Temporal and Spatial OBDA with Many-Dimensional Halpern-Shoham Logic

R. Kontchakov,[1] **L. Pandolfo,**[2] **L. Pulina,**[3] **V. Ryzhikov**[4] and **M. Zakharyaschev**[1]
[1]Department of Computer Science, Birkbeck, University of London, UK
[2]DIBRIS Department, University of Genoa, Italy
[3]POLCOMING Department, University of Sassari, Italy
[4]Faculty of Computer Science, Free University of Bozen-Bolzano, Italy

Abstract

We design an extension datalog$\mathcal{HS}_n^\square$ of datalog with hyperrectangle generalisations of Halpern-Shoham's modal operators on intervals and a corresponding query language. We prove that, over n-dimensional spaces comprised of $\mathbb{Z}$ and $\mathbb{R}$, finding certain answers to datalog$\mathcal{HS}_n^\square$ queries can be reduced to standard datalog query answering. We present experimental results showing the expressivity and efficiency of datalog$\mathcal{HS}_n^\square$ on historical data.

1 Introduction

The aim of ontology-based data access (OBDA) is to streamline the process of gathering information from various data sources by providing the user with a convenient high-level view of the data and thereby making IT middlemen redundant [Poggi et al., 2008]. Typical application scenarios for OBDA include querying databases with multiple tables and complex schemas [Kharlamov et al., 2015], in particular, with sensor and event data [Kharlamov et al., 2014]. Incidentally, these and other real-world use cases reveal that standard OBDA ontology and query languages, designed by the description logic, datalog and Semantic Web communities, do not provide any natural means to deal with numerical information such as historical, geospatial or sensor data.

There have been a number of attempts to add temporal or spatial constructs to conjunctive or SPARQL queries while keeping intact the standard OBDA ontology language OWL 2 QL. In the temporal case, this approach might seem to be justified because ontology axioms are often assumed to hold at all times; see, e.g., [Gutiérrez-Basulto and Klarman, 2012; Baader, Borgwardt, and Lippmann, 2013; Borgwardt, Lippmann, and Thost, 2013; Özcep et al., 2013; Klarman and Meyer, 2014; Özçep and Möller, 2014]. In the spatial ontology languages designed for OBDA [Özçep and Möller, 2012; Eiter, Krennwallner, and Schneider, 2013], domain individuals are associated with their spatial extensions via special functions [Lutz and Milicic, 2007]. However, axioms saying that those extensions stand in certain spatial relations easily ruin the main feature of OWL 2 QL—first-order rewritability of conjunctive queries (CQs) [Özçep and Möller, 2012].

The restriction of temporal and spatial constructs only to queries places the burden of using them on the user rather than on the ontology engineer, and thereby contradicts the spirit of the OBDA paradigm. First temporal extensions of OWL 2 QL and other logics in the *DL-Lite* family that enjoy FO-rewritability were proposed by Artale et al. [2013; 2015b]. These formalisms feature *LTL* temporal operators over $\mathbb{Z}$ on concepts and roles. One potential issue with using them in practice could be the complexity of FO-rewritings caused by the point-based temporal operators.

In this paper, we suggest a different approach to the design of ontology languages for OBDA with numerical data. We model such data by closed *intervals* $[x, y]$ of appropriate linear orders, say $(\mathbb{Z}, \leq)$ or $(\mathbb{R}, <)$, which can represent periods of time (as in temporal databases), temperature or altitude ranges, the length and width of rectangular (approximations of) spatial regions, etc. Thus, our 'spatio-temporal' entities are hyperrectangles in the n-dimensional Cartesian products $\mathfrak{X} = \prod_{\ell=1}^n \mathfrak{X}_\ell$ of linear orders $\mathfrak{X}_\ell = (X_\ell, \lhd_\ell)$. Relation algebras over such entities are well-known in temporal and spatial knowledge representation: Allen's [1983] interval algebra, the rectangle/block algebra RA [Balbiani, Condotta, and del Cerro, 2002], cardinal direction calculus [Goyal and Egenhofer, 2001]; see also [Navarrete et al., 2013; Cohn et al., 2014; Zhang and Renz, 2014] and references therein.

An expressive and elegant formalism for reasoning about intervals was suggested by Halpern and Shoham [1991] who used 'modal' operators of the form $[\mathsf{R}]$ and $\langle \mathsf{R} \rangle$, where R is one of Allen's interval relations. The full Boolean modal logic $\mathcal{HS}$ with these operators is undecidable over any non-trivial linear orders [Halpern and Shoham, 1991; Gabbay et al., 2003; Bresolin et al., 2014]. However, it has recently been shown that the *Horn fragment* of $\mathcal{HS}$ with the operators $[\mathsf{R}]$ over $(\mathbb{Z}, \leq)$ and the *reflexive* semantics of the interval relations turns out to be tractable [Artale et al., 2015a].

Here we define an ontology language datalog$\mathcal{HS}_n^\square$ whose programs consist of standard datalog rules with body atoms possibly prefixed by operators $\langle \mathsf{R} \rangle_\ell$ or $[\mathsf{R}]_\ell$ and head atoms by $[\mathsf{R}]_\ell$ over $\mathfrak{X}_\ell = (\mathbb{Z}, \leq)$ or $\mathfrak{X}_\ell = (\mathbb{R}, <)$. The atoms represent standard database relations that can hold at some hyperrectangles of $\mathfrak{X}$, and the rules are assumed to hold globally at all of them. Thus, a typical datalog$\mathcal{HS}_n^\square$ rule says that $P(\boldsymbol{t})$ holds at a hyperrectangle $\boldsymbol{\iota}$ if $Q_1(\boldsymbol{t}_1), \dots, Q_k(\boldsymbol{t}_k)$ hold at some (or all) hyperrectangles $\boldsymbol{\iota}_i$ such that $\mathsf{R}_i(\boldsymbol{\iota}, \boldsymbol{\iota}_i)$, where R_i is one of the basic relations of the n-dimensional block algebra RA. Sets of datalog$\mathcal{HS}_n^\square$ rules can define complex spatial relations such as, for example, P holds at $\boldsymbol{\iota}$ if the Q_i hold at some $\boldsymbol{\iota}_i$ with $\boldsymbol{\iota} = \bigcap \boldsymbol{\iota}_i$ (or with $\boldsymbol{\iota}$ being the smallest hyperrectangle containing all the $\boldsymbol{\iota}_i$). As a query language we suggest conjunctive queries (CQs) with atoms of the form $P(\boldsymbol{t})@\boldsymbol{\tau}$ and $\mathsf{R}_\ell(\tau, \tau')$, where $\boldsymbol{t}$ and $\boldsymbol{\tau}$ are tuples of individual and interval terms, respectively, and R_ℓ an interval relation over $\mathfrak{X}_\ell$.

The main result of the paper is that, given a CQ $\boldsymbol{q}(\boldsymbol{v}, \boldsymbol{\chi})$ and a datalog$\mathcal{HS}_n^\square$ program Π, one can construct in linear time a standard datalog program $\Pi^\dagger$ with a goal $G(\boldsymbol{v}, \boldsymbol{\chi})$ such that, for any data instance $\mathcal{D}$, a tuple $(\boldsymbol{c}, \boldsymbol{\delta})$ is a certain answer to $\boldsymbol{q}(\boldsymbol{v}, \boldsymbol{\chi})$ and Π over $\mathcal{D}$ iff $\Pi^\dagger, \mathcal{D} \models G(\boldsymbol{c}, \boldsymbol{\delta})$. To obtain this datalog rewritability result, we investigate the structure of the minimal model of $(\Pi, \mathcal{D})$ based on the space $\boldsymbol{\mathfrak{X}}$ and show that one can always construct a (non-standard) finite Kripke model of $(\Pi, \mathcal{D})$ that has polynomially-many worlds in the size of $\mathcal{D}$ and gives all certain answers to any CQ. On the other hand, we show that query evaluation for *propositional* datalog$\mathcal{HS}_1^\square$ programs is P-hard for data complexity. Note that since satisfiability of propositional datalog$\mathcal{HS}_n^\square$ programs is decidable in polynomial time, we obtain a tractable fragment of the RCC8 analogue of the Halpern-Shoham logic $\mathcal{HS}$ interpreted over hyperrectangles of $\mathbb{R}^n$, which is known to be not recursively enumerable [Lutz and Wolter, 2006].

We tested the expressivity and efficiency of datalog$\mathcal{HS}_n^\square$ on two real-world data sets: 1D historical data about the Italian Public Administration, and 2D (time × temperature) weather data. Our datalog$\mathcal{HS}_n^\square$ programs defined *complex* 'spatio-temporal' predicates, which were then used in *simple* user queries. Three off-the-shelf datalog tools demonstrated reasonable scalability on the resulting datalog rewritings.

2 Datalog$\mathcal{HS}_n^\square$

Let $\mathfrak{X} = (X, \lhd)$ be either $(\mathbb{Z}, \leq)$ or $(\mathbb{R}, <)$. Denote by $\mathsf{int}(\mathfrak{X})$ the set of closed intervals $[x_1, x_2]$ with $x_1 \lhd x_2$. We define the Allen *interval relations* A, B, E, D, L, O on $\mathsf{int}(\mathfrak{X})$ by taking

- $[x_1, x_2]\ \mathsf{A}\ [y_1, y_2]$ iff $x_2 = y_1$, (After)
- $[x_1, x_2]\ \mathsf{B}\ [y_1, y_2]$ iff $x_1 = y_1$ and $y_2 \lhd x_2$, (Begins)
- $[x_1, x_2]\ \mathsf{E}\ [y_1, y_2]$ iff $x_1 \lhd y_1$ and $x_2 = y_2$, (Ends)
- $[x_1, x_2]\ \mathsf{D}\ [y_1, y_2]$ iff $x_1 \lhd y_1$ and $y_2 \lhd x_2$, (During)
- $[x_1, x_2]\ \mathsf{L}\ [y_1, y_2]$ iff $x_2 \lhd y_1$, (Later)
- $[x_1, x_2]\ \mathsf{O}\ [y_1, y_2]$ iff $x_1 \lhd y_1 \lhd x_2 \lhd y_2$, (Overlaps)

and denote by $\bar{\mathsf{A}}$, $\bar{\mathsf{B}}$, $\bar{\mathsf{E}}$, $\bar{\mathsf{D}}$, $\bar{\mathsf{L}}$, $\bar{\mathsf{O}}$ their inverses. Note that $\mathsf{int}(\mathbb{Z}, \leq)$ contains *punctual* intervals while $\mathsf{int}(\mathbb{R}, <)$ does not. All of our results below would hold if we took any other discrete order with $\leq$ in place of $(\mathbb{Z}, \leq)$ and any other dense order in place of $(\mathbb{R}, <)$. However, for $(\mathbb{Z}, <)$ the logic we are about to define would be *undecidable* [Bresolin et al., 2016].

Fix some $n \geq 1$ and a linear order $\mathfrak{X}_\ell = (X_\ell, \lhd_\ell)$ as above, for $1 \leq \ell \leq n$. A *hyperrectangle* in the n-dimensional space $\boldsymbol{\mathfrak{X}} = \prod_{\ell=1}^n \mathfrak{X}_\ell$ is any n-tuple $\boldsymbol{\iota} = (\iota_1, \dots, \iota_n)$ such that $\iota_\ell \in \mathsf{int}(\mathfrak{X}_\ell)$, for $1 \leq \ell \leq n$. The set of hyperrectangles in $\boldsymbol{\mathfrak{X}}$ is denoted by $\mathsf{hyp}(\boldsymbol{\mathfrak{X}})$. Given $\boldsymbol{\iota}, \boldsymbol{\kappa} \in \mathsf{hyp}(\boldsymbol{\mathfrak{X}})$ and an interval relation R, we write $\boldsymbol{\iota}\mathsf{R}_\ell\boldsymbol{\kappa}$ if $\iota_\ell \mathsf{R} \kappa_\ell$ and $\iota_i = \kappa_i$, for $i \neq \ell$.

Syntax of datalog$\mathcal{HS}_n^\square$. A *data instance*, $\mathcal{D}$, is a finite set of *facts* of the form $P(\boldsymbol{c})@\boldsymbol{\iota}$, where P is an m-ary predicate symbol, $\boldsymbol{c}$ an m-tuple of individual constants, for some $m \geq 0$, and $\boldsymbol{\iota} \in \mathsf{hyp}(\boldsymbol{\mathfrak{X}})$. This fact says that $P(\boldsymbol{c})$ is true at $\boldsymbol{\iota}$. We denote by $\mathsf{ind}(\mathcal{D})$ the set of all individual constants in $\mathcal{D}$, by $\mathsf{num}_\ell(\mathcal{D})$ the set of $x_1, x_2 \in X_\ell$ with $\iota_\ell = [x_1, x_2]$, for some $\boldsymbol{\iota}$ mentioned in $\mathcal{D}$, and by $\mathsf{int}(\mathcal{D})$ the set of $[x_1, x_2] \in \mathsf{int}(\mathfrak{X}_\ell)$ with $x_1, x_2 \in \mathsf{num}_\ell(\mathcal{D})$, for $1 \leq \ell \leq n$.

An *individual term*, t, is an individual variable, v, or a constant, c. A datalog$\mathcal{HS}_n^\square$ *program*, Π, is a finite set of *rules* of the form

$$A^+ \leftarrow A_1 \wedge \cdots \wedge A_k \quad \text{or} \quad \bot \leftarrow A_1 \wedge \cdots \wedge A_k,$$

where $k \geq 1$, each A_i is either an inequality $t \neq t'$ with individual terms t and t' or defined by the grammar

$$A \ ::= \ P(t_1, \dots, t_m) \ \mid \ [\mathsf{R}]_\ell A \ \mid \ \langle \mathsf{R} \rangle_\ell A,$$

for an m-ary predicate P and individual terms t_i, and A^+ does not contain any diamond operators $\langle \mathsf{R} \rangle_\ell$. As usual, the atoms $A_1, \dots, A_k$ constitute the *body* of the rule, while A^+ or $\bot$ its *head*. We also impose other standard datalog restrictions on datalog$\mathcal{HS}_n^\square$ programs. However, we cannot allow $\langle \mathsf{R} \rangle_\ell$ in the heads as this would make our logic undecidable.

Semantics of datalog$\mathcal{HS}_n^\square$. An *interpretation*, $\mathfrak{M}$, is based on a *domain* $\Delta \neq \emptyset$ (for the individual variables and constants) and the space $\boldsymbol{\mathfrak{X}}$. For any m-ary predicate P, m-tuple $\boldsymbol{a}$ from Δ and $\boldsymbol{\iota} \in \mathsf{hyp}(\boldsymbol{\mathfrak{X}})$, $\mathfrak{M}$ specifies whether P is *true on* $\boldsymbol{a}$ *at* $\boldsymbol{\iota}$, in which case we write $\mathfrak{M}, \boldsymbol{\iota} \models P(\boldsymbol{a})$. Let ν be an *assignment* of elements of Δ to the individual variables (we adopt the standard name assumption: $\nu(c) = c$, for every individual constant c). We then set inductively:

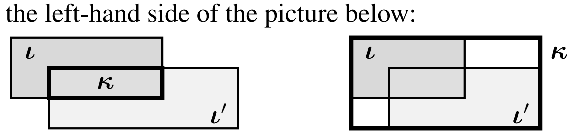
$$\begin{array}{lll} \mathfrak{M}, \boldsymbol{\iota} \models^\nu P(\boldsymbol{t}) & \text{iff} & \mathfrak{M}, \boldsymbol{\iota} \models P(\nu(\boldsymbol{t})), \\ \mathfrak{M}, \boldsymbol{\iota} \models^\nu t \neq t' & \text{iff} & \nu(t) \neq \nu(t'), \\ \mathfrak{M}, \boldsymbol{\iota} \models^\nu [\mathsf{R}]_\ell A & \text{iff} & \mathfrak{M}, \boldsymbol{\kappa} \models^\nu A \text{ for all } \boldsymbol{\kappa} \text{ with } \boldsymbol{\iota}\mathsf{R}_\ell\boldsymbol{\kappa}, \\ \mathfrak{M}, \boldsymbol{\iota} \models^\nu \langle \mathsf{R} \rangle_\ell A & \text{iff} & \mathfrak{M}, \boldsymbol{\kappa} \models^\nu A \text{ for some } \boldsymbol{\kappa} \text{ with } \boldsymbol{\iota}\mathsf{R}_\ell\boldsymbol{\kappa}. \end{array}$$

We say that $\mathfrak{M}$ *satisfies* Π *under* ν if, for all $\boldsymbol{\iota} \in \mathsf{hyp}(\boldsymbol{\mathfrak{X}})$ and all rules $A \leftarrow A_1 \wedge \cdots \wedge A_k$ in Π, we have

$$\mathfrak{M}, \boldsymbol{\iota} \models^\nu A \quad \text{whenever} \quad \mathfrak{M}, \boldsymbol{\iota} \models^\nu A_i \text{ for } 1 \leq i \leq k$$

(as usual $\mathfrak{M}, \boldsymbol{\iota} \not\models^\nu \bot$). $\mathfrak{M}$ is a *model* of Π and $\mathcal{D}$ if it satisfies Π under every assignment, and $\mathfrak{M}, \boldsymbol{\iota} \models P(\boldsymbol{c})$, for every fact $P(\boldsymbol{c})@\boldsymbol{\iota}$ in $\mathcal{D}$. Π and $\mathcal{D}$ are *consistent* if they have a model.

Example 1. Denote by $\langle \mathsf{Int} \rangle$ the binary modal operator such that $A \langle \mathsf{Int} \rangle A'$ holds at a hyperrectangle $\boldsymbol{\kappa}$ iff A holds at some $\boldsymbol{\iota}$, A' at some $\boldsymbol{\iota}'$, and $\boldsymbol{\kappa} = \boldsymbol{\iota} \cap \boldsymbol{\iota}'$. One can show that rules such as $B \leftarrow A \langle \mathsf{Int} \rangle A'$ are expressible as datalog$\mathcal{HS}_n^\square$ programs. For example, for $n = 2$, there are $13^2 = 169$ different relative positions of two rectangles; see, e.g., [Navarrete et al., 2013, Fig. 4] for an illustration. Those of them where the rectangles have non-empty intersection are encoded by datalog$\mathcal{HS}_n^\square$ rules such as $B \leftarrow \langle \bar{\mathsf{E}} \rangle_1 \langle \bar{\mathsf{B}} \rangle_2 A \wedge \langle \bar{\mathsf{B}} \rangle_1 \langle \bar{\mathsf{E}} \rangle_2 A'$ for the position on the left-hand side of the picture below:



Similarly, one can express the rule $B \leftarrow A \langle \mathsf{Cov} \rangle A'$ such that $A \langle \mathsf{Cov} \rangle A'$ holds at $\boldsymbol{\kappa}$ iff $\boldsymbol{\kappa}$ is the smallest hyperrectangle containing some $\boldsymbol{\iota}$ with A and $\boldsymbol{\iota}'$ with A'; see above right.

Ontology-mediated queries. An *interval term*, τ, is either an interval, ι, or an *interval variable*, χ. A *conjunctive query* (CQ) is a formula of the form

$$\boldsymbol{q}(\boldsymbol{v},\boldsymbol{\chi}) = \exists \boldsymbol{v}', \boldsymbol{\chi}'\, \Phi(\boldsymbol{v},\boldsymbol{v}',\boldsymbol{\chi},\boldsymbol{\chi}'), \qquad (1)$$

where Φ is a conjunction of atoms of two types:

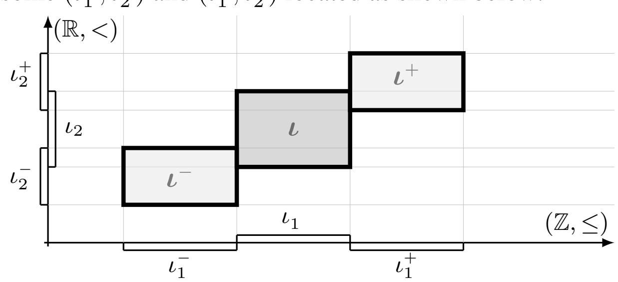
- $P(\boldsymbol{t})@\boldsymbol{\tau}$ with $\boldsymbol{t} = (t_1, \dots, t_m)$ and $\boldsymbol{\tau} = (\tau_1, \dots, \tau_n)$,
- $\mathsf{R}(\tau, \tau')$ with an interval relation R,

and all individual and interval variables in Φ are from $\boldsymbol{v} \cup \boldsymbol{v}'$ and $\boldsymbol{\chi} \cup \boldsymbol{\chi}'$, respectively. A datalog$\mathcal{HS}_n^\Box$ program Π and a CQ $\boldsymbol{q}(\boldsymbol{v},\boldsymbol{\chi})$ constitute an *ontology-mediated query* (OMQ) $\boldsymbol{Q}(\boldsymbol{v},\boldsymbol{\chi}) = (\Pi, \boldsymbol{q}(\boldsymbol{v},\boldsymbol{\chi}))$.

Example 2. Suppose $\mathfrak{X} = \mathfrak{X}_1 \times \mathfrak{X}_2$, where $\mathfrak{X}_1 = (\mathbb{Z}, \leq)$ represents time and $\mathfrak{X}_2 = (\mathbb{R}, <)$ temperature. Imagine that a turbine monitoring system is receiving from sensors a stream of data of the form $\mathsf{Blade}(\mathsf{ID140})@(\iota_1, \iota_2)$, where $\mathsf{ID140}$ is a blade ID and $\iota_2 \in \mathsf{int}(\mathbb{R}, <)$ is the observed temperature range during the time interval $\iota_1 \in \mathsf{int}(\mathbb{Z}, \leq)$. Then the rule

$$\mathsf{TempRise}(v) \leftarrow \langle \bar{\mathsf{A}} \rangle_1 \langle \bar{\mathsf{O}} \rangle_2 \mathsf{Blade}(v) \wedge \langle \mathsf{A} \rangle_1 \langle \mathsf{O} \rangle_2 \mathsf{Blade}(v)$$

says that the temperature of blade v is rising over a rectangle (ι_1, ι_2) if $\mathsf{Blade}(v)@(\iota_1^-, \iota_2^-)$ and $\mathsf{Blade}(v)@(\iota_1^+, \iota_2^+)$ hold at some (ι_1^-, ι_2^-) and (ι_1^+, ι_2^+) located as shown below:



The temperature drop is defined analogously:

$$\mathsf{TempDrop}(v) \leftarrow \langle \bar{\mathsf{A}} \rangle_1 \langle \mathsf{O} \rangle_2 \mathsf{Blade}(v) \wedge \langle \mathsf{A} \rangle_1 \langle \bar{\mathsf{O}} \rangle_2 \mathsf{Blade}(v).$$

To find the blades v and the time intervals χ such that the temperature of v was rising before χ, reaching 1500° in χ, and dropping after that, we can use the following CQ $\boldsymbol{q}(v, \chi)$

$$\exists \chi^-, \rho^-, \rho, \chi^+, \rho^+ \big[\mathsf{Blade}(v)@(\chi, \rho) \wedge \mathsf{TempRise}(v)@(\chi^-, \rho^-) \wedge \mathsf{TempDrop}(v)@(\chi^+, \rho^+) \wedge \mathsf{A}(\chi^-, \chi) \wedge \mathsf{A}(\chi, \chi^+) \wedge \mathsf{O}(\rho, [1500, 1600])\big].$$

Let $\boldsymbol{Q}(\boldsymbol{v},\boldsymbol{\chi}) = (\Pi, \boldsymbol{q}(\boldsymbol{v},\boldsymbol{\chi}))$ be an OMQ and $\mathcal{D}$ a data instance. A *certain answer to* $\boldsymbol{Q}(\boldsymbol{v},\boldsymbol{\chi})$ *over* $\mathcal{D}$ is any pair $(\boldsymbol{c}, \boldsymbol{\delta})$ satisfying the following conditions:

- $\boldsymbol{c} \subseteq \mathsf{ind}(\mathcal{D})$ and $\boldsymbol{\delta} \subseteq \mathsf{int}(\mathcal{D})$, with $|\boldsymbol{v}| = |\boldsymbol{c}|$, $|\boldsymbol{\chi}| = |\boldsymbol{\delta}|$;
- for every model $\mathfrak{M}$ of Π and $\mathcal{D}$, there is a map h of the individual terms in $\boldsymbol{q}$ to Δ and the interval terms to $\bigcup_\ell \mathsf{int}(\mathfrak{X}_\ell)$ preserving constants and dimensions such that $h(\boldsymbol{v}) = \boldsymbol{c}$, $h(\boldsymbol{\chi}) = \boldsymbol{\delta}$, $\mathfrak{M}, h(\boldsymbol{\tau}) \models P(h(\boldsymbol{t}))$ for every atom $P(\boldsymbol{t})@\boldsymbol{\tau}$ in $\boldsymbol{q}$, and $\mathsf{R}(h(\tau), h(\tau'))$ holds in the corresponding $\mathfrak{X}_\ell$, for every atom $\mathsf{R}(\tau, \tau')$ in $\boldsymbol{q}$.

We say that a program Π is in *normal form* if it does not contain occurrences of $\langle \mathsf{R} \rangle_\ell$ and nested $[\mathsf{R}]_\ell$. It is readily seen that any OMQ $\boldsymbol{Q} = (\Pi, \boldsymbol{q})$ can be transformed to an OMQ $\boldsymbol{Q}' = (\Pi', \boldsymbol{q})$ with Π' in normal form and of linear size in $|\Pi|$ such that $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ have the same certain answers over any data instance (in a given signature). For example, the rule $S(\boldsymbol{t}) \leftarrow \langle \mathsf{R} \rangle_\ell P(\boldsymbol{t}) \wedge Q(\boldsymbol{t})$ can be replaced with two rules $S(\boldsymbol{t}) \leftarrow P'(\boldsymbol{t}) \wedge Q(\boldsymbol{t})$ and $[\bar{\mathsf{R}}]_\ell P'(\boldsymbol{t}) \leftarrow P(\boldsymbol{t})$, for a fresh P'. We assume all our programs are in normal form.

Our aim now is to show that finding certain answers to an OMQ $\boldsymbol{Q}(\boldsymbol{v},\boldsymbol{\chi})$ over any given $\mathcal{D}$ can be reduced to finding answers to a standard datalog query $(\Pi^\dagger, G(\boldsymbol{v},\boldsymbol{\chi}))$ over $\mathcal{D}$. To do this, we first characterise the structure of models of datalog$\mathcal{HS}_n^\Box$ programs.

3 Canonical Models

Let Π be a datalog$\mathcal{HS}_n^\Box$ program and $\mathcal{C}$ a set of ground atoms of the form $P(\boldsymbol{c})@\boldsymbol{\iota}$. Denote by $\mathsf{cl}(\mathcal{C})$ the result of applying non-recursively the following rules to $\mathcal{C}$:

- if $A \leftarrow A_1 \wedge \dots \wedge A_k$ is a rule in Π and $\mathcal{C}$ contains $\nu(A_i)@\boldsymbol{\iota}$, for all $1 \leq i \leq k$ and some assignment ν, then we add $\nu(A)@\boldsymbol{\iota}$ to $\mathcal{C}$;
- if $[\mathsf{R}]_\ell A@\boldsymbol{\iota} \in \mathcal{C}$, then we add to $\mathcal{C}$ all $A@\boldsymbol{\kappa}$ with $\boldsymbol{\iota}\mathsf{R}_\ell\boldsymbol{\kappa}$;
- if $A@\boldsymbol{\kappa} \in \mathcal{C}$ for all $\boldsymbol{\kappa}$ with $\boldsymbol{\iota}\mathsf{R}_\ell\boldsymbol{\kappa}$, then add $[\mathsf{R}]_\ell A@\boldsymbol{\iota}$ to $\mathcal{C}$.

(Here, given an atom A and an assignment ν, we denote by $\nu(A)$ the result of replacing each variable v in A with $\nu(v)$; we also assume that $\mathcal{C}$ contains all inequalities $c \neq c'$ for distinct constants c and c'.) Then we set $\mathsf{cl}^0(\mathcal{C}) = \mathcal{C}$ and, for any successor ordinal $\xi + 1$ and limit ordinal ζ,

$$\mathsf{cl}^{\xi+1}(\mathcal{C}) = \mathsf{cl}(\mathsf{cl}^\xi(\mathcal{C})) \quad \text{and} \quad \mathsf{cl}^\zeta(\mathcal{C}) = \bigcup_{\xi<\zeta} \mathsf{cl}^\xi(\mathcal{C}).$$

Now, given a data instance $\mathcal{D}$, we set $\mathfrak{C}_{\Pi,\mathcal{D}} = \mathsf{cl}^{2^{\aleph_0}}(\mathcal{D})$. If $\bot@\boldsymbol{\iota} \notin \mathfrak{C}_{\Pi,\mathcal{D}}$ for all $\boldsymbol{\iota}$, we regard $\mathfrak{C}_{\Pi,\mathcal{D}}$ as an interpretation and call it the *canonical model* of Π and $\mathcal{D}$.

Theorem 3. (*i*) *If* $\bot@\boldsymbol{\iota} \in \mathfrak{C}_{\Pi,\mathcal{D}}$ *for any* $\boldsymbol{\iota}$*, then* Π *and* $\mathcal{D}$ *are inconsistent; otherwise,* Π *and* $\mathcal{D}$ *are consistent and* $\mathfrak{C}_{\Pi,\mathcal{D}}$ *is their minimal model in the sense that* $P(\boldsymbol{c})@\boldsymbol{\iota} \in \mathfrak{C}_{\Pi,\mathcal{D}}$ *implies* $\mathfrak{M}, \boldsymbol{\iota} \models P(\boldsymbol{c})$*, for any model* $\mathfrak{M}$ *of* Π *and* $\mathcal{D}$.

(*ii*) *A pair* $(\boldsymbol{c}, \boldsymbol{\delta})$ *is a certain answer to* $(\Pi, \boldsymbol{q}(\boldsymbol{v},\boldsymbol{\chi}))$ *over* $\mathcal{D}$ *that is consistent with* Π *iff* $\mathfrak{C}_{\Pi,\mathcal{D}} \models \boldsymbol{q}(\boldsymbol{c}, \boldsymbol{\delta})$.

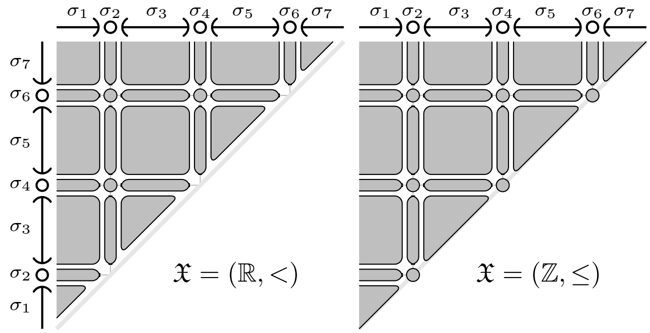
Our next aim is to describe the structure of $\mathfrak{C}_{\Pi,\mathcal{D}}$. Suppose $\mathfrak{X} = (X, \lhd)$ is a linear order and $K = \{k_1, \dots, k_m\}$ a finite subset of X with $k_j \lhd k_{j+1}$, for $1 \leq j < m$. Denote by $\mathsf{sec}_\mathfrak{X}(K)$ the following set of *sections* of X:

$$(-\infty, k_1), [k_1, k_1], (k_1, k_2), [k_2, k_2], \dots, [k_m, k_m], (k_m, \infty),$$

where $(-\infty, k) = \{x \in X \mid x \lhd k,\ x \neq k\}$ and symmetrically for (k, ∞). Given $\sigma_1, \sigma_2 \in \mathsf{sec}_\mathfrak{X}(K)$, we write $\sigma_1 \preceq \sigma_2$ if $[x_1, x_2] \in \mathsf{int}(\mathfrak{X})$, for some $x_1 \in \sigma_1$, $x_2 \in \sigma_2$. The relation $\preceq$ is a linear order, which is reflexive if $\lhd$ is reflexive (if $\lhd$ is irreflexive then $[k, k] \not\preceq [k, k]$). We now define a *partition* of $\mathsf{int}(\mathfrak{X})$ into non-empty *zones* of the form

$$\zeta_{\sigma_1,\sigma_2} = \{[x_1, x_2] \in \mathsf{int}(\mathfrak{X}) \mid x_1 \in \sigma_1,\ x_2 \in \sigma_2\},$$

for any $\sigma_1, \sigma_2 \in \mathsf{sec}_\mathfrak{X}(K)$ with $\sigma_1 \preceq \sigma_2$. The set of zones is denoted by $\mathsf{zone}_\mathfrak{X}(K)$. For $m = 3$, the partition of $\mathsf{int}(\mathfrak{X})$ into zones is shown below, where intervals $[x, y]$ are represented by points (x, y) in the Euclidean plane:



Note the following property of zones in $\mathsf{zone}_{\mathfrak{X}}(K)$:

(zone-uniformity) for any interval relation R and any zones ζ_1 and ζ_2, if $\iota_1 \mathsf{R} \iota_2$ for some $\iota_1 \in \zeta_1$ and $\iota_2 \in \zeta_2$, then for every $\iota_1' \in \zeta_1$ there is $\iota_2' \in \zeta_2$ such that $\iota_1' \mathsf{R} \iota_2'$.

It is not hard to see that, for $p = 2|K| + 1$,

$$|\mathsf{zone}_{\mathfrak{X}}(K)| = \begin{cases} (p^2+1)/2, & \text{if } \mathfrak{X} = (\mathbb{R}, <), \\ p(p+1)/2, & \text{if } \mathfrak{X} = (\mathbb{Z}, \leq). \end{cases}$$

Given a data instance $\mathcal{D}$, we define the set $\mathsf{zone}(\mathcal{D})$ of *hyperzones* of $\mathcal{D}$ as $\prod_{\ell=1}^{n} \mathsf{zone}_\ell(\mathcal{D})$, where $\mathsf{zone}_\ell(\mathcal{D}) = \mathsf{zone}_{\mathfrak{X}_\ell}(\mathsf{num}_\ell(\mathcal{D}))$. By using **(zone-uniformity)** for both R and $\bar{\mathsf{R}}$ along with the definition of $\mathfrak{C}_{\Pi,\mathcal{D}}$, we obtain:

Theorem 4. *If $P(\boldsymbol{c})@\boldsymbol{\iota} \in \mathfrak{C}_{\Pi,\mathcal{D}}$, then $P(\boldsymbol{c})@\boldsymbol{\kappa} \in \mathfrak{C}_{\Pi,\mathcal{D}}$ for all $\boldsymbol{\kappa}$ from the same hyperzone as $\boldsymbol{\iota}$.*

Note that **(zone-uniformity)** and Theorem 4 do not hold for $(\mathbb{Z}, <)$; see [Bresolin et al., 2016]. For example, consider $\Pi = \{[\mathsf{E}]q \leftarrow [\mathsf{B}]r,\ [\mathsf{E}]r \leftarrow [\mathsf{B}]q\}$ and $\mathcal{D} = \{q@[0,0]\}$: we have $q@[n,n] \in \mathfrak{C}_{\Pi,\mathcal{D}}$ for even $n \in \mathbb{N}$ but $q@[n,n] \notin \mathfrak{C}_{\Pi,\mathcal{D}}$ for odd n.

We can view each order $\mathfrak{X}_\ell$ as a Kripke frame $\mathfrak{I}_\ell$ with the set of worlds $\mathsf{int}(\mathfrak{X}_\ell)$ and accessibility relations R. The space $\boldsymbol{\mathfrak{X}} = \prod_{\ell=1}^{n} \mathfrak{X}_\ell$ gives rise to the n-dimensional product frame $\boldsymbol{\mathfrak{I}} = \prod_{\ell=1}^{n} \mathfrak{I}_\ell$ [Gabbay et al., 2003]. The canonical model $\mathfrak{C}_{\Pi,\mathcal{D}}$ can then be regarded as a first-order Kripke model with the first-order domain $\mathsf{ind}(\mathcal{D})$ and Kripke frame $\boldsymbol{\mathfrak{I}}$ where $P(\boldsymbol{c})$ holds at a world $\boldsymbol{\iota}$ iff $P(\boldsymbol{c})@\boldsymbol{\iota} \in \mathfrak{C}_{\Pi,\mathcal{D}}$.

Denote by $\mathfrak{Z}_\ell$ the Kripke frame with the set of worlds $\mathsf{zone}_\ell(\mathcal{D})$ and accessibility relations R over them such that $\zeta_1 \mathsf{R} \zeta_2$ iff $\iota_1 \mathsf{R} \iota_2$ for some $\iota_1 \in \zeta_1$ and $\iota_2 \in \zeta_2$. Now, define a Kripke model $\mathfrak{S}_{\Pi,\mathcal{D}}$ over the product frame $\boldsymbol{\mathfrak{Z}} = \prod_{\ell=1}^{n} \mathfrak{Z}_\ell$ as follows: $P(\boldsymbol{c})$ holds in a world $\boldsymbol{\zeta}$ iff $P(\boldsymbol{c})@\boldsymbol{\iota} \in \mathfrak{C}_{\Pi,\mathcal{D}}$ for some (equivalently, all) $\boldsymbol{\iota} \in \boldsymbol{\zeta}$. Let f_ℓ be a map from $\mathfrak{X}_\ell$ onto $\mathsf{zone}_\ell(\mathcal{D})$ sending all intervals in any zone ζ to ζ, and let $f(\boldsymbol{\iota}) = (f_1(\iota_1), \dots, f_n(\iota_n))$. By Theorem 4 and **(zone-uniformity)**, f is a p-morphism from $\mathfrak{C}_{\Pi,\mathcal{D}}$ onto $\mathfrak{S}_{\Pi,\mathcal{D}}$. This fact and Theorem 3 give the following theorem:

Theorem 5. *Fix some $\boldsymbol{\mathfrak{X}} = \prod_{\ell=1}^{n} \mathfrak{X}_\ell$ with $\mathfrak{X}_\ell = (\mathbb{Z}, \leq)$ or $\mathfrak{X}_\ell = (\mathbb{R}, <)$. For any OMQ $\boldsymbol{Q}(\boldsymbol{v}, \boldsymbol{\chi}) = (\Pi, \boldsymbol{q}(\boldsymbol{v}, \boldsymbol{\chi}))$ and any data instance $\mathcal{D}$ that is consistent with Π, a pair $(\boldsymbol{c}, \boldsymbol{\delta})$ is a certain answer to $\boldsymbol{Q}$ over $\mathcal{D}$ iff there is a tuple $\boldsymbol{\xi}$ of zones such that $\mathfrak{S}_{\Pi,\mathcal{D}} \models \boldsymbol{q}(\boldsymbol{c}, \boldsymbol{\xi})$ and $\delta_i \in \xi_i$, for $1 \leq i \leq |\boldsymbol{\delta}|$.*

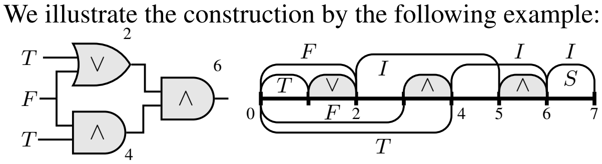
The model $\mathfrak{S}_{\Pi,\mathcal{D}}$ can be constructed in polynomial time using the same rules as in the definition of $\mathfrak{C}_{\Pi,\mathcal{D}}$, but with a quadratic number of hyperzones in place of hyperrectangles. This gives the upper bound of the next theorem:

Theorem 6. *The problem of checking whether $(\boldsymbol{c}, \boldsymbol{\delta})$ is a certain answer to $\boldsymbol{Q}(\boldsymbol{v}, \boldsymbol{\chi})$ over $\mathcal{D}$ is* P*-complete for data complexity and* ExpTime*-complete for combined complexity. For propositional datalog$\mathcal{HS}_n^\Box$ programs, the problem is* P*-complete for combined complexity.*

We establish the lower bound for data complexity by reduction of the monotone (fan-in 2) circuit value problem, which is known to be P-hard. Let $\boldsymbol{C}$ be a monotone circuit whose gates are enumerated by consecutive positive even integers so that if there is an edge from n to m then $n < m$. We encode $\boldsymbol{C}$ with an input $\boldsymbol{\alpha}$ by a data instance $\mathcal{D}$ with the facts:

- $V@([0, n-1])$ (or $V@([0, n])$) if the first (resp., second) input of gate n gets a value $V \in \{T, F\}$;
- $\mathsf{G}@([n-1, n])$, for every G-gate n, $\mathsf{G} \in \{\mathrm{AND}, \mathrm{OR}\}$;
- $I@([n, m-1])$ (or $I@([n, m])$) if the output of gate n is the first (resp., second) input of gate m;
- $S@([N, N+1])$ and $I@([N, N+1])$, where N is the last (output) gate in $\boldsymbol{C}$.

We illustrate the construction by the following example:



Let Π be the *propositional* program

$$T \leftarrow I \wedge \langle \bar{\mathsf{A}} \rangle (\mathrm{AND} \wedge \langle \bar{\mathsf{A}} \rangle T \wedge \langle \bar{\mathsf{E}} \rangle T),$$
$$F \leftarrow I \wedge \langle \bar{\mathsf{A}} \rangle (\mathrm{AND} \wedge \langle \bar{\mathsf{A}} \rangle T \wedge \langle \bar{\mathsf{E}} \rangle F),$$
$$F \leftarrow I \wedge \langle \bar{\mathsf{A}} \rangle (\mathrm{AND} \wedge \langle \bar{\mathsf{A}} \rangle F \wedge \langle \bar{\mathsf{E}} \rangle T),$$
$$F \leftarrow I \wedge \langle \bar{\mathsf{A}} \rangle (\mathrm{AND} \wedge \langle \bar{\mathsf{A}} \rangle F \wedge \langle \bar{\mathsf{E}} \rangle F),$$

whose clauses encode the truth-table for AND and dually for OR with F and T swapped (which can clearly be rewritten in datalog$\mathcal{HS}_1^\Box$). One can check that $\boldsymbol{C}(\boldsymbol{\alpha}) = 1$ iff the answer to the OMQ $(\Pi, \exists \chi\, (S@\chi \wedge T@\chi))$ over $\mathcal{D}$ is 'yes'. Note that this reduction works for both $(\mathbb{Z}, \leq)$ and $(\mathbb{R}, <)$. Note also that OMQs with $\mathit{LTL}_{horn}^\Box$ ontologies [Artale et al., 2015b] (which are syntactically similar to propositional datalog$\mathcal{HS}_1^\Box$ programs but with the underlying $\Box$-fragment of the linear time temporal logic) can be evaluated in AC^0 for data complexity. The difference between the data complexities of these formalisms can be explained by the fact that (classical) $\mathcal{HS}$ is essentially *two-dimensional* whereas *LTL* operates in one dimension; cf. [Gabbay et al., 2003].

We show now that the problem of finding certain answers to datalog$\mathcal{HS}_n^\Box$ OMQs can be reduced to computing answers to standard datalog programs over the same data instances.

4 Datalog Rewriting

Suppose we are given an OMQ $\boldsymbol{Q}(\boldsymbol{v}, \boldsymbol{\chi}) = (\Pi, \boldsymbol{q}(\boldsymbol{v}, \boldsymbol{\chi}))$. Our aim is to rewrite $\boldsymbol{Q}(\boldsymbol{v}, \boldsymbol{\chi})$ to a datalog program $\Pi^\dagger$ with a goal $G(\boldsymbol{v}, \boldsymbol{\chi})$ such that, for any data instance $\mathcal{D}$, a tuple $(\boldsymbol{c}, \boldsymbol{\delta})$ is a certain answer to $\boldsymbol{Q}(\boldsymbol{v}, \boldsymbol{\chi})$ over $\mathcal{D}$ iff $\Pi^\dagger, \mathcal{D} \models G(\boldsymbol{c}, \boldsymbol{\delta})$. To simplify presentation, we only consider the case of 1D space;

a generalisation to n dimensions is straightforward though cumbersome. Here we only show the rules for $\mathfrak{X} = (\mathbb{R}, <)$ and indicate the necessary modifications for $(\mathbb{Z}, \leq)$.

To operate with numbers in $\mathsf{num}(\mathcal{D})$, we first use the built-in $<$, $\neq$ and stratified negation to define in $\Pi^\dagger$ the predicates succ, max and min over $\mathsf{num}(\mathcal{D})$, where $\mathsf{succ}(s_1, s_2)$ iff s_2 is an immediate $<$-successor of s_1 in $\mathsf{num}(\mathcal{D})$, $\mathsf{max}(s)$ iff s is the maximal number in $\mathsf{num}(\mathcal{D})$, and similarly for $\mathsf{min}(s)$. We also use two constants $+\infty$ and $-\infty$ assuming that $-\infty < s < +\infty$, for any $s \in \mathsf{num}(\mathcal{D})$. Note that stratified negation (supported by most datalog tools) is not required elsewhere in $\Pi^\dagger$. In fact, a system implementing datalog$\mathcal{HS}_n^\Box$ could simply maintain the extensions of the above predicates, which are ontology- and query-independent.

We encode sections $\mathsf{sec}_{\mathfrak{X}}(\mathsf{num}(\mathcal{D}))$ of the partition of $\mathfrak{X}$ using a binary predicate $\mathsf{sec}(s_1, s_2)$: punctual sections correspond to $s_1 = s_2$ and open sections to $s_1 < s_2$. More formally, $\mathsf{sec}(s_1, s_2)$ is defined by the following rules:

$$\mathsf{sec}(s_1, s_2) \leftarrow (s_1 = s_2) \wedge (s_1 \neq -\infty) \wedge (s_2 \neq +\infty),$$
$$\mathsf{sec}(s_1, s_2) \leftarrow \mathsf{succ}(s_1, s_2),$$
$$\mathsf{sec}(s_1, +\infty) \leftarrow \mathsf{max}(s_1),$$
$$\mathsf{sec}(-\infty, s_2) \leftarrow \mathsf{min}(s_2).$$

We also require the immediate proper $\preceq$-successor relation on $\mathsf{sec}_{\mathfrak{X}}(\mathsf{num}(\mathcal{D}))$. Recall that for $(\mathbb{R}, <)$ each punctual section is followed by an open one and the other way round:

$$\mathsf{nx}(s_1, s_1, s_1, s_2) \leftarrow \mathsf{sec}(s_1, s_2) \wedge (s_1 < s_2) \wedge (s_1 \neq -\infty),$$
$$\mathsf{nx}(s_1, s_2, s_2, s_2) \leftarrow \mathsf{sec}(s_1, s_2) \wedge (s_1 < s_2) \wedge (s_2 \neq +\infty).$$

Thus, $\mathsf{nx}(s_1, s_2, s_1', s_2')$ holds just in case the section defined by (s_1', s_2') immediately follows the section of (s_1, s_2). Note that there is a third option for $(\mathbb{Z}, \leq)$: the punctual section $[s, s]$ can be followed by the punctual section $[s+1, s+1]$.

Each zone in $\mathsf{zone}_{\mathfrak{X}}(\mathsf{num}(\mathcal{D}))$ is characterised by a pair of sections in $\mathsf{sec}_{\mathfrak{X}}(\mathsf{num}(\mathcal{D}))$, and so we use quadruples of variables to specify zones and define a quaternary predicate zone for $(\mathbb{R}, <)$ by the rule

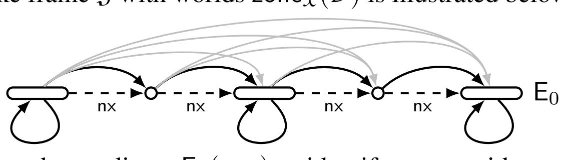
$$\mathsf{zone}(s_1, s_2, e_1, e_2) \leftarrow \mathsf{sec}(s_1, s_2) \wedge \mathsf{sec}(e_1, e_2) \wedge (s_1 < e_2).$$

For $(\mathbb{Z}, \leq)$, the inequality is $(s_1 \leq e_1) \wedge (s_2 \leq e_2)$.

Assuming that Π is in normal form, we replace each $P(\boldsymbol{t})$ in it with $P^*(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e})$ and each $[\mathsf{R}]P(\boldsymbol{t})$ with $P_{[\mathsf{R}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e})$, where two pairs $\boldsymbol{s} = (s_1, s_2)$ and $\boldsymbol{e} = (e_1, e_2)$ form a zone quadruple. To translate the facts from a data instance into zones, we use the rules

$$P^*(\boldsymbol{t}, s, s, e, e) \leftarrow P(\boldsymbol{t})@[s, e].$$

Thus, it remains to define the predicates of the form $P_{[\mathsf{R}]}$. We begin with $P_{[\mathsf{E}]}$. A slice of the accessibility relation E in the Kripke frame $\mathfrak{Z}$ with worlds $\mathsf{zone}_{\mathfrak{X}}(\mathcal{D})$ is illustrated below:



We use the predicate $\mathsf{E}_0(\boldsymbol{s}, \boldsymbol{e})$ to identify zones without any proper E-successors:

$$\mathsf{E}_0(s_1, s_2, e_1, e_2) \leftarrow \mathsf{zone}(s_1, s_2, e_1, e_2) \wedge (s_2 = e_2).$$

If $[\mathsf{E}]P$ holds at a zone, then both P^* and $[\mathsf{E}]P$ are propagated to all E-accessible zones using the rules

$$P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}', \boldsymbol{e}) \leftarrow P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \wedge \mathsf{nx}(\boldsymbol{s}, \boldsymbol{s}') \wedge \mathsf{zone}(\boldsymbol{s}', \boldsymbol{e}),$$
$$P^*(\boldsymbol{t}, \boldsymbol{s}', \boldsymbol{e}) \leftarrow P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \wedge \mathsf{nx}(\boldsymbol{s}, \boldsymbol{s}') \wedge \mathsf{zone}(\boldsymbol{s}', \boldsymbol{e}),$$
$$P^*(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \leftarrow P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \wedge (s_1 < s_2).$$

(The last rule reflects reflexivity of E in zones based on punctual sections.) For the converse direction, we require the following recursive rules for $P_{[\mathsf{E}]}$ of its fixpoint definition:

$$P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \leftarrow P^*(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \wedge \mathsf{E}_0(\boldsymbol{s}, \boldsymbol{e}),$$
$$P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \leftarrow (s_1 = s_2) \wedge \mathsf{nx}(\boldsymbol{s}, \boldsymbol{s}') \wedge P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}', \boldsymbol{e}),$$
$$P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \leftarrow (s_1 < s_2) \wedge P^*(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \wedge \mathsf{nx}(\boldsymbol{s}, \boldsymbol{s}') \wedge P_{[\mathsf{E}]}(\boldsymbol{t}, \boldsymbol{s}', \boldsymbol{e}).$$

We define $P_{[\bar{\mathsf{E}}]}$, $P_{[\mathsf{B}]}$ and $P_{[\bar{\mathsf{B}}]}$ in a similar manner. For example, we need

$$\bar{\mathsf{E}}_0(-\infty, s_2, e_1, e_2) \leftarrow \mathsf{min}(s_2) \wedge \mathsf{sec}(e_1, e_2),$$

and the rules for $P_{[\mathsf{B}]}$ and $P_{[\bar{\mathsf{B}}]}$ contain $\mathsf{nx}(\boldsymbol{e}, \boldsymbol{e}')$ in place of $\mathsf{nx}(\boldsymbol{s}, \boldsymbol{s}')$. These predicates are then used to define $P_{[\mathsf{R}]}$ for the remaining relations R. For example, for $P_{[\mathsf{A}]}$, we have

$$P_{[\mathsf{A}]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \leftarrow \mathsf{zone}(\boldsymbol{s}, \boldsymbol{e}) \wedge \mathsf{B}_0(\boldsymbol{e}, \boldsymbol{s}') \wedge P_{[\bar{\mathsf{B}}]}(\boldsymbol{t}, \boldsymbol{e}, \boldsymbol{s}'),$$
$$P_{[\bar{\mathsf{B}}]}(\boldsymbol{t}, \boldsymbol{e}, \boldsymbol{s}') \leftarrow \mathsf{B}_0(\boldsymbol{e}, \boldsymbol{s}') \wedge P_{[A]}(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}).$$

Finally, suppose $\boldsymbol{q}(\boldsymbol{v}, \boldsymbol{\chi}) = \exists \boldsymbol{v}', \boldsymbol{\chi}'\, \Phi(\boldsymbol{v}, \boldsymbol{v}', \boldsymbol{\chi}, \boldsymbol{\chi}')$, where Φ is a conjunction of atoms of the form $P(\boldsymbol{t})@[\tau_1, \tau_2]$ and $\mathsf{R}(\tau_1, \tau_2)$. For each such $P(\boldsymbol{t})@[\tau_1, \tau_2]$, we add to our program the rules

$$P^\dagger(\boldsymbol{t}, s, e) \leftarrow P^*(\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{e}) \wedge \mathsf{in}(s, \boldsymbol{s}) \wedge \mathsf{in}(e, \boldsymbol{e}),$$

where the ternary predicate in is defined by

$$\mathsf{in}(s, s_1, s_2) \leftarrow \mathsf{sec}(s_1, s_2) \wedge (s_1 = s = s_2),$$
$$\mathsf{in}(s, s_1, s_2) \leftarrow \mathsf{sec}(s_1, s_2) \wedge (s_1 < s < s_2).$$

We also add the rules defining the interval relations R such as

$$\mathsf{E}(x_1, x_2, y_1, y_2) \leftarrow (x_1 < y_1) \wedge (x_2 = y_2).$$

Finally, we add the rule with the head $G(\boldsymbol{v}, \boldsymbol{\chi})$ and the body containing the $P^\dagger(\boldsymbol{t}, \tau)$ and $\mathsf{R}(\tau_1, \tau_2)$ from the given CQ, and denote the resulting program by $\Pi^\dagger$.

Theorem 7. *For any data instance $\mathcal{D}$, a tuple $(\boldsymbol{c}, \boldsymbol{\delta})$ is a certain answer to $\boldsymbol{Q}(\boldsymbol{v}, \boldsymbol{\chi})$ over $\mathcal{D}$ iff $\Pi^\dagger, \mathcal{D} \models G(\boldsymbol{c}, \boldsymbol{\delta})$.*

Note that the size of $\Pi^\dagger$ is linear in the size of $\boldsymbol{Q}$. Indeed, the rewriting consists of a fixed number of rules to define the partition and a fixed number of rules for each $[\mathsf{R}]P$ in the program. The arity of predicates in $\Pi^\dagger$ is the original arity of predicates $+4n$ for n zone quadruples in n-dimensional case.

5 Case Studies

We tested the expressive power and efficiency of OBDA with datalog$\mathcal{HS}_n^\Box$ using two real-world scenarios. In our experiments, we used three off-the-shelf tools for (extensions of) datalog: CLASP [Gebser, Kaufmann, and Schaub, 2012] (v 3.1.4, with GRINGO v 4.5.4 as grounder), DLV [Leone et al., 2006] (v Dec 17 2012), and XSB [Sagonas, Swift, and Warren, 1994] (v 3.6). The three systems ran with default parameters on an Intel Xeon E3-1245 3.30 GHz workstation with 16GB of memory under the Ubuntu 12.04 64-bit OS.

5.1 Querying Historical Documents: 1D Case

In the first case study, we used datalog$\mathcal{HS}_n^\square$ to query the historical data in the STOLE[1] ontology, which extracts facts about the Italian Public Administration from journal articles [Adorni et al., 2015]. The dataset contains 1743 facts about institutions, legal systems, historical events and people obtained from articles published between 1850 and 1935. We represented the timestamps in years using the interval structure of $(\mathbb{Z}, \leq)$. The (atemporal) STOLE ontology was extended with datalog$\mathcal{HS}_n^\square$ rules such as

$$\mathit{RdSInst}(x) \leftarrow \mathit{Institution}(x) \land \langle \mathsf{B} \rangle \langle \bar{\mathsf{D}} \rangle \mathit{LegalSystem}(\mathit{RdS}),$$
$$\mathit{EventD\_J}(x, y) \leftarrow \mathit{HistoricalEvent}(x) \land \langle \bar{\mathsf{D}} \rangle \mathit{Journal}(y),$$

for institutions founded during the Regno di Sardegna period and historical event during the publication period of a journal,

$$\mathit{J\_EstIn}(x, y) \leftarrow \mathit{Journal}(x) \land \langle \mathsf{B} \rangle \langle \bar{\mathsf{D}} \rangle \mathit{LegalSystem}(y),$$
$$\mathit{J\_ClosedIn}(x, y) \leftarrow \mathit{Journal}(x) \land \langle \mathsf{E} \rangle \langle \bar{\mathsf{D}} \rangle \mathit{LegalSystem}(y)$$

relating journals to the legal system when they were established and closed, respectively, and

$$\mathit{CitedAftDeath}(x) \leftarrow \langle \bar{\mathsf{L}} \rangle \mathit{Person}(x) \land \mathit{Article}(y) \land \mathit{cites}(y, x)$$

for persons cited by journal articles after their death. We then used the new intensional predicates to formulate complex queries such as $\mathit{2SysJ}(x)$ given by the formula

$$\exists y, y' \big(\mathit{J\_EstIn}(x, y) \land \mathit{J\_ClosedIn}(x, y') \land (y \neq y')\big),$$

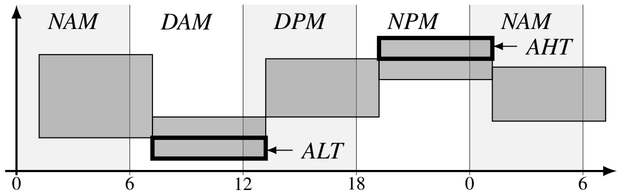
which finds journals published over two or more legal systems. The runtimes (in seconds) of the three systems on our queries are reported in table below:

query	CLASP	DLV	XSB	mat. facts
RdSInst	0.54	7.04	0.36	93 332
EventDuring_J	0.55	6.66	0.12	60 291
2SysJ	0.69	8.03	0.28	60 712
CitedAftDeath	3.34	47.61	0.41	77 718

The total number of zones for the dataset is 38 503, and the last column gives the number of facts in the materialisation.

5.2 Querying Meteorological Data: 2D Case

In the second case study, we used datalog$\mathcal{HS}_n^\square$ to query the US weather data collected by MesoWest[2] and containing sensor measurements of the temperature, precipitation, pressure, etc. by weather stations across the country. We mapped the MesoWest dataset tables to tuples $T^\circ(s, t_1, t_2, \tau_1, \tau_2)$, where s is the ID of a weather station, $[t_1, t_2]$ a time interval, and $[\tau_1, \tau_2]$ the minimal and maximal temperature observed during this interval. The measurements are taken every six hours, so that $t_2 - t_1 = 6$ for all s, but different s may have different $[t_1, t_2]$. We extended this data by tuples of the form NAM(0:00/10.01.2016, 6:00/10.01.2016, _, _) to indicate the night AM hours, and similar tuples *NPM* for night PM, *DAM* for day AM, and *DPM* for day PM hours. Normally, during sunny days, the day AM temperature is not lower than the night AM temperature and the night PM temperature is not higher than the day PM temperature. The picture below shows patterns with abnormally low and high temperatures:



Such patterns can be represented as datalog$\mathcal{HS}_n^\square$ rules:

$$\mathit{ALT}(x) \leftarrow \langle \bar{\mathsf{O}} \rangle_1 \langle \mathsf{U} \rangle_2 \mathit{DAM} \land \langle \bar{\mathsf{B}} \rangle_2 T^\circ(x) \land \langle \bar{\mathsf{A}} \rangle_1 \langle \mathsf{A} \rangle_2 T^\circ(x),$$
$$\mathit{AHT}(x) \leftarrow \langle \bar{\mathsf{O}} \rangle_1 \langle \mathsf{U} \rangle_2 \mathit{NPM} \land \langle \bar{\mathsf{E}} \rangle_2 T^\circ(x) \land \langle \bar{\mathsf{A}} \rangle_1 \langle \bar{\mathsf{A}} \rangle_2 T^\circ(x),$$

where U is the universal interval relation. We use $(\mathbb{Z}, \leq)$ for time and $(\mathbb{R}, <)$ for temperature intervals. The CPU time (in seconds) used by the systems to query for abnormally low temperatures is given in the table below:

days	CLASP	DLV	XSB	mat. facts	zones_1	zones_2
7	0.14	0.81	0.25	60 189	11 026	925
30	5.73	46.47	13.62	1 849 345	199 396	4 513
60	40.98	347.96	66.57	8 910 837	795 691	5 513
90	134.67	1209.88	MEM	24 757 694	1 758 750	8 065
120	315.43	MEM	MEM	51 119 769	3 148 795	10 513
180	MEM	MEM	MEM	–	7 097 028	13 613

MEM means that the system runs out of memory (16 GB). The total number of zones for the time and temperature dimensions is given in the last two columns. The experiments show that standard off-the-shelf systems scale relatively well even with such large numbers of zones. Specialised index data structures and optimisations should boost performance. For example, one could pre-compute zones and relations on them, and use highly scalable triple store RDFox [Nenov et al., 2015] (without the need to implement arithmetic).

6 Conclusions

We designed an ontology language datalog$\mathcal{HS}_n^\square$ with the aim of OBDA over many-dimensional interval-based spatio-temporal data. Datalog$\mathcal{HS}_n^\square$ ontologies are supposed to define complex spatio-temporal predicates (such as those in the examples above), which can then be used in user queries. We proved datalog rewritability of ontology-mediated queries, and demonstrated experimentally, using real-world data, that these rewritings are reasonably well executable by off-the-shelf datalog systems. This seems to be the first working application of a fragment of the Halpern-Shoham logic $\mathcal{HS}$, which is undecidable over any interesting linear order.

Note that the following are disallowed in datalog$\mathcal{HS}_n^\square$: (*i*) operators $\langle \mathsf{R} \rangle$ in the head of the rules, (*ii*) the strict semantics for R over $\mathbb{Z}$ (which can express *metric* constraints such as 'interval of length n'), and (*iii*) punctual intervals $[n, n]$ over $\mathbb{R}$. The addition of (*i*) or (*ii*) would make datalog$\mathcal{HS}_n^\square$ undecidable; the effect of (*iii*) remains unknown.

Propositional datalog$\mathcal{HS}_n^\square$ can be of interest in classical spatial representation as it provides a tractable formalism for reasoning about hyperrectangles. For example, it would be interesting to investigate whether our formalism can be adopted to handle *rotated* rectangles in the world of Angry Birds [Zhang and Renz, 2014], regions of arbitrary shape, and operators based on more expressive relations than Allen's.

[1] For STOria LEgislativa della pubblica amministrazione italiana.
[2] http://mesowest.utah.edu

Acknowledgments. This work has been supported by the EPSRC UK grant and EP/M012670/1 (iTract).

References

[Adorni et al., 2015] Adorni, G.; Maratea, M.; Pandolfo, L.; and Pulina, L. 2015. An ontology for historical research documents. In *Proc. RR*, 11–18.

[Allen, 1983] Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26(11):832–843.

[Artale et al., 2013] Artale, A.; Kontchakov, R.; Wolter, F.; and Zakharyaschev, M. 2013. Temporal description logic for ontology-based data access. In *Proc. IJCAI*, 711–717.

[Artale et al., 2015a] Artale, A.; Kontchakov, R.; Ryzhikov, V.; and Zakharyaschev, M. 2015a. Tractable interval temporal propositional and description logics. In *Proc. AAAI*.

[Artale et al., 2015b] Artale, A.; Kontchakov, R.; Kovtunova, A.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2015b. First-order rewritability of temporal ontology-mediated queries. In *Proc. IJCAI*, 2706–2712.

[Baader, Borgwardt, and Lippmann, 2013] Baader, F.; Borgwardt, S.; and Lippmann, M. 2013. Temporalizing ontology-based data access. In *Proc. CADE-24*, 330–344.

[Balbiani, Condotta, and del Cerro, 2002] Balbiani, P.; Condotta, J.; and del Cerro, L. F. 2002. Tractability results in the block algebra. *J. Log. Comput.* 12(5):885–909.

[Borgwardt, Lippmann, and Thost, 2013] Borgwardt, S.; Lippmann, M.; and Thost, V. 2013. Temporal query answering in description logic DL-Lite. In *Proc. FroCoS*.

[Bresolin et al., 2014] Bresolin, D.; Della Monica, D.; Goranko, V.; Montanari, A.; and Sciavicco, G. 2014. The dark side of interval temporal logic: marking the undecidability border. *Ann. Math. Artif. Intell.* 71(1-3):41–83.

[Bresolin et al., 2016] Bresolin, D.; Kurucz, A.; Muñoz-Velasco, E.; Ryzhikov, V.; Sciavicco, G.; Zakharyaschev, M. 2016. Horn fragments of Halpern-Shoham interval temporal logic. Tech. Rep., arxiv.org/abs/1604.03515

[Cohn et al., 2014] Cohn, A. G.; Li, S.; Liu, W.; and Renz, J. 2014. Reasoning about topological and cardinal direction relations between 2-dimensional spatial objects. *J. Artif. Intell. Res. (JAIR)* 51:493–532.

[Eiter, Krennwallner, and Schneider, 2013] Eiter, T.; Krennwallner, T.; and Schneider, P. 2013. Lightweight spatial conjunctive query answering using keywords. In *Proc. ESWC*, 243–258.

[Gabbay et al., 2003] Gabbay, D.; Kurucz, A.; Wolter, F.; and Zakharyaschev, M., 2003. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier.

[Gebser, Kaufmann, and Schaub, 2012] Gebser, M.; Kaufmann, B.; and Schaub, T. 2012. Conflict-driven answer set solving: From theory to practice. *AIJ* 187–188:52–89.

[Goyal and Egenhofer, 2001] Goyal, R., and Egenhofer, M.. 2001. Similarity of cardinal directions. In *Proc. SSTD*.

[Gutiérrez-Basulto and Klarman, 2012] Gutiérrez-Basulto, V., and Klarman, S. 2012. Towards a unifying approach to representing and querying temporal data in description logics. In *Proc. RR*, 90–105.

[Halpern and Shoham, 1991] Halpern, J., and Shoham, Y. 1991. A propositional modal logic of time intervals. *JACM* 38(4):935–962.

[Kharlamov et al., 2014] Kharlamov, E.; Solomakhina, N.; Özçep, Ö. L.; Zheleznyakov, D.; Hubauer, T.; Lamparter, S.; Roshchin, M.; Soylu, A.; and Watson, S. 2014. How semantic technologies can enhance data access at Siemens Energy. In *Proc. ISWC, Part I*, 601–619.

[Kharlamov et al., 2015] Kharlamov, E.; Hovland, D.; Jiménez-Ruiz, E.; Lanti, D.; Lie, H.; Pinkel, C.; Rezk, M.; Skjæveland, M. G.; Thorstensen, E.; Xiao, G.; Zheleznyakov, D.; and Horrocks, I. 2015. Ontology based access to exploration data at Statoil. In *Proc. ISWC*.

[Klarman and Meyer, 2014] Klarman, S., and Meyer, T. 2014. Querying temporal databases via OWL 2 QL. In *Proc. RR*, 92–107.

[Leone et al., 2006] Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.* 7(3):499–562.

[Lutz and Milicic, 2007] Lutz, C., and Milicic, M. 2007. A tableau algorithm for description logics with concrete domains and general TBoxes. *JAR* 38(1–3):227–259.

[Lutz and Wolter, 2006] Lutz, C., and Wolter, F. 2006. Modal logics of topological relations. *LMCS* 2(2).

[Navarrete et al., 2013] Navarrete, I.; Morales, A.; Sciavicco, G.; and Viedma, M. A. C. 2013. Spatial reasoning with rectangular cardinal relations — the convex tractable subalgebra. *Ann. Math. Artif. Intell.* 67(1):31–70.

[Nenov et al., 2015] Nenov, Y.; Piro, R.; Morik, B.; Horrocks, I.; Wu, Z.; and Banerjee, J. 2015. RDFox: A Highly-Scalable RDF Store. In *Proc. of ISWC*, 3–20.

[Özçep and Möller, 2012] Özçep, Ö., and Möller, R. 2012. Scalable geo-thematic query answering. In *Proc. ISWC*.

[Özçep and Möller, 2014] Özçep, Ö., and Möller, R. 2014. Ontology based data access on temporal and streaming data. In *Reasoning Web*, 279–312.

[Özcep et al., 2013] Özcep, O.; Möller, R.; Neuenstadt, C.; Zheleznyakov, D.; and Kharlamov, E. 2013. A semantics for temporal and stream-based query answering in an OBDA context. Tech. Rep., FP7-318338, EU.

[Poggi et al., 2008] Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *J. Data Semant.* X:133–173.

[Sagonas, Swift, and Warren, 1994] Sagonas, K.; Swift, T.; and Warren, D. S. 1994. XSB as an efficient deductive database engine. *ACM SIGMOD Record* 23:442–453.

[Zhang and Renz, 2014] Zhang, P., and Renz, J. 2014. Qualitative spatial representation and reasoning in Angry Birds: The extended rectangle algebra. In *Proc. KR*, 378–387.