Conditional Preferences: A New Semantics for Database Queries

Paolo Ciaccia
DEIS, University of Bologna, Italy
pciaccia@deis.unibo.it

Abstract. Preference queries aim to retrieve from large databases those objects that better match user's requirements. Approaches proposed so far in the DB field for specifying preferences are limited when one needs to consider *conditional*, rather than absolute, preferences (e.g., I prefer driving by car in winter, and by motorbike in summer), which are common in context-aware applications. CP-nets are a powerful formalism for concisely representing such preferences, which has its roots in decision making problems. However, CP-nets, being based on a *ceteris paribus* (all else being equal) interpretation, are hardly applicable in complex DB scenarios. In this paper we introduce a new *totalitarian* (i.e., not ceteris paribus) semantics for CP-nets. We prove that our semantics is equivalent to ceteris paribus for complete acyclic CP-nets, whereas it avoids some counterintuitive effects of ceteris paribus when the CP-net is partially specified.

1 Introduction

The trend towards the personalization of information systems functionalities requires new models and techniques able to provide users with the "right information" at the "right time" in the "right place". Context-aware applications are a remarkable step towards achieving this goal, the key idea being that of taking into account context information when processing user requests. In particular, ranking the result of a query should be based on the current user context, rather than on some absolute criterion.

Example 1. Consider the following database of hotels:

Name	Price	Stars	Rooms	Internet
Jolly	40	2	50	Yes
Continental	55	2	30	No
Excelsior	80	3	50	Yes
Rome	80	5	100	Yes
Holiday	60	4	20	No

When travelling for work, the user does not care about price and number of rooms, he preferring hotels with at least 4 stars and an Internet connection. In this case the best alternative is hotel Rome. However, if travelling for leisure, the user prefers small hotels (≤ 30 rooms) and whose price is at most 50 Euro. In this case no hotel satisfies both requirements, yet it can be argued that Continental, Jolly, and Holiday are the best available alternatives, since each of them satisfies one of the two user preferences. $\square$

Frameworks proposed so far in the DB field [Cho02,Kie02] have paid little attention to *conditional* preferences. On the other hand, these have been largely investigated by AI researchers, with a particular emphasis on *CP-nets* (Conditional Preference networks) [BBHP99,BBD$^+$04,Wil04,GLTW05], a graph-based formalism able to "factorize" the specification of preference statements over a set of attributes. A CP-net statement like $\varphi = p : a_i > a_j$ is given a *ceteris paribus* interpretation, i.e., "given p prefer a_i to a_j only if values of other attributes are *equal*".

In this paper we argue that the ceteris paribus semantics is unsuitable for real-world complex DB's, since it provides counterintuitive results whenever the DB is *incomplete*, i.e., it does not contain all the possible alternatives for the preference attributes, and the CP-net is not completely specified (see next section for a definition of complete CP-nets). With the aim of preserving the strong points of CP-nets, we provide an alternative, so called *totalitarian*, semantics for CP-nets. We first show that, rather surprisingly, the new semantics is equivalent to ceteris paribus for complete acyclic CP-nets. Then we prove that for complete DB's the two semantics, although leading to different preferences, always yield the same set of optimal results. Finally, we show that for incomplete DB's *and* CP-nets the new semantics excludes from the result those tuples that are apparently sub-optimal with respect to user preferences.

2 Background on CP-nets

A CP-net over a set of attributes $X = \{A_1, \ldots, A_n\}$ is a pair $N = (G, CPT)$, where $G = (X, E)$ is a directed graph and CPT is a function that associates to each $A_i \in X$ a *conditional preference table*, $CPT(A_i)$. If the arc $(A_j, A_i) \in E$, then A_j is a *parent* of A_i. Let P_i be the set of parents of A_i. Then, $CPT(A_i)$ consists of a set of preference statements φ of the form $\varphi = p : a_{i,1} > a_{i,2}$, where $p \in dom(P_i)$ and $a_{i,1}, a_{i,2} \in dom(A_i)$.[1] This expresses the conditional preference of $a_{i,1}$ with respect to $a_{i,2}$ given p. If A_i has no parents, then the statement simplifies to $\varphi = \perp: a_{i,1} > a_{i,2} \equiv a_{i,1} > a_{i,2}$, i.e., $a_{i,1}$ is unconditionally preferred to $a_{i,2}$.

Example 2. Figure 1 shows a simple CP-net over attributes RestaurantType (R), Table (T), and Price (P), thus $X = \{R, T, P\}$. For simplicity, all attributes have binary domains, in particular: $dom(R) = \{it, chn\}$ (italian or chinese), $dom(T) = \{in, out\}$ (inside or outside), and $dom(P) = \{low, high\}$. My preferences unconditionally go to italian restaurants ($it > chn$), for which I prefer to have a table inside ($it : in > out$) and pay the less ($it : low > high$). On other hand, in a chinese restaurant I prefer to sit outside ($chn : out > in$) and to pay more ($chn : high > low$). □

Definition 1 *A CP-net $N = (G, CPT)$ is:*

- acyclic *iff G is acyclic;*
- locally consistent *iff, for each attribute A_i, $CPT(A_i)$ does not include a "chain" of statements $\varphi_1, \ldots, \varphi_m$ ($m > 1$), such that: $p : a_{i,1} > a_{i,2} > \ldots > a_{i,1}$;*
- complete *iff, for each A_i and for each $p \in dom(P_i)$, $CPT(A_i)$ totally orders values in $dom(A_i)$, i.e., for each $a_{i,1}, a_{i,2}$ either $p : a_{i,1} > a_{i,2}$ or $p : a_{i,2} > a_{i,1}$.*

[1] Equivalently, each statement might specify a conjunction of pair orderings of the form $a_{i,j} > a_{i,k}$, given a set of values from $dom(P_i)$.

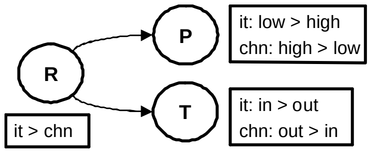


Fig. 1. A simple CP-net over 3 attributes

The CP-net in Figure 1 is acyclic and locally consistent. Further, it is also complete. Should we drop one statement (e.g., $it : in > out$) we would have an incomplete CP-net. If the CP-net is locally consistent, no contradiction is present as long as we consider preferences over any single attribute. In the following we only consider acyclic and locally consistent CP-nets.

The standard *ceteris paribus* interpretation of a statement $\varphi = p : a_{i,1} > a_{i,2}$, $\varphi \in CPT(A_i)$, is the set of pairs of tuples over X:

$$\varphi^*_{\mathsf{cp}} = \{((p, a_{i,1}, y), (p, a_{i,2}, y)) | y \in dom(X - P_i - \{A_i\})\} \tag{1}$$

in which y is any value of $dom(Y_i)$, Y_i being the set of attributes not involved in φ. Thus, each preference induced by φ concerns two tuples that differ *only* in the value of A_i. Let $\Phi^*_{A_i,\mathsf{cp}} = \bigcup_{\varphi \in CPT(A_i)} \varphi^*_{\mathsf{cp}}$ denote all preferences induced by $CPT(A_i)$. Since the CP-net is locally consistent, no conflicts are present in $\Phi^*_{A_i,\mathsf{cp}}$. Further, it is easy to see that, due to the cp semantics, any two tuples t_1 and t_2 are ordered by at most one $\Phi^*_{A_i,\mathsf{cp}}$ set. Taking the union of such sets leads to:

$$\Phi^*_{\mathsf{cpu}} = \bigcup_{A_i \in X} \Phi^*_{A_i,\mathsf{cp}} \tag{2}$$

Finally, let $\succ_{\mathsf{cpu}}$ stand for the order obtained by taking the transitive closure of Φ^*_{cpu}. We say that tuple t_1 *dominates* tuple t_2 (according to the *ceteris paribus union* (cpu) semantics) iff $t_1 \succ_{\mathsf{cpu}} t_2$, and that t_1 is *optimal* in a relation $r \subseteq dom(X)$ if it is undominated in r. A basic result on acyclic CP-nets is that $\succ_{\mathsf{cpu}}$ is always a *strict partial order*, thus not only transitive but also asymmetric (thus irreflexive). This guarantees that at least one optimal tuple exists. Further, if the CP-net is complete there is exactly one optimal tuple in $dom(X)$.

Example 3. Figure 2 shows the *preference graph* for the CP-net in Figure 1, where there is an arc from t_1 to t_2 iff the pair (t_1, t_2) is in Φ^*_{cpu}. Due to the ceteris paribus semantics, arcs exist only between tuples that differ in the value of a single attribute. There is a *path* in the graph from t_1 to t_2 iff $t_1 \succ_{\mathsf{cpu}} t_2$. Since the CP-net is complete there is one optimal tuple in $dom(X)$, namely (it, in, low). □

For lack of space, here we do not provide details on the proof procedure of CP-nets, needed to check if $t_1 \succ_{\mathsf{cpu}} t_2$. It suffices to say that for acyclic CP-nets its complexity can be exponential in the number of attributes, depending on the structure of the G graph and on how CPT's are specified.

3 Totalitarian Semantics for CP-nets

Although the cpu semantics is adequate in many situations, it is a fact that in most cases a *complete CP-net* is assumed. When preferences are over many attributes and/or

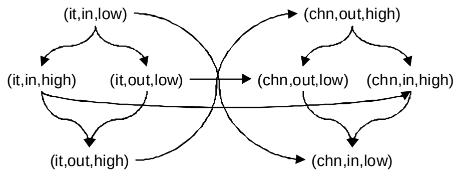
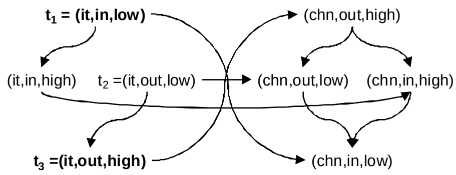


Fig. 2. The $\succ_{\text{cpu}}$ order induced over tuples by the CP-net in Figure 1

domains have large cardinalities, it is unrealistic to expect that a user will completely specify all the CPT's. The effect of having an incomplete CP-net can be seen by referring to our working example. Assume that $CPT(T)$ misses the entry for italian restaurants (i.e., $it : in > out$ is dropped from Figure 1), which is interpreted as "the user has no preference on where to sit". We are left with the following preference graph:



If the DB relation is *complete*, i.e., $r = dom(X)$, then the optimal tuples are $t_1 = (it, in, low)$ and $t_2 = (it, out, low)$, which is perfectly reasonable given the absence of preference on where to sit. Assume now that $r = \{t_1 = (it, in, low), t_3 = (it, out, high)\}$. Since $t_1 \not\succ_{\text{cpu}} t_3$ (there is no path from t_1 to t_3 in the above graph), we conclude that both t_1 and t_3 are optimal in r. We find this quite counterintuitive, since t_3 has a high price, which contradicts the preference $it : low > high$. Ideally, we would like to have that t_1 dominates t_3 even if the CP-net is incomplete.

We tackle the problem by redefining the semantics of preference statements *and* the way the so-resulting preferences have to be combined. We start with a first version of the *totalitarian* (as opposed to ceteris paribus) semantics of statements.

Definition 2 *Let $\varphi = p : a_{i,1} > a_{i,2}$ be a statement in $CPT(A_i)$. The* strong totalitarian *(st) interpretation of φ is the set of pairs of tuples:*

$$\varphi^*_{\mathit{st}} = \{((p, a_{i,1}, y), (p, a_{i,2}, y'))|y, y' \in dom(X - P_i - \{A_i\})\} \tag{3}$$

Thus, tuples ordered by φ differ in the value of A_i and, possibly, also in the values of attributes Y_i not involved in φ.

Since the CP-net is locally consistent, the sets $\Phi^*_{A_i,\text{st}} = \bigcup_{\varphi \in CPT(A_i)} \varphi^*_{\text{st}}$ of preferences induced by $CPT(A_i)$ still have no conflicts inside. However, two tuples t_1 and t_2 might be differently ordered by two $\Phi^*_{A_i,\text{st}}$ sets, thus taking their union could introduce cycles in the preference graph. As an example, given $\varphi = it : in > out$ and $\varphi' = it : low > high$ and the tuples $t_1 = (it, in, high)$ and $t_2 = (it, out, low)$, we have that $(t_1, t_2) \in \Phi^*_{T,\text{st}}$ and $(t_2, t_1) \in \Phi^*_{P,\text{st}}$, i.e., a cycle if we take the union of $\Phi^*_{T,\text{st}}$ and $\Phi^*_{P,\text{st}}$.

A way to preserve the strict partial order properties is to compose preferences in the $\Phi^*_{A_i,\text{st}}$ sets using a *Pareto rule*. Intuitively, this is to say that tuple t_1 dominates t_2 iff it does so over at least one attribute and is never the case that this is true also for t_2.

More precisely, we have that $(t_1, t_2) \in \Phi^*_{\mathsf{stp}}$ iff there exists an attribute A_i such that $(t_1, t_2) \in \Phi^*_{A_i,\mathsf{st}}$ and for no attribute A_j it is $(t_2, t_1) \in \Phi^*_{A_j,\mathsf{st}}$. The *strong totalitarian Pareto* (stp) order $\succ_{\mathsf{stp}}$ is then defined as the transitive closure of Φ^*_{stp}.

Theorem 1 *For any complete acyclic CP-net N, $\succ_{stp}$ is a strict partial order such that $\succ_{cpu} \subseteq \succ_{stp}$.*[2]

Above theorem shows that the strong totalitarian semantics includes *all* the ceteris paribus preferences. In many cases[3] it is also true that all the additional preferences in $\Phi^*_{\mathsf{stp}} - \Phi^*_{\mathsf{cpu}}$ are in the transitive closure of Φ^*_{cpu}, thus $\succ_{\mathsf{stp}} = \succ_{\mathsf{cpu}}$. For instance, this happens in our working example on restaurants. However, as the following example shows, this does not hold in general.

Example 4. Consider the CP-net in Figure 3, along with the preference graph of Φ^*_{cpu} (solid arcs). The figure also shows as dashed arcs 3 of the preferences in $\Phi^*_{\mathsf{stp}} - \Phi^*_{\mathsf{cpu}}$. While the one from (a_1, b_1, c_1) to (a_2, b_2, c_1), although not in Φ^*_{cpu} is in $\succ_{\mathsf{cpu}}$ (there is a path in the Φ^*_{cpu} graph) the other two are *not* derivable using the cpu semantics. For instance, consider the pair $(t, t') = ((a_1, b_2, c_1), (a_2, b_2, c_2))$. This is in Φ^*_{stp} since t is better than t' on A, $t[B] = t'[B]$, and on attribute C the two tuples cannot be compared, since they have different parent values ((a_1, b_2) and (a_2, b_2), respectively). □

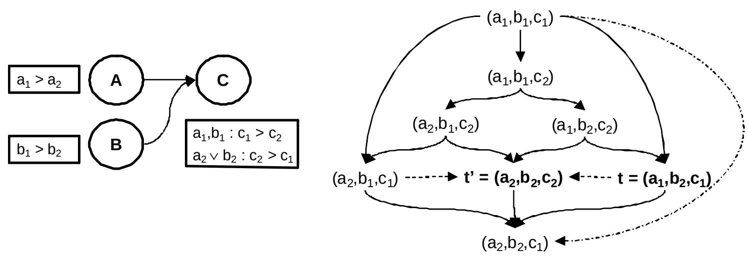


Fig. 3. A CP-net for which the cpu and stp semantics do not coincide

Is the stp semantics a "reasonable" one? We argue that stp is not completely exempt from problems, since it is unable to discover some *preference violations*. Refer to tuples t and t' in the above example and consider attribute C. Its CPT, written in the figure in a compact form, asserts that if $A = a_2$ or $B = b_2$ then preference is given to c_2 rather than to c_1. We have $t'[C] = c_2$ and $t[C] = c_1$, thus t' should be better than t on attribute C, yet stp is unable to discover it. This motivates the introduction of a new (weak) totalitarian semantics for interpreting the statements in a CPT.[4]

[2] For lack of space, proofs of formal results are omitted.

[3] A precise characterization of the CP-nets for which this occurs seems to be a difficult problem, since it depends not only on the net structure, but also on its CPT's.

[4] Indeed, this new semantics induces more preferences than the strong one from the CPT's. However, the net effect is that *less* preferences among tuples survive after the Pareto composition, as Theorem 2 proves. This is why we say it is "weak".

Definition 3 (Weak totalitarian Pareto semantics) *Let* $a_{i,1}, a_{i,2} \in dom(A_i)$ *and* t_1 *and* t_2 *be two tuples with* $t_1[A_i] = a_{i,1}$ *and* $t_2[A_i] = a_{i,2}$*. Let* P_i *be the parents of* A_i*, and* $p_1 = t_1[P_i]$*,* $p_2 = t_2[P_i]$*. If* $CPT(A_i)$ *includes statements (not necessarily distinct)* $\varphi_1 = p_1 : a_{i,1} > a_{i,2}$ *and* $\varphi_2 = p_2 : a_{i,1} > a_{i,2}$ *then* $(t_1, t_2) \in \Phi^*_{A_i,\mathsf{wt}}$*.*

The set of all preferences, Φ^*_{wtp}*, is the* n*-ary Pareto composition of the* $\Phi^*_{A_i,\mathsf{wt}}$ *sets, and the* weak totalitarian Pareto *(*wtp*) order* $\succ_{\mathsf{wtp}}$ *is the transitive closure of* Φ^*_{wt}*.*

Consider again Figure 3. In $CPT(C)$ there are two statements (once we write them in extended form), $\varphi_1 = a_1, b_2 : c_2 > c_1$ and $\varphi_2 = a_2, b_2 : c_2 > c_1$, from which we conclude, according to the above definition, that the pair $(t', t) \in \Phi^*_{C,\mathsf{wt}}$. Since $(t, t') \in \Phi^*_{A,\mathsf{wt}}$ still holds, it follows that $(t, t') \notin \Phi^*_{\mathsf{wtp}}$.

Given that we have redefined both statements' interpretation and the preference composition rule, the following is rather surprising:

Theorem 2 *For any complete acyclic CP-net* N *it is* $\succ_{\mathsf{cpu}} = \succ_{\mathsf{wtp}}$*.*

3.1 Incomplete CP-nets

Let us now analyze how wtp behaves on *incomplete* nets, which is the most relevant case for the DB scenarios we aim to consider. We start by showing that on incomplete CP-nets the equivalence of the cpu and wtp semantics breaks down (as required!):

Lemma 1 *For any, possibly incomplete, acyclic CP-net* N *it is* $\succ_{\mathsf{cpu}} \subseteq \succ_{\mathsf{wtp}}$*.*

For instance, in the example at the beginning of this section it is $t_1 = (it, in, low) \succ_{\mathsf{wtp}} t_3 = (it, out, high)$ even if the statement $it : in > out$ has not been specified. This follows since $(t_1, t_3) \in \Phi^*_{P,\mathsf{wt}}$ (t_1 is better on price than t_3) and no preferences over other attributes involve these two tuples. The fact that the optimal tuples in a relation r obtained from a CP-net N under the $\succ_{\mathsf{wtp}}$ semantics, denoted as $Opt_{\mathsf{wtp}}(r; N)$, are a subset of those of $\succ_{\mathsf{cpu}}$, $Opt_{\mathsf{cpu}}(r; N)$, is not a case.

Corollary 1. *For any acyclic CP-net* N *and any relation* r *it is* $Opt_{\mathsf{wtp}}(r; N) \subseteq Opt_{\mathsf{cpu}}(r; N)$*.*

The result immediately follows from $\succ_{\mathsf{cpu}}$ being a subset of $\succ_{\mathsf{wtp}}$, and can be refined in the case of complete relations, $r = dom(X)$.

Theorem 3 *For any acyclic CP-net* N *it is* $Opt_{\mathsf{wtp}}(dom(X); N) = Opt_{\mathsf{cpu}}(dom(X); N)$*.*

Besides above properties, how does CP-net incompleteness affect the wtp semantics? A first critical observation is that Definition 3 has to be properly extended in order to avoid cycles in the Φ^*_{wtp} graph.

Example 5. Consider the CP-net over attributes RestaurantType (R), Table (T), and SmokingArea (S), $dom(S) = \{yes, no\}$. As in Example 2, we have $it : in > out$ and $chn : out > in$, but now there is no preference on R (i.e., it and chn are not ordered). Preferences on S are conditional on T: if sitting inside, I do not want to stay in a smoking area ($in : no > yes$), but my preferences change should the table be outside ($out : yes > no$). According to Def. 3, we derive the following cycle of preferences:

1) $(it, in, yes) \succ_{\mathsf{wtp}} (it, out, yes)$ 2) $(it, out, yes) \succ_{\mathsf{wtp}} (chn, out, no)$
3) $(chn, out, no) \succ_{\mathsf{wtp}} (chn, in, no)$ 4) $(chn, in, no) \succ_{\mathsf{wtp}} (it, in, yes)$

Notice that 1) and 3) are also in $\succ_{\mathsf{cpu}}$, whereas this is not the case for 2) and 4). □

A simple solution to avoid above problem would be to inhibit ordering tuples when they have unordered values in some attributes. This is exactly what the cpu semantics would do and, as argued at the beginning of Section 3, is truly unsatisfactory.

The problem of allowing tuples to be ordered even if their attribute values are not completely ordered while, at the same time, preserving the strict partial order properties of $\succ_{\mathsf{wtp}}$, can be solved by: a) slightly revising the notion of what "being better on an attribute" means, and b) limiting the type of incompleteness in the CPT's. We discuss the two issues separately.

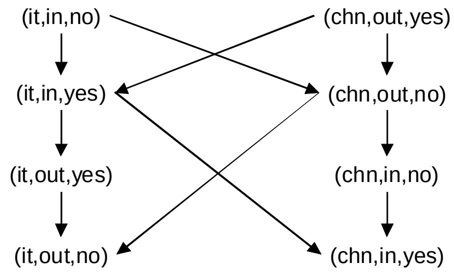
Consider first issue a). Referring to preference 2) in Example 5 (similar arguments hold for 4)), we see that the two tuples can be ordered only on S. However, looking at attribute T we might argue that, *since the parent values are unordered*, one should better consider comparing (it, out) and (chn, out) *as a whole*. Under this perspective, it seems natural to say that (chn, out) is *better* than (it, out), since the latter does not respect the corresponding statement $it : in > out$. In other terms, when being unable to order parents's values, one should look at how good is the attribute value under consideration (out) within the two different contexts (it and chn).

Let us now turn to issue b), i.e., the type of incompleteness in the CPT's, and, for the sake of definiteness, consider first a CP-net in which all attributes have no parents (thus, all preferences are unconditional). In the most "liberal" case, the statements in $CPT(A_i)$ might induce a generic strict partial order on $dom(A_i)$. However, it is well known [Cho02] that the Pareto composition of strict partial orders is *not* a strict partial order anymore. As a simple example, if we have attributes A and B, and statements $a_1 > a_2$, $a_3 > a_4$, $b_2 > b_3$, and $b_4 > b_1$, these would lead to the cycle $(a_1, b_1) \succ_{\mathsf{wtp}} (a_2, b_2) \succ_{\mathsf{wtp}} (a_3, b_3) \succ_{\mathsf{wtp}} (a_4, b_4) \succ_{\mathsf{wtp}} (a_1, b_1)$.

This immediately rules out the possibility of having an uncontrolled amount of incompleteness. On the positive side, if the $CPT(A_i)$ induce *weak orders*, their Pareto composition *is* a strict partial order. We remind that a weak order is a strict partial order that is also negatively transitive, i.e., for each triple of values a, b, c, if $a \not> b$ and $b \not> c$, then $a \not> c$. Clearly, a total order is also a weak order, but the converse is not necessarily true. More intuitively, a weak order can be viewed as a "linear order with ties". This is also to say that if $a_{i,1}$ and $a_{i,2}$ are not ordered, and $a_{i,1} > a_{i,3}$, then it should be $a_{i,2} > a_{i,3}$ as well. When attribute A_i has parents P_i, this restriction applies to each value of P_i. Clearly, if $p, p' \in dom(P_i)$ then the two weak orders they induce on $dom(A_i)$ need not to be the same. Finally, note that if no statement matching p is present in $CPT(A_i)$, then this induces a weak order in which all values are unordered.

Combining above considerations leads to extend the wtp semantics for the case of unordered values so that $\succ_{\mathsf{wtp}}$ is always a strict partial order. For lack of space we do not present here the formal definition, rather we show in the following figure the (transitively reduced) preference graph it induces for the CP-net in Example 5.

First, one should observe that tuples with unordered values can still be compared, yet no cycles arise. Second, it is interesting to see that optimal tuples for the two R contexts also dominate sub-optimal tuples in the other context (e.g., $(chn, out, yes) \succ_{\mathsf{wtp}} (it, in, yes)$). This is a further evidence that sub-optimal results are excluded when the relation is incomplete. Finally, it can be seen that $(chn, out, no) \not\succ_{\mathsf{wtp}} (it, out, yes)$, as expected, since the former is better on T, as explained above, whereas the latter is (still) better on S.



4 Conclusions

In this paper we have considered CP-nets as a viable tool to express user preferences in database queries, and have shown that their strength in compactly representing conditional preferences can be decoupled from the ceteris paribus (cpu) semantics. Our results show that one can use an alternative, weak totalitarian (wtp), semantics that overcomes the basic limitation of cpu when preferences are partially specified.

Being this the first work that investigates the use of (incomplete) CP-nets for querying databases, many issues need to be investigated. In particular, we need to develop a complete proof procedure to determine when $t \succ_{\mathsf{wtp}} t'$, which is at the basis of all algorithms for computing the optimal tuples in a relation [Cho02,Kie02,Cia06]. Second, it would be interesting to provide a characterization of optimal tuples in terms of the incompleteness of the CP-net (for the cpu semantics the optimal results of an incomplete CP-net N are just the union of the optimal results of the possible completions of N). Third, we would like to better understand the implications of having an explicit distinction between DB and context attributes, the intuition being that the latter are, for any given query, set to constant values (or to a set of constants).

References

[BBD+04] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research (JAIR)*, 21:135–191, 2004.

[BBHP99] Craig Boutilier, Ronen I. Brafman, Holger H. Hoos, and David Poole. Reasoning With Conditional Ceteris Paribus Preference Statements. In UAI '99, 71–80, 1999.

[Cho02] Jan Chomicki. Querying with Intrinsic Preferences. In EDBT 2002, 34–51, 2002.

[Cia06] Paolo Ciaccia. Processing Preference Queries in Standard Database Systems. In ADVIS 2006, 1–12, 2006. Invited paper.

[GLTW05] Judy Goldsmith, Jérôme Lang, Miroslaw Truszczynski, and Nic Wilson. The Computational Complexity of Dominance and Consistency in CP-nets. In IJCAI-05, 144–149, 2005.

[Kie02] Werner Kießling. Foundations of Preferences in Database Systems. In VLDB 2002, 311–322, 2002.

[Wil04] Nic Wilson. Extending CP-Nets with Stronger Conditional Preference Statements. In AAAI 2004, 735–741, 2004.