Condensed Representations for Sets of Mining Queries

Arnaud Giacometti[1],
Dominique Laurent[1], and Cheikh Talibouya Diop[1,2]

[1] LI, Université de Tours, 41000 Blois, FRANCE
{giaco,laurent}@univ-tours.fr
[2] Université Gaston Berger, Saint-Louis, SENEGAL
cdiop@ugb.sn

Abstract. In this paper, we propose a general framework for condensed representations of sets of mining queries. To this end, we adapt the standard notions of maximal, closed and key patterns introduced in previous works, including those dealing with condensed representations. Whereas these previous works concentrate on condensed representations of the answer to a *single* mining query, we consider the more general case of *sets* of mining queries defined by monotonic and anti-monotonic selection predicates.

1 Introduction

In the past decades, the problem of discovery of interesting patterns in large databases has motivated many research efforts. Whereas these works have focussed mainly on the efficiency of the algorithms [1,6,12,16], some other issues have been recently considered, among which the problem of efficient storage of the result of an extraction [4,14,15]. In this paper, we propose a general framework for condensed representations of the answers to a *set* of mining queries. More precisely, we assume that we are given:

1. A set $\mathbf{\Delta}$ of all data sets Δ from which the patterns are to be discovered.
2. A partially ordered set of patterns $\mathbb{L}$, where the partial ordering is denoted by $\preceq$.
3. A set of selection predicates $\mathbb{Q}$, a selection predicate being a boolean function defined over $\mathbb{L} \times \mathbf{\Delta}$.
4. A set of measure functions $\mathbb{F}$, a measure function being a real function defined over $\mathbb{L} \times \mathbf{\Delta}$.

Moreover, given a selection predicate q and a data set Δ in $\mathbf{\Delta}$, we say that a pattern φ in $\mathbb{L}$ is *interesting in* Δ *with respect to* q if $q(\varphi, \Delta)$ has the value *true*. Any selection predicate is also called a *simple mining query* and the set of interesting patterns in Δ with respect to q, denoted by $sol(q/\Delta)$, is called the *answer of* q in Δ.

R. Meo et al. (Eds.): Database Support for Data Mining Applications, LNAI 2682, pp. 250–269, 2004.
© Springer-Verlag Berlin Heidelberg 2004

We call *extended mining query* any pair of the form (q, f) where q is in $\mathbb{Q}$ and f is in $\mathbb{F}$. The answer in Δ to an extended mining query (q, f), denoted by $ans(q, f/\Delta)$, is the set of pairs $(\varphi, f(\varphi, \Delta))$ such that φ is in $sol(q/\Delta)$.

In the following example, that will be used as a running example throughout the paper, we illustrate these notions in the classical association rule mining problem of [1].

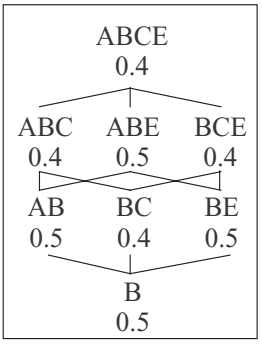
Running Example 1. *Given a set of items Items, the set of patterns* $\mathbb{L}$ *considered in our approach is the set of all subsets of Items, i.e.,* $\mathbb{L} = 2^{Items}$. *Moreover, the partial ordering over* $\mathbb{L}$ *that we consider is set inclusion: given two patterns* φ *and* φ' *in* $\mathbb{L}$, *we say that* $\varphi \preceq \varphi'$ *if* $\varphi' \subseteq \varphi$.

In this context, a data set Δ *is defined by a set of transactions* Tr *and a function it from* Tr *to* $\mathbb{L}$. *Given a transaction* $x \in Tr$, $it(x)$ *is the set of items in transaction* x. *The support of a pattern is an example of measure function of* $\mathbb{F}$. *More precisely, for every pattern* φ, *the support of* φ *in* Δ, *denoted by* $sup(\varphi, \Delta)$, *is defined by:*

$$sup(\varphi, \Delta) = |\{x \in Tr \mid it(x) \preceq \varphi\}|/|Tr|.$$

Note that, given a minimal support threshold minsup, we can consider the selection predicate q defined by: for every pattern $\varphi \in \mathbb{L}$, $q(\varphi, \Delta) = true$ *if* $sup(\varphi, \Delta) \geq minsup$.

In the rest of the paper, we consider the case where the set of items is $Items = \{A, B, C, D, E\}$ *and where the set of transactions is* $Tr = \{1, 2, \ldots, 10\}$. *For the sake of simplicity, sets of items are denoted by the concatenation of their elements, e.g. the set of items* $\{A, B, C\}$ *is denoted by* ABC. *The function it from* Tr *to* $\mathbb{L}$ *that defines the data set* Δ *is represented in the table of Figure 1.*



Tr	Set of Items
1	A
2	DE
3	ABCE
4	ABE
5	ABCDE
6	ACD
7	ABCE
8	AE
9	ABCDE
10	CD

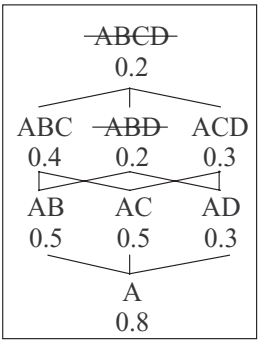


Fig. 1. Example of data set and sub-lattices of interesting patterns

Let m_1, m_2, a_1 *and* a_2 *be selection predicates defined for every pattern* φ *in* $\mathbb{L}$ *by:*

- $m_1(\varphi, \Delta) = true$ *if* $B \subseteq \varphi$, $m_2(\varphi, \Delta) = true$ *if* $A \subseteq \varphi$.
- $a_1(\varphi, \Delta) = \overline{a_1}(\varphi, \Delta) \wedge \widetilde{a_1}(\varphi, \Delta)$, *where* $\overline{a_1}(\varphi, \Delta) = true$ *if* $sup(\varphi, \Delta) \geq 0.4$, *and* $\widetilde{a_1}(\varphi, \Delta) = true$ *if* $\varphi \subseteq ABCE$.
- $a_2(\varphi, \Delta) = \overline{a_2}(\varphi, \Delta) \wedge \widetilde{a_2}(\varphi, \Delta)$ *where* $\overline{a_2}(\varphi, \Delta) = true$ *if* $sup(\varphi, \Delta) \geq 0.3$, *and* $\widetilde{a_2}(\varphi, \Delta) = true$ *if* $\varphi \subseteq ABCD$.

$q_1 = m_1 \wedge a_1$ *and* $q_2 = m_2 \wedge a_2$ *are simple mining queries. Moreover, it is easy to see from the table in Figure 1 that:*

$$sol(m_1 \wedge a_1/\Delta) = \{B, AB, BC, BE, ABC, ABE, BCE, ABCE\}$$
$$sol(m_2 \wedge a_2/\Delta) = \{A, AB, AC, AD, ABC, ACD\}$$

On the other hand, (q_1, sup) *and* (q_2, sup) *are examples of extended mining queries, and we have:*

$$ans(q_1, sup/\Delta) = \{(B, 0.5), (AB, 0.5), (BC, 0.4), (BE, 0.5), (ABC, 0.4), (ABE, 0.5), (BCE, 0.4), (ABCE, 0.4)\}$$
$$ans(q_2, sup/\Delta) = \{(A, 0.5), (AB, 0.5), (AC, 0.5), (AD, 0.3), (ABC, 0.4), (ACD, 0.3)\}$$

The answers in Δ *of* (q_1, sup) *and* (q_2, sup) *are also represented in Figure 1.* □

In the case of a simple mining query q, we recall that $sol(q/\Delta)$ can be computed *without any access* to Δ if only the maximal and minimal elements of $sol(q/\Delta)$ (with respect to the partial ordering on $\mathbb{L}$) are known [9,12]. Indeed, denoting these sets by $G(q/\Delta)$ and $S(q/\Delta)$, respectively, we know that a pattern φ is in $sol(q/\Delta)$ if and only if there exist $\varphi_g \in G(q/\Delta)$ and $\varphi_s \in S(q/\Delta)$ such that $\varphi_s \preceq \varphi \preceq \varphi_g$. Since $G(q/\Delta) \cup S(q/\Delta) \subseteq sol(q/\Delta)$, we say that $\{G(q/\Delta), S(q/\Delta)\}$ is a *condensed representation* of $sol(q/\Delta)$.

In our Running Example 1, it can be seen that $G(q_1/\Delta) = \{B\}$ and $S(q_1/\Delta) = \{ABCE\}$. Thus, $sol(q_1/\Delta)$ is the set of all itemsets φ such that $B \subseteq \varphi \subseteq ABCE$, and this can be computed independently from Δ.

On the other hand, in the case of extended mining queries, we adapt the notions of closed patterns and of key patterns ([3,4,16]) to our formalism, which allows us to obtain condensed representations of the set $ans(q, f/\Delta)$ (see Section 3.3). For instance, in our Running Example 1, for q_1 and the function *sup*, it will be seen that the answer $ans(q_1, f/\Delta)$ can be computed *without any access to* Δ, from the three sets $\{B\}$, $\{ABCE\}$, and $\{(ABE, 0.5), (ABCE, 0.4)\}$. In this case, we say that these three sets constitute an *extended condensed representation* of $ans(q_1, sup/\Delta)$.

As the main contribution of this paper, we consider the case of *sets* of mining queries (simple or extended). Noting that the union of condensed representations of different mining queries is *not* a condensed representation of the corresponding set of mining queries ([9]), we extend the notions of maximal, minimal, closed and key patterns to the case of sets of mining queries. Then, we propose condensed representations for such sets, in the sense that, given a set $\mathcal{Q}$ of mining queries, the answers in Δ of the queries in $\mathcal{Q}$ can be computed based *only* on the condensed representation, i.e., without any access to the data set Δ.

In the case of our Running Example 1, consider the set of simple mining queries $\mathcal{Q} = \{q_1, q_2\}$. Then, it will be seen in Section 4 that the sets of pairs

$\{(ABCE, q_1), (ACD, q_2)\}$ and $\{(B, q_1), (A, q_2)\}$ constitute a condensed representation of $sol(q_1/\Delta)$ and $sol(q_2/\Delta)$. We would like to emphasize that in the first set above, the maximal element ABC in $sol(q_2/\Delta)$ does not appear in the given condensed representation. Thus, in condensed representations of sets of mining queries, some maximal or minimal elements with respect to single mining queries can be omitted.

Comparing our approach to that of [6,7], we note that in [6,7] the authors consider conjunctive queries made of monotonic and anti-monotonic primitives, which correspond to what we call simple mining queries. Moreover, it is shown in [6,7] that the answer to one such query can be represented by its minimal and maximal elements only. However, contrary to the present paper, the case of *sets* of queries is not considered.

On the other hand, in [4], the authors also consider conjunctive queries. They use a caching technique to store condensed representations of the answers to these queries together with their supports. In our terminology, this corresponds to extended mining queries. However, in [4], each answer is condensed separately and stored in the cache, whereas our approach allows to benefit from relationships between the queries in order to further condense the answers to the queries.

Thus, our approach can be seen as an extension of [6,7] and [4]. In this paper, however, we do not consider computational aspects, such as the computation and the maintenance of condensed representations.

The paper is organized as follows: In Section 2, we give the formal definitions of the basic concepts of our approach, and in Section 3, mining queries, condensed representations as well as maximal, closed and key patterns are introduced. Section 4 deals with condensed representations of sets of mining queries. In Section 5, we conclude the paper and we propose further research directions based on this work.

2 Basic Definitions

In our formalism, we assume that we are given:

1. A set $\mathbf{\Delta}$ of all data sets from which the patterns are to be discovered. For instance, $\mathbf{\Delta}$ can be thought of as being the set of all instances of a given relation schema.
2. A set of patterns $\mathbb{L}$ and a partial ordering $\preceq$ over $\mathbb{L}$. Given two patterns φ_1, φ_2 in $\mathbb{L}$, we say that φ_1 is more specific than φ_2 (or that φ_2 is more general than φ_1) if we have $\varphi_1 \preceq \varphi_2$.
3. A set of selection predicates $\mathbb{Q}$, a selection predicate $q \in \mathbb{Q}$ being a boolean function defined over $\mathbb{L} \times \mathbf{\Delta}$. Moreover, given a pattern φ in $\mathbb{L}$ and a data set Δ in $\mathbf{\Delta}$, we say that φ is interesting in Δ with respect to q if $q(\varphi, \Delta) = true$.
4. A set of measure functions $\mathbb{F}$, a measure function being a function defined from $\mathbb{L} \times \mathbf{\Delta}$ to $\Re$.

Now, we define when a selection predicate is independent from $\mathbf{\Delta}$.

Definition 1 - Data Independency. *Let q be a selection predicate in $\mathbb{Q}$. q is* data independent *(or* independent *for short) if there exists a function $\widetilde{q}$ from $\mathbb{L}$ to $\{true, false\}$ such that for every data set Δ in $\mathbf{\Delta}$ and every pattern φ in $\mathbb{L}$, $q(\varphi, \Delta) = \widetilde{q}(\varphi)$.*

In our Running Example 1, it is easy to see that the selection predicates m_1, m_2, $\widetilde{a_1}$ and $\widetilde{a_2}$ are independent. In the following, we denote by $\widetilde{\mathbb{Q}}$ the set of all independent selection predicates, and by $\overline{\mathbb{Q}}$ the complement of $\widetilde{\mathbb{Q}}$ in $\mathbb{Q}$, i.e., $\overline{\mathbb{Q}} = \mathbb{Q} \setminus \widetilde{\mathbb{Q}}$.

In this paper, we consider only selection predicates that are monotonic or anti-monotonic, and measure functions that are monotonic increasing.

Definition 2 - Monotonicity. *Let q be a selection predicate.*

- *q is* monotonic *if for every data set Δ in $\mathbf{\Delta}$ and every pair of patterns (φ_1, φ_2) in $\mathbb{L}^2$, we have:*
$$\text{if } \varphi_1 \preceq \varphi_2 \text{ and } q(\varphi_2, \Delta) = true, \text{ then } q(\varphi_1, \Delta) = true.$$
- *q is* anti-monotonic *if for every data set Δ in $\mathbf{\Delta}$ and every pair of patterns (φ_1, φ_2) in $\mathbb{L}^2$, we have:*
$$\text{if } \varphi_1 \preceq \varphi_2 \text{ and } q(\varphi_1, \Delta) = true, \text{ then } q(\varphi_2, \Delta) = true.$$

Let f be a measure function. f is a monotonic increasing *function if for every data set Δ in $\mathbf{\Delta}$ and every pair of patterns (φ_1, φ_2) in $\mathbb{L}^2$, we have:*
$$\text{if } \varphi_1 \preceq \varphi_2, \text{ then } f(\varphi_1, \Delta) \leq f(\varphi_2, \Delta).$$

In our Running Example 1, it is easy to see that the selection predicates m_i $(i = 1, 2)$ are monotonic, whereas the selection predicates $\overline{a_i}$ and $\widetilde{a_i}$ $(i = 1, 2)$ are anti-monotonic. Moreover, the measure function *sup* is an example of monotonic increasing measure function.

In the following, we denote by $\mathbb{A}$ the set of all anti-monotonic selection predicates and by $\mathbb{M}$ the set of all monotonic selection predicates. Moreover, we denote by $\widetilde{\mathbb{A}}$ (respectively $\widetilde{\mathbb{M}}$) the set of all selection predicates in $\mathbb{A}$ (respectively $\mathbb{M}$) that are independent, and by $\overline{\mathbb{A}}$ (respectively $\overline{\mathbb{M}}$) the set of all selection predicates in $\mathbb{A}$ (respectively $\mathbb{M}$) that are not independent. Finally, we denote by $\mathbb{I}$ the set of all monotonic increasing measure functions.

In our approach, selection predicates are compared according to the following definition.

Definition 3 - Selectivity. *Let q_1 and q_2 be two selection predicates. q_1 is* more selective *than q_2, denoted by $q_1 \sqsubseteq q_2$, if for every data set Δ in $\mathbf{\Delta}$ and every pattern φ in $\mathbb{L}$, we have: if $q_1(\varphi, \Delta) = true$, then $q_2(\varphi, \Delta) = true$.*

In the context of our Running Example 1, let α_1 and α_2 be two support thresholds. For $i = 1, 2$, let a_i be the selection predicate defined by: for every pattern φ, $q_i(\varphi, \Delta) = true$ if $sup(\varphi, \Delta) \geq \alpha_i$. It is easy to see that if $\alpha_2 \geq \alpha_1$, then $q_2 \sqsubseteq q_1$.

In the rest of the paper, we consider a *fixed* data set Δ in $\mathbf{\Delta}$. Therefore, for notational convenience, we shall omit Δ in the subsequent definitions and propositions. For instance, referring to the previous two definitions, $q(\varphi, \Delta)$ and $f(\varphi, \Delta)$ will be simply denoted by $q(\varphi)$ and $f(\varphi)$, respectively.

3 Mining Query and Condensed Representations

3.1 Basic Definitions

In our approach, we define two types of mining query.

Definition 4 - Mining Query. *A* simple mining query *is a selection predicate q. Given a data set Δ, the* answer of q in Δ, *denoted by $sol(q)$, is defined by:*

$$sol(q) = \{\varphi \in \mathbb{L} \mid q(\varphi) = true\}.$$

$sol(q)$ denotes the set of all interesting patterns *in $\mathbb{L}$ with respect to q.*

An extended mining query *is a pair (q, f) where q is a selection predicate and f is a measure function. Given a data set Δ, the* answer of (q, f) in Δ, *denoted by $ans(q, f)$, is defined by:*

$$ans(q, f) = \{(\varphi, f(\varphi)) \mid \varphi \in sol(q)\}.$$

Note that an algorithm proposed in [5] can compute directly $sol(q)$ and $ans(q, f)$ if $q = m \wedge a$ with $m \in \mathbb{M}$ and $a \in \mathbb{A}$.

Let $Y = \{(y_1^i, y_2^i, \ldots, y_n^i) \mid i = 1, \ldots, p\}$ be a set of tuples whose first elements are patterns in $\mathbb{L}$. The projection of Y on $\mathbb{L}$, denoted by $\pi_{\mathbb{L}}(Y)$, is defined by: $\pi_{\mathbb{L}}(Y) = \{y_1^1, y_1^2, \ldots, y_1^p\}$. We note that $\pi_{\mathbb{L}}(Y) \subseteq \mathbb{L}$, and that for every $q \in \mathbb{Q}$ and $f \in \mathbb{F}$, $sol(q) = \pi_{\mathbb{L}}(ans(q, f))$.

We now introduce the notion of condensed representation.

Definition 5 - Condensed Representation. *Let $X_1, \ldots, X_K$ be sets of patterns, i.e., $X_k \subseteq \mathbb{L}$ $(k = 1, \ldots, K)$. Given a mining query $q \in \mathbb{Q}$ and a data set Δ, $\{X_1, \ldots, X_K\}$ is a* condensed representation *of $sol(q)$, denoted by $X_1, \ldots, X_K \models sol(q)$, if:*

- $(X_1 \cup \ldots \cup X_K) \subseteq sol(q)$, *and*
- *there exists a function F independent from Δ such that:* $sol(q) = F(X_1, \ldots, X_K)$.

Let Y be a set of pairs (φ, α) where φ is a pattern in $\mathbb{L}$ and α is a real. Given an extended mining query $(q, f) \in \mathbb{Q} \times \mathbb{F}$ and a data set Δ, $\{X_1, \ldots, X_K, Y\}$ is an extended condensed representation *of $ans(q, f)$, denoted by $X_1, \ldots, X_K, Y \models_e ans(q, f)$, if:*

- $(X_1 \cup \ldots \cup X_K \cup \pi_{\mathbb{L}}(Y)) \subseteq \pi_{\mathbb{L}}(ans(q, f))$, *and*
- *there exists a function F independent from Δ such that:* $ans(q, f) = F(X_1, \ldots, X_K, Y)$.

Given a simple mining query q and a measure function f, we now consider condensed representations of $sol(q)$ and extended condensed representations of $ans(q, f)$.

3.2 Maximal Patterns

In this paper, we consider only simple mining queries that are defined by conjunction of anti-monotonic and monotonic selection predicates. In this case, the answer of a simple mining query can be represented by its most specific and most general patterns [9,12].

Definition 6. *Let $q = m \wedge a$ be simple mining queries with $m \in \mathbb{M}$ and $a \in \mathbb{A}$.*

- *The set of* most specific patterns in $sol(q)$*, denoted by $S(q)$, is defined by:*
$$S(q) = min_{\preceq}(sol(q)) = \{\varphi \in sol(q) \mid (\nexists \varphi' \in sol(q))(\varphi' \prec \varphi)\}.$$
- *The set of* most general patterns in $sol(q)$*, denoted by $G(q)$, is defined by:*
$$G(q) = max_{\preceq}(sol(q)) = \{\varphi \in sol(q) \mid (\nexists \varphi' \in sol(q))(\varphi \prec \varphi')\}.$$

The following lemma, whose easy proof is omitted, shows that $sol(q)$ can be computed from $S(q)$ and $G(q)$.

Lemma 1. *Let $q = m \wedge a$ be a simple mining query with $m \in \mathbb{M}$ and $a \in \mathbb{A}$. We have: $sol(q) = \{\varphi \in \mathbb{L} \mid (\exists \varphi_s \in S(q))(\exists \varphi_g \in G(q))(\varphi_s \preceq \varphi \preceq \varphi_g)\}$.*

Therefore, we have the following proposition.

Proposition 1. *Let $q = m \wedge a$ be a simple mining query with $m \in \mathbb{M}$ and $a \in \mathbb{A}$. The set $\{G(q),\ S(q)\}$ is a condensed representation of $sol(q)$, i.e., $G(q),\ S(q) \models sol(q)$.*

Proof: *Let F be the function defined by: $F(X_1, X_2) = \{\varphi \in \mathbb{L} | (\exists \varphi_1 \in X_1)(\exists \varphi_2 \in X_2)(\varphi_1 \preceq \varphi \preceq \varphi_2)\}$. Using Lemma 1, we have $sol(q) = F(S(q), G(q))$. Moreover, F is independent from the data set Δ since $\preceq$ does not depend on Δ. Finally, we have $S(q) \cup G(q) \subseteq sol(q)$, which completes the proof.* □

We point out that algorithms for computing $S(m \wedge a)$ and $G(m \wedge a)$ directly have been proposed recently, e.g. the level-wise version space algorithm in [6].

Example 1. *Let q_1 and q_2 be the simple mining queries as given in our Running Example 1. We recall that: $G(q_1) = \{B\}$, $S(q_1) = \{ABCE\}$, $G(q_2) = \{A\}$, and $S(q_2) = \{ABC, ACD\}$. Applying Proposition 1, we have: $G(q_1),\ S(q_1) \models sol(q_1)$ and $G(q_2),\ S(q_2) \models sol(q_2)$.* □

It is important to note that $G(m) \models sol(m)$, $S(a) \models sol(a)$ and $sol(m \wedge a) = sol(m) \cap sol(a)$. Therefore, $sol(q)$ can be computed from $G(m)$ and $S(a)$. However, the set $\{G(m), S(a)\}$ is *not* always a condensed representation of $sol(q)$, since we can have $sol(q) \subset (G(m) \cup S(a))$. This is in particular the case for a query $q = m \wedge a$ such that $sol(q) = \emptyset$, $sol(m) \neq \emptyset$, and $sol(a) \neq \emptyset$.

On the other hand, in [12], the authors consider what they call the *positive* and the *negative* borders of the answer to a mining query. In our approach, given a simple mining query q, the corresponding positive and negative borders, respectively denoted by $Bd^+(q)$ and $Bd^-(q)$, can be defined as follows:

- $Bd^+(q) = \{S(q), G(q)\}$, where $S(q)$ and $G(q)$ have been defined previously
- $Bd^-(q) = \{S^-(q), G^-(q)\}$, where $S^-(q)$ and $G^-(q)$ are the following sets: $S^-(q) = max_{\preceq}\{\varphi \in sol(m) \mid \varphi \notin sol(a)\}$ and $G^-(q) = min_{\preceq}\{\varphi \in sol(a) \mid \varphi \notin sol(m)\}$.

Therefore, according to Definition 5, the positive border can be seen as a condensed representation of $sol(q)$, whereas the negative border can not. Indeed, although the sets $S^-(q)$ and $G^-(q)$ allow to recompte $sol(q)$ without any access to the data set, the first point of Definition 5 is not satisfied, since neither $S^-(q)$ nor $G^-(q)$ is a subset of $sol(q)$.

We shall not consider the case of negative borders in the rest of the paper, but we note in this respect that (i) storing $Bd^-(q)$ is not optimal in general (since its cardinality can be much greater than that of $sol(q)$), and (ii) $Bd^-(q)$ can be seen in our approach as a condensed representation of the set $sol(q) \cup Bd^-(q)$.

3.3 Closed and Key Patterns

In this section, we give alternative definitions of the notions of closed and key patterns introduced in [3,4,16]. To this end, given a measure function f, we consider the partial ordering $\leq_f$ defined for every pair of patterns (φ, φ') by:

$$\varphi \leq_f \varphi' \text{ if } \varphi \preceq \varphi' \text{ and } f(\varphi) = f(\varphi').$$

Definition 7. *Let q be a mining query in $\mathbb{Q}$ and f be a measure function in $\mathbb{F}$. Let Δ be a data set and φ be a pattern in $\mathbb{L}$.*

- *The set of all* interesting closed patterns *in Δ with respect to q and f, denoted by $SC(q, f)$, is defined by:*
$$SC(q, f) = min_{\leq_f}(sol(q)).$$
- *The set of all* interesting key patterns *in Δ with respect to q and f, denoted by $GK(q, f)$, is defined by:*
$$GK(q, f) = max_{\leq_f}(sol(q)).$$

It can be shown that our notions of interesting closed patterns and interesting key patterns coincide with those of [3,4,16] in the context of classical association rules mining [1].

Moreover, it is easily seen that for every extended mining query (q, f) with $q \in \mathbb{Q}$ and $f \in \mathbb{I}$, we have $S(q) \subseteq SC(q, f)$ and $G(q) \subseteq GK(q, f)$. More precisely, the following lemma holds.

Lemma 2. *Let q be a selection predicate in $\mathbb{Q}$ and f be a monotonic increasing measure function in $\mathbb{I}$. We have:*

$$S(q) = min_{\preceq}(SC(q, f)) \quad \textit{and} \quad G(q) = max_{\preceq}(GK(q, f)).$$

PROOF: *We first show that $S(q) \subseteq min_{\preceq}(SC(q, f))$. Let $\varphi \in S(q)$. There does not exist a pattern $\varphi' \in sol(q)$ such that $\varphi' \prec \varphi$. Therefore, there does not exist a pattern $\varphi' \in sol(q)$ such that $\varphi' \prec \varphi$ and $f(\varphi') = f(\varphi)$, which shows*

that $\varphi \in SC(q,f)$. Assume now that $\varphi \notin min_{\preceq}(SC(q,f))$. Then, there exists $\varphi' \in SC(q,f)$ such that $\varphi' \prec \varphi$, which is in contradiction with the hypothesis $\varphi \in S(q)$. Hence, we have $S(q) \subseteq min_{\preceq}(SC(q,f))$.

Now, we show that $min_{\preceq}(SC(q,f)) \subseteq S(q)$. Let $\varphi \in min_{\preceq}(SC(q,f))$. Assume that $\varphi \notin S(q)$. Then, there exists $\varphi' \in S(q)$ such that $\varphi' \prec \varphi$. Since it has been shown above that $S(q) \subseteq min_{\preceq}(SC(q,f))$, we have that $\varphi' \in SC(q,f)$. This is in contradiction with the hypothesis $\varphi \in min_{\preceq}(SC(q,f))$. Hence, we have $min_{\preceq}(SC(q,f)) \subseteq S(q)$.

Thus the proof that $S(q) = min_{\preceq}(SC(q,f))$ is complete. In the same way, it can be shown that $G(q) = max_{\preceq}(GK(q,f))$, which completes the proof. □

The following lemma states that given any pattern φ in $sol(q)$, $f(\varphi)$ can be computed based on $SC(q,f)$ or $GK(q,f)$.

Lemma 3. *Let q be a selection predicate in $\mathbb{Q}$ and f be a monotonic increasing measure function in $\mathbb{I}$. For every interesting pattern φ in $sol(q)$, we have:*

- $f(\varphi) = max\{f(\varphi') \mid \varphi' \in SC(q,f)$ *and* $\varphi' \preceq \varphi\}$, *and*
- $f(\varphi) = min\{f(\varphi') \mid \varphi' \in GK(q,f)$ *and* $\varphi \preceq \varphi'\}$

where min and max denote respectively the minimum and maximum functions according to the standard ordering of real numbers.

Proof: *Let $\varphi \in sol(q)$ and $X(\varphi) = \{\varphi' \in sol(q) \mid \varphi' \preceq \varphi$ and $f(\varphi') = f(\varphi)\}$. Since $\varphi \in X(\varphi)$, we know that $Y(\varphi) = min_{\preceq}(X(\varphi))$ is not empty. Given any $\varphi'' \in Y(\varphi)$, assume that $\varphi'' \notin SC(q,f)$. Then, there exists $\varphi' \in sol(q)$ such that $\varphi' \prec \varphi''$ and $f(\varphi') = f(\varphi'')$, which shows that $\varphi' \in X(\varphi)$ and contradicts the fact that φ'' is minimal in $X(\varphi)$. Hence, there exists $\varphi_c \in SC(q,f)$ such that $\varphi_c \preceq \varphi$ and $f(\varphi_c) = f(\varphi)$.*

On the other hand, for every $\varphi' \in SC(q,f)$ such that $\varphi' \preceq \varphi$, we have $f(\varphi') \leq f(\varphi)$. Therefore, we have $f(\varphi) = max\{f(\varphi') \mid \varphi' \in SC(q,f)$ and $\varphi' \preceq \varphi\}$. Since the fact that $f(\varphi) = min\{f(\varphi') \mid \varphi' \in GK(q,f)$ and $\varphi \preceq \varphi'\}$ can be shown in the same way, the proof is complete. □

Let (q,f) be an extended mining query. In the following, we denote by $SC^*(q,f)$ and $GK^*(q,f)$ the sets defined by:

- $SC^*(q,f) = \{(\varphi, f(\varphi)) \mid \varphi \in SC(q,f)\}$, and
- $GK^*(q,f) = \{(\varphi, f(\varphi)) \mid \varphi \in GK(q,f)\}$.

The following proposition follows from the previous two lemmas.

Proposition 2. *Let $q = m \wedge a$ be a simple mining query with $m \in \mathbb{M}$, $a \in \mathbb{A}$, and let f be a monotonic increasing measure function in $\mathbb{I}$. The sets $\{S(q),\ G(q),\ SC^*(q,f)\}$ and $\{S(q),\ G(q),\ GK^*(q,f)\}$ are extended condensed representations of $ans(q,f)$, i.e.,*

$S(q),\ G(q),\ SC^*(q,f) \models_e ans(q,f)$ *and* $S(q),\ G(q),\ GK^*(q,f) \models_e ans(q,f)$.

PROOF: *Let F be the function defined by:* $F(X_1, X_2, Y) = \{(\varphi, \alpha) \in \mathbb{L}\times\Re \mid (\exists\varphi_1 \in X_1)(\exists\varphi_2 \in X_2)(\varphi_1 \preceq \varphi \preceq \varphi_2)$ *and* $\alpha = max\{\alpha' \mid (\exists\varphi' \in \mathbb{L})((\varphi', \alpha') \in Y \wedge \varphi' \preceq \varphi)\}$. *Using Lemma 1 and Lemma 3, we have* $ans(q, f) = F(S(q), G(q), SC^*(q, f))$. *Moreover, F is independent from the data set Δ since $\preceq$ does not depend on Δ, and* $S(q) \cup G(q) \cup SC(q, f) \subseteq sol(q)$. *Therefore,* $\{S(q),\ G(q),\ SC^*(q, f)\}$ *is an extended condensed representation of* $ans(q, f)$. *Since the fact that* $S(q),\ G(q),\ GK^*(q, f) \models_e ans(q, f)$ *can be shown in the same way, the proof is complete.* □

Example 2. *Let q_1 be the simple mining query as defined in our Running Example 1. We can see that:*

$GK^*(q_1, sup) = \{(B, 0.5),\ (BC, 0.4)\}$ *and*
$SC^*(q_1, sup) = \{(ABE, 0.5), (ABCE, 0.4)\}$.

Recalling that $G(q_1) = \{B\}$ *and* $S(q_1) = \{ABCE\}$, *and using Proposition 2, we obtain that* $S(q_1),\ G(q_1),\ SC^*(q_1, sup) \models_e ans(q_1, sup)$ *and that* $S(q_1),\ G(q_1),\ GK^*(q_1, sup) \models_e ans(q_1, sup)$. □

4 Condensed Representations of Sets of Mining Queries

In this section, we extend the notions of condensed representation and of extended condensed representation to the case of sets of mining queries.

4.1 Definitions

Definition 8 - Set of Mining Queries. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries. Given a data set Δ, the* answer of $\mathcal{Q}$ in Δ, *denoted by $sol(\mathcal{Q})$, is the set defined by:*

$$sol(\mathcal{Q}) = \bigcup_{q\in\mathcal{Q}} \{(\varphi, q) \mid \varphi \in sol(q)\}.$$

Let f be a measure function in $\mathbb{F}$. The answer of $(\mathcal{Q}, f)$ in Δ, *denoted by $ans(\mathcal{Q}, f)$, is the set defined by:*

$$ans(\mathcal{Q}, f) = \bigcup_{q\in\mathcal{Q}} \{(\varphi, q, f(\varphi)) \mid \varphi \in sol(q)\}.$$

Definition 9 - Condensed Representation. *Let $\mathcal{X}_1, \ldots, \mathcal{X}_K$ be sets of pairs (φ, q) where $\varphi \in \mathbb{L}$ and $q \in \mathbb{Q}$. Given a set of mining queries $\mathcal{Q}$ and a data set Δ, $\{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$ is a* condensed representation *of $sol(\mathcal{Q})$, denoted by $\mathcal{X}_1, \ldots, \mathcal{X}_K \models sol(\mathcal{Q})$, if:*

- $\pi_{\mathbb{L}}(\mathcal{X}_1) \cup \ldots \cup \pi_{\mathbb{L}}(\mathcal{X}_K) \subseteq \pi_{\mathbb{L}}(sol(\mathcal{Q}))$, *and*
- *there exists a function F independent from Δ such that:* $sol(\mathcal{Q}) = F(\mathcal{X}_1, \ldots, \mathcal{X}_K)$.

Let Y be a set of pairs (φ, α) where φ is a pattern in $\mathbb{L}$ and α is a real. Given a set of mining queries $\mathcal{Q}$, a measure function f and a data set Δ, $\{\mathcal{X}_1, \ldots, \mathcal{X}_K, Y\}$ is an extended condensed representation *of $ans(\mathcal{Q}, f)$, denoted by $\mathcal{X}_1, \ldots, \mathcal{X}_K, Y \models_e ans(\mathcal{Q}, f)$, if:*

– $\pi_{\mathbb{L}}(\mathcal{X}_1) \cup \ldots \cup \pi_{\mathbb{L}}(\mathcal{X}_K) \cup \pi_{\mathbb{L}}(Y) \subseteq \pi_{\mathbb{L}}(ans(\mathcal{Q}, f))$, *and*
– *there exists a function* F *independent from* Δ *such that:* $ans(\mathcal{Q}, f) = F(\mathcal{X}_1, \ldots, \mathcal{X}_K, Y)$.

Let $\mathcal{C} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_K\}$ and $\mathcal{C}' = \{\mathcal{Z}'_1, \ldots, \mathcal{Z}'_K\}$ be two condensed representations (extended or not) having the same cardinality K. We say that $\mathcal{C}$ is *more concise* than $\mathcal{C}'$ if there exists a permutation θ of $\{1, \ldots, K\}$ such that for every $i = 1, \ldots, K$, $\mathcal{Z}_i \subseteq \mathcal{Z}'_{\theta(i)}$.

Given a set of mining queries $\mathcal{Q}$ and a measure function f, we study condensed representations of $sol(\mathcal{Q})$ and extended condensed representations of $ans(\mathcal{Q}, f)$.

4.2 Maximal Patterns

Given a set of mining queries $\mathcal{Q} = \{q_1, \ldots, q_n\}$, it is well known [9] that, although $\{S(q_i),\ G(q_i)\}$ is a condensed representation of $sol(q_i)$, for every $i = 1, \ldots, n$, the set $\{S(q_1) \cup \ldots \cup S(q_n),\ G(q_1) \cup \ldots \cup G(q_n)\}$ is *not* a condensed representation of $sol(q_1) \cup \ldots \cup sol(q_n)$.

However, if for every φ in $S(q_1) \cup \ldots \cup S(q_n)$ or in $G(q_1) \cup \ldots \cup G(q_n)$, we keep track of the query q_i the pattern φ comes from, then $sol(\mathcal{Q})$ and $ans(\mathcal{Q}, f)$ can be condensed. For this reason, we define the sets $\mathcal{S}(\mathcal{Q})$ and $\mathcal{G}(\mathcal{Q})$ as follows:

Definition 10. *Let* $\mathcal{Q} = \{q_1, \ldots, q_n\}$ *be a set of mining queries* $q_i \in \mathbb{Q}$ *(*$i = 1, \ldots, n$*). The sets* $\mathcal{S}(\mathcal{Q})$ *and* $\mathcal{G}(\mathcal{Q})$ *are defined by:*

$$\mathcal{S}(\mathcal{Q}) = \bigcup_{q \in \mathcal{Q}} \{(\varphi, q) \mid \varphi \in S(q)\} \quad \textit{and} \quad \mathcal{G}(\mathcal{Q}) = \bigcup_{q \in \mathcal{Q}} \{(\varphi, q) \mid \varphi \in G(q)\}.$$

Given these definitions, we have the following proposition.

Proposition 3. *Let* $\mathcal{Q} = \{q_1, \ldots, q_n\}$ *be a set of mining queries* $q_i = m_i \wedge a_i$ *with* $m_i \in \mathbb{M}$ *and* $a_i \in \mathbb{A}$ *(*$i = 1, \ldots, n$*). The set* $\{\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q})\}$ *is a condensed representation of* $sol(\mathcal{Q})$*, i.e.,* $\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q}) \models sol(\mathcal{Q})$.

Proof: *Let* F *be the function defined by:* $F(\mathcal{X}_1, \mathcal{X}_2) = \{(\varphi, q) \in \mathbb{L} \times \mathbb{Q} \mid (\exists(\varphi_1, q_1) \in \mathcal{X}_1)(\exists(\varphi_2, q_2) \in \mathcal{X}_2)(q_1 = q_2 = q \textit{ and } \varphi_1 \preceq \varphi \preceq \varphi_2)\}$*. Based on Lemma 1, we can easily see that* $sol(\mathcal{Q}) = F(\mathcal{S}(\mathcal{Q}), \mathcal{G}(\mathcal{Q}))$*. Moreover,* F *is independent from the data set* Δ *since* $\preceq$ *does not depend on* Δ*. Finally, we have* $\mathcal{S}(\mathcal{Q}) \subseteq sol(\mathcal{Q})$ *and* $\mathcal{G}(\mathcal{Q}) \subseteq sol(\mathcal{Q})$*, which completes the proof.* □

Example 3. *In the context of our Running Example 1, let* $q_3 = m_3 \wedge a_3$ *and* $q_4 = m_4 \wedge a_4$ *where* m_3*,* m_4*,* a_3 *and* a_4 *are selection predicates defined for every pattern* $\varphi \in \mathbb{L}$ *by:*

– $m_3(\varphi, \Delta) = true$ *if* $A \subseteq \varphi$*, and* $m_4(\varphi, \Delta) = true$ *if* $AC \subseteq \varphi$*,*
– $a_3(\varphi, \Delta) = true$ *if* $sup(\varphi, \Delta) \geq 0.4$ *and* $\varphi \subseteq ABC$*, and* $a_4(\varphi, \Delta) = true$ *if* $sup(\varphi, \Delta) \geq 0.3$ *and* $\varphi \subseteq ABCD$*.*

We note that m_3 and m_4 are monotonic selection predicates such that $m_4 \sqsubseteq m_3$, whereas a_3 and a_4 are anti-monotonic selection predicates such that $a_3 \sqsubseteq a_4$. We can see that $S(q_3) = \{ABC\}$, $S(q_4) = \{ABC, ACD\}$, $G(q_3) = \{A\}$ and $G(q_4) = \{AC\}$. Considering $\mathcal{Q} = \{q_3, q_4\}$, we have:

$\mathcal{S}(\mathcal{Q}) = \{(ABC, q_3), (ABC, q_4), (ACD, q_4)\}$ *and* $\mathcal{G}(\mathcal{Q}) = \{(A, q_3), (AC, q_4)\}$

Using Proposition 3, we can see that: $\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q}) \models sol(\mathcal{Q})$. □

In what follows, we show how to define condensed representations of $sol(\mathcal{Q})$ that are *more concise* than $\{\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q})\}$.

Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = m_i \wedge a_i$ with $m_i \in \mathbb{M}$ and $a_i \in \mathbb{A}$ $(i = 1, \ldots, n)$. We define two partial pre-orderings, denoted by $\leq_{\mathbb{A}}$ and $\leq_{\mathbb{M}}$, as follows: for all (φ_i, q_i) and (φ_j, q_j) in $\mathbb{L} \times \mathcal{Q}$:

$$(\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi_j, q_j) \ \text{ if } \ \varphi_i \preceq \varphi_j \text{ and } a_i \sqsubseteq a_j$$
$$(\varphi_i, q_i) \leq_{\mathbb{M}} (\varphi_j, q_j) \ \text{ if } \ \varphi_i \preceq \varphi_j \text{ and } m_j \sqsubseteq m_i.$$

Then, we denote by $\Sigma(\mathcal{Q})$ the set of all minimal pairs in $\mathcal{S}(\mathcal{Q})$ with respect to $\leq_{\mathbb{A}}$. Similarly, we denote by $\Gamma(\mathcal{Q})$ the set of all maximal pairs in $\mathcal{G}(\mathcal{Q})$ with respect to $\leq_{\mathbb{M}}$. That is:

$$\Sigma(\mathcal{Q}) = min_{\leq_{\mathbb{A}}}(\mathcal{S}(\mathcal{Q})) \ \text{ and } \ \Gamma(\mathcal{Q}) = max_{\leq_{\mathbb{M}}}(\mathcal{G}(\mathcal{Q})).$$

The following lemma states that, for every $q \in \mathcal{Q}$, $sol(q)$ can be computed based on $\Sigma(\mathcal{Q})$ and $\Gamma(\mathcal{Q})$, only.

Lemma 4. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = m_i \wedge a_i$ with $m_i \in \mathbb{M}$ and $a_i \in \mathbb{A}$ $(i = 1, \ldots, n)$. For every q in $\mathcal{Q}$, we have:*

$$sol(q) = \{\varphi \in \mathbb{L} \mid \ (\exists (\varphi_i, q_i) \in \Sigma(\mathcal{Q}))((\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi, q)) \text{ and } (\exists (\varphi_j, q_j) \in \Gamma(\mathcal{Q}))((\varphi, q) \leq_{\mathbb{M}} (\varphi_j, q_j))\}.$$

Proof: *Let $X(q)$ be the set defined by:*

$$X(q) = \{\varphi \in \mathbb{L} \mid \ (\exists (\varphi_i, q_i) \in \Sigma(\mathcal{Q}))((\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi, q)) \text{ and } (\exists (\varphi_j, q_j) \in \Gamma(\mathcal{Q}))((\varphi, q) \leq_{\mathbb{M}} (\varphi_j, q_j))\}.$$

We first show that $X(q) \subseteq sol(q)$. Let $\varphi \in X(q)$. There exist $(\varphi_i, q_i) \in \Sigma(\mathcal{Q})$ and $(\varphi_j, q_j) \in \Gamma(\mathcal{Q})$ such that $(\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi, q)$ and $(\varphi, q) \leq_{\mathbb{M}} (\varphi_j, q_j)$.

On one hand, we know that $q_i(\varphi_i) = true$. Thus, we have $a_i(\varphi_i) = true$. It follows that $a(\varphi_i) = true$ since $a_i \sqsubseteq a$, and that $a(\varphi) = true$ since $\varphi_i \preceq \varphi$ and a is anti-monotonic.

On the other hand, we know that $q_j(\varphi_j) = true$. Thus, we have $m_j(\varphi_j) = true$. It follows that $m(\varphi_j) = true$ since $m_j \sqsubseteq m$, and that $m(\varphi) = true$ since $\varphi \preceq \varphi_j$ and m is monotonic. Therefore, we have $a(\varphi) = true$ and $m(\varphi) = true$, which shows that $\varphi \in sol(q)$. Hence, we have: $X(q) \subseteq sol(q)$.

Now, we show that $sol(q) \subseteq X(q)$. Let $\varphi \in sol(q)$. There exist $\varphi_s \in S(q)$ and $\varphi_g \in G(q)$ such that $\varphi_s \preceq \varphi \preceq \varphi_g$. Thus, we have $(\varphi_s, q) \in \mathcal{S}(\mathcal{Q})$ and $(\varphi_g, q) \in \mathcal{G}(\mathcal{Q})$.

Given the definitions of $\Sigma(\mathcal{Q})$ and $\Gamma(\mathcal{Q})$, there exist $(\varphi_i, q_i) \in \Sigma(\mathcal{Q})$ and $(\varphi_j, q_j) \in \Gamma(\mathcal{Q})$ such that $(\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi_s, q)$ and $(\varphi_g, q) \leq_{\mathbb{M}} (\varphi_j, q_j)$. Moreover, we have $(\varphi_s, q) \leq_{\mathbb{A}} (\varphi, q)$ since $\varphi_s \preceq \varphi$, and $(\varphi, q) \leq_{\mathbb{M}} (\varphi_g, q)$ since $\varphi \preceq \varphi_g$. Thus, $(\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi, q)$ and $(\varphi, q) \leq_{\mathbb{M}} (\varphi_j, q_j)$, which shows that $\varphi \in X(q)$. Hence, we have $sol(q) \subseteq X(q)$, which completes the proof. □

As a consequence of Lemma 4 above, we have the following theorem:

Theorem 1. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = m_i \wedge a_i$ with $m_i \in \mathbb{M}$ and $a_i \in \mathbb{A}$ $(i = 1, \ldots, n)$. The set $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ is a condensed representation of $sol(\mathcal{Q})$, i.e., $\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}) \models sol(\mathcal{Q})$.*

Moreover, $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ is more concise than $\{\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q})\}$.

Proof: *Let F be the function defined by: $F(\mathcal{X}_1, \mathcal{X}_2) = \{(\varphi, q) \in \mathbb{L} \times \mathbb{Q} \mid (\exists(\varphi_1, q_1) \in \mathcal{X}_1)((\varphi_1, q_1) \leq_{\mathbb{A}} (\varphi, q))$ and $(\exists(\varphi_2, q_2) \in \mathcal{X}_2)((\varphi, q) \leq_{\mathbb{M}} (\varphi_2, q_2))\}$. Using Lemma 4, we can easily see that $sol(\mathcal{Q}) = F(\Sigma(\mathcal{Q}), \Gamma(\mathcal{Q}))$. Moreover, F is independent from the data set Δ since $\preceq$ and $\sqsubseteq$ do not depend on Δ.*

It is easily seen that we have $\Sigma(\mathcal{Q}) \subseteq \mathcal{S}(\mathcal{Q}) \subseteq sol(\mathcal{Q})$ and $\Gamma(\mathcal{Q}) \subseteq \mathcal{G}(\mathcal{Q}) \subseteq sol(\mathcal{Q})$. Therefore, $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ is more concise than $\{\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q})\}$ and thus, the proof is complete. □

Example 4. *We recall from Example 3 that we have:*

$$\mathcal{S}(\mathcal{Q}) = \{(ABC, q_3), (ABC, q_4), (ACD, q_4)\} \text{ and } \mathcal{G}(\mathcal{Q}) = \{(A, q_3), (AC, q_4)\}.$$

Since $a_3 \sqsubseteq a_4$, we have $(ABC, q_3) \leq_{\mathbb{A}} (ABC, q_4)$. On the other hand, (A, q_3) and (AC, q_4) are not comparable with respect to $\leq_{\mathbb{M}}$. It follows that:

$$\Sigma(\mathcal{Q}) = \{(ABC, q_3), (ACD, q_4)\} \quad \text{and} \quad \Gamma(\mathcal{Q}) = \{(A, q_3), (AC, q_4)\}$$

Using Theorem 1, we can see that $\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}) \models \mathcal{S}(\mathcal{Q})$. Moreover, since $\Sigma(\mathcal{Q}) \subset \mathcal{S}(\mathcal{Q})$ and $\Gamma(\mathcal{Q}) \subseteq \mathcal{G}(\mathcal{Q})$, $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ is more concise than $\{\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q})\}$. □

We end this subsection by showing how to optimize the computation of $\Sigma(\mathcal{Q})$ (respectively $\Gamma(\mathcal{Q})$) by stating that two pairs (φ_i, q_i) and (φ_j, q_j) in $\mathcal{S}(\mathcal{Q})$ (respectively $\mathcal{G}(\mathcal{Q})$) cannot be comparable with respect to $\leq_{\mathbb{A}}$ (respectively $\leq_{\mathbb{M}}$) if $\varphi_i \neq \varphi_j$.

Indeed, based on this result, it turns out that the computation of $\Sigma(\mathcal{S}) = min_{\leq_{\mathbb{A}}}(\mathcal{S}(\mathcal{Q}))$ (respectively $\Gamma(\mathcal{S}) = max_{\leq_{\mathbb{M}}}(\mathcal{G}(\mathcal{Q}))$) only requires to compare the pairs of $\mathcal{S}(\mathcal{Q})$ (respectively $\mathcal{G}(\mathcal{Q})$) that contain the same pattern.

Proposition 4. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = m_i \wedge a_i$ with $m_i \in \mathbb{M}$ and $a_i \in \mathbb{A}$ $(i = 1, \ldots, n)$.*

If (φ_i, q_i) and (φ_j, q_j) are two pairs in $\mathcal{S}(\mathcal{Q})$ (respectively $\mathcal{G}(\mathcal{Q})$) such that $(\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi_j, q_j)$ (respectively such that $(\varphi_i, q_i) \leq_{\mathbb{M}} (\varphi_j, q_j)$), then we have $\varphi_i = \varphi_j$.

Proof: *Let (φ_i, q_i) and (φ_j, q_j) be two pairs in $\mathcal{S}(\mathcal{Q})$ such that $(\varphi_i, q_i) \leq_{\mathbb{A}} (\varphi_j, q_j)$. Since $q_i(\varphi_i) = true$, we have $a_i(\varphi_i) = true$ and $a_j(\varphi_i) = true$ since $a_i \sqsubseteq a_j$. On the other hand, since $q_j(\varphi_j) = true$, $\varphi_i \preceq \varphi_j$ and m_j is monotonic,*

we have $m_j(\varphi_j) = true$ and $m_j(\varphi_i) = true$. Therefore, we have $q_j(\varphi_i) = true$, meaning that $\varphi_i \in sol(q_j)$. Moreover, since φ_j is minimal in $sol(q_j)$ with respect to $\preceq$ and $\varphi_i \preceq \varphi_j$, we necessarily have $\varphi_i = \varphi_j$. It can be shown in the same way that if (φ_i, q_i) and (φ_j, q_j) are two pairs in $\mathcal{G}(\mathcal{Q})$ such that $(\varphi_i, q_i) \leq_{\mathbb{M}} (\varphi_j, q_j)$, then $\varphi_i = \varphi_j$. Thus the proof is complete. □

4.3 Closed and Key Patterns

In this subsection, we consider the case of extended condensed representations of a set $\mathcal{Q} = \{q_1, \ldots, q_n\}$ of simple mining queries with $q_i \in \mathbb{Q}$ $(i = 1, \ldots, n)$ involving a monotonic increasing measure function f in $\mathbb{I}$. To this end, recalling that $SC(q_i, f)$ is the set of all interesting closed patterns in Δ with respect to q_i and f $(i = 1, \ldots, n)$, we define the sets $\mathcal{SC}(\mathcal{Q}, f)$ and $\mathcal{SC}^*(\mathcal{Q}, f)$ as follows:

$$\mathcal{SC}(\mathcal{Q}, f) = min_{\leq_f}(\bigcup_{q \in \mathcal{Q}} SC(q, f)) \text{ and } \mathcal{SC}^*(\mathcal{Q}, f) = \{(\varphi, f(\varphi)) \mid \varphi \in \mathcal{SC}(\mathcal{Q}, f)\}$$

Example 5. *Let $\mathcal{Q} = \{q_1, q_2\}$ be the set of simple mining queries as defined in our Running Example 1. We have:*

- $SC(q_1, f) = \{ABCE, ABE\}$ *and* $SC(q_2, f) = \{ABC, ACD, AB, AC, A\}$,
- $\mathcal{SC}(\mathcal{Q}, f) = \{ABCE, ABE, ACD, AC, A\}$ *and*
- $\mathcal{SC}^*(\mathcal{Q}, f) = \{(ABCE, 0.4), (ABE, 0.5), (ACD, 0.3), (AC, 0.5), (A, 0.8)\}$. □

Based on Lemma 3, we can state the following proposition:

Proposition 5. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of simple mining queries with $q_i \in \mathbb{Q}$ $(i = 1, \ldots, n)$ and f be a monotonic increasing measure function in $\mathbb{I}$. For every $i = 1, \ldots, n$ and $\varphi \in sol(q_i)$, we have:*

$$f(\varphi) = max\{f(\varphi') \mid \varphi' \in \mathcal{SC}(\mathcal{Q}, f) \text{ and } \varphi' \preceq \varphi\}.$$

PROOF: *Let φ_i in $sol(q_i)$. Using Lemma 3, we know that:*

$$f(\varphi_i) = max\{f(\varphi_i') \mid \varphi_i' \in SC(q_i, f) \text{ and } \varphi_i' \preceq \varphi_i\}$$

Let $\varphi_i' \in SC(q_i, f)$ such that $\varphi_i' \preceq \varphi_i$ and $f(\varphi_i') = f(\varphi_i)$. Given the definition of $\mathcal{SC}(\mathcal{Q}, f)$, there exists $\varphi_j' \in \mathcal{SC}(\mathcal{Q}, f)$ such that $\varphi_j' \leq_f \varphi_i'$, i.e., $\varphi_j' \preceq \varphi_i'$ and $f(\varphi_j') = f(\varphi_i')$. Thus, there exists $\varphi_j' \in \mathcal{SC}(\mathcal{Q}, f)$ such that $\varphi_j' \preceq \varphi_i$ and $f(\varphi_j') = f(\varphi_i)$. Finally, for every $\varphi' \in \mathcal{SC}(\mathcal{Q}, f)$ such that $\varphi' \preceq \varphi_i$, we have $f(\varphi') \leq f(\varphi_i)$ since f is a monotonic increasing function. It follows that: $f(\varphi_i) = f(\varphi_j') = max\{f(\varphi') \mid \varphi_i' \in \mathcal{SC}(\mathcal{Q}, f)$ and $\varphi' \preceq \varphi_i\}$ which completes the proof. □

The same idea applies for key patterns. Recalling that $GK(q_i, f)$ is the set of all interesting key patterns in Δ with respect to q_i and f $(i = 1, \ldots, n)$, we define the sets $\mathcal{GK}(\mathcal{Q}, f)$ and $\mathcal{GK}^*(\mathcal{Q}, f)$ by:

$$\mathcal{GK}(\mathcal{Q}, f) = max_{\leq_f}(\bigcup_{q \in \mathcal{Q}} GK(q, f)) \text{ and } \mathcal{GK}^*(\mathcal{Q}, f) = \{(\varphi, f(\varphi)) \mid \varphi \in \mathcal{GK}(\mathcal{Q}, f)\}$$

The following proposition states how to compute $f(\varphi)$ based on the set $\mathcal{GK}(\mathcal{Q}, f)$.

Proposition 6. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of simple mining queries with $q_i \in \mathbb{Q}$ $(i = 1, \ldots, n)$ and f be a monotonic increasing measure function in $\mathbb{I}$. For every $i = 1, \ldots, n$ and $\varphi \in sol(q_i)$, we have:*

$$f(\varphi) = min\{f(\varphi') \mid \varphi' \in \mathcal{GK}(\mathcal{Q}, f) \text{ and } \varphi \preceq \varphi'\}.$$

PROOF: *The proof uses similar arguments as that of Proposition 5, and thus is omitted.* □

Using propositions 5, 6 and Theorem 1, the following theorem holds.

Theorem 2. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries with $q_i = m_i \wedge a_i$ where $m_i \in \mathbb{M}$ and $a_i \in \mathbb{A}$ $(i = 1, \ldots, n)$. Let f be a monotonic increasing measure function in $\mathbb{I}$.*

The sets $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{SC}^*(\mathcal{Q}, f)\}$ and $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{GK}^*(\mathcal{Q}, f)\}$ are extended condensed representations of $ans(\mathcal{Q}, f)$, i.e.,

$\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{SC}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$, and

$\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{GK}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$.

PROOF: *Let F be the function defined as follows: for every triple $(\varphi, q, \alpha) \in \mathbb{L} \times \mathbb{Q} \times \Re$, $(\varphi, q, \alpha) \in F(\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y})$ if:*

- *there exists $(\varphi_1, q_1) \in \mathcal{X}_1$ such that $(\varphi_1, q_1) \leq_{\mathbb{A}} (\varphi, q)$, and*
- *there exists $(\varphi_2, q_2) \in \mathcal{X}_2$ such that $(\varphi, q) \leq_{\mathbb{M}} (\varphi_2, q_2)$, and*
- *$\alpha = max\{\alpha' \mid (\exists \varphi' \in \mathbb{L})((\varphi', \alpha') \in \mathcal{Y} \text{ and } \varphi' \preceq \varphi)\}$.*

Using Theorem 1 and Proposition 5, we can easily see that $ans(\mathcal{Q}, f) = F(\Sigma(\mathcal{Q}), \Gamma(\mathcal{Q}), \mathcal{SC}^*(\mathcal{Q}, f))$. Moreover, F is independent from the data set Δ since $\preceq$ and $\sqsubseteq$ do not depend on Δ. Finally, we have $\Sigma(\mathcal{Q}) \subseteq \mathcal{S}(\mathcal{Q}) \subseteq sol(\mathcal{Q})$, $\Gamma(\mathcal{Q}) \subseteq \mathcal{G}(\mathcal{Q}) \subseteq sol(\mathcal{Q})$ and $\pi_{\mathbb{L}}(\mathcal{SC}^*(\mathcal{Q}, f)) = \mathcal{SC}(\mathcal{Q}, f) \subseteq \pi_{\mathbb{L}}(sol(\mathcal{Q}))$, which shows that $\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{SC}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$. Using Theorem 1 and Proposition 6, it can be shown in the same way that $\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{GK}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$, thus the proof is complete. □

Example 6. *Let $\mathcal{Q} = \{q_1, q_2\}$ be the set of simple mining queries as defined in our Running Example 1. We recall from examples 1 and 5 that:*

- *$S(q_1) = \{ABCE\}$, $S(q_2) = \{ABC, ACD\}$, $G(q_1) = \{B\}$ and $G(q_2) = \{A\}$,*
- *$SC(q_1, f) = \{ABCE, ABE\}$ and $SC(q_2) = \{ABC, ACD, AB, AC, A\}$,*
- *$\mathcal{SC}(\mathcal{Q}, f) = \{ABCE, ABE, ACD, AC, A\}$, and*
- *$\mathcal{SC}^*(\mathcal{Q}, f) = \{(ABCE, 0.4), (ABE, 0.5), (ACD, 0.3), (AC, 0.5), (A, 0.8)\}$.*

Therefore, $\mathcal{S}(\mathcal{Q}) = \{(ABCE, q_1), (ABC, q_2), (ACD, q_2)\}$ and $\mathcal{G}(\mathcal{Q}) = \{(B, q_1), (A, q_2)\}$. Since $\mathcal{S}(\mathcal{Q})$ (respectively $\mathcal{G}(\mathcal{Q})$) contains no pairs comparable with respect to $\leq_{\mathbb{A}}$ (respectively $\leq_{\mathbb{M}}$), we have $\Sigma(\mathcal{Q}) = \mathcal{S}(\mathcal{Q})$ (respectively $\Gamma(\mathcal{Q}) = \mathcal{G}(\mathcal{Q})$).

Then using Theorem 2, we know that $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{SC}^*(\mathcal{Q}, f)\}$ is an extended condensed representation of $ans(\mathcal{Q}, f)$, i.e., $\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q}),\ \mathcal{SC}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$. Moreover, we note that $\mathcal{SC}^*(\mathcal{Q}, f) \subset SC(q_1, f) \cup SC(q_2, f)$. □

The previous example shows a case where the two condensed representations $\{\mathcal{S}(\mathcal{Q}),\ \mathcal{G}(\mathcal{Q})\}$ and $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ of $sol(\mathcal{Q})$ are equal. In the next subsection, we show that these condensed representations can be made more concise under additional hypotheses that are satisfied in the traditional case of association rules mining [1].

4.4 Further Improvement

We assume now that every query $q \in \mathcal{Q}$ is of the form $q = \overline{q} \wedge \widetilde{q}$ where $\widetilde{q}$ is an independent selection predicate. Intuively, in order to further condense $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$, we compare queries based on their 'non-independent parts,' since their 'independent parts' can be evaluated without considering the underlying data set.

To this end, given a set of mining queries $\mathcal{Q} = \{q_1, \ldots, q_n\}$ where $q_i = \overline{q_i} \wedge \widetilde{q_i}$ with $\overline{q_i} \in \overline{\mathbb{Q}}$ and $\widetilde{q_i} \in \widetilde{\mathbb{Q}}$, we define two partial pre-orderings, denoted by $\leq_{\overline{\mathbb{A}}}$ and $\leq_{\overline{\mathbb{M}}}$, as follows: for all (φ_i, q_i) and (φ_j, q_j) in $\mathbb{L} \times \mathcal{Q}$:

$$(\varphi_i, q_i) \leq_{\overline{\mathbb{A}}} (\varphi_j, q_j) \ \text{ if } \ \varphi_i \preceq \varphi_j \text{ and } \overline{a_i} \sqsubseteq \overline{a_j}$$
$$(\varphi_i, q_i) \leq_{\overline{\mathbb{M}}} (\varphi_j, q_j) \ \text{ if } \ \varphi_i \preceq \varphi_j \text{ and } \overline{m_j} \sqsubseteq \overline{m_i}.$$

Then, we introduce the following notations:

$$\overline{\Sigma}(\mathcal{Q}) = min_{\leq_{\overline{\mathbb{A}}}}(\mathcal{S}(\mathcal{Q})) \qquad \text{and} \qquad \overline{\Gamma}(\mathcal{Q}) = max_{\leq_{\overline{\mathbb{M}}}}(\mathcal{G}(\mathcal{Q})).$$

The following lemma states how, for every q in $\mathcal{Q}$, $sol(q)$ can be computed based on $\overline{\Sigma}(\mathcal{Q})$ and $\overline{\Gamma}(\mathcal{Q})$, assuming that the independent part $\widetilde{q}$ of q is known.

Lemma 5. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = \overline{q_i} \wedge \widetilde{q_i}$ where $\overline{q_i} = \overline{m_i} \wedge \overline{a_i}$ with $\overline{m_i} \in \overline{\mathbb{M}}$, $\overline{a_i} \in \overline{\mathbb{A}}$, and $\widetilde{q_i} = \widetilde{m_i} \wedge \widetilde{a_i}$ with $\widetilde{m_i} \in \widetilde{\mathbb{M}}$, $\widetilde{a_i} \in \widetilde{\mathbb{A}}$ $(i = 1, \ldots, n)$. For every q in $\mathcal{Q}$, we have:*

$$sol(q) = \{\varphi \in sol(\widetilde{q}) \mid (\exists(\varphi_i, q_i) \in \overline{\Sigma}(\mathcal{Q}))((\varphi_i, q_i) \leq_{\overline{\mathbb{A}}} (\varphi, q) \text{ and } (\exists(\varphi_j, q_j) \in \overline{\Gamma}(\mathcal{Q}))((\varphi, q) \leq_{\overline{\mathbb{M}}} (\varphi_j, q_j))\}.$$

Proof: *Let $\overline{X}(q)$ be the set defined by:*

$$\overline{X}(q) = \{\varphi \in sol(\widetilde{q}) \mid (\exists(\varphi_i, q_i) \in \overline{\Sigma}(\mathcal{Q}))((\varphi_i, q_i) \leq_{\overline{\mathbb{A}}} (\varphi, q) \text{ and } (\exists(\varphi_j, q_j) \in \overline{\Gamma}(\mathcal{Q}))((\varphi, q) \leq_{\overline{\mathbb{M}}} (\varphi_j, q_j))\}.$$

We first show that $\overline{X}(q) \subseteq sol(q)$. Let $\varphi \in \overline{X}(q)$. There exist $(\varphi_i, q_i) \in \overline{\Sigma}(\mathcal{Q})$ and $(\varphi_j, q_j) \in \overline{\Gamma}(\mathcal{Q})$ such that $(\varphi_i, q_i) \leq_{\overline{\mathbb{A}}} (\varphi, q)$ and $(\varphi, q) \leq_{\overline{\mathbb{M}}} (\varphi_j, q_j)$.

On one hand, we know that $q_i(\varphi_i) = true$. Thus, we have $\overline{a_i}(\varphi_i) = true$. It follows that $\overline{a}(\varphi_i) = true$ since $\overline{a_i} \sqsubseteq \overline{a}$, and so, $\overline{a}(\varphi) = true$ since $\varphi_i \preceq \varphi$ and $\overline{a}$ is anti-monotonic.

On the other hand, we know that $q_j(\varphi_j) = true$. Thus, we have $\overline{m_j}(\varphi_j) = true$. It follows that $\overline{m}(\varphi_j) = true$ since $\overline{m_j} \sqsubseteq \overline{m}$, and so, $\overline{m}(\varphi) = true$ since $\varphi \preceq \varphi_j$ and $\overline{m}$ is monotonic. Therefore, we have $\overline{q}(\varphi) = true$. Since $\varphi \in sol(\widetilde{q})$, we have $q(\varphi) = \overline{q}(\varphi) \wedge \widetilde{q}(\varphi) = true$, which shows that $\overline{X}(q) \subseteq sol(q)$.

Now, we show that $sol(q) \subseteq \overline{X}(q)$. Let $\varphi \in sol(q)$. There exist $\varphi_s \in S(q)$ and $\varphi_g \in G(q)$ such that $\varphi_s \preceq \varphi \preceq \varphi_g$. Moreover, we have $(\varphi_s, q) \in \mathcal{S}(\mathcal{Q})$ and $(\varphi_g, q) \in \mathcal{G}(\mathcal{Q})$.

Given the definitions of $\overline{\Sigma}(\mathcal{Q})$ and $\overline{\Gamma}(\mathcal{Q})$, there exist $(\varphi_i, q_i) \in \overline{\Sigma}(\mathcal{Q})$ and $(\varphi_j, q_j) \in \overline{\Gamma}(\mathcal{Q})$ such that $(\varphi_i, q_i) \leq_{\overline{\mathbb{A}}} (\varphi_s, q)$ and $(\varphi_g, q) \leq_{\overline{\mathbb{M}}} (\varphi_j, q_j)$. Moreover, we have $(\varphi_s, q) \leq_{\overline{\mathbb{A}}} (\varphi, q)$ since $\varphi_s \preceq \varphi$, and $(\varphi, q) \leq_{\overline{\mathbb{M}}} (\varphi_g, q)$ since $\varphi \preceq \varphi_g$. Thus, $(\varphi_i, q_i) \leq_{\overline{\mathbb{A}}} (\varphi, q)$ and $(\varphi, q) \leq_{\overline{\mathbb{M}}} (\varphi_j, q_j)$. As $\varphi \in sol(q)$ and as $sol(q) \subseteq sol(\widetilde{q})$, it follows that $\varphi \in \overline{X}(q)$, which entails that $sol(q) \subseteq \overline{X}(q)$. Thus, the proof is complete. □

Based on Lemma 5 above, we can state the following theorem.

Theorem 3. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = \overline{q_i} \wedge \widetilde{q_i}$ where $\overline{q_i} = \overline{m_i} \wedge \overline{a_i}$ with $\overline{m_i} \in \overline{\mathbb{M}}$, $\overline{a_i} \in \overline{\mathbb{A}}$, and $\widetilde{q_i} = \widetilde{m_i} \wedge \widetilde{a_i}$ with $\widetilde{m_i} \in \widetilde{\mathbb{M}}$, $\widetilde{a_i} \in \widetilde{\mathbb{A}}$ $(i = 1, \ldots, n)$. The set $\{\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q})\}$ is a condensed representation of $sol(\mathcal{Q})$, i.e., $\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q}) \models sol(\mathcal{Q})$.*

Proof: *Let us consider the function F defined by:*

$$F(\mathcal{X}_1, \mathcal{X}_2) = \{\ (\varphi, q) \in \mathbb{L} \times \mathbb{Q} \mid \varphi \in sol(\widetilde{q}) \text{ and } (\exists (\varphi_1, q_1) \in \mathcal{X}_1)((\varphi_1, q_1) \leq_{\overline{\mathbb{A}}} (\varphi, q)) \text{ and } (\exists (\varphi_2, q_2) \in \mathcal{X}_2)((\varphi, q) \leq_{\overline{\mathbb{M}}} (\varphi_2, q_2))\})$$

Using Lemma 5, we can easily see that $sol(\mathcal{Q}) = F(\overline{\Sigma}(\mathcal{Q}), \overline{\Gamma}(\mathcal{Q}))$. Moreover, F is independent from the data set Δ since $\preceq$ and $\sqsubseteq$ do not depend on Δ. Finally, for every pair (φ, q) in $\overline{\Sigma}(\mathcal{Q})$ or $\overline{\Gamma}(\mathcal{Q})$, we know that $(\varphi, q) \in sol(\mathcal{Q})$. Thus, we have $\pi_{\mathbb{L}}(\overline{\Sigma}(\mathcal{Q}) \cup \overline{\Gamma}(\mathcal{Q})) \subseteq \pi_{\mathbb{L}}(sol(\mathcal{Q}))$, which completes the proof. □

Unfortunately, as shown in the following example, the condensed representations $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ and $\{\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q})\}$ are not comparable in general. Intuitively, this is due to the fact that $(\varphi_1, q_1) \leq_{\mathbb{A}} (\varphi_2, q_2)$ can hold whereas $(\varphi_1, q_1) \leq_{\overline{\mathbb{A}}} (\varphi_2, q_2)$ does not, or conversely.

Example 7. *In the context of our Running Example 1, let $q_5 = m_5 \wedge a_5$ and $q_6 = m_6 \wedge a_6$ where m_5, m_6, a_5 and a_6 are defined for every $\varphi \in \mathbb{L}$ by:*

- *$m_5(\varphi, \Delta) = true$ if $sup(\varphi, \Delta) \leq 0.8$ and $A \subseteq \varphi$,*
- *$m_6(\varphi, \Delta) = true$ if $sup(\varphi, \Delta) \leq 0.9$ and $AC \subseteq \varphi$,*
- *$a_5(\varphi, \Delta) = true$ if $sup(\varphi, \Delta) \geq sup(AB, \Delta)$ and $\varphi \subseteq AC$,*
- *$a_6(\varphi, \Delta) = true$ if $sup(\varphi, \Delta) \geq sup(AC, \Delta)$ and $\varphi \subseteq ABC$.*

We note that m_5 and m_6 are monotonic, whereas a_5 and a_6 are anti-monotonic. Moreover, we can see that $S(q_5) = \{AC\}$, $S(q_6) = \{AC\}$, $G(q_5) = \{A\}$ and $G(q_6) = \{AC\}$.

Now, considering $\mathcal{Q} = \{q_5, q_6\}$, we have: $\mathcal{S}(\mathcal{Q}) = \{(AC, q_5), (AC, q_6)\}$ and $\mathcal{G}(\mathcal{Q}) = \{(A, q_5),\ (AC, q_6)\}$. Moreover, we have $(AC, q_5) <_{\mathbb{A}} (AC, q_6)$, whereas (A, q_5) and (AC, q_6) are not comparable with respect to $\leq_{\mathbb{M}}$. Therefore, we have:

$$\Sigma(\mathcal{Q}) = \{(AC, q_5)\} \text{ and } \Gamma(\mathcal{Q}) = \{(A, q_5), (AC, q_6)\}.$$

On the other hand, $(AC, q_6) <_{\overline{\mathbb{M}}} (A, q_5)$, whereas (AC, q_5) and (AC, q_6) are not comparable with respect to $\leq_{\overline{\mathbb{A}}}$. Therefore, we have:

$$\overline{\Sigma}(\mathcal{Q}) = \{(AC, q_5), (AC,\ q_6)\} \text{ and } \overline{\Gamma}(\mathcal{Q}) = \{(A, q_5)\}.$$

Hence, we have $\Sigma(\mathcal{Q}) \subset \overline{\Sigma}(\mathcal{Q})$ and $\overline{\Gamma}(\mathcal{Q}) \subset \Gamma(\mathcal{Q})$, which shows that $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$ and $\{\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q})\}$ are not comparable. □

The following lemma states a sufficient condition when $\{\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q})\}$ is more concise than $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$. Intuitively, according to this condition, the anti-monotonic (respectively monotonic) queries to be considered must satisfy the fact that if two queries are comparable, then their dependent part are comparable as well.

Lemma 6. *Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = \overline{q_i} \wedge \widetilde{q_i}$ where $\overline{q_i} = \overline{m_i} \wedge \overline{a_i}$ with $\overline{m_i} \in \overline{\mathbb{M}}$, $\overline{a_i} \in \overline{\mathbb{A}}$, and $\widetilde{q_i} = \widetilde{m_i} \wedge \widetilde{a_i}$ with $\widetilde{m_i} \in \widetilde{\mathbb{M}}$, $\widetilde{a_i} \in \widetilde{\mathbb{A}}$ $(i = 1, \ldots, n)$.*

If for every $(a_i, a_j) \in \mathbb{A}^2$ such that $a_i \sqsubseteq a_j$, we have $\overline{a_i} \sqsubseteq \overline{a_j}$, and for every $(m_i, m_j) \in \mathbb{M}^2$ such that $m_i \sqsubseteq m_j$, we have $\overline{m_i} \sqsubseteq \overline{m_j}$, then $\{\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q})\}$ is more concise than $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$.

Proof: *Assume that for every $(a_i, a_j) \in \mathbb{A}^2$ such that $a_i \sqsubseteq a_j$, we have $\overline{a_i} \sqsubseteq \overline{a_j}$. Then, for all pairs (φ_1, q_1) and (φ_2, q_2) in $\mathcal{S}(\mathcal{Q})$ such that $(\varphi_1, q_1) \leq_{\mathbb{A}} (\varphi_2, q_2)$, we also have $(\varphi_1, q_1) \leq_{\overline{\mathbb{A}}} (\varphi_2, q_2)$. Hence, we have $\overline{\Sigma}(\mathcal{Q}) \subseteq \Sigma(\mathcal{Q})$. In the same way, we can see that if for every $(m_i, m_j) \in \mathbb{M}^2$ such that $m_i \sqsubseteq m_j$, we have $\overline{m_i} \sqsubseteq \overline{m_j}$, then $\overline{\Gamma}(\mathcal{Q}) \subseteq \Gamma(\mathcal{Q})$. Thus, the proof is complete.* □

In what follows, we identify a case where the previous lemma applies. This case makes use of the notion of *dense* measure function, defined by:

Definition 11. *Let f be a measure function defined over $\Lambda \subseteq \Re$. We say that f is* dense *in Λ with respect to $\mathbb{L}$, if for every pair $(\lambda_1, \lambda_2) \in \Lambda^2$ such that $\lambda_1 < \lambda_2$ and every pattern $\varphi \in \mathbb{L}$, there exists a data set Δ such that $\lambda_1 < f(\varphi, \Delta) < \lambda_2$.*

Then, we have the following.

Proposition 7. *Let f be a increasing measure function defined from $\mathbb{L} \times \mathbf{\Delta}$ over $\Lambda \subseteq \Re$ such that f is dense in Λ with respect to $\mathbb{L}$.*

Let $\overline{\mathbb{Q}}_f = \overline{\mathbb{A}}_f \cup \overline{\mathbb{M}}_f$ where $\overline{\mathbb{A}}_f = \{\overline{a}_\lambda \mid \lambda \in \Lambda\}$ and $\overline{\mathbb{M}}_f = \{\overline{m}_\lambda \mid \lambda \in \Lambda\}$ are two sets of selection predicates defined by: for every data set Δ and every pattern $\varphi \in \mathbb{L}$, $\overline{a}_\lambda(\varphi, \Delta) = true$ if $f(\varphi, \Delta) \geq \lambda$, and $\overline{m}_\lambda(\varphi, \Delta) = true$ if $f(\varphi, \Delta) \leq \lambda$.

Let $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{M}}$ be two sets of independent selection predicates such that for every $\widetilde{a}$ in $\widetilde{\mathbb{A}}$ (respectively $\widetilde{m} \in \widetilde{\mathbb{M}}$), $\widetilde{a}$ is anti-monotonic (respectively $\widetilde{m}$ is monotonic) and $sol(\widetilde{a}) \neq \emptyset$ (respectively $sol(\widetilde{m}) \neq \emptyset$).

Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = \overline{q_i} \wedge \widetilde{q_i}$ where $\overline{q_i} = \overline{m_i} \wedge \overline{a_i}$ with $\overline{m_i} \in \overline{\mathbb{M}}_f$, $\overline{a_i} \in \overline{\mathbb{A}}_f$, and $\widetilde{q_i} = \widetilde{m_i} \wedge \widetilde{a_i}$ with $\widetilde{m_i} \in \widetilde{\mathbb{M}}$, $\widetilde{a_i} \in \widetilde{\mathbb{A}}$ $(i = 1, \ldots, n)$. Then, $\{\overline{\Sigma}(\mathcal{Q}),\ \overline{\Gamma}(\mathcal{Q})\}$ is more concise than $\{\Sigma(\mathcal{Q}),\ \Gamma(\mathcal{Q})\}$.

Proof: *Using the notation of the proposition, based on Lemma 6, we have to show that for every $i, j = \{1, \ldots, n\}$, if $a_i \sqsubseteq a_j$, then $\overline{a_i} \sqsubseteq \overline{a_j}$ and that if $m_i \sqsubseteq m_j$, then $\overline{m_i} \sqsubseteq \overline{m_j}$.*

Assuming that $a_i \sqsubseteq a_j$ and $\overline{a_i} \not\sqsubseteq \overline{a_j}$ implies that there exist two reals λ_i and λ_j such that for every pattern $\varphi \in \mathbb{L}$ and every data set Δ, $\overline{a}_i(\varphi, \Delta) = true$ if $f(\varphi, \Delta) \geq \lambda_i$ and $\overline{a}_j(\varphi, \Delta) = true$ if $f(\varphi, \Delta) \geq \lambda_j$. If $\overline{a_i} \not\sqsubseteq \overline{a_j}$, we necessarily have $\lambda_i < \lambda_j$.

Moreover, given a pattern $\varphi \in sol(\widetilde{a_i})$, there exists a data set Δ such that $\lambda_i < f(\varphi, \Delta) < \lambda_j$. Then, we have $\varphi \in sol(a_i/\Delta)$ and $\varphi \notin sol(a_j/\Delta)$, which contradicts the hypothesis $a_i \sqsubseteq a_j$.

Using similar arguments as above, it can shown that if $m_i \sqsubseteq m_j$, then $\overline{m_i} \sqsubseteq \overline{m_j}$, which completes the proof. □

Now, we note that the previous proposition applies in the traditional case of association rules where $\mathbb{L} = 2^{Items} \setminus \{\emptyset, Items\}$ and the measure function is the

function *sup*. Indeed, it is easy to see that the function *sup* is dense in $[0,1]$ with respect to $\mathbb{L} = 2^{Items} \setminus \{\emptyset, Items\}$.

The following example shows how Proposition 7 applies in the context of our Running Example 1.

Example 8. *Let $\mathcal{Q} = \{q_1, q_2\}$ be the set of simple mining queries $q_i = m_i \wedge a_i$ where m_i and a_i $(i = 1, 2)$ are defined in our Running Example 1.*

We recall from Example 6 that $\mathcal{S}(\mathcal{Q}) = \{(ABCE, q_1), (ABC, q_2), (ACD, q_2)\}$ and $\mathcal{G}(\mathcal{Q}) = \{(B, q_1), (A, q_2)\}$. Moreover, we also recall that $\Sigma(\mathcal{Q}) = \mathcal{S}(\mathcal{Q})$ and $\Gamma(\mathcal{Q}) = \mathcal{G}(\mathcal{Q})$.

Since $ABC \subseteq ABCE$ $(ABCE \preceq ABC)$ and $\overline{a_1} \sqsubseteq \overline{a_2}$, we have $(ABCE, q_1) \leq_{\overline{\mathbb{A}}} (ABC, q_2)$. Thus, the pair (ABC, q_2) does not belong to $\overline{\Sigma}(\mathcal{Q})$. Hence, we have

$$\overline{\Sigma}(\mathcal{Q}) = \{(ABCE, q_1), (ACD, q_2)\}.$$

Then, since the pairs (B, q_1) and (A, q_2) are not comparable with respect to $\leq_{\overline{\mathbb{M}}}$, we have $\overline{\Gamma}(\mathcal{Q}) = \overline{\mathcal{G}}(\mathcal{Q})$. In conclusion, using Theorem 3, we can see that $\overline{\Sigma}(\mathcal{Q})$, $\overline{\Gamma}(\mathcal{Q}) \models sol(\mathcal{Q})$. Moreover, since $\overline{\Sigma}(\mathcal{Q}) \subset \Sigma(\mathcal{Q})$ and $\overline{\Gamma}(\mathcal{Q}) \subseteq \Gamma(\mathcal{Q})$, it is easy to see that $\{\overline{\Sigma}(\mathcal{Q}), \overline{\Gamma}(\mathcal{Q})\}$ is more concise than $\{\Sigma(\mathcal{Q}), \Gamma(\mathcal{Q})\}$. □

Finally, regarding extended condensed representations, we can easily prove the following theorem, based on propositions 5 and 6 and on Theorem 3.

Theorem 4. *Let f be a monotonic increasing measure function in $\mathbb{I}$ and $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be a set of mining queries $q_i = \overline{q_i} \wedge \widetilde{q_i}$ where $\overline{q_i} = \overline{m_i} \wedge \overline{a_i}$ with $\overline{m_i} \in \overline{\mathbb{M}}$, $\overline{a_i} \in \overline{\mathbb{A}}$, and $\widetilde{q_i} = \widetilde{m_i} \wedge \widetilde{a_i}$ with $\widetilde{m_i} \in \widetilde{\mathbb{M}}$, $\widetilde{a_i} \in \widetilde{\mathbb{A}}$ $(i = 1, \ldots, n)$. The sets $\{\overline{\Sigma}(\mathcal{Q}), \overline{\Gamma}(\mathcal{Q}), \mathcal{SC}^*(\mathcal{Q}, f)\}$ and $\{\overline{\Sigma}(\mathcal{Q}), \overline{\Gamma}(\mathcal{Q}), \mathcal{GK}^*(\mathcal{Q}, f)\}$ are extended condensed representations of $ans(\mathcal{Q}, f)$, i.e., $\overline{\Sigma}(\mathcal{Q}), \overline{\Gamma}(\mathcal{Q}), \mathcal{SC}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$ and $\overline{\Sigma}(\mathcal{Q}), \overline{\Gamma}(\mathcal{Q}), \mathcal{GK}^*(\mathcal{Q}, f) \models_e ans(\mathcal{Q}, f)$.*

5 Conclusion

In this paper, we have considered the problem of defining condensed representations of sets of mining queries. To this end, we have first studied the case of a single mining query and we have extended previous works on version spaces by [9] so as to take into account the presence of measure functions in the query. This has been done based on the well known notions of closed and key patterns ([3,16]). Then, we have seen how to extend this approach to *sets* of mining queries. The main idea in this extension is that, in order to obtain condensed representations in this case, when storing a pattern, one must keep track of the query the pattern comes from.

Based on this work, we are currently investigating how condensed representations can be used to optimize the iterative computation of the answer of mining queries. This problem has been studied for standard association rules [2,10,13,14] and multi-dimensional association rules [8,15]. In our framework, this problem can be stated as follows: given a data set Δ, a set $\mathcal{Q} = \{q_1, \ldots, q_n\}$ of mining queries and a new extended mining query (q, f):

1. How to optimize the computation of $ans(q, f)$ using the extended condensed representations of $ans(\mathcal{Q}, f)$?

2. How to efficiently modify the extended condensed representation of $ans(\mathcal{Q}, f)$ so as to obtain an extended condensed representation of $ans(\mathcal{Q} \cup \{q\}, f)$?

Moreover, it is clear that some tests are necessary to compare the various condensed representations proposed in this paper. To this end, we are implementing our approach in the context of our previous work [8], where mining queries are composed through relational operators. We also investigate how our approach can be used to optimize the iterative computation of iceberg cubes [11].

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo (1996). *Fast Discovery of Association Rules.* In Advances in Knowledge Discovery and Data Mining, pp 309–328, AAAI-MIT Press.
2. E. Baralis and G. Psaila (1999). *Incremental Refinement of Mining Queries.* In Proc. of DAWAK'99, pp. 173–182, Florence.
3. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme and L. Lakhal (2000). *Mining Frequent Patterns with Counting Inference.* SIGKDD Explorations, 2(2), p. 66–75.
4. J.-F. Boulicaut, A. Bykowski and C. Rigotti (2000). *Approximation of Frequency Queries by Means of Free-Sets.* In Proc. of PKDD'00, LNCS vol. 1910, pp. 75–85, Springer-Verlag.
5. J.-F. Boulicaut (2001). *Habilitation thesis* (French). INSA-Lyon, France.
6. L. De Raedt and S. Kramer (2001). *The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding.* In Proc. of IJCAI'01, pp. 853–862.
7. L. De Raedt (2002). *Query execution and optimization for inductive databases.* In Proc. of International Workshop DTDM'02, In conjunction with EDBT 2002, pp. 19–28 (Extended Abstract), Praha, CZ.
8. C.T. Diop, A. Giacometti, D. Laurent and N. Spyratos (2002). *Composition of Mining Contexts for Efficient Extraction of Association Rules.* In Proc. of the EDBT'02, LNCS vol. 2287, pp. 106–123, Springer-Verlag.
9. H. Hirsh (1994). *Generalizing Version Spaces.* Machine Learning, Vol. 17(1), pp. 5–46, Kluwer Academic Publishers.
10. B. Jeudy, J-F. Boulicaut (2002). *Using condensed representations for interactive association rule mining.* In Proc. of ECML/PKDD 2002, Helsinki, LNAI vol. 2431, pp. 225–236, Springer-Verlag.
11. M. Laporte, N. Novelli, R. Cicchetti, L. Lakhal (2002). *Computing Full and Iceberg Datacubes Using Partitions.* In Proc. of ISMIS'2002, LNAI vol. 2366, pp. 244–254, Springer-Verlag.
12. H. Mannila, H. Toivonen (1997). *Levelwise Search and Borders of Theories in Knowledge Discovery.* Techn. Rep. C-1997-8, University of Helsinki.
13. T. Morzy, M. Wojciechowski and M. Zakrzewicz (2000). *Materialized Data Mining Views.* In Proc. of PKDD'2000, LNCS vol. 1910, pp. 65–74, Springer-Verlag.
14. B. Nag, P. Deshpande and D.J. DeWitt (1999). *Using a Knowledge Cache for Interactive Discovery of Association Rules.* In Proc. of KDD'99, pp. 244–253, San Diego, USA.
15. B. Nag, P. Deshpande and D.J. DeWitt (2001). *Caching for Multi-dimensional Data Mining Queries.* In Proc. of SCI'2001, Orlando, Florida.
16. N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal (1999). *Efficient Mining of Association Rules using Closed Itemsets Lattices.* Information Systems, Vol. 24(1), pp. 25–46, Elsevier Publishers.