Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS-Report

Preferential Query Answering in the Semantic Web with Possibilistic Networks (Extended Version)

Stefan Borgwardt, Bettina Fazzinga, Thomas Lukasiewicz, Akanksha Shrivastava, and Oana Tifrea-Marciuska

LTCS-Report 16-10

This is an extended version of the article in: Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI'16), pages 994–1000, AAAI Press, 2016.

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Preferential Query Answering in the Semantic Web with Possibilistic Networks (Extended Version)

Stefan Borgwardt
Faculty of Computer Science
Technische Universität Dresden, Germany
stefan.borgwardt@tu-dresden.de

Bettina Fazzinga
ICAR
National Research Council (CNR), Italy
fazzinga@icar.cnr.it

Thomas Lukasiewicz and **Akanksha Shrivastava** and **Oana Tifrea-Marciuska**
Department of Computer Science
University of Oxford, UK
{thomas.lukasiewicz,oana.tifrea}@cs.ox.ac.uk, akanksha.shrivstv@gmail.com

Abstract

In this paper, we explore how ontological knowledge expressed via existential rules can be combined with possibilistic networks (i) to represent qualitative preferences along with domain knowledge, and (ii) to realize preference-based answering of conjunctive queries (CQs). We call these combinations ontological possibilistic networks (OP-nets). We define skyline and k-rank answers to CQs under preferences and provide complexity (including data tractability) results for deciding consistency and CQ skyline membership for OP-nets. We show that our formalism has a lower complexity than a similar existing formalism.

1 Introduction

The abundance of information on the Web requires new personalized information filtering techniques that are able to retrieve resources that best fit users' interests and preferences. These systems should also manage the rapid change of users' preferences and have means for coping with trust and uncertainty on the Web. Moreover, the Web is evolving at an increasing pace towards the so-called Social Semantic Web (or Web 3.0), where classical linked information lives together with ontological knowledge and social interactions of users. While the former may allow for more precise and rich results in search and query answering tasks, the latter can be used to enrich the user profile, and it paves the way to more sophisticated personalized access to information. This requires new techniques for ranking search results, fully exploiting ontological and user-centered data, i.e., user preferences.

Conditional preferences are statements of the form "in the context of c, a is preferred over b", denoted $c\colon a \succ b$ [Ben Amor *et al.*, 2014; Boutilier *et al.*, 2004; Wilson, 2004]. Two preference formalisms that allow for representing such preferences are *possibilistic networks* and *CP-nets*.

Example 1 Bob wants to rent a car and (i) he prefers a new car over an old one, (ii) given he has a new car, he prefers it to be black over not black, and (iii) if he has an old car, he prefers it to be colorful over being black. We have two variables for car type (new (n) or old (o)) and car color (black (b) or colorful (c)), T and C, respectively, such that $Dom(T)=\{n,o\}$ and $Dom(C)=\{b,c\}$. Bob's preferences can be encoded as $\top\colon n \succ o$, $n\colon b \succ c$, and $o\colon c \succ b$. In CP-nets [Boutilier *et al.*, 2004], we have the following ordering of outcomes: $nb \succ nc \succ oc \succ ob$. That is, a new and colorful car is preferred over an old and colorful one, which is not a realistic representation of the given preferences. A more desirable order of outcomes for Bob would be $nb \succ oc \succ nc \succ ob$, which can be induced in possibilistic networks with an appropriate preference weighting in the possibility distribution. ■

In this paper, we propose a novel language for expressing preferences over the Web 3.0 using possibilistic networks. It has lower complexity compared to a similar existing formalism: OCP-theories [Di Noia *et al.*, 2015], which are an integration of Datalog+/$-$ with CP-theories [Wilson, 2004]. This is because deciding dominance in possibilistic networks can be done in polynomial time, while it is PSPACE-complete in CP-theories. Furthermore, every possibilistic network encodes a unique (numerical) ranking on the outcomes, while CP-theories encode a set of (qualitative) total orders on the outcomes. Additionally, our framework allows to specify the relative importance of preferences [Ben Amor *et al.*, 2014].

We choose existential rules in Datalog+/$-$ as ontology language for their intuitive nature, expressive power for rule-based knowledge bases, and the capability of performing query answering. Possibilistic networks are also a simple and natural way of representing conditional preferences and obtaining rankings on outcomes, and can be easily learned from data [Borgelt and Kruse, 2003]. The integration between the two formalisms is tight, as possibilistic network outcomes are constrained by the ontology, but they also dictate the ranking of answers to a query.

The main contributions of this paper are the following:

- We introduce a novel formalism, called ontological possibilistic networks (OP-nets), combining Datalog+/– with possibilistic networks, to encode preferences over atoms.
- We define skyline and k-rank answers for conjunctive queries (CQs) relative to the preferences encoded in OP-nets, and describe how to compute such answers.

• We analyze the complexity of deciding consistency and skyline membership of answers to CQs, for different types of complexity, and provide results for Datalog+/– languages. We also obtain several tractability results. Notably, these results hold for any preference formalism where dominance between two outcomes can be decided in polynomial time.

This is an extended version of [Borgwardt *et al.*, 2016].

2 Preliminaries

We first recall the basics on Datalog+/– [Calì *et al.*, 2012a] and on possibilistic networks.

2.1 Datalog+/–

Databases. Let Δ be a set of *constants*, Δ_N a set of *labeled nulls*, and $\mathcal{V}$ a set of *(regular) variables*. A *term* t is a constant, null, or variable. An *atom* has the form $p(t_1, \ldots, t_n)$, where p is an n-ary predicate, and $t_1, \ldots, t_n$ are terms. Conjunctions of atoms are often identified with the sets of their atoms. An *instance* I is a (possibly infinite) set of atoms $p(\mathbf{t})$, where $\mathbf{t}$ is a tuple of constants and nulls. A *database* D is a finite instance that contains only constants. A *homomorphism* is a substitution $h\colon \Delta \cup \Delta_N \cup \mathcal{V} \to \Delta \cup \Delta_N \cup \mathcal{V}$ that is the identity on Δ. We assume that the reader is familiar with *conjunctive queries (CQs)*. The set of answers to a CQ q over an instance I is denoted $q(I)$. A Boolean CQ (BCQ) q has a positive answer over I, denoted $I \models q$, if $q(I) \neq \emptyset$.

Dependencies. A *tuple-generating dependency (TGD)* (or *existential rule*) σ is a first-order formula $\forall \mathbf{X} \forall \mathbf{Y}\, \varphi(\mathbf{X}, \mathbf{Y}) \to \exists \mathbf{Z}\, p(\mathbf{X}, \mathbf{Z})$, where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \subseteq \mathcal{V}$, $\varphi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms, and $p(\mathbf{X}, \mathbf{Y})$ is an atom; $\varphi(\mathbf{X}, \mathbf{Y})$ is the *body* of σ, denoted $body(\sigma)$, while $p(\mathbf{X}, \mathbf{Z})$ is the *head* of σ, denoted $head(\sigma)$. For clarity, we consider single-atom-head TGDs; however, our results can be extended to TGDs with a conjunction of atoms in the head. An instance I satisfies σ, written $I \models \sigma$, if the following holds: for all homomorphisms h such that $h(\varphi(\mathbf{X}, \mathbf{Y})) \subseteq I$, there exists $h' \supseteq h|_{\mathbf{X} \cup \mathbf{Y}}$, where $h|_{\mathbf{X} \cup \mathbf{Y}}$ is the restriction of h to $\mathbf{X} \cup \mathbf{Y}$, such that $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. A *negative constraint (NC)* ν is a first-order formula of the form $\forall \mathbf{X}\, \varphi(\mathbf{X}) \to \bot$, where $\mathbf{X} \subseteq \mathcal{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms and is called the *body* of ν, denoted $body(\nu)$, and $\bot$ denotes the truth constant *false*. An instance I satisfies ν, written $I \models \nu$, if there is no homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. Given a set Σ of TGDs and NCs, I satisfies Σ, written $I \models \Sigma$, if I satisfies each TGD and NC of Σ.

Datalog+/– Ontologies. A *Datalog+/– ontology* $O = (D, \Sigma)$, where $\Sigma = \Sigma_T \cup \Sigma_{NC}$, consists of a finite database D over Δ, a finite set Σ_T of TGDs, and a finite set Σ_{NC} of NCs. The set of *models* of D and Σ, denoted $mods(D, \Sigma)$, contains all instances I with $I \supseteq D$ and $I \models \Sigma$. The ontology is *consistent* if this set is not empty.

Example 2 Consider the database D in Table 1, modeling the domain of an online car booking system. Moreover,

$$\begin{aligned}\Sigma = \{ & \mathit{offer}(V, P, S) \to \exists C, F, T\ \mathit{specs}(S, C, F, T),\\ & \mathit{offer}(V, P, S) \to \exists R\ \mathit{vendor}(V, R),\\ & \mathit{specs}(S, C, F, T) \to \mathit{color}(C) \wedge \mathit{type}(T),\\ & \mathit{specs}(S, C, F, T) \to \exists N\ \mathit{feature}(F, N),\\ & \mathit{offer}(V, P1, S) \wedge \mathit{offer}(V, P2, S) \to P1 = P2\ \}\end{aligned}$$

Table 1: Database D.

specs

	id	*color*	*feature*	*type*
t_1	s_1	b	f_1	o
t_2	s_2	c	f_2	n
t_3	s_3	c	f_2	o

feature

	id	*name*
t_7	f_1	ac
t_8	f_2	map
t_9	f_3	cd

offer

	vendor	*price*	*specs*
t_4	v_1	30	s_1
t_5	v_1	40	s_2
t_6	v_2	50	s_3

vendor

	id	*review*
t_{10}	v_1	p
t_{11}	v_2	n

says that every offer must have a specification and a vendor. It also says that there cannot be two equivalent offers from the same company with different prices (represented via a special *equality-generating dependency (EGD)*, which can be encoded as an NC [Calì *et al.*, 2012a]). We denote by $\mathbf{t}_1$ the term $specs(s_1, b, f_1, o)$ and by t_1 the tuple (s_1, b, f_1, o). ■

Conjunctive Query Answering. Given a Datalog+/– ontology $O = (D, \Sigma)$, we only consider answers that are true in *all* models of O. Formally, the set of *answers* to a CQ q w.r.t. D and Σ is $ans(q, D, \Sigma) := \bigcap_{I \in mods(D,\Sigma)} \{\mathbf{a} \mid \mathbf{a} \in q(I)\}$. The answer to a BCQ q is *positive*, denoted $D \cup \Sigma \models q$, if $ans(q, D, \Sigma) \neq \emptyset$. The problem of *CQ answering* is the following: given D, Σ, and q as above and a tuple of constants $\mathbf{a}$, decide whether $\mathbf{a} \in ans(q, D, \Sigma)$. Following Vardi's taxonomy (1982), the *combined complexity* of CQ answering is calculated by considering all the components, i.e., the database, the set of dependencies, and the query, as part of the input. The *bounded-arity combined (ba-combined) complexity* is calculated by assuming that the arity of the underlying schema is bounded by a constant. In description logics (DLs) [Bienvenu and Ortiz, 2015], the arity of the underlying schema is always bounded by 2. The *fixed-program combined (fp-combined) complexity* is calculated by considering the set of TGDs and NCs as fixed. Finally, for *data complexity*, we take only the size of the database into account.

2.2 Possibilistic Networks

We now recall possibilistic networks from [Ben Amor *et al.*, 2014], which are a direct counterpart of Bayesian networks from probability theory, the main differences being that possibilities *maximize* (rather than *summarize*) over disjoint events (thus, in the *normalized* case, one often assumes that their maximum (rather than their sum) over all disjoint elementary events is 1), and we measure the degree of potential surprise of an event, as opposed to the degree of its likelihood.

Syntax. Let $\mathcal{X}$ be a finite set of variables with pairwise disjoint, non-empty, finite *domains* $Dom(X)$, $X \in \mathcal{X}$. A possibilistic network Γ defines a possibility distribution over $\mathcal{X}$ using a combination of a graphical and a data component. The former is a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{E}$ is a set of edges encoding conditional (in)dependencies between variables. The data component associates a normalized conditional possibility distribution $\pi(X_i \mid pa(X_i))$ to each $X_i \in \mathcal{X}$, where $pa(X_i)$ is the set of *parents* of X_i in $\mathcal{G}$. The joint distribution over $\mathcal{X}=\{X_1,\ldots,X_n\}$ is then given by the chain rule [Ben Amor *et al.*, 2014; Benferhat *et al.*, 2000]:

$$\pi(X_1,\ldots,X_n) := \bigotimes_{i=1}^{n} \pi(X_i|\, pa(X_i)),$$

where $\otimes$ denotes the product (resp., minimum) in a quantitative (resp., qualitative) setting.

Semantics. A *value* u for a set of variables $U \subseteq \mathcal{X}$ assigns to each $X \in U$ an element $u(X) \in Dom(X)$, and the set of all such values u is called the domain of U, denoted $Dom(U)$. The empty set has a single value, denoted $\top$. Observe that $Dom(X)$ and $Dom(\{X\})$ are isomorphic, and hence the notation is consistent. The values $o \in Dom(\mathcal{X})$ are called *outcomes*. For two outcomes o, o', we say that o *dominates* o' (in Γ), denoted $o \succ o'$, if $\pi(o) > \pi(o')$.

Encoding Conditional Preferences. A *conditional preference* [Ben Amor *et al.*, 2014] has the form $\varphi = u\colon x \succ x'$, where $u \in Dom(U_\varphi)$ for some $U_\varphi \subseteq \mathcal{X}$, and $x, x' \in Dom(X_\varphi)$ for some $X_\varphi \in \mathcal{X} - U_\varphi$. The intention is that, given u and any $t \in Dom(T_\varphi)$, where $T_\varphi = \mathcal{X} - U_\varphi - \{X_\varphi\}$, we prefer x over x'. More formally, the outcome obtained from u, t, and x should dominate the one using x' instead. A *conditional preference theory* $\mathcal{P}$ is a finite set of conditional preferences.

As long as there are no cyclic dependencies between variables or cyclic preferences over the same variable X under the same precondition u, one can encode a conditional preference theory into a possibilistic network [Ben Amor *et al.*, 2014]: The conditional preference φ from above induces several directed edges in the DAG of the possibilistic network, one from each $X \in U_\varphi$ to X_φ. The conditional possibility measure must then be chosen such that $\pi(x|u) > \pi(x'|u)$.

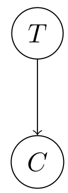
Example 3 Consider again the preference theory from Example 1: $\mathcal{P} = \{\top\colon n \succ o,\ n\colon b \succ c,\ o\colon c \succ b\}$, where $\mathcal{X} = \{T, C\}$, and the outcomes are denoted by nb, nc, ob, and oc. One possibilistic network expressing these conditional preferences is shown in Figure 1, where $\alpha, \beta, \gamma \in (0,1)$. To compare the outcomes, we compute their possibility values (using the quantitative semantics): $\pi(nb) = \pi(b|n) \cdot \pi(n) = 1$, $\pi(oc) = \alpha$, $\pi(nc) = \gamma$, and $\pi(ob) = \alpha \cdot \beta$. To obtain the desired total order $nb \succ oc \succ nc \succ ob$, it thus suffices to choose the values such that $\alpha > \gamma > \alpha \cdot \beta$. ∎

3 OP-Nets

We now introduce ontological possibilistic networks (OP-nets), which extend possibilistic networks by ontologies.

W.l.o.g., the set Δ_N of nulls is the set of all ground terms constructed from the set Δ of constants and a set $\mathcal{F}$ of functions used to skolemize all existential variables in TGDs. Let $O = (D, \Sigma)$ be a Datalog+/– ontology, and $\mathcal{X}_O$ be a finite set of variables, where each $X \in \mathcal{X}_O$ corresponds to a predicate from O, denoted $pred(X)$. Each $Dom(X)$ consists of at least two ground atoms of the form $p(c_1, \ldots c_k)$ with $p = pred(X)$ and $c_1, \ldots, c_k \in \Delta \cup \Delta_N$. Hence, every outcome $o \in Dom(\mathcal{X}_O)$ can be seen as a conjunction of ground atoms. An *ontological possibilistic network (OP-net)* is of the form (O, Γ), where Γ is a possibilistic network over $\mathcal{X}_O$.

Example 4 Consider the OP-net (O, Γ) given by the ontology O of Example 2, the DAG in Figure 2, and the conditional possibility distribution in Table 2. Here, we have



$\pi(n)$	$\pi(o)$
1	α

$\pi(C\|T)$	n	o
b	1	β
c	γ	1

Figure 1: Example 3.

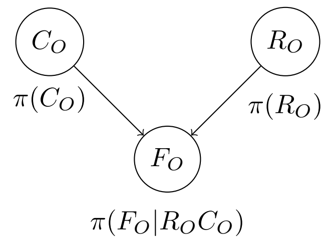


Figure 2: DAG for Example 4.

Table 2: Possibility distribution for Example 4.

$\pi(specs(t_1))$	$\pi(specs(t_2))$	$\pi(specs(t_3))$
1	0.5	0.4

$\pi(vendor(t_{10}))$	$\pi(vendor(t_{11}))$
1	0.4

$\pi(\cdot\|\cdot)$	$\mathbf{t}_1\mathbf{t}_{10}$	$\mathbf{t}_1\mathbf{t}_{11}$	$\mathbf{t}_2\mathbf{t}_{10}$	$\mathbf{t}_2\mathbf{t}_{11}$	$\mathbf{t}_3\mathbf{t}_{10}$	$\mathbf{t}_3\mathbf{t}_{11}$
$feature(t_7)$	1	0.3	0.2	0.2	0.2	0.2
$feature(t_8)$	0.7	0.5	0.7	1	0.4	0.3
$feature(t_9)$	0.5	0.3	0.5	0.3	1	0.2

$\mathcal{X}_O = \{C_O, R_O, F_O\}$ with the domains

$$Dom(C_O) = \{specs(t_1), specs(t_2), specs(t_3)\},$$
$$Dom(F_O) = \{feature(t_7), feature(t_8), feature(t_9)\},$$
$$Dom(R_O) = \{vendor(t_{10}), vendor(t_{11})\}.$$

The possibility distribution could either be learned or derived from explicit preferences, as shown in Section 3.2 below. The possibilities of outcomes are then computed as follows:

$$\pi(C_O R_O F_O) = \pi(F_O|C_O R_O) \otimes \pi(C_O) \otimes \pi(R_O).$$

For example, the outcome o given by $o(C_O) = specs(t_1)$, $o(R_O) = vendor(t_{10})$, and $o(F_O) = feature(t_7)$ encodes the conjunction $\mathbf{t}_1 \wedge \mathbf{t}_{10} \wedge \mathbf{t}_7$ and has the possibility 1. ∎

3.1 Consistency and Dominance

Since outcomes are conjunctions of ground atoms, some outcomes may be inconsistent, and some may be equivalent. This means that we need a notion of consistency for OP-nets.

An outcome o of (O, Γ) is *consistent*, if the ontology $O_o = O \cup \{o(X) \mid X \in \mathcal{X}_O\}$ is consistent. Two outcomes o and o' are *equivalent*, denoted $o \sim o'$, if O_o and $O_{o'}$ have the same models. The dominance $o \prec o'$ w.r.t. Γ is defined as in Section 2.2, and can be decided in polynomial time in the size of Γ by comparing the possibility values of o and o'.

An *interpretation* $\mathcal{I}$ for (O, Γ) is a total preorder over the consistent outcomes in $Dom(\mathcal{X}_O)$. It *satisfies* (or is a *model* of) (O, Γ), if, for all consistent outcomes o and o',

- if $o \prec o'$, then $(o, o') \in \mathcal{I}$ and $(o', o) \notin \mathcal{I}$, and
- if $o \sim o'$, then $(o, o'), (o', o) \in \mathcal{I}$.

An OP-net is *consistent*, if it has at least one consistent outcome, and it has a model.

Theorem 1 *An OP-net (O, Γ) is consistent iff (i) it has a consistent outcome, and (ii) there are no two equivalent consistent outcomes having different possibility values.*

3.2 Encoding Preferences with OP-Nets

In [Di Noia *et al.*, 2015], conditional preferences were generalized to the Datalog+/– setting as follows. Let $Dom^+(X)$ be the set of all atoms $p(t_1, \ldots, t_k)$, where each t_i is a term over Δ, $\mathcal{V}$, and $\mathcal{F}$. An *ontological conditional preference* φ over $\mathcal{X}$ is of the form $v : \xi \succ \xi'$, where

- $v \in Dom^+(U_\varphi)$ for some $U_\varphi \subseteq \mathcal{X}$, and
- $\xi, \xi' \in Dom^+(X_\varphi)$ for some $X_\varphi \in \mathcal{X} - U_\varphi$.

A *ground instance* $v\theta : \xi\theta \succ \xi'\theta$ of φ is obtained via a substitution θ such that $v\theta \in Dom(U_\varphi)$ and $\xi\theta, \xi'\theta \in Dom(X_\varphi)$. Under suitable acyclicity conditions, we can hence find an OP-net (O, Γ) that respects all ground instances of some given ontological conditional preferences in the same way as described in Section 2.2.

Example 5 Consider the ontological conditional preference $\mathit{specs}(I, C, F, o)$: $\mathit{vendor}(V_1, p) \succ \mathit{vendor}(V_2, n)$, which says that for an old car, it is preferable to have a vendor with positive feedback. One ground instance for this preference is $\mathit{specs}(t_1)$: $\mathit{vendor}(t_{10}) \succ \mathit{vendor}(t_{11})$. Thus, we could choose $\pi(\mathit{vendor}(t_{10})|\mathit{specs}(t_1)) = 1$ and $\pi(\mathit{vendor}(t_{11})|\mathit{specs}(t_1)) = \alpha < 1$. ∎

Although possibilistic networks are less expressive than full CP-theories, they allow for a compact encoding of conditional preferences over ground atoms and enable us to show lower complexity bounds (see Section 5).

4 Query Answering under OP-Nets

Using the notions of consistency and dominance, we can define the semantics of query answering, as well as skyline and k-rank answers, in the context of OP-nets. We first formalize query answering for a given consistent OP-net (O, Γ). Since the semantics of OP-nets is similar to that of OCP-theories [Di Noia *et al.*, 2015], the definitions are similar. Let $q(\mathbf{X}) = \exists \mathbf{Y} \phi(\mathbf{X}, \mathbf{Y})$ be a CQ. To extract answers based on the outcomes of a possibilistic network, the atoms in the query must be related to the atoms in conditional preferences. For this purpose, we assume a bijection β from a set of atoms $\phi_\beta(\mathbf{X}, \mathbf{Y}) \subseteq \phi(\mathbf{X}, \mathbf{Y})$ in q to a set of variables of (O, Γ), such that for every atom $p(\mathbf{Z}) \in \phi_\beta(\mathbf{X}, \mathbf{Y})$ there exists some variable X in (O, Γ) with $pred(X) = p$ and $\beta(p(\mathbf{Z})) = X$. We collect in $\mathbf{Y}_\beta$ those quantified variables from $\mathbf{Y}$ that occur in the atoms $\phi_\beta(\mathbf{X}, \mathbf{Y})$, and denote by $\mathbf{Y}_{\overline{\beta}}$ the set of all remaining variables from $\mathbf{Y}$. When ϕ_β is empty, i.e., the query atoms are not related to the preferences, then the answers for the query are standard CQ answers w.r.t. O.

Definition 1 *Let (O, Γ) with $O = (D, \Sigma)$ be a consistent OP-net, $q(\mathbf{X}) = \exists \mathbf{Y} \phi(\mathbf{X}, \mathbf{Y})$ be a CQ, and o be a consistent outcome of (O, Γ). An* answer *to q w.r.t. (O, Γ) and o is a tuple $\mathbf{a}$ over $\Delta \cup \Delta_N$ for which there exists a homomorphism $h\colon \mathbf{X} \cup \mathbf{Y}_\beta \to \Delta \cup \Delta_N$ with (i) $h(\mathbf{X}) = \mathbf{a}$, (ii) $D \cup \Sigma \models \exists \mathbf{Y}_{\overline{\beta}}\, h(\phi(\mathbf{X}, \mathbf{Y}))$, and (iii) $h(a) = o(\beta(a))$ for all $a \in \phi_\beta(\mathbf{X}, \mathbf{Y})$. The set of all such answers is denoted by $ans(q, O, \Gamma, o)$.*

We want to point out that $\exists \mathbf{Y}_{\overline{\beta}}\, h(\phi(\mathbf{X}, \mathbf{Y}))$ is a BCQ that uses elements from $\Delta \cup \Delta_N \cup \mathcal{V}$ as arguments in its atoms. In the following, we call such queries BCQ^{N}s. Since the values of the homomorphism h on $\mathbf{Y}_\beta$ are determined by the outcome o, BCQ^{N} answering w.r.t. o has the same complexity as classical BCQ^{N} answering. k-rank answers are obtained by iteratively computing sets of skyline answers until k answers have been found. However, a tuple may be an answer under more than one outcome. To avoid repetition of answers, we need to keep track of exhausted outcomes and answers.

Definition 2 (Skyline Answer) *A* skyline answer *to q w.r.t. (O, Γ) outside a given set $\mathcal{Y} \subseteq Dom(\mathcal{X}_O)$ of outcomes is any tuple $\mathbf{a} \in ans(q, O, \Gamma, o)$ for some consistent outcome $o \notin \mathcal{Y}$ such that there exists no consistent outcome $o' \notin \mathcal{Y}$ with $o' \succ o$ and $ans(q, O, \Gamma, o') \neq \emptyset$. A* skyline answer *to q w.r.t. (O, Γ) is a skyline answer to q w.r.t. (O, Γ) and $\emptyset$.*

Definition 3 (k-Rank Answer) *A* k-rank answer *to Q w.r.t. (O, Γ) outside $\mathcal{Y}$ and outside a given set of ground tuples S is a sequence $\langle \mathbf{a}_1, \ldots, \mathbf{a}_k \rangle$ such that either $\mathbf{a}_1, \ldots, \mathbf{a}_k$ are k skyline answers to Q w.r.t. (O, Γ) outside $\mathcal{Y}$ that do not belong to S; or $\mathbf{a}_1, \ldots, \mathbf{a}_i$ are all the skyline answers to Q w.r.t. (O, Γ) outside $\mathcal{Y}$ that do not belong to S and $\langle \mathbf{a}_{i+1}, \ldots, \mathbf{a}_k \rangle$ is a $(k - i)$-rank answer to Q w.r.t. (O, Γ) outside $\mathcal{Y} \cup \{o\}$ and $S \cup \{\mathbf{a}_1, \ldots, \mathbf{a}_i\}$, where o is an undominated outcome w.r.t. (O, Γ). A* k-rank answer *to Q w.r.t. (O, Γ) is a k-rank answer to Q w.r.t. (O, Γ) outside $\emptyset$ and $\emptyset$.*

Example 6 Consider the consistent OP-net (O, Γ) of Example 4 and the CQ $q(C, F, T, N) = \exists I\, \mathit{specs}(I, C, F, T) \wedge \mathit{feature}(F, N)$. Then, $\langle$b, f_1, o, ac$\rangle$ is the skyline answer under the consistent outcome $\mathbf{t}_1 \wedge \mathbf{t}_{10} \wedge \mathbf{t}_7$. The skyline answer for $q'(C, T) = \exists N\, q(C, f_2, T, N)$ is $\langle$c, n$\rangle$ with possibility $\pi(\mathbf{t}_2\mathbf{t}_{10}\mathbf{t}_8) = 0.5 \cdot 1 \cdot 0.7 = 0.35$, while the 2-rank answer is $\langle\langle$c, n$\rangle$, $\langle$c, o$\rangle\rangle$. Hence, if feature f_2 is mandatory, the offered new and colorful car is preferred over the old and colorful one, mainly due to positive feedback about vendor v_1. ∎

5 Computational Complexity

We now analyze the computational complexity of the following problems, and delineate some tractable special cases:

Consistency: Is a given OP-net (O, Γ) consistent?

CQ Skyline Membership: For an OP-net (O, Γ), a CQ q, and a tuple $\mathbf{a}$ over $\Delta \cup \Delta_N$, is $\mathbf{a}$ a skyline answer to q w.r.t. (O, Γ)?

5.1 Complexity Classes

We assume some familiarity with the complexity classes AC^0, P, NP, co-NP, Σ_2^{P}, Π_2^{P}, PSPACE, EXP, and 2EXP. The class $\text{D}^{\text{P}} = \text{NP} \wedge \text{co-NP}$ (resp., $\text{D}_2^{\text{P}} = \Sigma_2^{\text{P}} \wedge \Pi_2^{\text{P}}$) is the class of all problems that are the intersection of a problem in NP (resp., Σ_2^{P}) and a problem in co-NP (resp., Π_2^{P}). The class Δ_2^{P} (resp., Δ_3^{P}) is the class of all problems that can be computed in polynomial time with an oracle for NP (resp., Σ_2^{P}). The above complexity classes and their inclusion relationships (which are all currently believed to be strict) are shown below:

$$\text{AC}^0 \subseteq \text{P} \subseteq \text{NP}, \text{co-NP} \subseteq \text{D}^{\text{P}} \subseteq \Delta_2^{\text{P}} \subseteq \Sigma_2^{\text{P}}, \Pi_2^{\text{P}} \subseteq \text{D}_2^{\text{P}} \subseteq \Delta_3^{\text{P}} \subseteq \text{PSPACE} \subseteq \text{EXP} \subseteq 2\text{EXP}.$$

5.2 Decidability Paradigms

The main (syntactic) conditions on TGDs that guarantee the decidability of BCQ answering are guardedness [Calì *et al.*, 2013], stickiness [Calì *et al.*, 2012b], and acyclicity. Interestingly, each such condition has its "weak" counterpart: weak guardedness [Calì *et al.*, 2013], weak stickiness [Calì *et al.*, 2012b], and weak acyclicity [Fagin *et al.*, 2005], respectively.

A TGD σ is *guarded* if an atom $\mathbf{a} \in body(\sigma)$ exists that contains (or "guards") all the body variables of σ. The class of guarded TGDs, denoted G, is defined as the family of all possible sets of guarded TGDs. A key subclass of guarded TGDs are linear TGDs with just one body atom (which is automatically a guard), and the corresponding class is denoted L. *Weakly-guarded* TGDs extend guarded TGDs by requiring only "harmful" body variables to appear in the guard; the associated class is denoted WG. Notice that $\mathsf{L} \subset \mathsf{G} \subset \mathsf{WG}$.

Stickiness is inherently different from guardedness, and its central property is as follows: variables that appear more than once in a body (i.e., join variables) are always propagated (or "stick") to the inferred atoms. A set of TGDs that enjoys the above property is *sticky*, and the corresponding class is denoted S. Weak-stickiness is a relaxation of stickiness where only "harmful" variables are taken into account. A set of TGDs that enjoys weak-stickiness is *weakly-sticky*, and the associated class is denoted WS. Observe that $\mathsf{S} \subset \mathsf{WS}$.

A set Σ of TGDs is *acyclic* if its predicate graph is acyclic, and the underlying class is denoted A. In fact, an acyclic set of TGDs can be seen as nonrecursive. We say Σ is *weakly-acyclic* if its dependency graph enjoys a certain acyclicity condition, which actually guarantees the existence of a finite canonical model; the associated class is denoted WA. We have $\mathsf{A} \subset \mathsf{WA} \subset \mathsf{WS}$.

Another key fragment of TGDs are *full* TGDs, i.e., TGDs without existentially quantified variables, and the corresponding class is denoted F. If full TGDs enjoy linearity, guardedness, stickiness, or acyclicity, then we obtain the classes LF, GF, SF, and AF, respectively. Note that $\mathsf{F} \subset \mathsf{WA}$ and $\mathsf{F} \subset \mathsf{WG}$.

5.3 Overview of Results

Our complexity results for the consistency and the CQ skyline membership problem for OP-nets over the decidable Datalog+/– languages mentioned above are compactly summarized in Tables 3 and 4, respectively. Observe that compared to OCP-theories [Di Noia *et al.*, 2015], we obtain lower complexities for L, LF, AF, G, S, F, GF, SF, WS, and WA in the fp-combined complexity (completeness for D^{P} and Δ_2^{P}, respectively, rather than PSPACE), and for L, LF, AF, S, F, GF, and SF in the ba-complexity (completeness for D_2^{P} and Δ_3^{P}, respectively, rather than PSPACE). Notice also that the complexity theorems below are generic results, applying also to Datalog+/– languages beyond the ones in Tables 3 and 4. Their proofs even apply to arbitrary preference formalisms, as long as dominance between two outcomes can be decided in polynomial time, e.g., rankings computed by Information Retrieval methods [Joachims, 2002].

5.4 Combined Complexity

We first show some generic upper bounds for the complexity of consistency and CQ skyline membership w.r.t. OP-nets.

Theorem 2 *Let $\mathcal{T}$ be a class of OP-nets (O, Γ). If checking non-emptiness of the answer set of a CQ^N w.r.t. O is in a complexity class $\mathcal{C}$, then consistency in $\mathcal{T}$ is in* $\text{NP}^{\mathcal{C}} \wedge \text{co-NP}^{\mathcal{C}}$ *and CQ skyline membership in $\mathcal{T}$ is in* $\text{P}^{\text{NP}^{\mathcal{C}}}$.

If $\mathcal{C} = \text{NP}$ and we consider the fp-combined complexity, then consistency in $\mathcal{T}$ is in D^{P} *and CQ skyline membership in $\mathcal{T}$ is in* Δ_2^{P}.

Proof. To check consistency of OP-nets, it suffices to do the following: guess an outcome o in polynomial time and verify its consistency (for which we need to check whether BCQ$^{\text{N}}$s corresponding to the NCs have a negative answer w.r.t. O_o); then verify that, for all equivalent pairs o, o' of consistent outcomes, their possibility values are the same. Equivalence can be decided in $\mathcal{C}$ (by answering ground atomic BCQ$^{\text{N}}$s), while computing the possibility values is in P.

For CQ skyline membership, we first calculate the possibility value of the skyline using a binary search over the space of possibility values. Starting from the interval $[0, 1]$, in each step we compute one bit of the skyline value, thereby halving the search space, and the maximal precision needed is bounded by the product of the number of variables in the OP-net and the number of bits needed to represent any possibility value of the input. Hence, the search needs only polynomially many steps. In each step, we need to guess an outcome in polynomial time, check that the outcome has a non-empty answer to the considered CQ (in $\mathcal{C}$), check that the outcome is consistent (in co-$\mathcal{C}$), and that its possibility value is in the considered interval (in P).

If $\mathcal{C} = \text{NP}$ and we consider the fp-combined complexity, then BCQ$^{\text{N}}$ answering for ground atomic queries and fixed queries (e.g., those corresponding to the fixed NCs of Σ) w.r.t. O is in P, and hence consistency of OP-nets is decidable in $\text{NP} \wedge \text{co-NP}$. For CQ skyline membership, the needed NP^{NP}-oracle is then actually NP-oracle. This is because consistency of a given outcome can be decided in P, and hence guessing a consistent outcome in the correct possibility interval and checking if it has a non-empty set of answers is overall in NP. □

In particular, for $\mathcal{C} = \text{PSPACE}$, we obtain inclusion in PSPACE for both problems, and the same for any deterministic complexity class above PSPACE. For $\mathcal{C} = \text{NP}$, we get the classes D_2^{P} and Δ_3^{P}. We now provide some matching lower bounds.

Theorem 3 *Let $\mathcal{T}$ be a class of OP-nets (O, Γ). If ground atomic BCQ^N answering w.r.t. O is $\mathcal{C}$-hard, where $\mathcal{C} \supseteq$ PSPACE is a deterministic complexity class, then consistency and CQ skyline membership in $\mathcal{T}$ are $\mathcal{C}$-hard.*

Proof. Note that consistency and equivalence of outcomes are as powerful as checking entailment of arbitrary ground BCQ$^{\text{N}}$s. □

Theorem 4 *For OP-nets whose underlying ontology is defined in a Datalog+/– language $\mathcal{T}$ that allows for NCs, deciding consistency is* D^{P}*-hard in the fp-combined complexity.*

Proof. We give a reduction from the following D^{P}-complete problem: given two propositional formulas $\varphi = c_1 \vee \cdots \vee c_m$ and $\psi = d_1 \wedge \cdots \wedge d_n$ in 3-DNF and 3-CNF, respectively, decide whether φ is a tautology and ψ is satisfiable. W.l.o.g., φ and ψ use different variables. We first construct two OP-nets whose consistency is equivalent to the satisfiability of ψ

Table 3: Combined, ba-combined, fp-combined, and data complexity of deciding consistency for OP-nets with different classes of TGDs.

Class	Comb.	ba-comb.	fp-comb.	Data
L, LF, AF	PSPACE	D_2^P	D^P	in AC^0
G	2EXP	EXP	D^P	P
WG	2EXP	EXP	EXP	EXP
S, SF	EXP	D_2^P	D^P	in AC^0
F, GF	EXP	D_2^P	D^P	P
WS, WA	2EXP	2EXP	D^P	P

Table 4: Combined, ba-combined, fp-combined, and data complexity of deciding CQ skyline membership for OP-nets with different classes of TGDs.

Class	Comb.	ba-comb.	fp-comb.	Data
L, LF, AF	PSPACE	Δ_3^P	Δ_2^P	in AC^0
G	2EXP	EXP	Δ_2^P	P
WG	2EXP	EXP	EXP	EXP
S, SF	EXP	Δ_3^P	Δ_2^P	in AC^0
F, GF	EXP	Δ_3^P	Δ_2^P	P
WS, WA	2EXP	2EXP	Δ_2^P	P

and the validity of ϕ, respectively, and then describe how to combine them into one OP-net.

For every propositional variable x_i of ψ, we use the variable X_i in the possibilistic network with the 2-ary predicate v (where $v(i,t)$ encodes that variable x_i has the truth value t) and the domain $\{v(i,0), v(i,1)\}$. The NC $v(i,0) \wedge v(i,1) \rightarrow \perp$ enforces that all variables have exactly one truth value. In this way, each outcome uniquely represents a variable assignment. Moreover, all variables are independent and all atomic values have possibility 1.

For each disjunction d_j in ψ, we put one tuple $s(j, i_1, t_1, i_2, t_2, i_3, t_3)$ into the database D, where t_l is 0 if x_{i_l} occurs positively in d_j, and t_l is 1 if x_{i_l} occurs negatively in d_j (note the inversion of truth values). We express the non-satisfaction of any disjunction by an NC:

$$v(i_1,t_1) \wedge v(i_2,t_2) \wedge v(i_3,t_3) \wedge s(j, i_1, t_1, i_2, t_2, i_3, t_3) \rightarrow \perp$$

Here, j, i_l, and t_l are variables. Hence, the consistent outcomes of the resulting OP-net uniquely represent the satisfying valuations of ψ. Since there are no equivalent outcomes with different possibility values, the consistency of the OP-net is equivalent to the satisfiability of ψ.

For the second OP-net, we similarly introduce, for each variable y_i of φ, a variable Y_i with domain $\{u(i,0), u(i,1), u(i,2)\}$ and the NC $u(i,0) \wedge u(i,1) \rightarrow \perp$. The goal is to have exactly one additional outcome, namely $u(1,2), \ldots, u(n,2)$, that does not represent a truth value assignment. For this purpose, we use the additional NCs $u(i_1,0) \wedge u(i_2,2) \rightarrow \perp$ and $u(i_1,1) \wedge u(i_2,2) \rightarrow \perp$.

We also use an additional variable W with predicate w and domain $\{w(0), w(1)\}$. The variable X_1 depends on W and we set $\pi(u(1,0)|w(0)) = \pi(u(1,1)|w(0)) = 0.5$. There are no other dependencies between variables and all other (conditional) possibility values are 1. By putting the atoms $w(0)$ and $w(1)$ into the database, we ensure that any two outcomes that differ only in $w(0)$ and $w(1)$ are equivalent. If the rest of these outcomes represents a variable assignment, then these outcomes have different possibility values. However, the equivalent outcomes $(u(1,2), \ldots, u(n,2), w(0))$ and $(u(1,2), \ldots, u(n,2), w(1))$ both have possibility 1.

For each conjunction c_j in φ, we put one tuple $r(j, i_1, t_1, i_2, t_2, i_3, t_3)$ into D, where t_l is 1 iff x_{i_l} occurs positively in c_j, and t_l is 0 iff x_{i_l} occurs negatively in c_j. Additionally, we use the following NC to express the satisfaction of any conjunction:

$$u(i_1,t_1) \wedge u(i_2,t_2) \wedge u(i_3,t_3) \wedge r(j, i_1, t_1, i_2, t_2, i_3, t_3) \rightarrow \perp$$

This ensures that φ is a tautology iff all equivalent outcomes with different possibility values are inconsistent. Moreover, the outcome $(u(1,2), \ldots, u(1,2), w(0))$ is clearly consistent since the above NC is not applicable.

To combine these two OP-nets, we take the union of the constructed variables, possibilistic networks, databases, and ontologies. As they are formulated over disjoint signatures, we obtain a single OP-net that is consistent iff both the original OP-nets are consistent, which is the case iff ψ is satisfiable and φ is a tautology. Finally, note that the NCs we have constructed are independent of the actual form of φ and ψ. □

Theorem 5 *For OP-nets whose underlying ontology is defined in a Datalog+/– language $\mathcal{T}$ that allows for NCs, deciding consistency is D_2^P-hard in the ba-combined complexity.*

Proof. We reduce the validity problem of $\Phi \wedge \Psi$, where $\Phi = \exists \mathbf{X} \forall \mathbf{Y} \varphi(\mathbf{X}, \mathbf{Y})$, $\Psi = \forall \mathbf{Z} \exists \mathbf{Y} \psi(\mathbf{Z}, \mathbf{Y})$, $\phi(\mathbf{X}, \mathbf{Y})$ is a propositional 3-DNF formula, and $\psi(\mathbf{Z}, \mathbf{Y})$ is a propositional 3-CNF formula. W.l.o.g., $\mathbf{X}$ and $\mathbf{Z}$ are disjoint. As in the previous proof, we construct two disjoint OP-nets for each of the subproblems, and then combine them.

For the first OP-net, for each variable $x_i \in \mathbf{X}$, we again use a variable X_i with the predicate v, the domain $\{v(1,0), v(i,1)\}$, and the NC $v(i,0) \wedge v(i,1) \rightarrow \perp$. We illustrate the encoding of the conjunctions of $\varphi(\mathbf{X}, \mathbf{Y})$ on the example $c_j = x_3 \wedge \neg x_5 \wedge y_6$ with $x_3, x_5 \in \mathbf{X}$ and $y_6 \in \mathbf{Y}$. We use the variables $C_{j,0}, C_{j,1}$, where $\mathit{pred}(C_{j,0}) = \mathit{pred}(C_{j,1}) = s_j$ is a binary predicate that encodes the truth value of c_j in dependence of that of y_6, i.e., $s_j(t_j, v_6)$ expresses that (under fixed values for the variables X_i), the truth value v_6 of y_6 implies the truth value t_j of c_j. Their domains are $\mathit{Dom}(C_{j,0}) = \{s_j(0,0), s_j(1,0)\}$ and $\mathit{Dom}(C_{j,1}) = \{s_j(0,1),\ s_j(1,1)\}$. We constrain this predicate using the following NCs:

$$s_j(0, v_6) \wedge s_j(1, v_6) \rightarrow \perp$$
$$v(3,1) \wedge v(5,0) \wedge s_j(0,1) \rightarrow \perp$$
$$s_j(1,0) \rightarrow \perp$$
$$v(3,0) \wedge s_j(1, v_6) \rightarrow \perp$$
$$v(5,1) \wedge s_j(1, v_6) \rightarrow \perp$$

In this order, they express that

- c_j cannot be true and false at the same time, regardless of the value of y_6;

- if x_3 and y_6 are true and x_5 is false, then c_j is true;
- if y_6 is false, then c_j is false;
- if x_3 is false, then c_j is false; and
- if x_5 is true, then c_j is false.

We similarly encode all other conjunctions c_j in $\varphi(\mathbf{X}, \mathbf{Y})$ using predicates s_j whose arity reflects the cardinality of $\mathbf{Y}_{c_j}$ (the set of all variables from $\mathbf{Y}$ that occur in c_j). For example, if $\mathbf{Y}_{c_j} = \{y_1, y_2, y_3\}$, then $s_j(1, 0, 0, 0)$ expresses that c_j is true if all three variables are false. Correspondingly, we have to use 8 variables: $C_{j,0,0,0}, \ldots, C_{j,1,1,1}$.

Finally, we use the NC $\bigwedge_j s_j(0, \mathbf{Y}_{c_j}) \rightarrow \bot$ to enforce that, given an outcome fixing the truth value assignment for $\mathbf{X}$, the existence of a single assignment for $\mathbf{Y}$ that falsifies all conjunctions in $\varphi(\mathbf{X}, \mathbf{Y})$ is equivalent to the inconsistency of the outcome. Hence, the existence of a consistent outcome is equivalent to the validity of Φ. Moreover, no two equivalent consistent outcomes with different possibility values exist.

For the second OP-net, we again use a similar construction. We introduce the variables Z_i for each variable $z_i \in \mathbf{Z}$, the corresponding domains $\{u(i, 0), u(i, 1), u(i, 2)\}$, and the NCs $u(i, 0) \wedge u(i, 1) \rightarrow \bot$, $u(i_1, 0) \wedge u(i_2, 2) \rightarrow \bot$, and $u(i_1, 1) \wedge u(i_2, 2) \rightarrow \bot$, in addition to the variable W with predicate w and domain $\{w(0), w(1)\}$. As before, Z_1 depends on W, $\pi(u(1, 0)|w(0)) = \pi(u(1, 1)|w(0)) = 0.5$, and all other possibility values are 1. We also add the atoms $w(0)$ and $w(1)$ to the database.

We encode the disjunctions in $\psi(\mathbf{Z}, \mathbf{Y})$ similarly as above; e.g., $d_j = z_3 \vee \neg z_5 \vee y_6$ with $z_3, z_5 \in \mathbf{Z}$ and $y_6 \in \mathbf{Y}$ is expressed via the predicate r_j, the variables $D_{j,0}$, $D_{j,1}$ with $\mathit{Dom}(D_{j,0}) = \{r_j(0, 0),\ r_j(1, 0),\ r_j(2, 0)\}$, $\mathit{Dom}(D_{j,1}) = \{r_j(0, 1), r_j(1, 1), r_j(2, 1)\}$, and the following NCs:

$$
\begin{aligned}
& r_j(0, v_6) \wedge r_j(1, v_6) \rightarrow \bot \\
& r_j(0, v_6) \wedge r_j(2, v_6') \rightarrow \bot \\
& r_j(1, v_6) \wedge r_j(2, v_6') \rightarrow \bot \\
& u(i, 0) \wedge r_j(2, v_6) \rightarrow \bot \\
& u(i, 1) \wedge r_j(2, v_6) \rightarrow \bot \\
& u(i, 2) \wedge r_j(0, v_6) \rightarrow \bot \\
& u(i, 2) \wedge r_j(1, v_6) \rightarrow \bot \\
& u(3, 0) \wedge u(5, 1) \wedge r_j(1, 0) \rightarrow \bot \\
& r_j(0, 1) \rightarrow \bot \\
& u(3, 1) \wedge r_j(0, v_6) \rightarrow \bot \\
& u(5, 0) \wedge r_j(0, v_6) \rightarrow \bot \\
& \textstyle\bigwedge_j r_j(1, \mathbf{Y}_{d_j}) \rightarrow \bot
\end{aligned}
$$

The last NC ensures that, given any outcome representing a truth value assignment for $\mathbf{Z}$, the existence of an assignment for $\mathbf{Y}$ that satisfies all disjunctions in $\psi(\mathbf{Z}, \mathbf{Y})$ is equivalent to inconsistency of the outcome. The inconsistency of all equivalent outcomes with different possibility values is hence equivalent to the validity of Ψ. Note that we again have exactly two consistent outcomes, namely the ones that do not represent a truth value assignment for $\mathbf{Z}$, which differ only in $w(0)$ and $w(1)$. Since they both have the same possibility, validity of Ψ is even equivalent to the consistency of the constructed OP-net.

We can now again combine these two OP-nets into one whose consistency is equivalent to the original problem. Notice also that the used predicates' arity is bounded by 4. $\square$

Theorem 6 *For OP-nets whose underlying ontology is defined in a Datalog+/– language $\mathcal{T}$ that allows for NCs, deciding CQ skyline membership is Δ_2^{P}-hard in the fp-combined complexity.*

Proof. We give a polynomial transformation from the following Δ_2^{P}-complete problem [Krentel, 1988] (cf. Theorem 5.9). Given a satisfiable 3-CNF formula $\psi = d_1 \wedge \cdots \wedge d_m$ over the variables $x_1, \ldots, x_n$, decide whether the lexicographically maximal truth assignment satisfying ψ maps x_n to *true*.

For every propositional variable x_i, we use the variable X_i in the possibilistic network with the 2-ary predicate v (where $v(i, t)$ encodes that variable x_i has the truth value t) and the domain $\{v(i, 0), v(i, 1)\}$. The NC $v(i, 0) \wedge v(i, 1) \rightarrow \bot$ enforces that all variables have exactly one truth value. Moreover, all variables are independent, and the possibility values of $v(i, 0)$ and $v(i, 1)$ are $2^{-2^{n-i}}$ and 1, respectively. This ensures that the resulting possibilistic order on the outcomes coincides with the lexicographic order.

For each disjunction d_j in ψ, we put one tuple $s(j, i_1, t_1, i_2, t_2, i_3, t_3)$ into the database D, where t_l is 0 if x_{i_l} occurs positively in d_j, and t_l is 1 if x_{i_l} occurs negatively in d_j (note the inversion of truth values). As in the previous proof, we express the non-satisfaction of any disjunction by an NC:

$$
v(i_1, t_1) \wedge v(i_2, t_2) \wedge v(i_3, t_3) \wedge \\
r(j, i_1, t_1, i_2, t_2, i_3, t_3) \rightarrow \bot
$$

Hence, the consistent outcomes uniquely represent the satisfying valuations of ψ. Furthermore, $v(n, 1)$ is in the skyline for the CQ $v(n, t)$ iff $v(n, 1)$ holds in the consistent outcome with the highest possibility. As argued above, the latter is in turn equivalent to x_n being true in the lexicographically maximal satisfying truth assignment. $\square$

Theorem 7 *For OP-nets whose underlying ontology is defined in a Datalog+/– language $\mathcal{T}$ that allows for NCs, deciding CQ skyline membership is Δ_3^{P}-hard in the ba-combined complexity.*

Proof. We give a polynomial transformation from the following Δ_3^{P}-hard problem [Krentel, 1992]. Given a valid QBF $\Phi = \exists \mathbf{X} \forall \mathbf{Y} \varphi(\mathbf{X}, \mathbf{Y})$, where $\varphi(\mathbf{X}, \mathbf{Y}) = c_1 \vee \cdots \vee c_m$ is in 3-DNF, decide whether the lexicographically maximal truth assignment for $\mathbf{X} = \{x_1, \ldots, x_n\}$ that satisfies $\forall \mathbf{Y} \varphi(\mathbf{X}, \mathbf{Y})$ maps x_n to *true*. For this, we combine the proofs of Theorems 5 and 6.

We use the same encoding of the variables in $\mathbf{X}$ and conjunctions in $\varphi(\mathbf{X}, \mathbf{Y})$ using the variables $X_i, C_{j,t,\ldots}$ and predicates v, s_j, and the NCs from the proof of Theorem 5. Hence, the consistent outcomes uniquely represent those assignments for $\mathbf{X}$ that satisfy $\forall \mathbf{Y} \varphi(\mathbf{X}, \mathbf{Y})$. Moreover, since Φ is valid, there must exist at least one consistent outcome, and there do not exist two different outcomes that are equivalent. This means that the constructed OP-net is consistent.

Using the possibility distribution from the proof of Theorem 6, namely $\pi(v(i, 0)) = 2^{-2^{n-i}}$ and $\pi(v(i, 1)) = 1$, we can again identify the lexicographically maximal assignment via the outcome with the highest possibility value. Thus, the CQ $v(n,t)$ has $v(n,1)$ as a skyline answer iff x_n is true in the lexicographically maximal truth assignment for $\mathbf{X}$ that satisfies $\forall \mathbf{Y} \varphi(\mathbf{X},\mathbf{Y})$. □

From the known complexity results for ontology languages of the Datalog+/– family (see, e.g., [Di Noia *et al.*, 2015]), we obtain the complexity results w.r.t. combined, ba-combined, and fp-combined complexity listed in Tables 3 and 4.

5.5 Data Complexity

We now show that tractability in data complexity for deciding consistency and CQ skyline membership for OP-nets carries over from classical BCQ^N answering. Here, data complexity means that Σ and the variables and possibility distributions of Γ are both fixed, while D is part of the input.

Theorem 8 *Let $\mathcal{T}$ be a class of OP-nets (O,Γ) for which BCQ^N answering in O is possible in polynomial time (resp., in* AC^0*) in the data complexity. Then, deciding consistency and CQ skyline membership in $\mathcal{T}$ is possible in polynomial time (resp., in* AC^0*) in the data complexity.*

Proof. We can decide these problems in the same way as in the proof of Theorem 2. Under data complexity assumptions, however, we can enumerate all outcomes in constant time, or incorporate them into a constant-depth circuit, without affecting the complexity class. Since both P and AC^0 are closed under complementation and conjunction, it is easy to construct a P-Turing machine or an AC^0-circuit to check consistency of (O,Γ). Moreover, the binary search used for CQ skyline membership also needs only constantly many steps now, and hence can be encoded into the Turing machine or circuit as well. □

As a corollary, we obtain the data tractability results listed in the last column of Tables 3 and 4. Note that all memberships in P are also P-hard, due to a standard reduction of propositional logic programming to guarded full TGDs. These results do not apply to WG, where BCQ^N answering is data complete for EXP, and data hardness holds even for ground atomic BCQs; however, data completeness for EXP can be proved similarly to Theorems 2 and 3.

6 Related Work

Preferences have long been studied in many disciplines, prominently in philosophy, databases, and AI. In philosophy, the research mainly deals with *preference logics*, where preferences are usually expressed over mutually exclusive worlds like truth assignments to formulas and the research focus lies on axiomatizations. One of the earliest works on modeling preferences in databases is [Lacroix and Lavency, 1987], which extends the relational calculus with preference modeling mechanisms for query answering. Since then, many approaches go in this direction [Stefanidis *et al.*, 2011]. In AI, preference modeling is more concerned with compact representation and computational issues. In this regard, [Bienvenu *et al.*, 2010] bridges the gaps between the two streams of preference modeling and suggests that most AI formalisms are fragments of a prototypical preference logic. CP-nets [Boutilier *et al.*, 2004] are one of the most widely used preference representation languages. Possibilistic logic [Benferhat *et al.*, 2001; Dubois and Prade, 2004] has recently also been discovered as a useful tool, and a lot of work has been done in bridging the differences between possibilistic logic and CP-nets [Dubois *et al.*, 2013]. More recently, possibilistic networks [Ben Amor *et al.*, 2014] have been advocated as a natural encoding of preferences. Having some computational and expressive benefits over CP-nets, possibilistic networks look very promising.

The work closest in spirit to this paper is perhaps [Di Noia *et al.*, 2015], which is based on CP-theories [Wilson, 2004]. CP-theories admit preferences of the type "given c, we prefer a to b, irrespective of the value of W", which realize a weakening of the ceteris paribus condition. Although possibilistic networks do not allow for such indifference between values of some variables W, they are also based on a weakening of the ceteris paribus condition. This is because possibilistic networks represent total preorders of outcomes, based on the given conditional preferences and the choice of their relative importance (via their conditional possibility), which can only be expressed in possibilistic networks. CP-theories (and CP-nets), in contrast, can handle to some extent cyclic preference dependency graphs, while possibilistic networks assume that these graphs are acyclic. Possibilistic networks can also express certain types of preferences that CP-theories cannot, as we have seen in Example 1. Clearly, all these semantic properties of possibilistic networks, compared to CP-theories (and CP-nets), are inherited by OP-nets. Moreover, OP-nets sometimes have lower combined, ba-combined, and fp-combined complexity than OCP-theories [Di Noia *et al.*, 2015], since consistency and dominance in CP-theories are already PSPACE-hard problems [Goldsmith *et al.*, 2008]. In summary, OP-nets have advantages over OCP-theories, as they are computationally less expensive, while retaining most of the expressivity, and even allow representing some preferences that cannot be represented in OCP-theories.

Other combinations of Semantic Web formalisms with preference representation and reasoning include the work by Lukasiewicz and Schellhase [2007], which presents a system to rank-order ontologically annotated objects, using a ranking function based on conditional preferences. Lukasiewicz *et al.* [2013] focus on preference-based query answering on ontological data by extending Datalog+/− with preference management capabilities, called *PrefDatalog+/−*. Preferences in PrefDatalog+/− have the form of general first-order sentences, and so have a higher complexity. Data tractability results also hold only for disjunctions of atomic queries and not conjunctive queries. Di Noia *et al.* [2013] use ontological axioms to restrict CP-net outcomes. In information retrieval, in [Boubekeur *et al.*, 2007], Wordnet is used to add semantics to CP-net variables. Possibilistic networks have also been used for information retrieval in [Boughanem *et al.*, 2009], where the possibility and the necessity measure are used to evaluate (i) the extent to which a given document is relevant to a query, and (ii) the reasons of eliminating irrelevant documents.

7 Summary and Outlook

We have introduced OP-nets, which are a novel combination of Datalog+/– ontologies with possibilistic networks. We have then defined skyline and k-rank answers for this framework. Furthermore, we have provided a host of complexity (including several data tractability) results for deciding consistency and CQ skyline membership for OP-nets. Due to the lower (polynomial) complexity of dominance testing in possibilistic networks, compared to CP-theories, several resulting complexities for OP-nets are lower than for OCP-theories.

Note that the complexity results and these lower complexities are actually independent of possibilistic networks; they hold for all rankings on outcomes where each rank can be computed in polynomial time. For example, they are also applicable to combinations of Datalog+/– with rankings computed by standard Information Retrieval and Machine Learning approaches [Joachims, 2002].

Interesting topics of ongoing and future research include the implementation and experimental evaluation of the presented approach, as well as a generalization based on possibilistic logic [Benferhat *et al.*, 2002] to gain more expressivity and some new features towards nonmonotonic reasoning and belief revision [Ben Amor *et al.*, 2014]; moreover, an apparent relation between possibilistic logic and quantitative choice logic [Benferhat *et al.*, 2004] may also be exploited.

Acknowledgments. This work was supported by the UK EPSRC grants EP/J008346/1, EP/L012138/1, and EP/M025268/1, by a Google European Doctoral Fellowship, and by the grant PON02_00563_3470993 ("VINCENTE").

References

[Ben Amor *et al.*, 2014] N. Ben Amor, D. Dubois, H. Gouider, and H. Prade. Possibilistic networks: A new setting for modeling preferences. In *Proc. SUM*, pp. 1–7, 2014.

[Benferhat *et al.*, 2000] S. Benferhat, D. Dubois, L. Garcia, and H. Prade. Directed possibilistic graphs and possibilistic logic. In *Information, Uncertainty and Fusion*, pp. 365–379. Springer, 2000.

[Benferhat *et al.*, 2001] S. Benferhat, D. Dubois, and H. Prade. Towards a possibilistic logic handling of preferences. *Appl. Intell.*, 14(3):303–317, 2001.

[Benferhat *et al.*, 2002] S. Benferhat, D. Dubois, L. Garcia, and H. Prade. On the transformation between possibilistic logic bases and possibilistic causal networks. *Int. J. Approx. Reason.*, 29(2):135–173, 2002.

[Benferhat *et al.*, 2004] S. Benferhat, G. Brewka, and D. Le Berre. On the relationship between qualitative choice logic and possibilistic logic. In *Proc. IPMU*, pp. 951–957, 2004.

[Bienvenu and Ortiz, 2015] M. Bienvenu and M. Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web*, pp. 218–307. Springer, 2015.

[Bienvenu *et al.*, 2010] M. Bienvenu, J. Lang, and N. Wilson. From preference logics to preference languages, and back. In *Proc. KR*, pp. 414–424, 2010.

[Borgelt and Kruse, 2003] C. Borgelt and R. Kruse. Operations and evaluation measures for learning possibilistic graphical models. *Artif. Intell.*, 148(1/2):385-418, 2003.

[Borgwardt *et al.*, 2016] S. Borgwardt, B. Fazzinga, T. Lukasiewicz, A. Shrivastava, and O. Tifrea-Marciuska. Preferential query answering in the Semantic Web with possibilistic networks. In *Proc. IJCAI*, pp. 994–1000, 2016.

[Boubekeur *et al.*, 2007] F. Boubekeur, M. Boughanem, and L. Tamine-Lechani. Semantic information retrieval based on CP-nets. In *Proc. FUZZ-IEEE*, pp. 1–7, 2007.

[Boughanem *et al.*, 2009] M. Boughanem, A. Brini, and D. Dubois. Possibilistic networks for information retrieval. *Int. J. Approx. Reason.*, 50(7):957–968, 2009.

[Boutilier *et al.*, 2004] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.*, 21:135–191, 2004.

[Calì *et al.*, 2012a] A. Calì, G. Gottlob, and T. Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.

[Calì *et al.*, 2012b] A. Calì, G. Gottlob, and A. Pieris. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.*, 193:87–128, 2012.

[Calì *et al.*, 2013] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive integrity constraints. *J. Artif. Intell. Res.*, 2013.

[Di Noia *et al.*, 2013] T. Di Noia, T. Lukasiewicz, and G. I. Simari. Reasoning with semantic-enabled qualitative preferences. In *Proc. SUM*, pp. 374–386, 2013.

[Di Noia *et al.*, 2015] T. Di Noia, T. Lukasiewicz, M. V. Martinez, G. I. Simari, and O. Tifrea-Marciuska. Combining existential rules with the power of CP-theories. In *Proc. IJCAI*, pp. 2918–2925, 2015.

[Dubois and Prade, 2004] D. Dubois and H. Prade. Possibilistic logic: A retrospective and prospective view. *Fuzzy Set. Syst.*, 144(1):3–23, 2004.

[Dubois *et al.*, 2013] D. Dubois, H. Prade, and F. Touazi. Conditional preference nets and possibilistic logic. In *Proc. ECSQARU*, pp. 181–193, 2013.

[Fagin *et al.*, 2005] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[Goldsmith *et al.*, 2008] J. Goldsmith, J. Lang, M. Truszczynski, and N. Wilson. The computational complexity of dominance and consistency in CP-nets. *J. Artif. Intell. Res.*, 33:403–432, 2008.

[Joachims, 2002] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. SIGKDD*, pp. 133–142, 2002.

[Krentel, 1988] M. W. Krentel. The complexity of optimization problems. *J. Comput. Syst. Sci.*, 36(3):490–509, 1988.

[Krentel, 1992] M. W. Krentel. Generalizations of Opt P to the polynomial hierarchy. *Theor. Comput. Sci.*, 97:183–198, 1992.

[Lacroix and Lavency, 1987] M. Lacroix and P. Lavency. Preferences: Putting more knowledge into queries. In *Proc. VLDB*, pp. 217–225, 1987.

[Lukasiewicz and Schellhase, 2007] T. Lukasiewicz and J. Schellhase. Variable-strength conditional preferences for ranking objects in ontologies. *J. Web Sem.*, 5(3):180–194, 2007.

[Lukasiewicz *et al.*, 2013] T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Preference-based query answering in Datalog+/– ontologies. In *Proc. IJCAI*, pp. 1017–1023, 2013.

[Nin and Torra, 2010] J. Nin and V. Torra. Possibilistic reasoning for trust-based access control enforcement in social networks. In *Proc. FUZZ-IEEE*, pp. 1–6, 2010.

[Stefanidis *et al.*, 2011] K. Stefanidis, G. Koutrika, and E. Pitoura. A survey on representation, composition and application of preferences in database systems. *ACM T. Database Syst.*, 36(3):19, 2011.

[Vardi, 1982] M. Y. Vardi. The complexity of relational query languages. In *Proc. STOC*, pp. 137–146, 1982.

[Wilson, 2004] N. Wilson. Extending CP-nets with stronger conditional preference statements. In *Proc. AAAI*, pp. 735–741, 2004.