

## Chapter 2

# Models for Representing User Preferences

The study of how preferences can be modeled and leveraged in different kinds of application domains has been the subject of a wide range of works in many different disciplines. Most relevant to our work are the developments in the computer science literature, and in particular the incorporation of preferences into different kinds of query answering systems. Even within this specialized application, several approaches have been developed centered around different goals. In this chapter, we provide a brief overview of the approaches that are most relevant to our work. Before doing so, we present some basic notation that will be used here and in following chapters.

### 2.1 Basic Definitions and Notation

In general, we will refer to preferences over *elements*, which—as we will see below—can be either simple objects or more complex ones. In the following, we refer to a general set  $S$  of such elements, keeping their description abstract. Preferences can then be abstracted simply as binary relations representing orders; we use the symbol  $>$  to denote such relations. Different kinds of relations arise according to what subset of the following properties they satisfy [37]:

- Reflexive:  $a > a$ , for all  $a \in S$ .  
If  $R$  does not satisfy this property, we say that it is *non-reflexive* (not to be confused with *irreflexive*).
- Irreflexive:  $a \not> a$ , for all  $a \in S$ .
- Symmetric: If  $a > b$ , then  $b > a$ , for all  $a, b \in S$ .  
If  $R$  does not satisfy this property, we say that it is *non-symmetric* (not to be confused with *asymmetric* or *antisymmetric*).
- Asymmetric: If  $a > b$ , then  $b \not> a$ , for all  $a, b \in S$ .
- Antisymmetric: If  $a > b$  and  $b > a$ , then  $a = b$ , for all  $a, b \in S$ .

- **Transitive:** If  $a \succ b$  and  $b \succ c$ , then  $a \succ c$ , for all  $a, b, c \in S$ .  
If  $R$  does not satisfy this property, we say that it is *non-transitive* (not to be confused with *negatively transitive*).
- **Negatively transitive:** If  $a \succ c$ , then  $a \succ b$  or  $b \succ c$ , for all  $a, b, c \in S$ .
- **Strongly Complete:**  $a \succ b$  or  $b \succ a$ , for all  $a, b \in S$ .
- **Complete:**  $a \succ b$  or  $b \succ a$ , for all  $a, b \in S$  such that  $a \neq b$ .

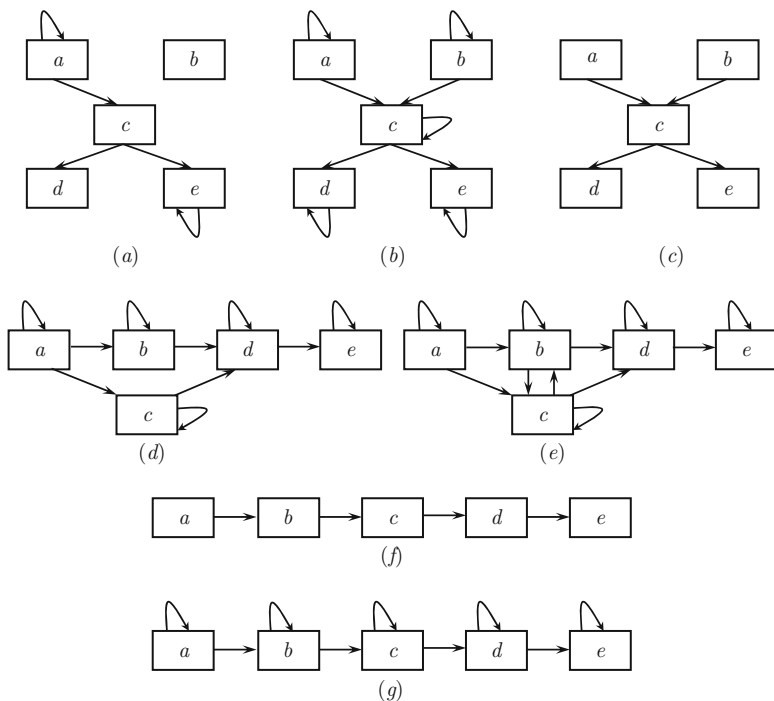
Clearly, these properties are not completely independent of each other; for instance, if both transitivity and symmetry hold, then so does reflexivity (if  $a \succ b$  and  $b \succ a$ , then  $a \succ a$ ). There are therefore different characterizations for the different kinds of relations that usually appear in the literature. The main ones are the following [37]:

- **Order:** transitive and antisymmetric.
- **Partial Order:** reflexive, transitive, and antisymmetric.
- **Strict Partial Order:** transitive and asymmetric. Intuitively, the relation can be represented by a directed acyclic graph.
- **Linear Order:** transitive, asymmetric, and strongly complete. Intuitively, all elements in the set can be sorted in ascending or descending order. These relations are also sometimes referred to as *total* orders.
- **Strict Linear Order:** transitive, asymmetric, and complete.
- **Weak Order:** transitive and strongly complete. Intuitively, these orders are like linear orders that allow “ties”, and all pairs of elements that are “tied at the same level” are in the relation.
- **Strict Weak Order:** asymmetric and negatively transitive. Intuitively, these orders are like linear orders that allow “ties”.

To better understand how this naming convention works, we can informally say that *strict* refers to irreflexive relations, *weak* refers to allowing “ties”, and *partial* means that there are incomparable pairs of elements. A graphical representation of the different kinds of orders is shown in Fig. 2.1.

In the rest of this chapter (and book in general), we will assume that a “*preference relation*” (denoted with the symbol  $\succ$ , as above), is defined over a set  $S$  of elements and is a strict partial order (SPO) over  $S$  (so, a relation that can be visualized as a directed acyclic graph)—these are generally considered to be the minimal requirements for a reasonable preference relation. If  $a \succ b$ , we say that  $a$  is *preferred* to  $b$ . The *indifference relation* induced by  $\succ$ , denoted with  $\sim$ , is defined as follows: for any  $a, b \in S$ ,  $a \sim b$  iff  $a \not\succ b$  and  $b \not\succ a$ .

A *stratification* of  $S$  relative to  $\succ$  is an ordered sequence  $S_1, \dots, S_k$ , where each  $S_i$  is a maximal subset of  $S$  such that for every  $a \in S_i$ , there is no  $b \in \bigcup_{j=i}^k S_j$  with  $b \succ a$ . Intuitively,  $S_1$  contains the most preferred elements in  $S$  relative to  $\succ$ ; then,  $S_2$  contains the most preferred elements of  $S - S_1$ , and so on. Stratifications always exist, are unique, and are a partition of  $S$ . Elements in stratum  $S_i$  have *rank*  $i$ . The rank of an element  $a \in S$  relative to  $\succ$  is denoted as  $\text{rank}(a, \succ)$ .



**Fig. 2.1** A graphical depiction of the different kinds of orders, where *transitive edges are implicit*. (a) Order; (b) Partial order; (c) Strict partial order; (d) Weak order; (e) Strict weak order; (f) Strict linear order; and (g) Linear order

**Qualitative vs. Quantitative Preferences Models** One of the most basic distinctions between preference models and the underlying preference relations that they represent is that of *qualitative* vs. *quantitative*. Essentially, quantitative models are equivalent to the assignment of a numerical score to each element in the set of interest—depending on whether ties are allowed or not, this gives rise to either a strict weak order or a linear order, respectively.

On the other hand, qualitative models allow preferences to be expressed by referring to different aspects of the elements, such as location and size in the case of apartments. Since there may exist, for instance, a large apartment in a bad location and another one that is small but in a good location, the result can be a relation with incomparable elements. Clearly, qualitative models are richer than quantitative ones, since the relations that the former are capable of representing strictly contain those of the latter.

**Individual vs. Group Preferences** Another important aspect that distinguishes approaches to preference modeling is whether they aim to represent single users or groups. See Sect. 2.4 for a brief discussion and references to relevant literature on this aspect.

In the rest of this chapter we will provide a short overview of some of the models that have been developed in the literature, dividing them into two main groups: preferences over simple elements like database tuples or ground atoms, and over more complex ones such as truth assignments to formulas, or possible worlds in some theory.

## 2.2 Preferences *à la* Databases

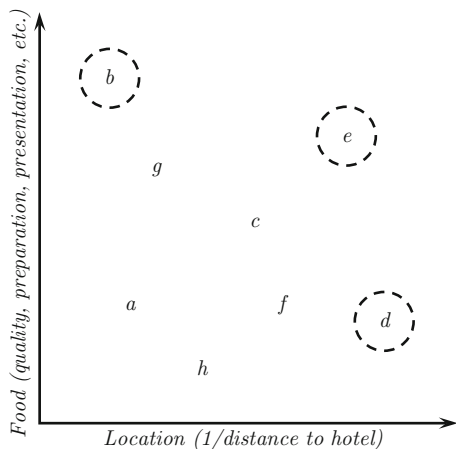
In databases and related fields, preferences are typically defined over single tuples or atomic ground atoms—the motivation behind this is that *answers to queries* need to be ranked according to users’ preferences. We will begin with a brief overview of preferences in relational databases, and then go on to the more closely related formalisms of ontological languages for the Semantic Web.

### 2.2.1 Relational Databases

The seminal work in general preference-based query answering—not assuming the availability of numerical information about values in the database—was carried out three decades ago in [22], where the authors recognized the need to deal with situations in which a given query either has no answers or too many—in the first case, conditions can be weakened, while in the latter they can be strengthened. Their solution was to extend the SQL language to incorporate user preferences via a new `PREFER` clause that can be either simple or compound, as well as nested for richer combinations. The expressivity of the resulting language is determined via a mapping to the domain relational calculus [23], and the authors briefly comment on a prototype Prolog implementation. This work subsequently inspired an entire line of research, as well as implemented systems. For instance [21], describes the Preference SQL system, which has been commercially available since 1999.

**The Skyline Operator** An important development in this line of research is the well-known *skyline* operator, which was first introduced in [5] as a way to characterize *interesting points*—essentially, a point (a tuple, in this setting) is considered to be interesting, if it is not dominated by any other point. When preferences are specified in terms of multiple aspects, such as location and price for a restaurant, then a dominance relation can be simply defined to hold between two points whenever the dominant one has a better value (or at least one that is just as good) than the dominated one in *every* aspect. The term “skyline” arises from the visualization of a cityscape—the points that form the skyline in a given dataset resemble the buildings that stand out against the sky because they are either very tall or they are closer than the others to the viewpoint (in this example, the dimensions are thus height and distance to the viewpoint). Figure 2.2 shows a simple illustration

**Fig. 2.2** Set of restaurants plotted with respect to their evaluation in terms of two dimensions of interest to a hypothetical user: *Location* and *Food*; both metrics are assumed to be defined so that higher values are better. The circled points comprise the skyline



of this concept: restaurants are evaluated in terms of location with respect to the user's hotel and food (for instance, according to the restaurant's average "Food" rating on TripAdvisor). Restaurants *b*, *e*, and *d* comprise the skyline; *b* has a great rating in terms of food, though it is quite bad in terms of location; *d* is somewhat the opposite, since it is among the worst in terms of food, but very close to the hotel; finally, *e* clearly dominates all non-skyline points.

In [5], the authors develop an extension of SQL with an optional SKYLINE OF clause in which users can express their desire to either minimize or maximize different dimensions (table attributes), or that values should be different in the result. Seven algorithms are provided for the physical implementation of the new clause (based on block nested loops and divide and conquer approaches, plus a special one that only works for two-dimensional cases). Skyline queries can also be implemented directly in standard SQL, though performance suffers greatly.

Skylines play an important role in preference-based query answering systems; in general, they are used in combination with top-*k* answers to yield the user's preferred answers.

**Preference Formulas** In [11], the *preference formula* formalism was introduced as a way to flexibly specify qualitative preferences, as well as embed them into queries. Preference formulas have the form:

$$t_1 \succ_C t_2 \text{ iff } C(t_1, t_2),$$

where  $t_1$  and  $t_2$  are database tuples, and  $C(t_1, t_2)$  is a first-order formula that may contain equality (or inequality) constraints and/or rational-order constraints (equality, inequality, and less/greater than comparisons with rational numbers). A more flexible variant was also used in [26], where "if" is used instead of "iff" in the definition. Chomicki refers to formulas with only built-in predicates as *intrinsic preference formulas* (ipfs), and studies the complexity of checking properties of

relations specified with them. Assuming that formulas are in DNF, the *width* is the number of disjuncts, and the *span* is the maximum number of conjuncts in a disjunct. The following complexity results hold in this case [11]; given a preference relation defined using an ipf  $C$  containing only atomic constraints over a single domain, with  $\text{width}(C) \leq m$  and  $\text{span}(C) \leq n$ , the time complexity of checking properties is:

- Irreflexivity:  $O(m \cdot n)$
- Asymmetry:  $O(m^2 \cdot n)$
- Transitivity:  $O(m^2 \cdot n^m \cdot \max(m, n))$
- Negative transitivity:  $O(m \cdot n^{2m} \cdot \max(m, n))$
- Completeness:  $O(k \cdot m \cdot n^{2m})$

Note that the above complexity results are characterized in terms of the size of the formula, and not in terms of the size of the database.

Other interesting properties of preference formulas are those resulting from their *composition*, since the model flexibly allows their combination via operators, such as union, intersection, and difference. Interestingly, neither weak nor total orders are preserved by such operations; on the other hand, strict partial orders are preserved by intersection, though not by union or difference.

More complex composition operations can also be defined [11]:

- Prioritized composition: Prefer according to  $\succ_2$ , unless  $\succ_1$  is applicable:

$$t_1 \succ_{1,2} t_2 \equiv t_1 \succ_1 t_2 \vee (t_1 \sim_1 t_2 \wedge t_1 \succ_2 t_2).$$

This kind of composition is associative and distributes over union. Weak and total orders are preserved, but strict partial orders are not.

- Pareto composition: Defined over the Cartesian product of two relations:

$$(t_1, t_2) \succ_P (t'_1, t'_2) \equiv (t_1 \succeq_1 t'_1) \wedge (t_2 \succeq_2 t'_2) \wedge ((t_1 \succ_1 t'_1) \vee (t_2 \succ_2 t'_2)).$$

Pareto composition does not preserve strict partial order, weak order, or total order.

- Lexicographic composition: Defined over the Cartesian product of two relations:

$$(t_1, t_2) \succ_L (t'_1, t'_2) \equiv (t_1 \succeq_1 t'_1) \vee (t_1 \sim_1 t'_1 \wedge t_2 \succ_2 t'_2).$$

This kind of composition preserves weak and total orders, but not strict partial ones.

The *winnow* operator is defined as a companion to a preference formula  $C$  over a database instance  $r$  as follows:

$$\omega_C(r) = \{t \in r \mid \nexists t' \in r \text{ such that } t' \succ_C t\}.$$

Clearly, the winnow operator characterizes the skyline of a database instance with respect to a preference relation, as discussed above; the main difference is that skylines as proposed in [5] are defined for single relations. Several algebraic properties of this operator are investigated in [11], such as commutativity, commutativity with selection, commutativity with projection, distribution over Cartesian product, and distribution over union and difference.

**Other Notes** Studies of preferences related to (active) databases have also been done in classical logic programming [18, 19] as well as answer set programming frameworks [7]. For a fairly recent survey of preference-based query answering formalisms in databases, we refer the reader to [41].

### 2.2.2 Ontology Languages

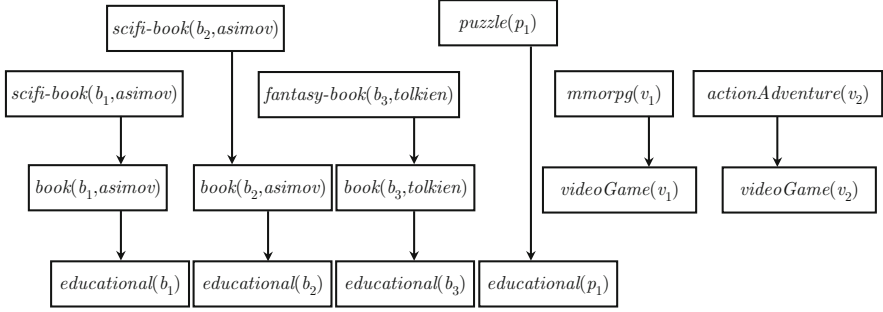
There have been only a few approaches to adding preferences to ontology languages. The first one—to our knowledge—was [39], where an extension is developed so that users can add their preferences to SPARQL queries via a new `PREFERRING` solution sequence modifier that allows to return either just skyline answers or soft constraints (where preference is given to answers that satisfy them, but they can also be relaxed if necessary). Other early approaches for preference-based querying RDF graphs include [8, 9, 32].

The approach that is most closely related to the one used in this book is the `PrefDatalog+/-` formalism, which was first presented in [26]; later, it was extended to work in combination with preference models based on probability values [30], as well as groups of users [29]. We will now provide a brief overview, starting with a toy ontology that will be used as a running example.

*Example 2.1* Consider the following simple ontology  $O = (D, \Sigma)$  describing gift ideas for friends:

$$D = \{scifi-book(b_1, asimov), scifi(b_2, asimov), fantasy-book(b_3, tolkien), \\ puzzle(p_1), mmorpg(v_1), actionAdventure(v_2)\}.$$

$$\Sigma = \{scifi-book(T, A) \rightarrow book(T, A), \\ fantasy-book(T, A) \rightarrow book(T, A), \\ mmorpg(X) \rightarrow videoGame(X), \\ actionAdventure(X) \rightarrow videoGame(X), \\ book(T, A) \rightarrow educational(T), \\ puzzle(N) \rightarrow educational(N), \\ book(X) \wedge videoGame(X) \rightarrow \perp\}.$$



**Fig. 2.3** Chase graph for the ontology from Example 2.1—arrows indicate TGD applications; nodes without incoming edges are part of the database

This ontology specifies a taxonomy with two kinds of book genres (science fiction and fantasy) and two video game genres (massively multi-player online role playing and action/adventure); it also states that books and puzzles are categorized under an “educational” label. The database  $D$  contains several instances for each of these genres. Finally, the last formula is a negative constraint stating that an item cannot be both a book and a video game.

Figure 2.3 shows the chase graph that arises from applying the TGDs in  $\Sigma$  over the database  $D$ . ■

Though the PrefDatalog+/- formalism does not commit to a specific preference framework, in this overview, we adopt the *preference formulas* approach discussed above. Consider the following example.

**Example 2.2** Continuing with Example 2.1, consider the following formulas representing a specific user’s preferences for children’s gifts:

- $C_1$ :  $\text{educational}(X) \succ \text{videoGame}(Y)$  if  $\top$ ;
- $C_2$ :  $\text{book}(T_1, A_1) \succ \text{book}(T_2, A_2)$  if  $\text{scifi\_book}(T_1, A_1) \wedge \text{fantasy\_book}(T_2, A_2)$ ;
- $C_3$ :  $\text{book}(T_1, A_1) \succ \text{book}(T_2, A_2)$  if  $(T_1 = b_1) \wedge (T_2 = b_2)$ ;
- $C_4$ :  $\text{educational}(X) \succ \text{educational}(Y)$  if  $\text{puzzle}(X) \wedge \text{fantasy\_book}(Y)$ ;
- $C_5$ :  $\text{videoGame}(X) \succ \text{videoGame}(Y)$  if  $\text{actionAdventure}(X) \wedge \text{mmorpg}(Y)$ .

The formula  $C_1$  states that educational gifts are preferable to video games (unconditionally),  $C_2$  states that sci-fi books are preferred over fantasy ones,  $C_3$  refers specifically to books  $b_1$  and  $b_2$  (the former is preferred),  $C_4$  states that an educational gift that is a puzzle is preferable to one that is a fantasy book, and  $C_5$  specifies preference of action/adventure video games over massively multiplayer online role playing ones. ■

In the following, for a preference formula  $C : t_1 \succ t_2$  if  $C(t_1, t_2)$ , we call  $C(t_1, t_2)$  the *condition* of  $C$ , and denote it with  $\text{cond}(C)$ .

**Syntax and Semantics of PrefDatalog+/-** As discussed in Chap. 1, for classical Datalog+/- we have an infinite universe of constants  $\Delta_{Ont}$ , an infinite set of



variables  $\mathcal{V}_{Ont}$ , and a finite set of predicate names  $\mathcal{R}_{Ont}$ . Analogously, for the preference model, we have a finite set of constants  $\Delta_{Pref}$ , an infinite set of variables  $\mathcal{V}_{Pref}$ , and a finite set of predicate names  $\mathcal{R}_{Pref}$ . In the following, we assume w.l.o.g. that  $\mathcal{R}_{Pref} \subseteq \mathcal{R}_{Ont}$ ,  $\Delta_{Pref} \subseteq \Delta_{Ont}$ , and  $\mathcal{V}_{Pref} \subseteq \mathcal{V}_{Ont}$ . These sets give rise to corresponding *Herbrand bases* consisting of all possible ground atoms that can be formed, which we denote by  $\mathcal{H}_{Ont}$  and  $\mathcal{H}_{Pref}$ , respectively. Clearly, we have  $\mathcal{H}_{Pref} \subseteq \mathcal{H}_{Ont}$ , meaning that preference relations are defined over a subset of the possible ground atoms.

Let  $O$  be a Datalog+/- ontology and  $P$  be a set of preference formulas with Herbrand bases  $\mathcal{H}_{Ont}$  and  $\mathcal{H}_{Pref}$ , respectively. A *preference-based Datalog+/- ontology* (PrefDatalog+/- ontology, or knowledge base) is of the form  $KB = (O, P)$ , where  $\mathcal{H}_{Pref} \subseteq \mathcal{H}_{Ont}$ . The semantics of PrefDatalog+/- arises as a direct combination of the semantics of Datalog+/- and that of preference formulas. A knowledge base  $KB = (O, P)$  satisfies  $a_1 \succ_P a_2$ , denoted  $KB \models a_1 \succ_P a_2$ , if and only if:

1.  $O \models a_1$  and  $O \models a_2$ , and
2.  $\models \bigvee_{pf_i \in P} cond(pf_i)(a_1, a_2)$ .

Intuitively, the consequences of  $KB$  are computed in terms of the chase for the classical Datalog+/- ontology  $O$ , and the set of preference formulas  $P$  describes the preference relation over pairs of atoms in  $\mathcal{H}_{Ont}$ . Considering  $KB = (O, P)$ , where  $O$  is the ontology, and  $P$  is the set of preference formulas from the running example, we have that:

$$KB \models educational(b_1) \succ_P videoGame(v_1),$$

since  $O \models educational(b_1)$ ,  $O \models videoGame(v_1)$ , and  $C_1 \in P$  allows us to conclude that  $educational(b_1) \succ_P videoGame(v_1)$ . On the other hand,

$$KB \not\models educational(b_1) \succ_P educational(b_2),$$

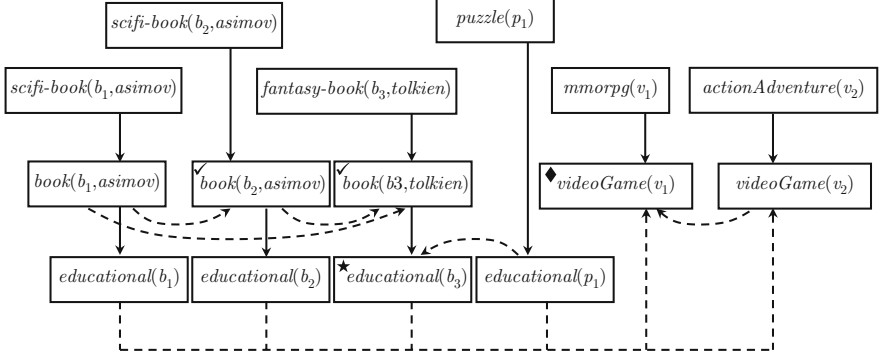
since the second condition is not satisfied in this case.

**Preference-Based Queries** There are two kinds of preference-based queries that can be issued over PrefDatalog+/- ontologies: *skyline* and *k-rank*. Answers to such queries are defined as usual via *substitution* (functions from variables to variables or constants) and *most general unifiers* [25]. We consider two kinds of classical queries: disjunctive atomic queries (disjunctions of atoms—DAQs) and conjunctive queries (CQs). We begin with the former: let  $Q(\mathbf{X}) = q_1(\mathbf{X}_1) \vee \dots \vee q_n(\mathbf{X}_n)$ , where the  $q_i$ 's are atoms and  $\mathbf{X}_1 \cup \dots \cup \mathbf{X}_n = \mathbf{X}$ :

- The set of *skyline answers* to  $Q$  is defined as:

$$\{\theta q_i \mid O \models \theta q_i \text{ and } \nexists \theta' \text{ such that } O \models \theta' q_j \text{ and } \theta' q_j \succ_P \theta q_i, \text{ with } 1 \leq i, j \leq n\},$$

where  $\theta, \theta'$  are most general unifiers for the variables in  $Q(\mathbf{X})$ .



**Fig. 2.4** Chase graph from Fig. 2.3, augmented with information on preferences between atoms based on the preference formulas from Example 2.2—dashed arrows denote the preference relation that arises from that model. Marks on the upper left-hand corner indicate dominance marks associated with different queries

- A  $k$ -rank answer to  $Q(\mathbf{X})$  is defined for transitive relations  $\succ_p$  as a sequence of maximal length of mgu's for  $\mathbf{X}$ :  $S = (\theta_1, \dots, \theta_{k'})$  such that  $O \models \theta_i Q$  for  $1 \leq i \leq k' \leq k$ , and  $S$  is built by subsequently appending the skyline answers to  $Q$ , removing these atoms from consideration, and repeating the process until either  $S = k$  or no more answers to  $Q$  remain.

Note that  $k$ -rank answers are only defined when the preference relation is transitive; this kind of answer can be seen as a generalization of traditional top- $k$  answers [41] that are still defined when  $\succ_p$  is not a weak order, and their name arises from the concept of *rank* introduced in [11].

Intuitively, for DAQs, both kinds of answers can be seen as atomic consequences of  $O$  that satisfy the query: the skyline answers can be seen as sets of atoms that are not dominated by any other such atom, while  $k$ -rank answers are  $k$ -tuples sorted according to the preference relation. We refer to these as *answers in atom form*.

Returning to the running example, consider the queries:

$$Q_1(X, Y) = \text{book}(X, Y) \quad \text{and} \quad Q_2(X) = \text{educational}(X).$$

The  $\succ_p$  relation is depicted in Fig. 2.4 (the arcs with dashed lines denote the ordered pairs in the relation). The set of skyline answers to  $Q_1$  is:  $\{\text{book}(b_1, \text{asimov})\}$ , while for  $Q_2$ , it is  $\{\text{educational}(p_1), \text{educational}(b_1), \text{educational}(b_2)\}$ . A 3-rank answer to  $Q_1$  is:

$$(\text{book}(b_1, \text{asimov}), \text{book}(b_2, \text{asimov}), \text{book}(b_3, \text{tolkien})).$$

For  $Q_2$ , a 3-rank answer is

$$(\text{educational}(p_1), \text{educational}(b_1), \text{educational}(b_2)).$$

Finally, the query

$$Q_3 = \text{puzzle}(X) \vee \text{videoGame}(X)$$

yields  $\{\text{puzzle}(p_1), \text{videoGame}(v_2)\}$  as skyline answers, and a 3-rank answer is

$$\{\text{puzzle}(p_1), \text{videoGame}(v_2), \text{videoGame}(v_1)\}.$$

In the case of (non-atomic) conjunctive queries, the substitutions in answers no longer yield single atoms but rather *sets* of atoms. Therefore, to answer such queries relative to a preference relation, we must extend the preference specification framework to take into account sets of atoms instead of individual ones. One such approach was proposed in [45], where a mechanism to define a preference relation over tuple sets  $\succ_{PS}: 2^{\mathcal{H}_{Pref}} \times 2^{\mathcal{H}_{Pref}}$  is introduced. We will not discuss this further here; [26] includes a treatment of their complexity and briefly describes how methods from the relational databases literature can be applied to answer them.

**The Preference-Augmented Chase (*prefChase*)** To compute skyline and  $k$ -rank answers to queries over a PrefDatalog+/- ontology  $KB = ((D, \Sigma), P)$ , an *augmented* chase forest is used, which is comprised of the necessary finite part of the chase forest relative to a given query that is augmented with an additional kind of edge called *preference edges*—these occur between nodes labeled with  $a, b \in \text{chase}(D, \Sigma)$  if and only if  $a \succ_P b$ . Finally, when an edge is introduced *between nodes whose labels satisfy  $Q$* , the node with the incoming edge is *marked*. Figure 2.4 shows  $\text{prefChase}(KB, Q)$  for the PrefDatalog+/- ontology from the running example; for illustrative reasons, markings for three different queries have been included in the figure:  $\text{book}(X, Y)$  (check mark),  $\text{educational}(X)$  (star), and  $\text{videoGame}(X)$  (diamond).

This data structure can directly be used to answer queries. Obtaining the node markings consists of almost all of the work towards answering a skyline query; all that remains to be done is to go through the structure and find the nodes whose labels satisfy the query and, if unmarked, add them to the output. For  $k$ -rank queries, the query answering process involves iterating through the computation of the skyline answers, updating the result by appending these answers in arbitrary order, and removing the nodes and edges involved in the result from the chase structure; finally, before the next iteration, the node markings need to be updated.

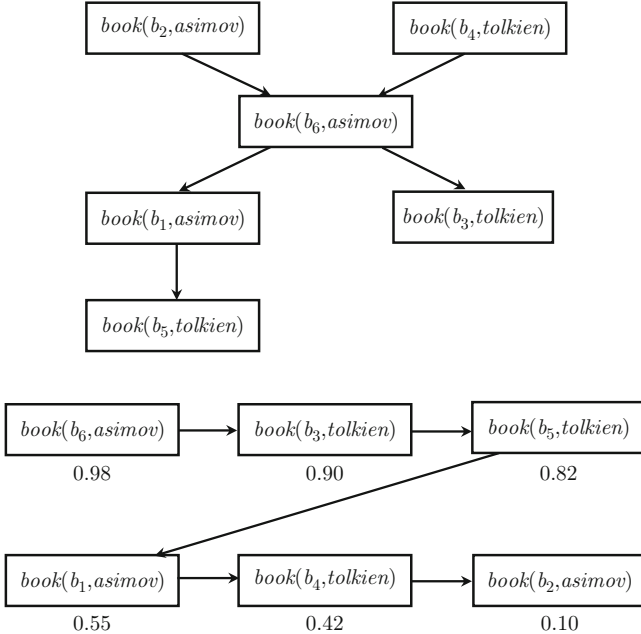
As an upper bound on the cost of these procedures, both kinds of queries can be answered in time quadratic in the cost of building the classical chase (in the data complexity) [26]—this cost depends on the fragment of Datalog+/- used in the underlying ontology, but the overhead imposed by the markings and preference edges to obtain the preference-augmented chase is clearly at most quadratic.

### 2.2.3 Models for Uncertain Preferences

There are several ways in which uncertainty can be incorporated into models for preference-based reasoning in the databases tradition. We will provide a brief overview of the two that are most relevant to this work.

**PP-Datalog+/-** The first model we will review is called *Probabilistic Preference-based Datalog+/-* [27, 30]. It is an extension of PrefDatalog+/-, where the main idea is to combine it with the probabilistic extension proposed in [17]; for the purposes of this discussion, the only important aspect of the latter is that any atomic inference can be assigned a probability with which it holds—such probability values naturally induce a weak order over the elements, under the assumption that more probable inferences are more preferable in that model.

The presence of two preference models—an SPO provided by the user and a weak order induced by the probability values—then poses the problem of how to order the results of a query. Figure 2.5 illustrates this situation in the domain of Example 2.1. As usual, the SPO represents the user’s preferences, while the probability values could indicate, for instance, the probability that each book will be delivered in time. Note that books  $b_2$  and  $b_4$ , which are the top two picks, have the



**Fig. 2.5** *Top*: Strict partial order provided by the user; *Bottom*: Weak order arising from the probability values shown (under the assumption that more probable inferences are preferred in that model). For clarity, edges arising from the transitive closure are not shown

lowest probabilities; in this case, it seems that the user would be best off choosing book  $b_6$ , which is the next preferred and has a very high probability of reaching them in time.

In order to formalize this, the concept of *preference combination operators* is introduced in [30]. The goal of such operators is to produce a new preference relation that satisfies a set of basic properties. Two classes of operators are proposed: (1) *Egalitarian* operators allow the resulting relation to eventually resemble either the SPO or the probability-based orders; (2) *User-biased* operators, on the other hand, base the resulting preference relation on the user's preferences, and use the probabilistic model as a secondary source of "advice". Of the specific operators defined for each family in [30], though the theoretical complexity of the algorithms is similar, the actual properties of specific inputs—such as edge density, size of skylines in the SPO, and number of ties in the weak order—greatly affect their running time in practice.

**PPLNs** The other model that we would like to mention here is that of *Probabilistic Preference Logic Networks* [28], which is also known as *Markov Models over Weighted Orderings* [31]. Like PP-PrefDatalog+/-, PPLNs combine user's preferences with probabilistic uncertainty; however, this is done in a very different way: users provide preference formulas that are Boolean combinations of atomic statements of the form " $a \succ b$ ", and each formula receives a *weight*. This allows us to model situations that are characterized by: (1) the fact that we only have information on certain pairs of elements; and (2) the uncertainty underlying the information provided—users are much more likely to express preferences that are subject to exceptions than ones that hold all the time.

The semantics of PPLNs is based on *Markov Random Fields* (MRFs), a classical model for representing distributions over possible worlds; instead of having possible worlds arising from truth assignments to Boolean variables, possible worlds in the PPLN approach are *linear orderings*, so their number is factorial in the number of elements instead of exponential. Computing the probability of a preference statement issued as a query to a PPLN has been shown to be #P-hard [28], so two approaches based on results from the mathematical branch of order theory are investigated to tackle intractability: approximations via approximate model counting (yielding a fully polynomial randomized approximation scheme under fixed-parameter assumptions) and exact approaches via exact counting that work for specific fragments (yielding a fixed-parameter polynomial time algorithm).

Another approach to tractable query answering in these models was proposed in [31], where variable elimination algorithms inspired by those for related models like Bayesian networks were investigated and empirically evaluated. For *chain* models (in which weighted preference statements taken as a whole form sets of concatenated formulas), a cubic time algorithm is proposed and empirically shown to scale well—queries can be answered in under a minute for models of up to 2500 elements. For the more general case of models consisting of weighted atomic statements (and queries/evidence consisting of conjunctions of such formulas), an algorithm is proposed that, although of exponential running time, has the *linear cut*

*size* of the model (a measure similar to treewidth) as dominating factor. Empirical evaluations of this algorithm show that tractability is much more elusive than for chain models; however, even simple heuristics based on minimizing linear cut size have a large impact both on running time and the size of the data structures that must be maintained.

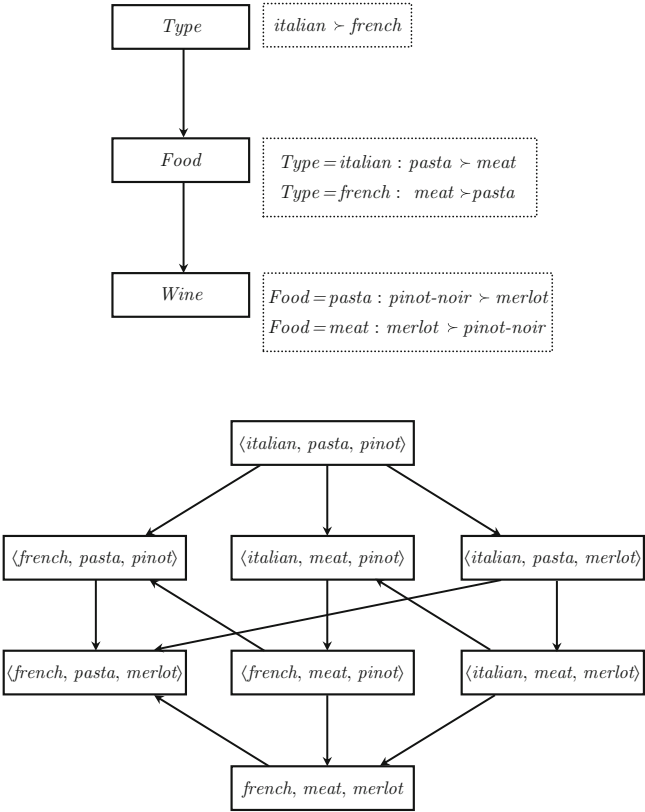
## 2.3 Preferences *à la* Philosophy and Related Disciplines

Preferences in philosophy have received much attention, as far back as Aristotle, perhaps because they constitute a purely subjective evaluation of the elements in question. In contrast to the approach taken in databases as described in the previous section, the tradition in philosophy and related disciplines is to consider preferences over elements that are mutually exclusive—the most common sets of elements, especially in disciplines borne out of philosophy, such as logic and decision theory, are *states of the world*, which can be represented as vectors of values for a set of variables. For instance, one might prefer sunny weather on weekends over sunny weather on weekdays. Note that, in general, this leads to large sets of alternatives, since these are comprised of the Cartesian product of the domains of all variables. This is quite different from the simpler view taken in databases and related approaches, where atomic elements are typically (but not always) referred to in preference relations. In contrast, the philosophical tradition does not exclude reasoning about atomic elements—the separation implied by dividing the discussion into two separate sections of this chapter is merely a practical one.

Since this conception of preference modeling is somewhat less related to the tools developed in this book, here, we will only review the *CP-nets* model.

**CP-Nets** This model, first proposed in [6], is designed to represent preferences over possible worlds much in the same way as probabilities are represented in Bayesian networks. Their name comes from either *conditional preference* or *ceteris paribus* (meaning “all else being equal”)—variables have corresponding nodes in a directed graph, and conditional tables state preferences between the different values that they can take, given values for the variables that are directly connected by incoming edges in the graph. The set of *outcomes* is comprised of all possible settings of the variables in the model. The semantics of CP-nets is given by the concept of *worsening flip* between outcomes; given two outcomes that differ in only one value, the values that remain fixed allow the ordering of the pair according to the relevant conditional preference table(s)—this is where “*ceteris paribus*” comes into play. An outcome is said to *dominate* another, if there is a sequence of worsening flips from the first to the second.

The two main reasoning tasks associated with CP-nets are: deciding whether one outcome dominates another (*dominance query*), and deciding whether a given outcome is optimal (*outcome optimization*). The following example illustrates these concepts.



**Fig. 2.6** *Top*: A simple CP-net specifying preferences over restaurant types, foods, and wine. *Bottom*: Graph of all possible outcomes; edges denote preference between outcomes

*Example 2.3* Consider the toy example CP-net depicted in Fig. 2.6 (top), containing three variables relevant to the restaurant domain: restaurant type, foods, and wine—all variables are binary in order to keep the example simple.

According to this CP-net, the user prefers Italian restaurants over French ones; if they are in an Italian restaurant, then pasta is preferred over meat, and the other way around, if the restaurant is French. Finally, when having pasta, Pinot Noir is preferred over Merlot, whereas the opposite holds when having meat.

Figure 2.6 (bottom) specifies all possible worsening flips between outcomes, from which the full preference relation can be obtained. For example, outcome  $\langle \text{italian}, \text{meat}, \text{merlot} \rangle$  is preferred over  $\langle \text{italian}, \text{meat}, \text{pinot} \rangle$ , because of the first entry in the conditional preference table for the *Food* variable, which specifies that when having meat, Merlot is preferred over Pinot Noir.

For this model, note that outcome  $\langle \text{italian}, \text{pasta}, \text{merlot} \rangle$  dominates outcome  $\langle \text{french}, \text{meat}, \text{merlot} \rangle$  since there is a sequence of worsening flips of length two that goes through  $\langle \text{italian}, \text{meat}, \text{merlot} \rangle$ . Finally, outcome  $\langle \text{italian}, \text{pasta}, \text{pinot} \rangle$  is optimal, since there is no other outcome that dominates it. ■

CP-nets were later generalized by *CP-theories* [43], which are sets of conditional preference statements that allow the specification of a set of variables for which the value does not matter—CP-nets are thus captured by the case in which these variable sets are empty.

**Extensions of CP-Nets and CP-Theories with Ontologies** In [13], the *ontological CP-net* model is presented, in which variables correspond to description logic axioms whose values are simply satisfied/not satisfied. Later, in [14], a similar approach was developed to inform how answers to queries over Datalog+/- ontologies should be ranked, focusing on skyline and *k*-rank answers. Finally, this line of research was extended to work with CP-theories in [15], studying the data and combined computational complexity of the different reasoning tasks for different fragments of Datalog+/-.

**CP-Nets for Modeling Preferences Under Uncertainty** CP-nets have also more recently been extended with probabilistic uncertainty [4, 12], both over the structure of the graph as well as the preference tables. The main reasoning tasks in probabilistic CP-nets are the computation of the probability of dominance of one outcome over another, computing the probability that a given outcome is optimal, finding the most probable optimal outcome, and computing the most probable induced (classical) CP-net.

## 2.4 Final Notes

**Preferences and Groups** The area of modeling different kinds of groups is quite related to the study of preferences. For instance, *social choice* focuses on mechanisms for finding the decision that is best for a group as a whole by combining the opinion of individuals; this has been the topic of study in different fields, like mathematics, economics, politics, and sociology for decades [35, 42]. Other areas related to social choice are multiagent systems [44], recommender systems [2, 34, 40], rank aggregation [16], and combining incomplete preferences [1, 24, 36]. Finally, there is also an approach based on Datalog+/- for query answering with group preferences that was recently proposed in [29].

**Preferences and Provenance** This work is also closely connected to the study and use of *provenance* in information systems and, in particular, the Semantic Web and social media [3, 33]—provenance refers to the description of the history of data in its life cycle, and it is also sometimes referred to as *lineage*. Research in provenance distinguishes between *data* and *workflow* provenance: the former explores the data flow within applications in a fine-grained way, while the latter is coarse-grained and does not consider the flow of data. Another classification is typically considered in the provenance literature is the *why*, *how*, and *where* framework [10]. As we will see in Chap. 3, our data models incorporate provenance information via registers that store information about the origin of each report, allowing us to take into account



where evaluations within a social media system come from (such as information about who has issued the report, their origin, and what their preferences were at the time), and leverage this information to allow users to make informed provenance-based decisions.

For a more comprehensive review of the many aspects associated with reasoning with preferences, the interested reader is referred to [38] and [20].

## References

1. M. Ackerman, S. Choi, P. Coughlin, E. Gottlieb, J. Wood, Elections with partially ordered preferences. *Public Choice* **157**(1/2), 145–168 (2013)
2. S. Amer-Yahia, S.B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: semantics and efficiency. *Proc. VLDB Endow.* **2**(1), 754–765 (2009)
3. G. Barbier, Z. Feng, P. Gundechea, H. Liu, *Provenance Data in Social Media* (Morgan and Claypool, San Rafael, CA, 2013)
4. D. Bigot, B. Zanuttini, H. Fargier, J. Mengin, Probabilistic conditional preference networks, in *Proceedings of the Conference on Uncertainty in Artificial Intelligence UAI* (2013)
5. S. Börzsönyi, D. Kossmann, K. Stocker, The Skyline operator, in *Proceedings of the International Conference on Data Engineering (ICDE)* (2001), pp. 421–430
6. C. Boutilier, R.I. Brafman, C. Domshlak, H.H. Hoos, D. Poole, CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.* **21**, 135–191 (2004)
7. G. Brewka, Preferences, contexts and answer sets, in *Proceedings of the International Conference on Logic Programming (ICLP)* (2007), p. 22
8. G. Chamiel, M. Pagnucco, Exploiting ontological information for reasoning with preferences, in *Multidisciplinary Workshop on Advances in Preference Handling* (2008)
9. L. Chen, S. Gao, K. Anyanwu, Efficiently evaluating skyline queries on RDF databases, in *The Semantic Web: Research and Applications* (2011), pp. 123–138
10. J. Cheney, L. Chiticariu, W. Tan, Provenance in databases: why, how, and where. *Found. Trends Databases* **1**(4), 379–474 (2009)
11. J. Chomicki, Preference formulas in relational queries. *ACM Trans. Database Syst.* **28**(4), 427–466 (2003)
12. C. Cornelio, J. Goldsmith, N. Mattei, F. Rossi, K.B. Venable, Dynamic probabilistic CP-nets, in *Proceedings of the Multidisciplinary Workshop on Advances in Preference Handling* (2013), pp. 1–7
13. T. Di Noia, T. Lukasiewicz, G.I. Simari, Reasoning with semantic-enabled qualitative preferences, in *Proceedings of the International Conference on Scalable Uncertainty Management (SUM)* (2013), pp. 374–386
14. T. Di Noia, T. Lukasiewicz, M.V. Martinez, G.I. Simari, O. Tifrea-Marcuska, Computing k-rank answers with ontological CP-nets, in *Proceedings of the Workshop on Logics for Reasoning about Preferences, Uncertainty, and Vagueness (PRUV)* (2014), pp. 74–87
15. T. Di Noia, T. Lukasiewicz, M.V. Martinez, G.I. Simari, O. Tifrea-Marcuska, Combining existential rules with the power of CP-Theories, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (2015), pp. 2918–2925
16. R. Fagin, R. Kumar, D. Sivakumar, Comparing top k lists. *SIAM J. Discret. Math.* **17**(1), 134–160 (2003)
17. G. Gottlob, T. Lukasiewicz, M.V. Martinez, G.I. Simari, Query answering under probabilistic uncertainty in Datalog+/- ontologies. *Ann. Math. Artif. Intell.* **69**, 37–72 (2013)
18. K. Govindarajan, B. Jayaraman, S. Mantha, Preference logic programming, in *Proceedings of the International Conference on Logic Programming (ICLP)* (1995), pp. 731–745

19. K. Govindarajan, B. Jayaraman, S. Mantha, Preference queries in deductive databases. *N. Gener. Comput.* **19**(1), 57–86 (2001)
20. S. Kaci, in *Working with Preferences: Less Is More*. Cognitive Technologies (Springer, Berlin/Heidelberg, 2011)
21. W. Kießling, M. Endres, F. Wenzel, The preference SQL system: an overview. *IEEE Data Eng. Bull.* **34**(2), 11–18 (2011)
22. M. Lacroix, P. Lavency, Preferences: putting more knowledge into queries, in *Proceedings of the International Conference on Very Large Databases (VLDB)* (1987), pp. 217–225
23. M. Lacroix, A. Pirotte, ILL: an English structured query language for relational data bases. *ACM SIGART Bull.* (61), 61–63 (1977), <http://dl.acm.org/citation.cfm?id=1045335>
24. J. Lang, M.S. Pini, F. Rossi, D. Salvagnin, K.B. Venable, T. Walsh, Winner determination in voting trees with incomplete preferences and weighted votes. *J. Auton. Agent. Multi-Agent Syst.* **25**(1), 130–157 (2012)
25. J.W. Lloyd, *Foundations of Logic Programming*, 2nd edn. (Springer, Berlin/Heidelberg, 1987)
26. T. Lukasiewicz, M.V. Martinez, G.I. Simari, Preference-based query answering in Datalog+/- ontologies, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (2013), pp. 1017–1023
27. T. Lukasiewicz, M.V. Martinez, G.I. Simari, Preference-based query answering in probabilistic Datalog+/- ontologies, in *Proceedings of the International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE)* (2013), pp. 501–518
28. T. Lukasiewicz, M.V. Martinez, G.I. Simari, Probabilistic preference logic networks, in *Proceedings of the European Conference on Artificial Intelligence (ECAI)* (IOS Press, Amsterdam, 2014), pp. 561–566
29. T. Lukasiewicz, M.V. Martinez, G.I. Simari, O. Tifrea-Marcuska, Ontology-based query answering with group preferences. *ACM Trans. Internet Technol.* **14**(4), 25 (2014)
30. T. Lukasiewicz, M.V. Martinez, G.I. Simari, O. Tifrea-Marcuska, Preference-based query answering in probabilistic Datalog+/- ontologies. *J. Data Semant.* **4**(2), 81–101 (2015)
31. T. Lukasiewicz, M.V. Martinez, D. Poole, G.I. Simari, Probabilistic models over weighted orderings: fixed-parameter tractable variable elimination, in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)* (2016), pp. 494–504
32. S. Magliacane, A. Bozzon, E. Della Valle, Efficient execution of top-k SPARQL queries, *Proceedings of the International Semantic Web Conference (ISWC)* (2012), pp. 344–360
33. L. Moreau, The foundations for provenance on the Web. *Found. Trends Web Sci.* **2**(2/3), 99–241 (2010)
34. E. Ntoutsi, K. Stefanidis, K. Nørvgå, H. Kriegel, Fast group recommendations by applying user clustering, in *Proceedings of the International Conference on Conceptual Modelling (ER)* (Springer, Berlin, 2012), pp. 126–140
35. P.K. Pattanaik, *Voting and Collective Choice: Some Aspects of the Theory of Group Decision-Making* (Cambridge University Press, Cambridge, 1971)
36. M.S. Pini, F. Rossi, K.B. Venable, T. Walsh, Aggregating partially ordered preferences. *Int. J. Log. Comput.* **19**(3), 475–502 (2008)
37. F. Roberts, B. Tesman, *Applied Combinatorics* (CRC Press, Boca Raton, FL, 2009)
38. F. Rossi, K.B. Venable, T. Walsh, in *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Synthesis Lectures on Artificial Intelligence and Machine Learning (Morgan & Claypool Publishers, San Rafael, CA, 2011)
39. W. Siberski, J.Z. Pan, U. Thaden, Querying the Semantic Web with preferences, in *Proceedings of the International Semantic Web Conference (ISWC)* (Springer, Berlin, 2006), pp. 612–624
40. B. Smith, G. Linden, Two decades of recommender systems at Amazon.com. *IEEE Internet Comput.* **21**(3), 12–18 (2017)
41. K. Stefanidis, G. Koutrika, E. Pitoura, A survey on representation, composition and application of preferences in database systems. *ACM Trans. Database Syst.* **36**(3), 19:1–19:45 (2011)
42. A.D. Taylor, *Social Choice and the Mathematics of Manipulation* (Cambridge University Press, Cambridge, 2005)

- 43. N. Wilson, Extending CP-nets with stronger conditional preference statements, in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 4 (2004), pp. 735–741
- 44. M. Wooldridge, *An Introduction to Multiagent Systems* (Wiley, Hoboken, NJ, 2009)
- 45. X. Zhang, J. Chomicki, Preference queries over sets, in *Proceedings of the International Conference on Data Engineering (ICDE)* (2011), pp. 1019–1030

Ontology-Based Data Access Leveraging Subjective  
Reports

Simari, G.I.; Molinaro, C.; Vanina Martinez, M.;

Lukasiewicz, Th.; Predoiu, L.

2017, VIII, 77 p. 32 illus., Softcover

ISBN: 978-3-319-65228-3