a small model property for our fragment.

Keywords: Modal logic · Data ordinals · Freeze logic · Proof system · Hypersequent calculus

1 Introduction

Data Streams. Many applications can generate *data streams*, such as traces of a program's execution [21], system logs [7], XML streams [18], or intrusions detection [20], which motivates the study of *data words* and *data ω-words* in order to be able to formally reason about such streams. They consist respectively of finite and infinite sequences in which each position carries a *label* from a finite alphabet and a *datum* from an infinite domain.

Consider for instance a system where multiple processes could be editing the same file on some server. The log of their execution can be represented as an infinite data word, the datum being an integer identifying the process, and the label representing their action: b for the beginning of a process, e for its ending, and r (resp. w) when a process reads (resp. writes) the file. On such a data ω-word, we could want to verify various properties:

1. Every process does not do anything after it stops or before it starts, i.e. for every datum, the corresponding subword belongs to $b(r+w)^*e$.
2. For every position labelled by w, there exists a previous position labelled by r with the same datum such that there is no a position in-between labelled by w and carrying a different datum.

On the following infinite data word, only the first property is respected.

$$(b,1)(r,1)(b,2)(r,2)(w,1)(w,2)(e,2)(e,1)(b,3)(r,3)(e,3)\ \cdots$$

* Funded by ANR-14-CE28-0005 PRODAQ. I am grateful to David Baelde and Sylvain Schmitz for their many valuable advices, and for proofreading this paper.

Data Logics. Among the many logics developed to reason about data words, freeze LTL [14,17,30] extends linear temporal logic [36] with *freeze quantifiers*: a formula $\downarrow_r \varphi$ stores the current datum in register r and evaluates φ; in this scope, $\uparrow_r$ is satisfied if the current datum matches the one stored in r. As always with data logics, the satisfiability problem for freeze LTL is undecidable and its known decidable fragments are untractable [14,17].

Contributions. In this paper, we investigate the *freeze tense logic* over data ordinals, which we call $\mathbf{K_t^{\downarrow} L_{\ell}.3}$, and which combines freeze quantifiers à la Demri and Lazić [14] with the tense logic over ordinals $\mathbf{K_t L_{\ell}.3}$. Our temporal core is thus Prior's tense logic [39,9], which only features the strict 'past' H and 'future' G temporal modalities (and their duals P and F), but this is sufficient for many modelling tasks [41], and is known to lead to an NP-complete satisfiability problem over arbitrary linear time flows [35], over ω-words [40], and over arbitrary ordinals [3]. For instance, property 1 above can be expressed by

$$\begin{aligned}&\mathsf{G}\,\big(b \supset \downarrow_r \big(\mathsf{H}\,\neg\uparrow_r \wedge\, \mathsf{G}\,(b \supset \neg\uparrow_r) \wedge \mathsf{F}\,(\uparrow_r \wedge\, e \wedge \mathsf{G}\,\neg\uparrow_r \wedge \mathsf{H}\,(e \supset \neg\uparrow_r))\big)\big)\\ \wedge\, &\mathsf{G}\,((e \vee r \vee w) \supset \downarrow_r \mathsf{P}\,(\uparrow_r \wedge\, b))\end{aligned}$$

The full freeze tense logic $\mathbf{K_t^{\downarrow} L_{\ell}.3}$ is already undecidable with a single register, just like freeze LTL over finite words [14,17]. We present a decidable fragment, dubbed $\mathbf{K_t^{d} L_{\ell}.3}$, in which the use of registers is further restricted, and which is exactly as expressive as the two-variable fragment of the first order logic on data words [6]. We show in particular that

1. the satisfiability problem for $\mathbf{K_t^{d} L_{\ell}.3}$ over the class of ordinals is NP-complete,
2. a formula φ of $\mathbf{K_t^{d} L_{\ell}.3}$ has a well-founded linear model if and only if it has a model of order type α for some $\alpha < \omega \cdot (4 \cdot |\varphi|^2 + |\varphi| + 2)$; this should be contrasted with the corresponding $\omega \cdot (|\varphi| + 2)$ bound proven in [3, Prop. 4.1] for the underlying data-free logic $\mathbf{K_t L_{\ell}.3}$.

These results are however just by-products of our main contribution, which is a sound and complete proof system for $\mathbf{K_t^{d} L_{\ell}.3}$ in which proof search is in coNP.

Moreover, our system allows to work not only with data ω-words but with arbitrary data ordinals, which provides greater modelling flexibility: for instance, Demri and Novak [15] use ordinals to model Zeno behaviours in physical systems [15], and Godefroid and Wolper [19] to model-check n concurrent executions while avoiding exploring their $n!$ interleavings.

Algorithmic Approach. Algorithmic results for data logics are often obtained via *automata-theoretic* techniques [42,14,6], by building an enriched automaton recognising the models of a given formula and testing it for emptiness. However, this kind of approach might not be modular as the type of enriched automata is often tailored for each specific logic. Moreover, if one's interest is to check that a formula φ is valid, the automata-theoretic approach does not yield a 'natural' certificate that could be checked by simple independent means.

All these considerations motivate our use of *proof-theoretic* techniques. The primary example of proof system amenable to automated reasoning is Gentzen's

sequent calculus. However, it is often too limited for modal logics, it has been enriched in various ways, using e.g. labelled sequents [34], display calculus [5,28], nested sequents [27,8,37,38,31], or hypersequents [1,22,29,23]. These enriched formalisms remain quite modular and sustain extensions simply by adding a few rules. They can be exploited to provide optimal complexity solutions to the validity problem directly by proof search [25,33,4,12,2,3], which may sometimes avoid the worst-case complexity of the problem and rely in practice on various heuristics. Finally, this approach obviously yields a proof of validity as a certificate in case of success.

Specifically, we use the framework of ordered hypersequents *with clusters* introduced in [2] as an elaboration, with terminating proof search, of Indrzejczak's ordered hypersequent calculus for $\mathbf{K_t 4.3}$ [23,24], and which was then generalised in [3] to work over ordinals. Conceptually, re-using the framework required to adapt it to work with data ordinals, to use additional rules to deal with registers, and to develop a strategy to make sure that proof search always produces proof attempts of polynomial depth. Moreover, this framework uses *annotations* to bound the proof search, and we managed to handle them as a new type of formulæ rather than just as an artefact of the proof system.

Furthermore, as in [3], our proof system can easily adapted to also address the more precise problems of validity over all the data ordinals of order type $\beta < \alpha + 1$ for a given α and of order type exactly $\alpha < \omega^2$—which is recalled in Section 6. Such a result seems out of reach of axiomatisations, and yields for instance a coNP decision procedure for validity over data ω-words.

2 Freeze Tense Logic over Ordinals

Syntax. Our logic, called $\mathbf{K_t^{\downarrow} L_\ell .3}$, features two unary temporal operators from the tense logic, and countably many freeze operators, over a countable set Φ of propositional variables, with the following syntax:

$$\varphi ::= \ \bot \mid p \mid \varphi \supset \varphi \mid \mathsf{G}\,\varphi \mid \mathsf{H}\,\varphi \mid \downarrow_r \varphi \mid \uparrow_r \qquad \text{(where } p \in \Phi \text{ and } r \in \mathbb{N}\text{)}$$

Formulæ $\mathsf{G}\,\varphi$ and $\mathsf{H}\,\varphi$ are called *modal formulæ*. Intuitively, $\mathsf{G}\,\varphi$ expresses that φ holds 'globally' in all future worlds, while $\mathsf{H}\,\varphi$ expresses that φ holds 'historically' in all past worlds. Other Boolean connectives may be encoded from $\bot$ and $\supset$, and as usual $\mathsf{F}\,\varphi = \neg\mathsf{G}\,\neg\varphi$ expresses that φ will hold 'in the future' and $\mathsf{P}\,\varphi = \neg\mathsf{H}\,\neg\varphi$ that it held 'in the past'. Formulæ $\downarrow_r \varphi$ are called *freeze formulæ*, and atoms $\uparrow_r$ are called *thaw formulæ*. Intuitively, $\downarrow_r \varphi$ stores the datum of the current world in the register r, and evaluates φ, and $\uparrow_r$ tests if the current world has the same datum as the one stored in the register r. Any occurrence of a thaw $\uparrow_r$ within the scope of a freeze quantifier $\downarrow_r$ is bounded by it; otherwise, that thaw is *free*.

Furthermore, in order to guide the proof search, our calculus will have to manipulate a different kind of future formulæ called *annotations*: these formulæ will be written $(\mathsf{G}\,\varphi)$, where $\mathsf{G}\,\varphi$ is a future modal formula, and will express that $\mathsf{G}\,\varphi$ holds starting from a specific later position. Remark that such formulæ cannot ever appear as a subformula.

Data Ordinal Semantics. Recall that an ordinal α is seen set-theoretically as $\{\beta \in \mathrm{Ord} \mid \beta < \alpha\}$. An ordinal is either 0 (the empty linear order), a *limit* ordinal λ (such that for all $\beta < \lambda$ there exists γ with $\beta < \gamma < \lambda$), or a *successor* ordinal $\alpha + 1$. In the case of $\mathbf{K}_{\mathbf{t}}^{\downarrow}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$, our formulæ shall be evaluated on *data ordinals*, which are couples (α, δ) with α an ordinal and δ a function mapping elements from α to a datum from an infinite[1] domain $\mathbb{D}$. Models of our logic are Kripke *structures* $\mathfrak{M} = (\mathfrak{F}, V, \nu)$, where the frame $\mathfrak{F} = (\alpha, \delta)$ is a data ordinal, $V : \Phi \to \alpha$ is a valuation of the propositional variables, and ν is a finite partial map from $\mathbb{N}$ to $\mathbb{D}$ called a *register valuation*. The domain of such a ν must contain all the free registers that appear in the formulæ it evaluates.

Given a structure $\mathfrak{M} = ((\alpha, \delta), V)$ and a register valuation ν, we define the *satisfaction* relation $\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \varphi$, where $\beta < \alpha$, $\theta < \alpha$ and φ is a formula, by structural induction on φ. Notice that θ is only used for the annotations.

$$
\begin{array}{ll}
\mathfrak{M}, \beta \not\models_{\nu}^{(\theta)} \bot & \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} p & \text{iff } \beta \in V(p) \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \varphi \supset \psi & \text{iff if } \mathfrak{M}, \beta \models_{\nu}^{(\theta)} \varphi \text{ then } \mathfrak{M}, \beta \models_{\nu}^{(\theta)} \psi \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \mathsf{G}\,\varphi & \text{iff } \mathfrak{M}, \gamma \models_{\nu}^{(\theta)} \varphi \text{ for all } \beta < \gamma < \alpha \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \mathsf{H}\,\varphi & \text{iff } \mathfrak{M}, \gamma \models_{\nu}^{(\theta)} \varphi \text{ for all } \gamma < \beta \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \downarrow_r \varphi & \text{iff } \mathfrak{M}, \beta \models_{\nu[r \mapsto \delta(\beta)]}^{(\theta)} \varphi \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \uparrow_r & \text{iff } \delta(\beta) = \nu(r) \\
\mathfrak{M}, \beta \models_{\nu}^{(\theta)} (\mathsf{G}\,\varphi) & \text{iff } \beta < \theta, \text{and } \mathfrak{M}, \gamma \models_{\nu}^{(\theta)} \varphi \\
 & \text{for all } \gamma \text{ such that } \theta \leq \gamma < \alpha
\end{array}
$$

When $\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \varphi$, we say that $(\mathfrak{M}, \nu, \beta, (\theta))$ is a *model* of φ. Remark that, since annotations cannot appear as subformulæ, we have $\mathfrak{M}, \beta \models_{\nu}^{(\theta)} \varphi$ if and only if $\mathfrak{M}, \beta \models_{\nu}^{(\theta')} \varphi$ for any θ', when φ is not an annotation.

Substitutions. We note $[x/y](\varphi)$ for the formula φ where every free occurrence of the register y is replaced by the register x. More formally, we define it by structural induction as follows:

$$
\begin{array}{rlrl}
[x/y](\bot) &= \bot & [x/y](p) &= p \\
[x/y](\mathsf{G}\,\varphi) &= \mathsf{G}\,[x/y](\varphi) & [x/y](\mathsf{H}\,\varphi) &= \mathsf{H}\,[x/y](\varphi) \\
[x/y](\varphi_1 \supset \varphi_2) &= [x/y](\varphi_1) \supset [x/y](\varphi_2) & & \\
[x/y](\uparrow_r) &= \uparrow_r \text{ if } r \neq y & [x/y](\uparrow_y) &= \uparrow_x \\
[x/y](\downarrow_r \varphi) &= \downarrow_r [x/y](\varphi) \text{ if } r \neq y \text{ and } r \neq x & [x/y](\downarrow_y \varphi) &= \downarrow_y \varphi \\
[x/y](\downarrow_x \varphi) &= \downarrow_r [x/y]([r/x](\varphi)) \text{ where } r \text{ is fresh} & &
\end{array}
$$

[1] Since we will only be able to perform equality tests between data values, we can assume that $\mathbb{D}$ is countable.

Example 1. Even though the underlying data-free logic $\mathbf{K_t L_\ell.3}$ cannot express that a model is of order type at least ω^2 [3], this can be done with $\mathbf{K_t^{\downarrow} L_\ell.3}$, even without using any propositional variable. Consider for this $\varphi_1 = \mathsf{G}\,(\downarrow_r \mathsf{F} \uparrow_r)$, $\varphi_2 = \mathsf{G}\,(\downarrow_r \mathsf{F} \neg\uparrow_r)$, and $\varphi_3 = \mathsf{G}\,(\downarrow_r \mathsf{G}\,(\mathsf{F}\uparrow_r \supset \uparrow_r))$. Then, $\varphi = \mathsf{F}\top \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3$ is satisfiable, and any model of φ is of order type at least ω^2.

Thanks to the conjunct $\mathsf{F}\top$, the other formulæ do not quantify over an empty set of future positions: there exists at least a future β_1. The conjunct φ_1 forces that every datum appears infinitely many times, and φ_3 forces that every such infinite sequence of positions carrying the same datum is continuous (two such sequences for two different data cannot be interleaved). Hence, any model of φ starts with at least ω positions carrying $d_1 = \delta(\beta_1)$. In turn, φ_2 forces the existence of β_2 carrying a datum d_2 such that $d_1 \neq d_2$, which due to φ_3 must be after the positions carrying d_1; and because of φ_2 we must have at least ω positions carrying the datum d_2. Again, φ_2 forces the existence of β_3 carrying d_3 different from d_1 and d_2, and thus ω positions carrying d_3 must exist, etc. By repeating this reasoning, any model of φ must comprise at least ω positions carrying the datum d, for infinitely many $d \in \mathbb{D}$, so is of order type at least ω^2.

Moreover, φ is indeed satisfied by a model of order type ω^2, where the ith ω carries d_i, for an enumeration $(d_i)_{i\in\mathbb{N}}$ of $\mathbb{D}$.

3 Hypersequents with Clusters

As is often the case with modal logics, Gentzen's sequent calculus does not provide a rich enough framework to obtain complete proof systems. The extension we consider is to use *hypersequents* [1], which are essentially sets of sequents logically interpreted as a disjunction. Indrzejczak has moved to *ordered* hypersequents [23,24] (which are lists of hypersequents) to obtain a sound and complete calculus for $\mathbf{K_t 4.3}$. We have further enriched the structure of his ordered hypersequents with *clusters* and annotations to obtain calculi for $\mathbf{K_t 4.3}$ [2] and $\mathbf{K_t L_\ell.3}$ [3] for which proof search terminates and, in fact, yields an optimal complexity decision procedure. We keep the same structure in the present work, but simplify the annotation mechanism, and add rules to handle freeze formulæ. In Section 4, we present our proof system, and prove that it is sound for $\mathbf{K_t^{\downarrow} L_\ell.3}$. In Section 5, we focus on a decidable fragment of $\mathbf{K_t^{\downarrow} L_\ell.3}$, and prove that our calculus is complete for that fragment, and has a proof strategy of optimal complexity.

3.1 Annotated Hypersequents with Clusters

Sequents. A *sequent* (denoted S) is a couple consisting of two finite sets Γ, Δ of formulæ, written $\Gamma \vdash \Delta$. It is satisfied by worlds β and θ of a structure $\mathfrak{M}$ if there exists a register valuation ν such that $\mathfrak{M}, \beta \models_\nu^{(\theta)} \bigwedge \Gamma \supset \bigvee \Delta$ (where $\bigwedge \Gamma$ and $\bigvee \Delta$ denote respectively the conjunction of the formulæ of Γ and the disjunction of the formulæ of Δ).

Hypersequents. A *hypersequent* is a list of *cells*, each cell being either a sequent or a non-empty list of sequents called a (syntactic) *cluster*. We shall use the

following abstract syntax, where both operators ';' and '‖' are associative with unit '•':

$$H ::= C \mid H\,;H \qquad \text{(hypersequents)}$$
$$C ::= \bullet \mid S \mid \{Cl\} \qquad \text{(cells)}$$
$$Cl ::= S \mid Cl \parallel Cl \qquad \text{(cluster contents)}$$

Note that this definition allows for empty cells and hypersequents '•', but these notational conveniences will never arise in actual proofs—and should not be confused with the empty sequent '⊢'. We will see that the order of cells in a hypersequent is semantically relevant, but the order of sequents inside a cluster is not. Nevertheless, assuming an ordering as part of the syntactic structure of clusters is useful in order to refer to specific sequents or positions.

3.2 Semantics

The semantics of an ordered hypersequent with clusters relies on a notion of embedding, building on a view of hypersequents as partially ordered structures.

Partial Order of a Hypersequent. Let H be a hypersequent containing n sequents, counting both the sequents found directly in its cells and those in its clusters. In this context, any $i \in [1;n]$ is called a *position* of H, and we write $H(i)$ for the i-th sequent of H. We define a partial order $\precsim$ on the positions of H by setting $i \precsim j$ if and only if either the i-th and j-th sequents are in the same cluster, or the i-th sequent is in a cell that lies strictly to the left of the cell of the j-th sequent. We write $i \prec j$ when $i \precsim j$ but $j \not\precsim i$, i.e. j lies strictly to the right of i in H. We write $i \sim j$ when $i \precsim j \precsim i$. Finally, the *domain* of H is defined as $\mathrm{dom}(H) = ([1;n], \precsim)$; note that empty cells are ignored in $\mathrm{dom}(H)$.

While a hypersequent is syntactically a finite partial order, its semantics will refer to a linear well-founded order, obtained by 'bulldozing' its clusters into copies of ω. This defines the *order type* $o(H)$ of H by induction on its structure: for cells, $o(\bullet) = 0$, $o(S) = 1$, and $o(\{Cl\}) = \omega$, and for hypersequents, $o(H_1\,; H_2) = o(H_1) + o(H_2)$. Thus, $o(H) = \omega \cdot k + m$ where k is the number of clusters in H and m the number of non-empty cells to the right of the rightmost cluster.

Embeddings. Let H be a hypersequent and α an ordinal. We say that $\mu : \mathrm{dom}(H) \to \alpha + 1 \setminus \{0\}$ is an *embedding* of H into α, written $H \hookrightarrow_\mu \alpha$, if

- for all $i, j \in \mathrm{dom}(H)$, $i \prec j$ implies $\mu(i) < \mu(j)$ and $i \sim j$ implies $\mu(i) = \mu(j)$
- and for all $i \in \mathrm{dom}(H)$, i is in a cluster if and only if $\mu(i)$ is a limit ordinal.

Observe that, if $H \hookrightarrow_\mu \alpha$, then $o(H) < \alpha + 1$.

Semantics. A structure $\mathfrak{M}$ is a *model* of a hypersequent H if there exists a register valuation ν, an embedding $\mathfrak{M} \hookrightarrow_\mu H$, and a position i of H such that for all $d \in \mathbb{D}$ there exists an ordinal $\beta_d < \mu(i)$ such that for all γ such that $\beta_d \leq \gamma < \mu(i)$ and $\delta(\gamma) = d$, we have $\mathfrak{M}, \gamma \models_\nu^{(\mu(i))} H(i)$. In that case, we write $\mathfrak{M}, \nu, \mu \models H$.

Following this definition, we say that a hypersequent is *valid* if for any $\mathfrak{M} = ((\alpha, \delta), V)$, any embedding $H \hookrightarrow_\mu \mathfrak{M}$ and any register valuation ν, $\mathfrak{M}, \nu, \mu \models H$. A formula φ is valid in the usual sense (i.e., satisfied in every world of every ordinal structure) if and only if the hypersequent $\vdash \varphi$ is valid in our sense.

If a hypersequent H is not valid, then it has a *counter-model*, that is a structure $\mathfrak{M} = ((\alpha, \delta), V)$, an embedding $H \hookrightarrow_\mu \mathfrak{M}$ and a register valuation ν such that for every $i \in \mathrm{dom}(H)$ there exists $d_i \in \mathbb{D}$ such that for every $\beta < \mu(i)$, there exists γ with $\beta \leq \gamma < \mu(i)$ and $\delta(\gamma) = d_i$ such that $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(i))} H(i)$. For the positions $i \in \mathrm{dom}(H)$ that are not in clusters, $\mu(i)$ is a successor ordinal $\gamma + 1$ and this amounts to asking that $\mathfrak{M}, \gamma \not\models_\nu^{(\gamma+1)} H(i)$. When i is in a cluster, the condition implies the existence of an infinite increasing sequence $(\gamma_j)_j$ of ordinals carrying the same datum, and with limit $\mu(i) = \sup_j \gamma_j$, such that $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(i))} H(i)$ for all j.

4 Proof System

We now present our proof system for $\mathbf{K}_\mathbf{t}^{\downarrow}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$, called $\mathbf{HK}_\mathbf{t}^\mathbf{d}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$. The rules of $\mathbf{HK}_\mathbf{t}^\mathbf{d}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ are given in figures 1 to 3: the first group includes the usual propositional rules, the second deals with modalities, the third deals with annotations, and the last one with freeze formulæ. The figures make use of some notations which we explain next, before commenting on the rule definitions themselves.

4.1 Notations

First, we use hypersequents with *holes*. One-placeholder hypersequents, cells, and clusters are defined by the following syntax:

$$H\,[] ::= H\,;C\,[]\,;H \quad C\,[] ::= \star \mid \{\ Cl[]\ \} \quad Cl[] ::= Cl_\bullet \parallel \star \parallel Cl_\bullet \quad Cl_\bullet ::= \bullet \mid Cl$$

Two-placeholder cells and hypersequents have two holes identified by $\star_1$ and $\star_2$:

$$H\,[]\,[] ::= H\,;C\,[]\,[]\,;H \mid H[\star_1]\,;H[\star_2]$$
$$C\,[]\,[] ::= \{\ Cl[\star_1] \parallel Cl[\star_2]\ \} \mid \{\ Cl[\star_2] \parallel Cl[\star_1]\ \}$$

As usual, $C\,[S]$ (resp. $C\,[Cl]$) denotes the same cell with S (resp. Cl) substituted for $\star$; two-placeholder cells and hypersequents with holes behave similarly. In terms of the partial orders underlying hypersequents with two holes, observe that the positions i and j associated resp. to $\star_1$ and $\star_2$ are exactly such that $i \precsim j$.

Second, we use a convenient notation for *enriching* a sequent: if S is a sequent $\Gamma \vdash \Delta$, then $S \ltimes (\Gamma' \vdash \Delta')$ is the sequent $\Gamma, \Gamma' \vdash \Delta, \Delta'$. Moreover, we sometimes need to enrich an arbitrary sequent of a cluster $\{Cl\}$ with a sequent S; then $\{Cl\} \ltimes S$ denotes the cluster with its leftmost sequent enriched.

Finally, we write $[x/y](H)$ for the hypersequent H where the operator $[x/y]$ has been applied to every formula.

$$\frac{}{H\,[\varphi, \Gamma \vdash \Delta, \varphi]}\ (\mathsf{ax}) \qquad \frac{H\,[\varphi \supset \psi, \Gamma \vdash \Delta, \varphi] \quad H\,[\varphi \supset \psi, \psi, \Gamma \vdash \Delta]}{H\,[\varphi \supset \psi, \Gamma \vdash \Delta]}\ (\supset \vdash)$$

$$\frac{}{H\,[\Gamma, \bot \vdash \Delta]}\ (\bot) \qquad \frac{H\,[\varphi, \Gamma \vdash \Delta, \psi, \varphi \supset \psi]}{H\,[\Gamma \vdash \Delta, \varphi \supset \psi]}\ (\vdash \supset)$$

Fig. 1. Propositional rules of $\mathbf{HK^d_t L_\ell.3}$.

$$\frac{H\,[\mathsf{G}\,\varphi, \Gamma \vdash \Delta]\,[\varphi, \mathsf{G}\,\varphi, \Pi \vdash \Sigma]}{H\,[\mathsf{G}\,\varphi, \Gamma \vdash \Delta]\,[\Pi \vdash \Sigma]}\ (\mathsf{G}\vdash) \qquad \frac{H_1; \{Cl_\bullet \parallel \varphi, \mathsf{G}\,\varphi, \Gamma \vdash \Delta \parallel Cl'_\bullet\}; H_2}{H_1; \{Cl_\bullet \parallel \mathsf{G}\,\varphi, \Gamma \vdash \Delta \parallel Cl'_\bullet\}; H_2}\ (\{\mathsf{G}\vdash\})$$

$$\frac{H\,[\varphi, \mathsf{H}\,\varphi, \Pi \vdash \Sigma]\,[\mathsf{H}\,\varphi, \Gamma \vdash \Delta]}{H\,[\Pi \vdash \Sigma]\,[\mathsf{H}\,\varphi, \Gamma \vdash \Delta]}\ (\mathsf{H}\vdash) \qquad \frac{H_1; \{Cl_\bullet \parallel \varphi, \mathsf{H}\,\varphi, \Gamma \vdash \Delta \parallel Cl'_\bullet\}; H_2}{H_1; \{Cl_\bullet \parallel \mathsf{H}\,\varphi, \Gamma \vdash \Delta \parallel Cl'_\bullet\}; H_2}\ (\{\mathsf{H}\vdash\})$$

$$\frac{\begin{array}{l} H_1\,;\,C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi]\,;\,(\mathsf{G}\,\varphi) \vdash \varphi\,;\,C'\,;\,H_2 \\ H_1\,;\,C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi]\,;\,\{(\mathsf{G}\,\varphi) \vdash \varphi\}\,;\,C'\,;\,H_2 \\ H_1\,;\,C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi \parallel (\mathsf{G}\,\varphi) \vdash \varphi]\,;\,C'\,;\,H_2 \quad \text{if } C \neq \star \\ H_1\,;\,C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi]\,;\,C' \ltimes (\vdash \mathsf{G}\,\varphi)\,;\,H_2 \quad \text{if } C' \neq \bullet \\ H_1\,;\,C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi]\,;\,C' \ltimes ((\mathsf{G}\,\varphi) \vdash \varphi)\,;\,H_2 \text{ if } C' \neq \bullet \text{ and } C' \neq \{Cl\} \end{array}}{H_1\,;\,C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi]\,;\,C'\,;\,H_2}\ (\vdash\mathsf{G})$$

$$\frac{\begin{array}{l} H_2\,;\,C'\,;\,\mathsf{H}\,\varphi \vdash \varphi\,;\,C\,[\Gamma \vdash \Delta, \mathsf{H}\,\varphi]\,;\,H_1 \\ H_2\,;\,C' \ltimes (\vdash \mathsf{H}\,\varphi)\,;\,C\,[\Gamma \vdash \Delta, \mathsf{H}\,\varphi]\,;\,H_1 \quad \text{if } C' \neq \bullet \\ H_2\,;\,C' \ltimes (\mathsf{H}\,\varphi \vdash \varphi)\,;\,C\,[\Gamma \vdash \Delta, \mathsf{H}\,\varphi]\,;\,H_1 \text{ if } C' \neq \bullet \text{ and } C' \neq \{Cl\} \end{array}}{H_2\,;\,C'\,;\,C\,[\Gamma \vdash \Delta, \mathsf{H}\,\varphi]\,;\,H_1}\ (\vdash\mathsf{H})$$

Fig. 2. Modal rules of $\mathbf{HK^d_t L_\ell.3}$. In $(\vdash\mathsf{G})$ and $(\vdash\mathsf{H})$, we allow $C' = \bullet$ only when $H_2 = \bullet$.

4.2 Rules

We now comment on our rules. The rules from figures 1 and 2 are the same as in [3]. The propositional rules of Figure 1 are straightforward: they are the usual ones applied to an arbitrary sequent of the hypersequent.

The left modal rules of Figure 2 should not be surprising. For instance, in $(\mathsf{G}\vdash)$, if the conclusion has a counter-model, then $\mathsf{G}\,\varphi$ holds at some ordinal and thus both φ and $\mathsf{G}\,\varphi$ must also hold at strictly greater ordinals. The rule also applies to two distinct sequents inside the same cluster. The $(\{\mathsf{G}\vdash\})$ rule allows to proceed in the same way inside a cluster when the sequent 'further to the right' is the original sequent itself, something that our notations do not allow in $(\mathsf{G}\vdash)$. Finally, $(\mathsf{H}\vdash)$ and $(\{\mathsf{H}\vdash\})$ are symmetric to the two previous rules.

The rules $(\vdash\mathsf{G})$ and $(\vdash\mathsf{H})$ are the most complex ones. We shall not try to justify their soundness at this point, but simply make a few remarks that are important to understand their definition. First, these rules are the only ones that may introduce new cells in hypersequents. In $(\vdash\mathsf{G})$, new cells come with the annotation $(\mathsf{G}\,\varphi)$ of the principal formula $\mathsf{G}\,\varphi$, which will help bounding the proof search, as we will see in Lemma 1.

Second, the principal cell $C\,[\Gamma \vdash \Delta, \mathsf{G}\,\varphi]$ in $(\vdash\mathsf{G})$ may be the rightmost cell of the conclusion hypersequent, in which case both C' and H_2 are empty, and

$$\frac{H_1\,[(\mathsf{G}\,\varphi), \Gamma \vdash \Delta]\,;H_2\,[\varphi, \mathsf{G}\,\varphi, \Pi \vdash \Sigma]}{H_1\,[(\mathsf{G}\,\varphi), \Gamma \vdash \Delta]\,;H_2\,[\Pi \vdash \Sigma]}\ ((\mathsf{G}))\qquad \frac{[x/y](H\,[\uparrow_x, \Gamma \vdash \Delta])}{H\,[\uparrow_x, \uparrow_y, \Gamma \vdash \Delta]}\ (\uparrow \vdash)$$

$$\frac{\begin{array}{ll} H\,[\Gamma, \downarrow_r \varphi, \uparrow_x, [x/r](\varphi) \vdash \Delta] & \text{if } \forall y \in \mathbb{N}, \uparrow_y \notin \Gamma, \text{ with } x \text{ fresh} \\ H\,[\Gamma, \downarrow_r \varphi, [x/r](\varphi) \vdash \Delta] & \text{if } \uparrow_x \text{ is the only thaw atom in } \Gamma \end{array}}{H\,[\Gamma, \downarrow_r \varphi \vdash \Delta]}\ (\downarrow \vdash)$$

$$\frac{\begin{array}{ll} H\,[\Gamma, \uparrow_x \vdash [x/r](\varphi), \downarrow_r \varphi, \Delta] & \text{if } \forall y \in \mathbb{N}, \uparrow_y \notin \Gamma, \text{ with } x \text{ fresh} \\ H\,[\Gamma \vdash [x/r](\varphi), \downarrow_r \varphi, \Delta] & \text{if } \uparrow_x \text{ is the only thaw atom in } \Gamma \end{array}}{H\,[\Gamma \vdash \downarrow_r \varphi, \Delta]}\ (\vdash \downarrow)$$

Fig. 3. Annotation, freeze, and thaw rules of $\mathbf{HK^d_t L_\ell.3}$. By fresh, we mean that x does not appear as a free register anywhere in the conclusion.

the rule has two or three premises depending on whether the principal cell is a cluster or not. When the principal cell is not rightmost, then C' is not allowed to be empty, and the rule has one or two extra premises depending on whether C' is a cluster or not. The situation is symmetric for $(\vdash \mathsf{H})$.

The annotations rules from [3] are now subsumed by the new rule $((\mathsf{G}))$ which is similar to $(\mathsf{G} \vdash)$, the difference being that an annotation $(\mathsf{G}\,\varphi)$ cannot affect the current cluster.

The other rules from Figure 3 are new. The rule $(\uparrow \vdash)$ unify two registers when they must contain the same datum, and is helpful to bound the number of registers appearing in the proof search. The rules $(\downarrow \vdash)$ and $(\vdash \downarrow)$ both handle the freeze quantifier $\downarrow_r$ by adding a version of φ where r has been replaced by either an already used register matching the current datum if any, or a fresh one otherwise.

Finally, we have designed our rules so that they are all *invertible*: by keeping in premises all the formulæ from the conclusion, we ensure that validity is never lost by applying a rule; this is proved in Appendix A.1.

Proposition 1 (invertibility). *In any rule instance, if a premise has a counter-model, then so does its conclusion.*

In practice, keeping all formulæ can be unnecessarily heavy. Fortunately, it is easy to see that the following weakening rules are admissible:

$$\frac{H\,[\Gamma \vdash \Delta]}{H\,[\Gamma, \varphi \vdash \Delta]}\ (\text{weak} \vdash)\qquad \frac{H\,[\Gamma \vdash \Delta]}{H\,[\Gamma \vdash \varphi, \Delta]}\ (\vdash \text{weak})$$

4.3 Soundness

Our calculus is sound w.r.t $\mathbf{K^d_t L_\ell.3}$, which is proved in Appendix A.2.

Proposition 2. *The rules of* $\mathbf{HK^d_t L_\ell.3}$ *are sound: if the premises of a rule instance are valid, then so is its conclusion.*

Invertibility is not enough to obtain completeness since proof search does not terminate, $\mathbf{K^{\downarrow}_t L_\ell.3}$ being undecidable. We now investigate a decidable fragment for which $\mathbf{HK^d_t L_\ell.3}$ is complete and has a proof strategy of optimal complexity.

5 Restricted Logic and Completeness

The previous logic is known to be undecidable, even with only one register [14] and some restrictions regarding the use of that register [17]. Here, we consider another restriction of the logic, and prove that our calculus is complete for that fragment, with proof search in coNP.

5.1 Restricted Syntax

We consider the following fragment of $\mathbf{K}_{\mathbf{t}}^{\downarrow}\mathbf{L}_{\ell}\mathbf{.3}$, that we call $\mathbf{K}_{\mathbf{t}}^{\mathbf{d}}\mathbf{L}_{\ell}\mathbf{.3}$:

$$\begin{aligned}\varphi ::= \ & \bot \mid p \mid \varphi \supset \varphi \mid \mathsf{G}\,\varphi \mid \mathsf{H}\,\varphi \\ & \mid \downarrow_r \mathsf{G}\,(\uparrow_r \supset \varphi) \mid \downarrow_r \mathsf{G}\,(\neg\uparrow_r \supset \varphi) \\ & \mid \downarrow_r \mathsf{H}\,(\uparrow_r \supset \varphi) \mid \downarrow_r \mathsf{H}\,(\neg\uparrow_r \supset \varphi) \qquad \text{(where } p \in \Phi \text{ and } r \in \mathbb{N})\end{aligned}$$

Because the use of registers is restricted to such specific formulæ, we define the following syntactic sugar:

$$\mathsf{G}_{=r}\,\varphi = \mathsf{G}\,(\uparrow_r \supset \varphi) \qquad \mathsf{G}_{\neq r}\,\varphi = \mathsf{G}\,(\neg\uparrow_r \supset \varphi)$$
$$\mathsf{H}_{=r}\,\varphi = \mathsf{H}\,(\uparrow_r \supset \varphi) \qquad \mathsf{H}_{\neq r}\,\varphi = \mathsf{H}\,(\neg\uparrow_r \supset \varphi)$$

Intuitively, $\mathsf{G}_{=r}\,\varphi$ (resp. $\mathsf{G}_{\neq r}\,\varphi$) expresses the fact that φ holds in every future position with the same (resp. a different) datum as the one stored in the register r; and $\mathsf{H}_{=r}\,\varphi$, $\mathsf{H}_{\neq r}\,\varphi$ express the same for past positions. Moreover, since a negation before a freeze quantifier can be moved inside its scope, e.g. $\neg\downarrow_r \mathsf{G}_{\neq r}\,\neg\varphi \equiv \downarrow_r \neg \mathsf{G}_{\neq r}\,\neg\varphi$, we can also define their dual diamond modularities, e.g. $\mathsf{F}_{\neq r}\,\varphi = \neg\mathsf{G}_{\neq r}\,\neg\varphi$. Formulæ of the form $\downarrow_r \mathsf{G}_{=r}\,\varphi$ (resp. $\downarrow_r \mathsf{G}_{\neq r}\,\varphi$) corresponds to formulæ denoted $\Box_{=}\varphi$ (resp. $\Box_{\neq}\varphi$) from [4], which works over data trees.

Example 2. The formula from Example 1, forcing its models to have order type at least ω^2, does not belong to this fragment. Furthermore, there is no equivalent formula belonging to $\mathbf{K}_{\mathbf{t}}^{\mathbf{d}}\mathbf{L}_{\ell}\mathbf{.3}$, as we will show later that satisfiable formulæ from this fragment always have a model of order type strictly below ω^2.

Property 2 from the introduction does not seem expressible either, since it would require to perform nested data tests. However, property 1 can be expressed by the following formula:

$$\begin{aligned}&\mathsf{G}\,(b \supset (\neg\downarrow_r \mathsf{P}_{=r}\,\top \wedge \neg\downarrow_r \mathsf{F}_{=r}\,b \wedge \downarrow_r \mathsf{F}_{=r}\,(e \wedge \neg\downarrow_r \mathsf{F}_{=r}\,\top \wedge \neg\downarrow_r \mathsf{P}_{=r}\,e))) \\ \wedge &\mathsf{G}\,((e \vee r \vee w) \supset \downarrow_r \mathsf{P}_{=r}\,b)\end{aligned}$$

From now on, we only consider formulæ from $\mathbf{K}_{\mathbf{t}}^{\mathbf{d}}\mathbf{L}_{\ell}\mathbf{.3}$.

5.2 Completeness and Complexity

As in [2,3], completeness is a by-product of a rather simple proof-search strategy. As already stated in 1, all the rules are invertible; and as we shall see, our strategy only produce proof trees with branches that are polynomially bounded for the restricted logic, as it will avoid any pitfall that could happen. Thus it is useless to backtrack during proof-search. Moreover, proof attempts result in finite (polynomial depth) partial proofs, whose unjustified leaves yield counter-models

that amount (by invertibility) to counter-models of the conclusion. Hence the completeness of our calculus. We detail this argument below, and its corollary: proof-search yields an optimal coNP procedure for validity.

We characterise next the proof attempts that we consider for proof search, and show how to extract counter-models when such attempts fail.

Lemma 1. *If a hypersequent H satisfies one of these conditions, then H is provable (and we say that H is* immediately provable*).*

(a) There exists a formula φ, and two positions $i \prec j$ of H such that $H(i)$ and $H(j)$ both contain $(\mathsf{G}\,\varphi) \vdash \varphi$.
(b) There exists a formula φ, and two positions $i \prec j$ of H such that $H(i)$ and $H(j)$ both contain $\mathsf{H}\,\varphi \vdash \varphi$.
(c) There exists a formula φ, three positions $i \prec j \prec k$ of H, and three registers $x, y, z \in \mathbb{N}$ such that:
  – $H(i)$ contains $(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\!\uparrow_x \supset \varphi$.
  – $H(j)$ contains $(\mathsf{G}_{\neq y}\,\varphi) \vdash \neg\!\uparrow_y \supset \varphi$.
  – $H(k)$ contains $(\mathsf{G}_{\neq z}\,\varphi) \vdash \neg\!\uparrow_z \supset \varphi$.
(d) There exists a formula φ, three positions $i \prec j \prec k$ of H, and three registers $x, y, z \in \mathbb{N}$ such that:
  – $H(i)$ contains $\mathsf{H}_{\neq x}\,\varphi \vdash \neg\!\uparrow_x \supset \varphi$.
  – $H(j)$ contains $\mathsf{H}_{\neq y}\,\varphi \vdash \neg\!\uparrow_y \supset \varphi$.
  – $H(k)$ contains $\mathsf{H}_{\neq z}\,\varphi \vdash \neg\!\uparrow_z \supset \varphi$.

We provide a proof tree for every case in Appendix A.3. The intuition behind (d) is the following: if there exists γ where $\mathsf{H}_{\neq z}\,\varphi$ holds, and $\gamma' < \gamma$ where $\mathsf{H}_{\neq y}\,\varphi$ holds, and if y and z stores different data, then φ holds in every past position of γ' (at any position, either $\neg\!\uparrow_z$ or $\neg\!\uparrow_y$ holds) and thus any $\mathsf{G}_{\neq x}\,\varphi$ holds in the past. The intuition behind (c) is similar. This reasoning fails if y and z store the same datum, but this cannot be assumed during proof search only when $\uparrow_y$ and $\uparrow_z$ appear on the left-hand side of the same sequent, and in this case we should apply $(\uparrow\vdash)$ in priority.

Partial Proofs. We characterise now the proof attempts that we consider for proof search, and show how to extract counter-models when such attempts fail. We call *partial proof* a finite derivation tree whose internal nodes correspond to rule applications, but whose leaves may be unjustified hypersequents, and that satisfies two conditions:

(a) no rule application should be such that, if H is the conclusion hypersequent,
  (i) one of the premises is also H, or
  (ii) the rule being applied is $(\vdash\mathsf{G})$ on a formula $\mathsf{G}\,\varphi$ at position i such that there exists $j \sim i$ such that $H(j)$ contains $(\mathsf{G}\,\varphi) \vdash \varphi$, or
  (iii) the rule being applied is $(\vdash\mathsf{G})$ on a formula $\mathsf{G}\,(\neg\!\uparrow_x \supset \varphi)$ at position i such that there exists $j \sim i$ and $y \neq x$ such that $H(j)$ contains $\big(\mathsf{G}\,(\neg\!\uparrow_y \supset \varphi)\big) \vdash \neg\!\uparrow_y \supset \varphi$ and does not contain $\uparrow_x$ on its left-hand side.
(b) If the rule $(\uparrow\vdash)$ is applicable, or if the rule $(\vdash\supset)$ is applicable on a formula of the form $\uparrow_x \supset \varphi$, then the other rules cannot be applied.
(c) immediately provable hypersequents must be proven immediately as sketched in the proof of Lemma 1.

Finally, we call *failure hypersequent* a hypersequent on which any rule application would not respect condition (a).

The second part of condition (b) is there to optimize the use of its first part, which in turn is there to bound the number of registers our calculus manipulates during a proof search. Conditions (a) and (c) amount to a simple proof search strategy that avoids loops, and addresses especially loops arising from repeated applications of $(\vdash \mathsf{H})$ or $(\vdash \mathsf{G})$, in branches where several new cells are created for the same modal formula (up to maybe a different register): this results either in immediately provable hypersequents from Lemma 1, or failure hypersequents on which the proof strategy is stuck and for which we prove next that we can always construct a counter-model.

Example 3. The annotation rules from [3] are subsumed by our new version of the rule $((\mathsf{G}))$. For instance, if H has a position i that is not in a cluster such that $H(i)$ contains $(\mathsf{G}\,\varphi) \vdash \mathsf{G}\,\varphi$, the branch can be immediately closed by some rule from [3]. Let us show that such an H is provable by $\mathbf{HK}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$. First of all, since $(\mathsf{G}\,\varphi) \in H(i)$, then either $H(i)$ contains $(\mathsf{G}\,\varphi) \vdash \varphi$, or there exist $j \prec i$ such that $H(j)$ contains it. Then:

- If $(\vdash \mathsf{G})$ cannot be applied on $\mathsf{G}\,\varphi$, it is either because $H(i+1)$ contains $(\mathsf{G}\,\varphi) \vdash \varphi$, and then H is immediately provable, or because $\mathsf{G}\,\varphi$ also appears on the right-hand side of $H(i+1)$, and the same formula can also be sent on its left-hand side (if not already present) by applying $((\mathsf{G}))$ on the annotation $(\mathsf{G}\,\varphi)$, and then (ax) can be used.
- Else, we apply $(\vdash \mathsf{G})$ on $\mathsf{G}\,\varphi$. All premises are immediately provable, except for the premise sending $\mathsf{G}\,\varphi$ on the right-hand side of $H(i+1)$ which can be proved as in the previous case.

Proposition 3. *Any failure hypersequent H has a counter-model.*

Proof (sketch). The detailed proof is provided in Appendix A.4. We first construct a counter-model $\mathfrak{M} = ((\alpha, \delta), V)$ of H with $\alpha = o(H)$ and a straightforward embedding $\mathfrak{M} \hookrightarrow_\mu H$. We also describe a function $\mathrm{pos} : \alpha \to \mathrm{dom}(H)$ which will act as the reverse of μ. Then, δ is chosen such that every pair of worlds from $\mathfrak{M}$ carry distinct data unless their corresponding sequent in H carries the same atomic $\uparrow_r$ on their left-hand side; and V is constructed similarly: every atom p is false except in worlds for which the corresponding sequent carries p on its left-hand side. We finally prove that $\mathfrak{M}$ is a counter-model of H by structural induction on the subformulæ of H.

Example 4. Consider the following partial proof of the sequent $\vdash \downarrow_r \mathsf{G} \uparrow_r$:

$$\dfrac{\dfrac{\uparrow_x \vdash \downarrow_r \mathsf{G}\uparrow_r, \mathsf{G}\uparrow_x \,;\, (\mathsf{G}\uparrow_x) \vdash \uparrow_x \qquad \dfrac{\cdots \quad \uparrow_x \vdash \downarrow_r \mathsf{G}\uparrow_r, \mathsf{G}\uparrow_x \,;\, \{(\mathsf{G}\uparrow_x) \vdash \uparrow_x, \mathsf{G}\uparrow_x\}}{\uparrow_x \vdash \downarrow_r \mathsf{G}\uparrow_r, \mathsf{G}\uparrow_x \,;\, \{(\mathsf{G}\uparrow_x) \vdash \uparrow_x\}}\,(\vdash \mathsf{G})}{\uparrow_x \vdash \downarrow_r \mathsf{G}\uparrow_r, \mathsf{G}\uparrow_x}\,(\vdash \downarrow)}{\vdash \downarrow_r \mathsf{G}\uparrow_r}\,(\vdash \mathsf{G})$$

The first leaf is a failure hypersequent, since case (i) of condition (a) prevents any rule application. Its corresponding counter-model consists of to worlds with distinct data. The other branch will reach immediately provable hypersequent (not displayed on the figure), and another failure hypersequent, since cases (i) and (ii) of condition (a) prevent any rule application. Its corresponding counter-model consists of a first world with a datum d_x, followed by an infinite sequence of worlds all carrying another datum $d_y \neq d_x$.

We now turn to establishing that proof search terminates, and always produces branches of polynomial length. For a hypersequent H, let $\mathsf{len}(H)$ be its number of sequents (i.e., the size of $\mathrm{dom}(H)$), and $|H|$ the number of distinct subformulæ occurring in H.

Lemma 2. *For any partial proof of a hypersequent H, any branch of the proof is of length at most* $2|H|(4|H| + \mathsf{len}(H))((4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1)$.

Proof (sketch). Each creation of a new position along a branch of the proof search could lead to the creation of a new register later in the branch, which in turn could create a new renamed copy of some subformula of H, which then could lead to the creation of another position. We must make sure that such a process cannot go ad infinitum: we first bound the numbers of free registers that can appear along a branch respecting our proof strategy, then the size of the hypersequents of such a branch, and finally the number of rules that can be applied. The details are presented in Appendix A.5.

Example 5. If we did not follow our strategy, a bad case such as described at the beginning of the previous proof could happen on the hypersequent $H = \vdash ; \varphi \vdash$ with $\varphi = \mathsf{H}(\neg\downarrow_r \mathsf{H}_{=r} \bot)$. In practice, φ can send the subformula $\downarrow_r \mathsf{H}_{=r} \bot$ on the right-hand side of any past position by using $(\mathsf{H}\vdash)$ and then handling the negation. A proof of H will start as follows:

$$\dfrac{\dfrac{\dfrac{\mathsf{H}_{=x} \bot \vdash \uparrow_x \supset \bot ; \varphi, \uparrow_x \vdash \mathsf{H}_{=x} \bot, \downarrow_r \mathsf{H}_{=r} \bot ; \varphi \vdash}{\varphi, \uparrow_x \vdash \mathsf{H}_{=x} \bot, \downarrow_r \mathsf{H}_{=r} \bot ; \varphi \vdash}\,(\vdash\mathsf{H})}{\varphi \vdash \downarrow_r \mathsf{H}_{=r} \bot ; \varphi \vdash}\,(\vdash \downarrow)}{\vdash ; \varphi \vdash}$$

Our strategy would now force to handle the formula $\uparrow_x \supset \bot$. If we do not respect that, and instead send $\downarrow_r \mathsf{H}_{=r} \bot$ on the right-hand side of the leftmost position and apply $(\vdash \downarrow)$ again, a new register y would be created, along with the formula $\mathsf{H}_{=y} \bot$, which in turn will create a new position more in the past when applying $(\vdash\mathsf{H})$. If we never deal with a formula of the form $\uparrow_x \supset \bot$, this process could go on ad infinitum, alternating between creating a new register and creating a new position. However, if we respect our strategy, the proof search will reach an immediately provable hypersequent after creating a fourth position. It is not surprising, since we can prove that a counter-model of H should be such that every datum appearing in the past do so infinitely many times, which is impossible as our models are well-founded.

We now conclude that $\mathbf{HK}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ is complete for $\mathbf{K}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$, and also enjoys optimal complexity proof search.

Theorem 1 (completeness). *Our calculus is complete for* $\mathbf{K_t^d L_\ell.3}$*: every valid hypersequent* H *has a proof in* $\mathbf{HK_t^d L_\ell.3}$.

Proof. Assume that H is not provable. Consider a partial proof $\mathcal{P}$ of H that cannot be expanded any more: its unjustified leaves are failure hypersequents. Such a partial proof exists by Lemma 2. Any unjustified leaf of that partial proof has a counter-model by Proposition 3, and by invertibility shown in Proposition 1 it is also a counter-model of H.

Proposition 4. *Proof search in* $\mathbf{HK_t^d L_\ell.3}$ *is in* coNP.

Proof. Proof search can be implemented in an alternating Turing machine maintaining the current hypersequent on its tape, with only universal states (choosing a premise of the rule): by Proposition 1, we can choose an arbitrary order in which to apply rules; and the choice of a fresh x by any application of $(\downarrow \vdash)$ or $(\vdash \downarrow)$ does not matter (e.g., x can be taken as the next unused integer). Moreover, by Lemma 2, the computation branches are of length bounded by a polynomial, hence the Turing machine is in coNP.

6 Restricted Logic on Given Ordinals

We have designed a proof system that is sound and complete for $\mathbf{K_t^d L_\ell.3}$, and enjoys optimal complexity proof search. Moreover, as in [3], we can derive a small model property from the proof of completeness: the logic $\mathbf{K_t^d L_\ell.3}$ can only distinguish ordinals up to ω^2, as the underlying data-free logic [3].

Proposition 5 (small model property). *If a hypersequent* H *has a counter-model, then it has one of order type* $\alpha < \omega \cdot ((4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1)$.

Proof. This is a corollary of Theorem 1. By the proof of Lemma 2, the hypersequents in a failure branch—which are not immediately provable—have at most $(4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1$ non-empty sequents. The counter-model extracted in Proposition 3 from a failure hypersequent H' is over $o(H') < \omega \cdot ((4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1)$. A counter-model for H is then obtained by Proposition 1, with a different embedding but the same structure.

In particular, for a formula φ, the hypersequent $H = \vdash \varphi$ has $|H| = |\varphi|$ and $\mathsf{len}(H) = 1$, hence the $\omega \cdot (4 \cdot |\varphi|^2 + |\varphi| + 2)$ bound announced in the introduction.

Furthermore, as in [3], we can easily enrich our calculus by the following rule to obtain a proof system for tense logic over data ordinals below a certain type α.

$$\frac{}{H}\ (\mathsf{ord}_\alpha) \text{ if } o(H) > \alpha$$

We can also capture validity at a fixed ordinal $\alpha < \omega^2$, by padding the input with enough empty sequents to start with a hypersequent H of order type α, and enriching our calculus with rule (ord_α) to forbid larger ordinals (as in [3]). The only catch is that we should check that the formula of interest

in valid in all possible positions, i.e. considering all possible paddings leading to a hypersequent of order type α. When checking validity of a formula φ in all structures of order type exactly α, we must prove in $\mathbf{HK}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ extended with (ord_α) all hypersequents of order type α containing one sequent $\vdash \varphi$ and otherwise only empty sequents. For instance, when $\alpha = \omega$ we must check $\vdash \varphi;\{\vdash\}$.

7 Related Work and Conclusion

We have investigated $\mathbf{K}^{\downarrow}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$—the freeze tense logic over ordinals—and proposed a decidable fragment, namely $\mathbf{K}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$, for which we designed a sound and complete proof system.

Thanks to Indrzejczak's ordered hypersequents [23], enriched with clusters and annotations as in [2,3], our system enjoys optimal coNP proof search, allows to derive small model properties, and can be extended into a proof system for variants of the logic over bounded or fixed data ordinals.

First-Order Logic with Two Variables. Bojańczyk et al. [6] have shown that validity in first-order logic with two variables over data words and data ω-words is in coNEXP. The same statement can be derived from our results, since $\mathbf{K}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ is exactly as expressive as $\mathbf{FO}^2(<, \sim)$. We detail this aspect in Appendix B: converting a first-order formulæ into an equivalent $\mathbf{K}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ formulæ can be done by adapting the proof from [16]—which involves an exponential blow-up—, we can then apply Theorem 4 to get a coNEXP decision procedure.

Other logics. Our fragment can be encoded in the Logic of Repeating Values with Past from [13] (PLRV), for which the satisfiability problem is equivalent to the problem of reachability in VASS, which is TOWER-hard [11], and with an ACKERMANN complexity upper bound [32]. $\mathbf{K}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ can also be encoded in the fragment of XPath with data tests and navigation among siblings which has been proved undecidable [17].

In both cases, the main difference is that $\mathbf{K}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ cannot perform nested data tests. However, this restriction allowed us to get a logic for which the satisfiability problem has a smaller complexity (NP), as established by our proof system. The complexities of various logics on data words and their inclusions is summed up in Appendix C.

The systems most closely related to $\mathbf{HK}^{\mathbf{d}}_{\mathbf{t}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ is obviously the calculus for $\mathbf{K_t4.3}$ [2] and $\mathbf{K_tL}_{\boldsymbol{\ell}}\mathbf{.3}$ [3] in which we respectively introduced the notions of clusters and annotations, and adapted them to work over ordinals. The main contribution of the paper is being able to maintain the small branch property of the calculus with the addition of data registers. Another contribution is the shift of perspective about the annotations. In [3], they were only considered as an artefact of the proof system being able to guide the proof search; but in this paper, they are treated as a new kind of formula, and generalising the notion of immediately provable hypersequent introduced in [3] allowed us to mimic the syntactic condition they were previously bound to.

References

1. Avron, A.: Hypersequents, logical consequence and intermediate logics for concurrency. Annals of Mathematics and Artificial Intelligence **4**(3–4), 225–248 (1991). https://doi.org/10.1007/BF01531058
2. Baelde, D., Lick, A., Schmitz, S.: A hypersequent calculus with clusters for linear frames. In: AiML '18. Advances in Modal Logic, vol. 12, pp. 36–55. College Publications (2018), https://hal.inria.fr/hal-01756126
3. Baelde, D., Lick, A., Schmitz, S.: A hypersequent calculus with clusters for tense logic over ordinals. In: FSTTCS '18. Leibniz International Proceedings in Informatics, vol. 122, pp. 15:1–15:19. LZI (2018). https://doi.org/10.4230/LIPIcs.FSTTCS.2018.15
4. Baelde, D., Lunel, S., Schmitz, S.: A sequent calculus for a modal logic on finite data trees. In: CSL '16. Leibniz International Proceedings in Informatics, vol. 62, pp. 32:1–32:16. LZI (2016). https://doi.org/10.4230/LIPIcs.CSL.2016.32
5. Belnap, N.D.: Display logic. J. Philos. Logic **11**(4), 375–417 (1982). https://doi.org/10.1007/BF00284976
6. Bojańczyk, M., David, C., Muscholl, A., Schwentick, T., Segoufin, L.: Two-variable logic on data words. ACM Transactions on Computational Logic **12**(4), 27:1–27:26 (2011). https://doi.org/10.1145/1970398.1970403
7. Bollig, B.: An automaton over data words that captures EMSO logic. In: Concur '11. Lecture Notes in Computer Science, vol. 6901, pp. 171–186. Springer (2011). https://doi.org/10.1007/978-3-642-23217-6_12
8. Brünnler, K.: Deep sequent systems for modal logic. Archiv für Mathematische Logik und Grundlagenforschung **48**(6), 551–577 (2009). https://doi.org/10.1007/s00153-009-0137-3
9. Cocchiarella, N.B.: Tense and Modal Logic: a Study in the Topology of Temporal Reference. Ph.D. thesis, University of California, Los Angeles (1965)
10. Colcombet, T., Manuel, A.: Generalized data automata and fixpoint logic. In: FSTTCS '14. Leibniz International Proceedings in Informatics, vol. 29, pp. 267–278. LZI (2014). https://doi.org/10.4230/LIPIcs.FSTTCS.2014.267
11. Czerwiński, W., Lasota, S., Lazić, R., Leroux, J., Mazowiecki, F.: The reachability problem for Petri nets is not elementary. In: STOC '19. ACM (2019), to appear
12. Das, A., Pous, D.: A cut-free cyclic proof system for Kleene algebra. In: TABLEAUX '17. Lecture Notes in Computer Science, vol. 10501, pp. 261–277. Springer (2017). https://doi.org/10.1007/978-3-319-66902-1_16
13. Demri, S., Figueira, D., Praveen, M.: Reasoning about data repetitions with counter systems. Logical Methods in Computer Science **12**(3), 1 (2016). https://doi.org/10.2168/LMCS-12(3:1)2016
14. Demri, S., Lazić, R.: LTL with the freeze quantifier and register automata. ACM Transactions on Computational Logic **10**(3), 16 (2009). https://doi.org/10.1145/1507244.1507246
15. Demri, S., Nowak, D.: Reasoning about transfinite sequences. International Journal of Fundations of Computer Science **18**(01), 87–112 (2007). https://doi.org/10.1142/S0129054107004589
16. Etessami, K., Vardi, M.Y., Wilke, T.: First-order logic with two variables and unary temporal logic. Information and Computation **179**(2), 279–295 (2002). https://doi.org/10.1006/inco.2001.2953
17. Figueira, D., Segoufin, L.: Future-looking logics on data words and trees. In: MFCS '09. Lecture Notes in Computer Science, vol. 5734, pp. 331–343. Springer (2009). https://doi.org/10.1007/978-3-642-03816-7_29

18. Gauwin, O., Niehren, J., Tison, S.: Queries on XML streams with bounded delay and concurrency. Information and Computation **209**(3), 409–442 (2011). https://doi.org/10.1016/j.ic.2010.08.003
19. Godefroid, P., Wolper, P.: A partial approach to model checking. Information and Computation **110**(2), 305–326 (1994). https://doi.org/10.1006/inco.1994.1035
20. Goubault-Larrecq, J., Olivain, J.: On the efficiency of mathematics in intrusion detection: The NetEntropy case. In: FPS '14. Lecture Notes in Computer Science, vol. 8352, pp. 3–16. Springer (2014). https://doi.org/10.1007/978-3-319-05302-8_1
21. Grigore, R., Distefano, D., Petersen, R.L., Tzevelekos, N.: Runtime verification based on register automata. In: TACAS '13. Lecture Notes in Computer Science, vol. 7795, pp. 260–276. Springer (2013). https://doi.org/10.1007/978-3-642-36742-7_19
22. Indrzejczak, A.: Cut-free hypersequent calculus for S4.3. Bull. Sec. Logic **41**(1/2), 89–104 (2012)
23. Indrzejczak, A.: Linear time in hypersequent framework. Bulletin of Symbolic Logic **22**(1), 121–144 (2016). https://doi.org/10.1017/bsl.2016.2
24. Indrzejczak, A.: Cut elimination theorem for non-commutative hypersequent calculus. Bull. Sec. Logic **46**(1/2), 135–149 (2017). https://doi.org/10.18778/0138-0680.46.1.2.10
25. Kanovich, M.: The multiplicative fragment of linear logic is NP-complete. Tech. Rep. X-91-13, Institute for Language, Logic, and Information (1991)
26. Kara, A., Schwentick, T., Zeume, T.: Temporal logics on words with multiple data values. In: FSTTCS '10. Leibniz International Proceedings in Informatics, vol. 8, pp. 481–492. LZI (2010). https://doi.org/10.4230/LIPIcs.FSTTCS.2010.481
27. Kashima, R.: Cut-free sequent calculi for some tense logics. Studia Logica **53**(1), 119–135 (1994). https://doi.org/10.1007/BF01053026
28. Kracht, M.: Power and weakness of the modal display calculus. In: Proof Theory of Modal Logic, pp. 93–121. Springer (1996). https://doi.org/10.1007/978-94-017-2798-3_7
29. Kurokawa, H.: Hypersequent calculi for modal logics extending S4. In: JSAI-isAI '13. Lecture Notes in Computer Science, vol. 8417, pp. 51–68. Springer (2014). https://doi.org/10.1007/978-3-319-10061-6_4
30. Lazić, R.: Safety alternating automata on data words. ACM Transactions on Computational Logic **12**(2), 10:1–10:24 (2011). https://doi.org/10.1145/1877714.1877716
31. Lellmann, B.: Linear nested sequents, 2-sequents and hypersequents. In: TABLEAUX '15. Lecture Notes in Computer Science, vol. 9323, pp. 135–150. Springer (2015). https://doi.org/10.1007/978-3-319-24312-2_10
32. Leroux, J., Schmitz, S.: Reachability in vector addition systems is primitive-recursive in fixed dimension. In: LICS '19. IEEE (2019), http://arxiv.org/abs/1903.08575, to appear
33. Lincoln, P., Mitchell, J., Scedrov, A., Shankar, N.: Decision problems for propositional linear logic. Annals of Pure and Applied Logic **56**(1–3), 239–311 (1992). https://doi.org/10.1016/0168-0072(92)90075-B
34. Negri, S.: Proof analysis in modal logic. J. Philos. Logic **34**(5), 507–544 (2005). https://doi.org/10.1007/s10992-005-2267-3
35. Ono, H., Nakamura, A.: On the size of refutation Kripke models for some linear modal and tense logics. Studia Logica **39**(4), 325–333 (1980). https://doi.org/10.1007/BF00713542

36. Pnueli, A.: The temporal logic of programs. In: FOCS '77. pp. 46–57. IEEE (1977). https://doi.org/10.1109/SFCS.1977.32
37. Poggiolesi, F.: The method of tree-hypersequents for modal propositional logic. In: Trends in Logic IV. Trends in Logic, vol. 28, pp. 31–51. Springer (2009). https://doi.org/10.1007/978-1-4020-9084-4_3
38. Poggiolesi, F.: A purely syntactic and cut-free sequent calculus for the modal logic of provability **2**(4), 593–611 (2009). https://doi.org/10.1017/S1755020309990244
39. Prior, A.N.: Time and Modality. Oxford University Press (1957)
40. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. J. ACM **32**(3), 733–749 (1985). https://doi.org/10.1145/3828.3837
41. Sistla, A.P., Zuck, L.D.: Reasoning in a restricted temporal logic. Information and Computation **102**(2), 167–195 (1993). https://doi.org/10.1006/inco.1993.1006
42. Vardi, M.Y., Wolper, P.: An automata-theoretic approach to automatic program verification. In: LICS '86. pp. 322–331. IEEE (1986)

A Detailed Proofs

A.1 Invertibility

Proposition 1 (invertibility). *In any rule instance, if a premise has a counter-model, then so does its conclusion.*

Proof. Considering a rule instance with a counter-model $(\mathfrak{M}, \nu, \mu)$ of a premise H, we build a counter-model $(\mathfrak{M}, \nu', \mu')$ of the conclusion H'. Depending on the rule that is applied, H and H' will either have exactly the same structure, or H will have a new cell. Accordingly, we take μ' to be the restriction of μ to the positions of H' (and adapt it accordingly for the positions that have been shifted). It is indeed a proper embedding of H' into $\mathfrak{M}$. Moreover, except for the rule $(\uparrow \vdash)$, all the free $\uparrow_r$ of H' are also free registers of H, so taking $\nu' = \nu$ suffices in these cases. For the case of $(\uparrow \vdash)$, H' must have two formulæ $\uparrow_x$ and $\uparrow_y$ on the left-hand side of some sequent such that $\uparrow_y$ is not a subformula of H, and we can take $\nu' = \nu[y \mapsto \nu(x)]$. It is then easy to see that $(\mathfrak{M}, \nu', \mu')$ is a counter-model of H', since any sequent $H'(i)$ is contained in the corresponding sequent $H(j)$ (up to some register renaming for the rule $(\uparrow \vdash)$): for any β, $\mathfrak{M}, \beta \not\models_{\nu}^{(\mu(j))} H(j)$ implies $\mathfrak{M}, \beta \not\models_{\nu'}^{(\mu'(i))} H'(i)$.

A.2 Soundness

Not surprisingly, the following substitution lemma holds, which will be used in the next proof.

Lemma 3. *For every formula φ, and every model $(\mathfrak{M}, \nu, \beta, (\theta))$ of φ:*

- *if $\nu(x) = \nu(y)$, then $\mathfrak{M}, \beta \models_{\nu}^{(\theta)} [x/y](\varphi)$.*
- *if no free occurrence of $\uparrow_x$ appears in φ, then $\mathfrak{M}, \beta \models_{\nu[x \mapsto \nu(y)]}^{(\theta)} [x/y](\varphi)$.*

Proposition 2. *The rules of $\mathbf{HK}_{\mathbf{t}}^{\mathbf{d}}\mathbf{L}_{\boldsymbol{\ell}}\mathbf{.3}$ are sound: if the premises of a rule instance are valid, then so is its conclusion.*

Proof. We show the contrapositive: considering an application of a rule with a conclusion hypersequent H and a counter-model $(\mathfrak{M}, \nu, \mu)$ of H with $\mathfrak{M} = (\alpha, V)$ and $H \hookrightarrow_{\mu} \alpha$ an embedding, we provide a counter-model of one of the premises (or a contradiction when there is no premise).

Since we will often have to extend an embedding with a value for a new position, we define $\mu + (i \mapsto \alpha)$ as the mapping μ' such that $\mu'(i) = \alpha$, $\mu'(k) = \mu(k)$ for $k < i$ and $\mu'(k+1) = \mu(k)$ for $k \geq i$ in the domain of μ.

The case of propositional rules (Figure 1) is immediate: The usual reasoning applies to the principal sequent, and the same embedding is used to obtain a counter-model of one of the premises.

Next we turn to the modal rules of Figure 2:

- Consider the case of $(\mathsf{G}\vdash)$, applied with $\mathsf{G}\,\varphi, \Gamma \vdash \Delta$ at position i and $\Pi \vdash \Sigma$ at position j such that $i \precsim j$. Remark that the rule ensures that $i \neq j$, but we do not need this assumption to justify it. We show that $(\mathfrak{M}, \nu, \mu)$ is an counter-model of the premise H', concentrating on the only difference with H, at position j. For clarity we distinguish two cases:
  - When $i \prec j$, we also have $\mu(i) < \mu(j)$. Since $(\mathfrak{M}, \nu, \mu)$ is a counter-model of H, by taking an arbitrary $\beta_i < \mu(i)$ we obtain γ_i such that $\beta_i \leq \gamma_i < \mu(i)$ such that $\mathfrak{M}, \gamma_i \not\models_\nu^{(\mu(i))} H(i)$. In particular, $\mathfrak{M}, \gamma_i \models_\nu^{(\mu(i))} \mathsf{G}\,\varphi$. Now, considering an arbitrary $\beta < \mu(j)$ we need to exhibit γ such that $\beta \leq \gamma < \mu(j)$ and $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(j))} H'(j)$. By taking $\beta_j = \max(\beta, \mu(i)) < \mu(j)$ we obtain γ_j such that $\beta_j \leq \gamma_j < \mu(j)$ and $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H(j)$. Furthermore, since $\gamma_i < \mu(i) \leq \beta_j \leq \gamma_j$ and $\mathfrak{M}, \gamma_i \models_\nu^{(\mu(i))} \mathsf{G}\,\varphi$, we also have $\mathfrak{M}, \gamma_j \models_\nu^{(\mu(j))} \varphi$ and $\mathfrak{M}, \gamma_j \models_\nu^{(\mu(j))} \mathsf{G}\,\varphi$, hence $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H'(j)$.
  - When $i \sim j$ we have that $\mu(i) = \mu(j)$ and it is a limit ordinal because we are considering positions in a cluster. Consider an arbitrary $\beta < \mu(i)$. There exists γ_i such that $\beta \leq \gamma_i < \mu(i)$ and $\mathfrak{M}, \gamma_i \not\models_\nu^{(\mu(i))} H(i)$. Because $\mu(i)$ is a limit ordinal, $\gamma_i + 1 < \mu(i) = \mu(j)$. Again, there exists γ_j such that $\gamma_i + 1 \leq \gamma_j < \mu(j)$ and $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H(j)$. But, since $\gamma_i < \gamma_j$ we also have that γ_j satisfies φ and $\mathsf{G}\,\varphi$, hence $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H'(j)$.
- The case of rule $(\{\mathsf{G}\vdash\})$ is covered by the second part of the previous argument, by taking $i = j$. Indeed, we have $i \sim i$ when $(\{\mathsf{G}\vdash\})$ applies at position i.
- Consider now an application of rule $(\mathsf{H}\vdash)$ with $\Pi \vdash \Sigma$ at position i and $\mathsf{H}\,\varphi, \Gamma \vdash \Delta$ at j. We have $i \precsim j$, hence $\mu(i) \leq \mu(j)$. Consider an arbitrary $\beta < \mu(i)$. There exists γ_i such that $\beta \leq \gamma_i < \mu(i)$ and $\mathfrak{M}, \gamma_i \not\models_\nu^{(\mu(i))} H(i)$. We claim, as before, that there exists γ_j such that $\gamma_i < \gamma_j < \mu(j)$ and $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H(j)$. Indeed, if $\mu(i) < \mu(j)$ then there exists γ_j with $\mu(i) \leq \gamma_j < \mu(j)$ that falsifies $H(j)$. Otherwise $\mu(i) = \mu(j)$ but then this must be a limit ordinal and, by considering $\gamma_i + 1 < \mu(i) = \mu(j)$ we obtain $\gamma_i < \gamma_j < \mu(j)$ that invalidates $H(j)$. Having $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H(j)$, we also have $\mathfrak{M}, \gamma_j \models_\nu^{(\mu(j))} \mathsf{H}\,\varphi$. Thus γ_i satisfies φ and $\mathsf{H}\,\varphi$, and $\mathfrak{M}, \gamma_i \not\models_\nu^{(\mu(i))} H'(i)$ as needed.
- The case of $(\{\mathsf{H}\vdash\})$ is covered by the previous argument.
- Consider an application of $(\vdash\mathsf{G})$ with $\Gamma \vdash \Delta, \mathsf{G}\,\varphi$ at position i. For any $\beta_i < \mu(i)$ there exists γ_i with $\beta_i \leq \gamma_i < \mu(i)$ such that $\mathfrak{M}, \gamma_i \not\models_\nu^{(\mu(i))} H(i)$, and thus $\mathfrak{M}, \gamma_i \not\models_\nu^{(\mu(i))} \mathsf{G}\,\varphi$. Hence there also exists $\gamma_i' > \gamma_i$ such that $\mathfrak{M}, \gamma_i' \not\models_\nu^{(\theta)} \varphi$ for any θ. Let γ be the least ordinal that is strictly bigger than all such γ_i'. We have that $\mu(i) \leq \gamma$.
  We now distinguish several cases regarding γ. When $C'; H_2$ is not empty let j be the first position of the conclusion hypersequent that is in C'.
  - If $\mu(i) = \gamma$, then $\mu(i)$ must be a limit ordinal (for every $\beta_i < \mu(i)$, we can find $\beta_i < \gamma_i' < \gamma = \mu(i)$). Hence $C \neq \star$ and the third premise H_3' is available. We construct a counter-model $(\mathfrak{M}, \nu, \mu')$ for it by taking

$\mu' = \mu + (k \mapsto \gamma)$, where $k = i + 1$ is the new position in H'_3. Indeed, we have that for any $\beta' < \mu'(k)$ there exists γ' with $\beta' \leq \gamma' < \mu'(k)$ and $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu'(k))} \varphi$ (the inequality can even be made strict). Moreover, $\mathfrak{M}, \gamma' \models_\nu^{(\mu'(k))} (\mathsf{G}\,\varphi)$ by definition of $\gamma = \mu'(k)$: there cannot be any $\lambda \geq \gamma$ such that $\mathfrak{M}, \lambda \not\models_\nu^{(\gamma)} \varphi$.

- If $C'; H_2$ is empty, or $\gamma < \mu(j)$, we conclude by observing that $(\mathfrak{M}, \nu, \mu')$ is a counter-model of one of the first two premises with $\mu' = \mu + (k \mapsto \gamma)$ where k is the position of the new cell in these premises. We check that μ' is monotone, because $\mu(i) < \gamma$, and $\gamma < \mu(j)$ when it is defined. If γ is a successor ordinal, $(\mathfrak{M}, \nu, \mu')$ is a counter-model of the first premise simply because the predecessor of γ invalidates φ and satisfies $(\mathsf{G}\,\varphi)$; both hold by construction. If γ is a limit ordinal we have a counter-model $(\mathfrak{M}, \nu, \mu')$ of the second premise: we do have that for any $\beta' < \mu'(k)$ there exists γ' with $\beta' \leq \gamma' < \mu'(k)$ that invalidates φ, and $(\mathsf{G}\,\varphi)$ is satisfied by construction.
- Otherwise $\mu(j) \leq \gamma$.
  - If $\mu(j) < \gamma$, we obtain a counter-model $(\mathfrak{M}, \nu, \mu)$ of the fourth premise H'_4. We check it for the only position whose sequent has changed between H and H'_4, that is position j. Take any $\beta_j < \mu(j)$. We know that there exists γ_j with $\beta_j \leq \gamma_j < \mu(j)$ such that $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H(j)$. But, since $\gamma_j < \mu(j) < \gamma$, there exists γ' such that $\gamma_j < \gamma' < \gamma$ and $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(j))} \varphi$. Thus $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} \mathsf{G}\,\varphi$, and $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H'_4(j)$.
  - If $\mu(j) = \gamma$ and is a limit ordinal, we also obtain a counter-model $(\mathfrak{M}, \nu, \mu)$ of the fourth premise. This time, for any $\beta_j < \mu(j)$, we know that there exists γ_j with $\beta_j \leq \gamma_j < \mu(j)$ such that $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H(j)$. But, since $\gamma_j < \gamma$ and γ is a limit ordinal, there still exists γ' such that $\gamma_j < \gamma' < \gamma$ and $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(j))} \varphi$. Thus $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} \mathsf{G}\,\varphi$, and $\mathfrak{M}, \gamma_j \not\models_\nu^{(\mu(j))} H'_4(j)$.
  - Finally, if $\mu(j) = \gamma$ and is not a limit ordinal, then the position j is not in a cluster, so the fifth premise is available. We claim that it admits $(\mathfrak{M}, \nu, \mu)$ as a counter-model. Let θ be the predecessor of $\gamma = \theta + 1$, which satisfies $\mathfrak{M}, \theta \not\models_\nu^{(\mu(j))} \varphi$ by definition of γ. Since $(\mathfrak{M}, \nu, \mu)$ is a counter-model of H we also have $\mathfrak{M}, \theta \not\models_\nu^{(\mu(j))} H(j)$. This allows us to conclude, together with the fact that, as before, the new annotation $(\mathsf{G}\,\varphi)$ is satisfied by definition of γ (there cannot be any $\lambda \geq \gamma$ such that $\mathfrak{M}, \lambda \not\models_\nu^{(\gamma)} \varphi$).

– We now consider an application of rule $(\vdash \mathsf{H})$ with $\Gamma \vdash \Delta, \mathsf{H}\,\varphi$ at position i. Let j be the first position of C', if it exists. For any $\beta_i < \mu(i)$ there exists γ_i with $\beta_i \leq \gamma_i < \mu(i)$ that invalidates $H(i)$, thus there exists $\gamma'_i < \gamma_i < \mu(i)$ such that $\mathfrak{M}, \gamma'_i \not\models_\nu^{(\mu(i))} \varphi$. Let γ be the successor of the least ordinal among all such γ'_i. We have $\gamma < \mu(i)$.

- If $H_2; C'$ is empty, or $\mu(j) < \gamma$, then $(\mathfrak{M}, \nu, \mu')$ is a counter-model of the first premise with $\mu' = \mu + (k \mapsto \gamma)$ where k is the new position

in that premise. We do have that the predecessor of γ satisfies $\mathsf{H}\,\varphi$ (by minimality) but not φ (by definition).
- If $\mu(j) = \gamma$ then C' cannot be a cluster, because γ is a successor. In that case $(\mathfrak{M}, \nu, \mu)$ directly yields a counter-model of the third premise.
- Otherwise $\gamma < \mu(j)$ and $(\mathfrak{M}, \nu, \mu)$ is a counter-model of the second premise.

We now consider the case of the annotation rule from Figure 3. Consider an application of $((\mathsf{G}))$ with $(\mathsf{G}\,\varphi), \Gamma \vdash \Delta$ at position i and $\Pi \vdash \Sigma$ at position j, with $i \prec j$. Because of the annotation $(\mathsf{G}\,\varphi)$ we have that, for all $\lambda \geq \mu(i)$, $\mathfrak{M}, \lambda \models_{\nu}^{(\mu(i))} \varphi$. Hence $(\mathfrak{M}, \nu, \mu)$ is a counter-model of the premise.

We finally consider the case of freeze rules (Figure 3):

- Consider an application of $(\downarrow \vdash)$ with $H(i) = \Gamma, \downarrow_r \varphi \vdash \Delta$. If $\uparrow_x \in \Gamma$, then $(\mathfrak{M}, \nu, \mu)$ is also a counter-model of the premise. Else, since $(\mathfrak{M}, \nu, \mu)$ is a counter-model of H, there exists $d_i \in \mathbb{D}$ such that for all $\beta < \mu(i)$ there exists γ such that $\beta \leq \gamma < \mu(i)$, $\delta(\gamma) = d_i$, and $\mathfrak{M}, \gamma \not\models_{\nu}^{(\mu(i))} H(i)$. In particular, $\mathfrak{M}, \gamma \models_{\nu}^{(\mu(i))} \downarrow_r \varphi$, so $\mathfrak{M}, \gamma \models_{\nu[r \mapsto d_i]}^{(\mu(i))} \uparrow_r \wedge\, \varphi$. Let us take $\nu' = \nu[x \mapsto d_i]$. Since x is fresh, it is also the case that $\mathfrak{M}, \gamma \not\models_{\nu'}^{(\mu(i))} H(i)$; and by the second case of Lemma 3, $\mathfrak{M}, \gamma \models_{\nu'}^{(\mu(i))} \uparrow_x \wedge\, [x/r](\varphi)$. Hence, $(\mathfrak{M}, \nu', \mu)$ is a counter-model of the premise.
- The case of $(\vdash \downarrow)$ is similar.
- Consider the application of $(\uparrow \vdash)$ with $H(i) = \uparrow_x, \uparrow_y, \Gamma \vdash \Delta$. Since there exists γ such that $\mathfrak{M}, \gamma \models_{\nu}^{(\mu(i))} \uparrow_x$ and $\mathfrak{M}, \gamma \models_{\nu}^{(\mu(i))} \uparrow_y$, then $\nu(x) = \nu(y) = \delta(\gamma)$. Hence, by the first case of Lemma 3, $(\mathfrak{M}, \nu, \mu)$ is a counter-model of the premise.

A.3 Immediately provable

Lemma 1. *If a hypersequent H satisfies one of these conditions, then H is provable (and we say that H is* immediately provable*).*

(a) There exists a formula φ, and two positions $i \prec j$ of H such that $H(i)$ and $H(j)$ both contain $(\mathsf{G}\,\varphi) \vdash \varphi$.
(b) There exists a formula φ, and two positions $i \prec j$ of H such that $H(i)$ and $H(j)$ both contain $\mathsf{H}\,\varphi \vdash \varphi$.
(c) There exists a formula φ, three positions $i \prec j \prec k$ of H, and three registers $x, y, z \in \mathbb{N}$ such that:
  - *$H(i)$ contains $(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi$.*
  - *$H(j)$ contains $(\mathsf{G}_{\neq y}\,\varphi) \vdash \neg\uparrow_y \supset \varphi$.*
  - *$H(k)$ contains $(\mathsf{G}_{\neq z}\,\varphi) \vdash \neg\uparrow_z \supset \varphi$.*

(d) There exists a formula φ, three positions $i \prec j \prec k$ of H, and three registers $x, y, z \in \mathbb{N}$ such that:
  - *$H(i)$ contains $\mathsf{H}_{\neq x}\,\varphi \vdash \neg\uparrow_x \supset \varphi$.*
  - *$H(j)$ contains $\mathsf{H}_{\neq y}\,\varphi \vdash \neg\uparrow_y \supset \varphi$.*

– $H(k)$ *contains* $\mathsf{H}_{\neq z}\,\varphi \vdash \neg\uparrow_z \supset \varphi$.

Proof. For every case, we show how to prove H.

(a) Such a hypersequent can be proved as follows:

$$\dfrac{\dfrac{}{H_1\,[\Gamma, (\mathsf{G}\,\varphi) \vdash \varphi, \Delta]\,;\, H_2\,[\Gamma', (\mathsf{G}\,\varphi), \mathsf{G}\,\varphi, \varphi \vdash \varphi, \Delta']}\,(\mathsf{ax})}{H_1\,[\Gamma, (\mathsf{G}\,\varphi) \vdash \varphi, \Delta]\,;\, H_2\,[\Gamma', (\mathsf{G}\,\varphi) \vdash \varphi, \Delta']}\,((\mathsf{G}))$$

(b) This case is similar, roles of i and j being reverted, and using $(\vdash\mathsf{H})$ instead of $((\mathsf{G}))$:

$$\dfrac{\dfrac{}{H_1\,[\Gamma, \mathsf{H}\,\varphi, \varphi \vdash \varphi, \Delta]\,;\, H_2\,[\Gamma', \mathsf{H}\,\varphi \vdash \varphi, \Delta']}\,(\mathsf{ax})}{H_1\,[\Gamma, \mathsf{H}\,\varphi \vdash \varphi, \Delta]\,;\, H_2\,[\Gamma', \mathsf{H}\,\varphi \vdash \varphi, \Delta']}\,(\vdash\mathsf{H})$$

(c) Let us first establish the following: if there is a formula φ and a register r such that a sequent of H contains $\neg\uparrow_r \supset \varphi \vdash \varphi$, then we can make $\uparrow_r$ appears on the left-hand side of this sequent:

$$\dfrac{\dfrac{}{H\,[\Gamma, \varphi \vdash \varphi, \Delta]}\,(\mathsf{ax}) \qquad \dfrac{H\,[\Gamma, \neg\uparrow_r \supset \varphi, \uparrow_r \vdash \varphi, \Delta]}{H\,[\Gamma, \neg\uparrow_r \supset \varphi \vdash \neg\uparrow_r, \varphi, \Delta]}}{H\,[\Gamma, \neg\uparrow_r \supset \varphi \vdash \varphi, \Delta]}\,(\supset\vdash)$$

We now sketch in Figure 4 how to prove a hypersequent satisfying condition (c). As the weakening rules are admissible, we omit other formulæ that could appear at positions i, j or k, in order to make the figure more readable. The omitted steps correspond to the ones described above, for registers x and y. The last hypersequent satisfies condition (a), hence it is provable.

(d) This case is similar to (c), with the roles of positions i and k reverted (as well as roles of registers x and z), and using $(\vdash\mathsf{H})$ instead of $((\mathsf{G}))$; and reducing to an instance of (b).

$$\dfrac{\dfrac{\dfrac{\dfrac{[x/y](H_1)\,[(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi]\,;\, [x/y](H_2)\,[(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi]\,;\, [x/y](H_3)\,[(\mathsf{G}_{\neq z}\,\varphi), \uparrow_x, \neg\uparrow_z \vdash \varphi]}{H_1\,[(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi]\,;\, H_2\,[(\mathsf{G}_{\neq y}\,\varphi) \vdash \neg\uparrow_y \supset \varphi]\,;\, H_3\,[(\mathsf{G}_{\neq z}\,\varphi), \uparrow_x, \uparrow_y, \neg\uparrow_z \vdash \varphi]}\,(\uparrow\vdash)}{\vdots}}{H_1\,[(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi]\,;\, H_2\,[(\mathsf{G}_{\neq y}\,\varphi) \vdash \neg\uparrow_y \supset \varphi]\,;\, H_3\,[(\mathsf{G}_{\neq x}\,\varphi), (\mathsf{G}_{\neq y}\,\varphi), (\mathsf{G}_{\neq z}\,\varphi), \neg\uparrow_x \supset \varphi, \neg\uparrow_y \supset \varphi, \neg\uparrow_z \vdash \varphi]}\,(\vdash\supset)}{H_1\,[(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi]\,;\, H_2\,[(\mathsf{G}_{\neq y}\,\varphi) \vdash \neg\uparrow_y \supset \varphi]\,;\, H_3\,[(\mathsf{G}_{\neq x}\,\varphi), (\mathsf{G}_{\neq y}\,\varphi), (\mathsf{G}_{\neq z}\,\varphi), \neg\uparrow_x \supset \varphi, \neg\uparrow_y \supset \varphi \vdash \neg\uparrow_z \supset \varphi]}\,((\mathsf{G}))}{H_1\,[(\mathsf{G}_{\neq x}\,\varphi) \vdash \neg\uparrow_x \supset \varphi]\,;\, H_2\,[(\mathsf{G}_{\neq y}\,\varphi) \vdash \neg\uparrow_y \supset \varphi]\,;\, H_3\,[(\mathsf{G}_{\neq x}\,\varphi), (\mathsf{G}_{\neq y}\,\varphi), \neg\uparrow_x \supset \varphi \vdash \neg\uparrow_z \supset \varphi]}\,((\mathsf{G}))$$

Fig. 4. Proof tree sketch for a hypersequent satisfying condition (c).

A.4 Failure Hypersequents

Proposition 3. *Any failure hypersequent H has a counter-model.*

Proof. Let $\alpha = o(H)$. We define $\mu : \mathrm{dom}(H) \to \alpha + 1 \setminus \{0\}$ as follows:

$$\begin{array}{ll} \mu(i) = m & \text{if } i \text{ is the } m\text{-th cell of } H \\ & \text{and appears before its first cluster;} \\ \mu(i) = \omega \cdot k & \text{if } i \text{ belongs to the } k\text{-th cluster of } H; \\ \mu(i) = \omega \cdot k + m & \text{if } i \text{ is the } m\text{-th cell appearing between} \\ & \text{the } k\text{-th and the next cluster (if any).} \end{array}$$

Now let $\mathrm{pos} : \alpha \to \mathrm{dom}(H)$ be a function such that:

(a) $\forall \beta < \beta' < \alpha,\ \mathrm{pos}(\beta) \precsim \mathrm{pos}(\beta')$
(b) $\forall \beta < \alpha,\ \forall i \in \mathrm{dom}(H),\ \beta < \mu(i) \Leftrightarrow (\mathrm{pos}(\beta) \precsim i \text{ or } \mathrm{pos}(\beta) = i)$
(c) $\forall \beta < \alpha,\ \forall i \in \mathrm{dom}(H),\ \mathrm{pos}(\beta) \precsim i \Rightarrow \exists \beta < \gamma < \mu(i),\ i = \mathrm{pos}(\gamma)$

There always exists one such function. Its choice is quite constrained due to the definitions of α and μ. Positions i that are not in a cluster will be such that $i = \mathrm{pos}(\beta)$ for a single β, typically the predecessor of $\mu(i)$. A position i appearing in a cluster must correspond to an infinite sequence of ordinals of limit $\mu(i)$, so that for all $i \sim j$ and β, if $\mathrm{pos}(\beta) = i$ then there exists γ with $\beta < \gamma < \mu(i) = \mu(j)$ such that $\mathrm{pos}(\gamma) = j$; informally, this ensures that positions i and j inside a cluster are 'infinitely interleaved' within $\mu(i) = \mu(j)$.

We now define the data assignment δ of α. Since the rule $(\uparrow \vdash)$ cannot be applied on H, each position of H must have at most one atomic formula of the form $\uparrow_r$ on its left-hand side. For each position i of H, we chose a datum $d_i \in \mathbb{D}$ with the following constraints:

- If $H(i)$ and $H(j)$ have the same atomic $\uparrow_r$ on their left-hand side, then $d_i = d_j$;
- Else, $d_i \neq d_j$.

We can now define δ by $\delta(\beta) = d_{\mathrm{pos}(\beta)}$, for all $\beta < \alpha$. From this, we fix a fresh datum $d_\perp$ different from all the d_i, and we can define a register valuation ν defined for every free register that appears in H by:

- $\nu(r) = d_i$ if $\uparrow_r$ appears on the left-hand side of $H(i)$,
- $\nu(r) = d_\perp$ otherwise.

We finally define a valuation $V : \Phi \to \wp(\alpha)$ by $V(p) = \{\beta < \alpha \mid \exists \Gamma, \Delta\ .\ H(\mathrm{pos}(\beta)) = (p, \Gamma \vdash \Delta)\}$ and let $\mathfrak{M} = ((\alpha, \delta), V)$. We now claim that $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} H(\mathrm{pos}(\gamma))$ for all $\gamma < \alpha$: we prove by induction on ψ that, if ψ appears in the left-hand (resp. right-hand) side of the turnstile in $H(\mathrm{pos}(\gamma))$, then $\mathfrak{M}, \gamma \models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \psi$ (resp. $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \psi$).

- If ψ is an atom $p \in V$ the results follow by definition of V, and because (ax) does not apply to H. The propositional cases are obtained by induction hypothesis, because the corresponding rules of Figure 1 have already been applied.
- If ψ is an atomic formula $\uparrow_r$, then:

- If ψ appears on the left-hand side of the turnstile, the results follows by definition of ν.
- If ψ appears on the right-hand side of the turnstile at position i, then either ψ also appears on the left-hand side of some position j, and $i \neq j$ because (ax) does not apply, so $\nu(r) = d_j \neq d_i$; or ψ never appears on the left-hand side of some sequent of H and $\nu(r) = d_\perp \neq d_i$. Either way, $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(i))} \uparrow_r$ for γ such that $\mathrm{pos}(\gamma) = i$.

– Because $(\downarrow\vdash)$ and $(\vdash\downarrow)$ do not apply on H, there are no formula of the form $\downarrow_r\psi'$ anywhere in H for which $(\downarrow\vdash)$ or $(\vdash\downarrow)$ has not been applied (such rules could also be prevented by having two formulæ $\uparrow_x$ and $\uparrow_y$ on the left-hand side of a sequent, but this cannot happen here since $(\uparrow\vdash)$ does not apply). This means that every such $\downarrow_r\psi'$ appears along with $[x/r](\psi')$ on the same side of the turnstile, and $\uparrow_x$ on the left-hand side of the turnstile, for some x. Let us assume that $\downarrow_r\psi'$ and $[x/r](\psi')$ appear on the right-hand side of $H(\mathrm{pos}(\gamma))$ (the other case is similar). By induction hypothesis, $\mathfrak{M}, \gamma \models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \uparrow_x$ and $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} [x/r](\psi')$, hence $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \downarrow_r\psi'$.
– Assume that $\psi = (\mathsf{G}\,\varphi)$ appears on the left-hand side of the turnstile in $H(\mathrm{pos}(\gamma))$ (an annotation cannot appear on the right-hand side). Then, because $((\mathsf{G}))$ does not apply, φ appears on the left-hand side of $H(i)$ for any i such that $\mathrm{pos}(\gamma) \prec i$, so $\mathfrak{M}, \gamma' \models_\nu^{(\theta)} \varphi$ for every $\gamma' \geq \mu(\mathrm{pos}(\gamma))$ and for any θ, hence $\mathfrak{M}, \gamma \models_\nu^{(\mu(\mathrm{pos}(\gamma)))} (\mathsf{G}\,\varphi)$.
– The cases of modal formulæ on the left-hand side are similar, we only detail that of H. If $\psi = \mathsf{H}\,\varphi$ occurs on the left-hand side of $H(\mathrm{pos}(\gamma))$ then by $(\mathsf{H}\vdash)$ and $(\{\mathsf{H}\vdash\})$, the formula φ must occur on the left-hand side of any $H(i)$ with $i \precsim \mathrm{pos}(\gamma)$. Moreover, for all $\gamma' < \gamma$, we have $\mathrm{pos}(\gamma') \precsim \mathrm{pos}(\gamma)$ by **(a)**, so $\mathfrak{M}, \gamma' \models_\nu^{(\mu(\mathrm{pos}(\gamma')))} \varphi$, and thus $\mathfrak{M}, \gamma \models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \psi$.
– Assume that $\psi = \mathsf{H}\,\varphi$ occurs on the right of $H(\mathrm{pos}(\gamma))$. We prove by a sub-induction on $\mathrm{pos}(\gamma)$ that $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \mathsf{H}\,\varphi$. Since $(\vdash\mathsf{H})$ does not apply, and since the first premise necessarily differs from the conclusion, it must be that there is a cell C' preceding the cell that contains $\mathrm{pos}(\gamma)$, and that the last two premises (if available) would coincide with H. Let i be the first position in C'. Take an arbitrary $\lambda < \mu(i)$ such that $\mathrm{pos}(\lambda) = i$ (such a λ always exists, thanks to **(b)** and **(c)** instantiated with $\beta = 0$). Since $i \prec \mathrm{pos}(\gamma)$ it must be that $\lambda < \gamma$. As noted above, we have either that $\mathsf{H}\,\varphi$ belongs to the right-hand side of $H(i)$, or that φ belongs to its left-hand side. In the first case, we obtain $\mathfrak{M}, \lambda \not\models_\nu^{(\mu(\mathrm{pos}(\lambda)))} \mathsf{H}\,\varphi$ by induction hypothesis on $i < \mathrm{pos}(\gamma)$. In the second case we directly have $\mathfrak{M}, \lambda \not\models_\nu^{(\mu(\mathrm{pos}(\lambda)))} \varphi$. We conclude either way that $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \mathsf{H}\,\varphi$.
– Assume finally that $\psi = \mathsf{G}\,\varphi$ occurs on the right-hand side of $H(\mathrm{pos}(\gamma))$. Let us first assume that there does not exist any $i \succ \mathrm{pos}(\gamma)$ such that $\mathsf{G}\,\varphi$ appears on the right-hand side of $H(i)$.
  - If $(\vdash\mathsf{G})$ does not apply on ψ because of case (i) of condition (a), it cannot be because of the fourth premise by the previous assumption, so

an annotation $(\mathsf{G}\,\varphi)$ must appear at some position i in H on the left-hand side of the turnstile, along with φ on its right-hand side. By rule $((\mathsf{G}))$ we must have $\mathrm{pos}(\gamma) \precsim i$ (or else (ax) could by applied). By **(c)**, there exists $\gamma' > \gamma$ such that $i = \mathrm{pos}(\gamma')$. We then have $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(\mathrm{pos}(\gamma')))} \varphi$, thus $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \mathsf{G}\,\varphi$.

- If $(\vdash\mathsf{G})$ does not apply on ψ because of case (ii) of condition (a), there exists $i \sim \mathrm{pos}(\gamma)$ such that $H(i)$ contains $(\mathsf{G}\,\varphi) \vdash \varphi$, and we can conclude the same way we did above for case (i).
- If $(\vdash\mathsf{G})$ does not apply on ψ because of case (iii) of condition (a), then φ is of the form $\neg\uparrow_x \supset \varphi'$, and there exists $i \sim \mathrm{pos}(\gamma)$ and $y \neq x$ such that $H(i)$ contains $(\mathsf{G}_{\neq y}\,\varphi') \vdash \neg\uparrow_y \supset \varphi'$, and does not contain $\uparrow_x$ on its left-hand side. By **(c)**, there exists $\gamma' > \gamma$ such that $i = \mathrm{pos}(\gamma')$; and by definition of ν and δ, $\nu(x) \neq \delta(\gamma')$ (because $\uparrow_x$ does not appear on the left-hand side of $H(\mathrm{pos}(\gamma'))$); hence $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(i))} \uparrow_x$. Moreover, $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(i))} \neg\uparrow_y \supset \varphi'$, so in particular $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(i))} \varphi'$. Thus, in the end, $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(i))} \neg\uparrow_x \supset \varphi'$, and we have $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \mathsf{G}\,(\neg\uparrow_x \supset \varphi')$.

Now, if there exists a position $i \succ \mathrm{pos}(\gamma)$ such that $\mathsf{G}\,\varphi$ appears on the right-hand side of $H(i)$, since we just proved that $\mathfrak{M}, \gamma' \not\models_\nu^{(\mu(i))} \mathsf{G}\,\varphi$ for any γ' with $\mathrm{pos}(\gamma') = i$ (in particular, $\gamma' > \gamma$), then we indeed have $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(\mathrm{pos}(\gamma)))} \mathsf{G}\,\varphi$.

We can check that $H \hookrightarrow_\mu \alpha$: the conditions of an embedding are met by construction.

Finally, $(\mathfrak{M}, \nu, \mu)$ is a counter-model of H. Indeed, for all $i \in \mathrm{dom}(H)$ and $\beta < \mu(i)$ there exists γ with $\beta \leq \gamma < \mu(i)$ such that $\mathrm{pos}(\gamma) = i$, and hence $\mathfrak{M}, \gamma \not\models_\nu^{(\mu(i))} H(i)$: if $\mathrm{pos}(\beta) = i$, we can take $\gamma = \beta$, else **(b)** enforces $\mathrm{pos}(\beta) \precsim i$, and **(c)** provides one such γ.

A.5 Small branch property

Lemma 2. *For any partial proof of a hypersequent H, any branch of the proof is of length at most* $2|H|(4|H| + \mathsf{len}(H))((4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1)$.

Proof. Let H be a hypersequent, $\mathcal{P}$ a partial proof of it, and B a branch of $\mathcal{P}$. We note Φ_H the set of subformulæ of H. Remark that all the formulæ that appear in B belongs to Φ_H, up to the renaming of some registers that appear in B. We have to be careful about the following: each creation of a new position along B could lead to the creation of a new register later in B, which in turn could create a new renamed copy of some formula of Φ_H, which then could lead to the creation of another position. We must make sure that such a process cannot go ad infinitum. Let us first establish that the number of free registers in hypersequents of B is bounded by $4|H| + \mathsf{len}(H)$. Because we always unify registers with $(\uparrow\vdash)$ as soon as possible in B (condition (b) of being a partial proof), and because the only way to introduce new registers is via rules $(\downarrow\vdash)$ and $(\vdash\downarrow)$ when no thaw appear on the left-hand side, we always have less free registers than positions. We have at most $\mathsf{len}(H)$ positions initially; and we must

now bound the creations of positions that can effectively lead to the creation of a new register later in B (new positions can only be created by rules ($\vdash \mathsf{H}$) and ($\vdash \mathsf{G}$)):

- For any $\mathsf{H}\,\varphi$ among the subformulæ of B that do not contain a free register (so among $|H|$ formulæ), a new position can only be created once without creating an immediately provable hypersequent.
- For any $\mathsf{G}\,\varphi$ among the subformulæ of B that do not contain a free register, a new position can only be created once in the same cluster (because of case (ii) of condition (a)). A second position cannot be created elsewhere either without creating an immediately provable hypersequent.
- For any $\downarrow_r \mathsf{H}_{\neq r}\,\varphi \in \Phi_H$, many formulæ of the form $\mathsf{H}_{\neq x}\,\varphi$ could appear along B (as many as free registers), and they could all lead to the creation of a new position. However, only 2 such positions can be created along B (each time for a different x) without creating an immediately provable hypersequent.
- Similarly, for any $\downarrow_r \mathsf{G}_{\neq r}\,\varphi \in \Phi_H$, many formulæ of the form $\mathsf{G}_{\neq x}\,\varphi$ could appear along B. Let us prove that a new position can only be created at most 4 times by such formulæ (each time for a different x) without creating an immediately provable hypersequent (the worst case being two different clusters, both containing two such sequents). First of all, we cannot create 3 such positions with $i \prec j \prec k$ without creating an immediately provable hypersequent, so the worst case indeed involve at most two clusters. Moreover, we cannot create more than two such positions in the same cluster. Let us look at the evolution of such a cluster along B. A first position i is created for a formula $\mathsf{G}_{\neq x}\,\varphi$, and a second position $j \sim i$ could be created later for a formula $\mathsf{G}_{\neq y}\,\varphi$ (with $y \neq x$) only if the ith position of the current hypersequent has $\uparrow_y$ on its left-hand side. But then, a third creation of such a position (for a formula $\mathsf{G}_{\neq z}\,\varphi$, with $z \neq y$) will be prevented since j is an instance of (iii) (the jth position of the current hypersequent cannot contain $\uparrow_z$ on its left-hand side, since it already contains $\uparrow_y$ and we always unify registers as soon as possible).
- One more (overall) position could be created, leading to an immediately provable hypersequent. In such a case, a new register will not be created since the hypersequent as to be proved immediately as sketched in the proof of Lemma 1.
- Because of condition (b), whenever a new cell is created by a formula $\mathsf{H}_{=x}\,\varphi$ or $\mathsf{G}_{=x}\,\varphi$, $\uparrow_x$ will appear on its left-hand side from the next step in B. Hence, such new cells will not lead to new registers, and the number of registers is bounded by $4|H| + \mathsf{len}(H)$. Moreover, at most one such cell can be created for every φ (among $|H|$ formulæ), and every register appearing in B, i.e. $(4|H| + \mathsf{len}(H))|H|$ in total.

This also proves that the number of positions of hypersequents in B is at most $(4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1$. Now, any other rule application adds some subformulæ among $(4|H| + \mathsf{len}(H))|H|$ to the left or to the right of the turnstile at a position among $(4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1$, hence with $2(4|H| +$

$\mathsf{len}(H))|H|((4|H| + \mathsf{len}(H))|H| + \mathsf{len}(H) + 1)$ choices. Thus B is of length at most $2|H|(4|H|+\mathsf{len}(H))((4|H|+\mathsf{len}(H))|H|+\mathsf{len}(H)+1)$.

B First-Order Logic with Two Variables

In this section, we show that $\mathbf{K_t^d L_\ell.3}$ is exactly as expressive as the two-variable fragment of first-order logic over data ordinals from [6].

B.1 Syntax and Semantics

We consider first-order formulæ with two variables x and y over the signature $(=, \sim, <, (p)_{p\in\Phi})$ where $=$, $<$ and $\sim$ are binary relational symbols and each p is a unary relational symbol:

$$\psi ::= z = z' \mid z < z' \mid z \sim z' \mid p(z) \mid \bot \mid \psi \supset \psi \mid \forall z.\psi \qquad \text{(first-order formulæ)}$$

where z, z' range over $\{x, y\}$ and p over Φ. We call this logic $\mathrm{FO}^2(\sim, <)$.

We interpret our formulæ over structures $\mathfrak{M} = ((\alpha, \delta), V)$ where $=$ is interpreted as the equality over α, $<$ as the canonical strict total ordering of α, $\sim$ as the equality with respect to δ, and each p as $V(p)$ for the valuation $V : \Phi \to 2^W$.

That is, we say that $\mathfrak{M}$ satisfies ψ under an assignment $\sigma : \{x, y\} \to \alpha$, written $\mathfrak{M}, \sigma \models \psi$, in the following inductive cases:

$$\begin{array}{ll}
\mathfrak{M}, \sigma \not\models \bot & \\
\mathfrak{M}, \sigma \models z = z' & \text{if } \sigma(z) = \sigma(z') \\
\mathfrak{M}, \sigma \models z < z' & \text{if } \sigma(z) < \sigma(z') \\
\mathfrak{M}, \sigma \models z \sim z' & \text{if } \delta(\sigma(z)) = \delta(\sigma(z')) \\
\mathfrak{M}, \sigma \models p(z) & \text{if } \sigma(z) \in V(p) \\
\mathfrak{M}, \sigma \models \psi \supset \psi' & \text{if } \mathfrak{M}, \sigma \models \psi \text{ implies } \mathfrak{M}, \sigma \models \psi' \\
\mathfrak{M}, \sigma \models \exists z.\psi & \text{if } \exists w \in W,\ \mathfrak{M}, \sigma[w/z] \models \psi
\end{array}$$

where $\sigma[w/z]$ is the updated assignment mapping z to w and the remaining variable $z' \in \{x, y\} \setminus \{z\}$ to $\sigma(z')$.

B.2 Equivalence with $\mathbf{K_t^d L_\ell.3}$

Given an $\mathrm{FO}^2(\sim, <)$ formula $\psi(z)$ with one free variable z, we show how to construct a $\mathbf{K_t^d L_\ell.3}$ formula φ such that, for all data ordinals $\mathfrak{M} = ((\alpha, \delta), V)$, $\mathfrak{M}, [w/z] \models \psi$ if and only if $\mathfrak{M}, w \models \varphi$, where $[w/z]$ is the variable assignment mapping z to w.

Theorem 2. *Every* $\mathrm{FO}^2(\sim, <)$ *formula* $\varphi(x)$ *can be converted to an equivalent* $\mathbf{K_t^d L_\ell.3}$ *formula* φ' *with* $|\varphi'| \in 2^{poly(|\varphi|)}$.

Proof. The proof from [16] consist first in putting $\varphi(x)$ in Scott normal form, and then constructing its translation by structural induction. After multiples steps—involving an exponential blow-up—, they obtain the following formula equivalent to $\varphi(x)$:

$$\bigvee_{\bar{\gamma}\in\{\top,\bot\}^s}\left(\bigwedge_{i<s}(\xi_i(x)\leftrightarrow\gamma_i)\wedge\bigvee_{\tau\in\Upsilon}\exists y\,.\,(\tau(x,y)\wedge\beta^\tau(y,\bar{\gamma}))\right)$$

where each of the ξ_i have a *quantifier depth* strictly lower than φ (hence can be translated by induction hypothesis), and where $\tau(x,y)$ is what they call an *order type*, and expresses which relations hold between x and y (in [16], $\tau(x,y)$ expresses which order relation holds between x and y, but it can now also express which relation holds between the data of x and y). By β^τ, we denote the formula β where every atomic order formula have been replaced by either $\top$ or $\bot$, according to τ.

At this point, assuming by induction hypothesis that ψ' is the translation of some formula $\psi(x)$, we need to provide translation to a formula of the form $\exists y(\tau(x,y)\wedge\psi(y))$. We consider 9 mutually exclusive cases of such $\tau(x,y)$ in the following table, where $\tau\langle\psi\rangle$ denotes the translation of $\exists y(\tau(x,y)\wedge\psi(y))$:

$\tau(x,y)$	$\tau\langle\psi\rangle$
$x=y$	ψ'
$x\sim y$	$\psi'\vee\downarrow_r\mathsf{F}_{=r}\,\psi'\vee\downarrow_r\mathsf{P}_{=r}\,\psi'$
$\neg(x\sim y)$	$\downarrow_r\mathsf{F}_{\neq r}\,\psi'\vee\downarrow_r\mathsf{P}_{\neq r}\,\psi'$
$x<y$	$\mathsf{F}\,\psi'$
$x<y\wedge x\sim y$	$\downarrow_r\mathsf{F}_{=r}\,\psi'$
$x<y\wedge\neg(x\sim y)$	$\downarrow_r\mathsf{F}_{\neq r}\,\psi'$
$y<x$	$\mathsf{P}\,\psi'$
$y<x\wedge x\sim y$	$\downarrow_r\mathsf{P}_{=r}\,\psi'$
$y<x\wedge\neg(x\sim y)$	$\downarrow_r\mathsf{P}_{\neq r}\,\psi'$

Any other $\tau(x,y)$ can be reduced to either these cases, or trivially to $\bot$ or $\top$.

Conversely, $\mathbf{K_t^dL_\ell.3}$ formulæ can be easily translated into $\mathrm{FO}^2(\sim,<)$ formulæ. Hence, $\mathbf{K_t^dL_\ell.3}$ and $\mathrm{FO}^2(\sim,<)$ are equally expressive.

Thus, Theorem 4 yield an optimal NEXP upper bound for the satisfiability of $\mathrm{FO}^2(\sim,<)$.

C Other Data Logics

On the following figure, $\mathbf{K_t^dL_\ell.3}$ is contained in the Logic of Repeating Values with Past navigation (PLRV) [13], which is evaluated on linear structures with multiple attributes (when working with only one attribute, these structures coincide with data words). The logic is able to navigate to a future position with an attribute equal (resp. different) to an attribute of the current position. Demri

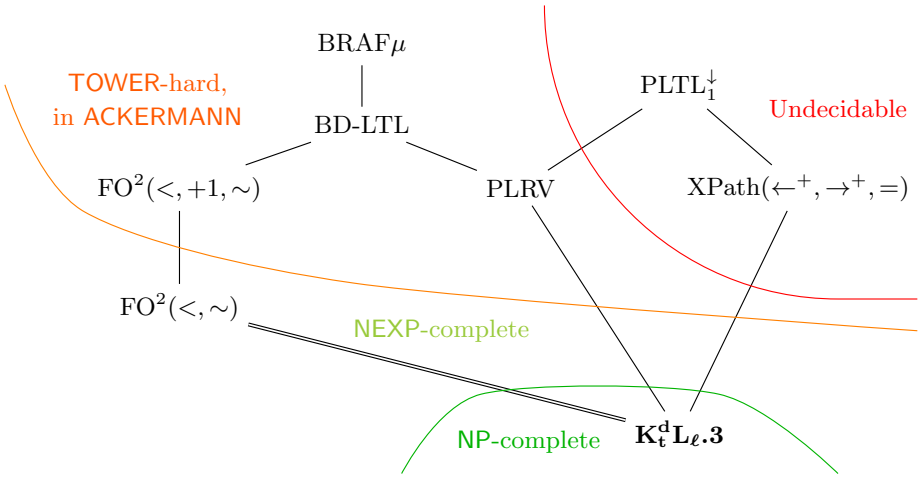

Fig. 5. Inclusions and complexities of some logics on data words.

et al. [13] shows that the satisfiability problem of LRV is equivalent to the reachability problem in VASS, which is currently known to be TOWER-hard [11] and in ACKERMANN [32]. Our fragment is also contained in the fragment of XPath on words XPath$(\leftarrow^+,\rightarrow^+,=)$ [17], which features data joins and can navigate along the `following-sibling` and `preceding-sibling` axes, which is undecidable [17]. They are both contained in Freeze LTL with Past (PLTL$_1^{\downarrow}$) [14], which is also undecidable.

$\mathbf{K_t^d L_\ell.3}$ is equally expressive as FO$^2(<,\sim)$ [6], but is only NP-complete whereas this two variable fragment is NEXP-complete. When enriched with the *next* operator $+1$, the satisfiability problem for this logic also becomes equivalent to the reachability problem in VASS [6].

PLRV and FO$^2(<,+1,\sim)$ are also contained in BD-LTL from [26], which is itself contained in the bounded-reversal alternation-free fragment of the μ-calculus from [10]—denoted by BRAFμ on Figure 5.