Combining Dependent Annotations for Relational Algebra

Egor V. Kostylev
University of Edinburgh
ekostyle@inf.ed.ac.uk

Peter Buneman
University of Edinburgh
opb@inf.ed.ac.uk

ABSTRACT

Annotation is some form of data that is added to an existing database. It could be additional data that for whatever reason cannot be stored in the original database, or it could be some form of metadata such as comments, probabilities, timestamps that are not normally regarded part of the basic database design. It has recently been observed that, in order to determine how annotations should be propagated through database queries, we need to have some structure on them. Although various forms of annotation have been considered in some detail, each form has been considered in isolation.

In this paper we consider what happens when different forms of annotation are combined. We show that there are many cases in which annotations, for quite natural reasons, depend on one another. What is the appropriate structure to place on such annotations? We provide a method for combining different forms and provide a normal form which is useful in deciding whether two or more combined annotations are equivalent.

Categories and Subject Descriptors

H.2.1 [**Database Management**]: Logical Design; H.2.4 [**Database Management**]: Systems—*Relational databases*; F.m [**Theory of Computation**]: Miscellaneous

General Terms

Design, Theory

Keywords

Annotation, provenance, semirings

1. INTRODUCTION

Annotation is a major industry. The main value of curated databases [14] lies in the addition of annotations by experts; the whole idea of linked data [4] is arguably about the annotation of existing structures; and several extensions to relational databases, such as probabilistic databases, c-tables, various forms of provenance, etc. [18] can be seen as annotations on the tuples of existing tables.

An observation that has emerged relatively recently is that annotations have *structure* [18, 17]. The understanding of this structure is prompted by the following central question: how should annotations be propagated through queries [10, 11]? For example, if annotations are sets of user *comments* and we have tuples $r \in R$ and $s \in S$ that are respectively annotated with sets of comments A^r and A^s. A tuple t resulting from the "merge" of r and s in $R \cup S$ when $r = s$ would carry the annotation $A^r \cup A^s$. Similarly if t is the result of concatenating tuples r and s in $R \bowtie S$, t would also carry the same union of annotations. However if B^r and B^s are sets of people who *believe* the tuples r and s to be true, we would probably take the annotation to be $B^r \cup B^s$ for a union operation on r and s but $B^r \cap B^s$ for join. Thus we would use different algebraic structures for propagating comment annotations through queries and for propagating belief annotations. In [18] a quite general *provenance semiring* structure is proposed for describing annotations on tuples. It is capable of representing a wide variety of forms of annotation. In fact, it is claimed to be the most general structure for annotating the relational algebra, because any other annotation structure is a homomorphic image of it. But what happens if we have two kinds of annotations? How do they interact? This is the question we address in this paper.

It is often tacitly assumed that different kinds of annotation are *independent* of each other. In curated databases different people with different expertise can annotate some underlying database. In this case one would assume the annotations to be independent, as it is in the next example.

Example 1. Suppose we have a bibliography on a topic, represented by a table $\textbf{Publication}(\underline{\mathsf{Id}}, \mathsf{Title}, \mathsf{Authors}, \ldots)$. One could imagine one group of experts annotating each entry with a set of keywords by creating $\textbf{Keywords}(\mathsf{Term}, \mathsf{Id})$ and another group or system annotating each entry with citations $\textbf{Cites}(\mathsf{CId}, \mathsf{Id})$. We expect these two annotations to operate independently, and that whatever algebraic rules we use for combining keywords or citations will operate independently of each other.

What we show in this paper is that there are many natural cases in which annotations are *not* independent and that we need to ask what is the appropriate algebraic structure when there is a dependency between annotations. Even the question of whether two dependent annotations are equivalent is non-trivial. Here is an simple example of such annotations taken from [9].

Example 2. Suppose we have the information on export partners of countries such as given in Fig. 1(a). Here we think of the base data as a table **Export**(CName, Goods) of countries with the goods they export. Added to each tuple are two annotations, one containing the period for which this tuple is valid and another a list of the export partners for those goods. If we project on CName in a query **Q** shown in Fig. 1(b) and combine the annotations for Time and Customers independently we get the annotated tuple indicating that the UK, Germany, Italy and Cyprus were export customers from 2004-2010. This would make sense if the two annotations were independent, but it is incorrect because the Customers annotation clearly depends on Time. That is, Fig. 1(b) is an incorrect annotation, and we want to formulate a correct semantics for combining annotations.

An obvious response to this example is that we have represented the information incorrectly. If instead we had made customers and times a part of the basic data by creating a table **Export**(CName, Goods, Customer, Year) in a first normal form in which time intervals are represented as the set of years it spans, we would not have seen this anomaly. The problem is that we can only do this when the annotation is a finite set, and this cannot be done for annotations such as probabilities, degree of trust, cardinalities, etc. Moreover, even when the annotation is a set, the representation is inaccurate because we want to be able to annotate a tuple with the empty set without it disappearing.[1] Finally, following our comparison above of comment and belief annotations, even when the annotation is a set and we can create a tabular representation, we need to take care in the interpretation of results of queries on such tables.

There are many practical examples of dependent annotations. The main strength of the major protein database SWISS-PROT [5] is the manual annotation of existing sequence data. Interestingly, in this database the times of both the last sequence update and the last annotation update are recorded, perhaps indicating an awareness by the curators of the interaction between these two forms of annotation. There are numerous examples of interacting annotations in belief databases ([15]) and in web sites where users annotate other annotations with, for example, some indication of trust.

Hence, the main goal of this paper is to develop the machinery for combining dependent annotations for the relational algebra. In the preliminaries we summarize the basic algebraic structures for annotations on tuples in the positive relational algebra introduced in [18]. In Sec. 3 we look on the general question how annotations can interact between each other. In Sec. 4 the definition of combined annotations is given and an equivalence relation on them introduced which represents the fact that different combined annotations carry the same information. In Sec. 5 the algebraic operations for combined annotations are defined, and it is shown that they form a semiring structure. In Sec. 6 a normal form for combined annotations is defined and a normalization algorithm is presented. We conclude with a discussion of some of the practical issues in Sec. 7.

[1]One could also use a separate table and an inclusion dependency, but even this is problematic when one considers more than two dependent annotations.

Exports:

CName	Goods	Time	Customers
Greece	Food	2004-2008	UK, Germany
Greece	Textile	2007-2010	Germany, Italy, Cyprus

(a)

$\mathbf{Q} = \pi_{\mathsf{CName}}(\mathbf{Exports})$:

CName	Time	Customers
Greece	2004-2010	UK, Germany, Italy, Cyprus

(b)

Figure 1: (a) Relation with two annotations and (b) Query if the annotations considered independent

The proofs for propositions and theorems are given in the Appendix.

2. PRELIMINARIES

An algebraic structure $\mathcal{K} = \langle K, \oplus, \otimes, \mathbb{0}, \mathbb{1}\rangle$ with binary operations *sum* $\oplus$ and *product* $\otimes$ and constants $\mathbb{0}$ and $\mathbb{1}$ is a (*commutative*) *semiring* iff $\langle K, \oplus, \mathbb{0}\rangle$ and $\langle K, \otimes, \mathbb{1}\rangle$ are commutative monoids with identities $\mathbb{0}$ and $\mathbb{1}$ correspondingly,[2] $\otimes$ is distributive over $\oplus$, and for each $v \in K$ it holds that $v \otimes \mathbb{0} = \mathbb{0}$.

Commutative semirings are shown ([18]) to be essential structures for annotation domains of a positive relational algebra. The simplest example of a semiring used for annotation is $\mathcal{B} = \langle\{\mathsf{false}, \mathsf{true}\}, \vee, \wedge, \mathsf{false}, \mathsf{true}\rangle$. Databases annotated with this semiring is a model for usual relational databases without any annotations: in this case each tuple is annotated with true to indicate whether it is "in" the current version or false to indicate whether it is not. In order to determine whether a tuple should be "in" the result of a positive query we apply $\vee$ and $\wedge$ to annotations of source tuples of union and join operations, correspondingly.

Example 3. The following are semirings which we use for annotations in examples of this paper.
(1) The *bag semiring* $\mathcal{N} = \langle\mathbb{N}_0, +, \times, 0, 1\rangle$ where $\mathbb{N}_0$ is the set of natural numbers with 0. This semiring is used to model the standard bag semantics.
(2) The *boolean algebra* $\mathcal{A}_S = \langle 2^S, \cup, \cap, \emptyset, S\rangle$ where S is a set of arbitrary keys. Particular examples of finite boolean algebras are $\mathcal{A}_\Omega$ used to model probabilistic databases ([13, 23]), where Ω is a finite set of events; $\mathcal{A}_B$ where B is the set of people who may believe in tuples; and $\mathcal{A}_E$ where E is the set of possible export partners. The latter one is the domain for the second annotations Customers in Ex. 2. Another example of a boolean algebra is the *time semiring* $\mathcal{A}_T$ where T is a set of time stamps which can be either finite, countable or even uncountable (for a continuous time), but in the latter two cases the set 2^T may be restricted to any subset closed under $\cup$ and $\cap$ for representation purposes; particularly, in Ex. 2 the first annotations Time come from the time semiring $\mathcal{A}_Y$ where Y is the set of years.

[2]Recall that a commutative monoid is a set with an associative and commutative binary operation and an identity element.

(3) The *fuzzy semiring* [20] $\mathcal{K}_{[0,1]} = \langle [0,1], \mathsf{max}, \otimes, 0, 1\rangle$ where $\otimes$ is any *t-norm* which distributes over max; in this paper we consider the t-norm to be the usual multiplication.
(4) The *lineage semiring* [12, 6] $\mathcal{L}_X = \left\langle 2^X \cup \{\bot\}, +, \cdot, \bot, \emptyset \right\rangle$ where X is a set of variables and for each S and T it holds that $\bot + S = S + \bot = S, \bot \cdot S = S \cdot \bot = \bot$ and $S + T = S \cdot T = S \cup T$ if $S, T \neq \bot$. If C is a set of atomic comments then $\mathcal{L}_C$ can be used to model databases with comments.
(5) The *provenance semiring* [18] $\mathcal{P}_X = \langle \mathbb{N}[X], +, \times, 0, 1\rangle$ where $\mathbb{N}[X]$ is the set of all positive polynomials with variables from a finite set X.

Other interesting semirings used for annotations include the *c-table semiring* [19], the *distance* or *tropical semiring*, the *why semiring* [6], the *security semiring* [3], etc.

We will refer to semirings used for annotating databases as *annotation domains*.

Formally, we have the following definitions for the relational model [1]. A *U-tuple* is a function $t : U \to \mathbb{D}$, where U is a finite set of *attributes* and $\mathbb{D}$ is a *domain of values*. The set of U-tuples is denoted by U-Tup. For a semiring $\mathcal{K} = \langle K, \oplus, \otimes, \mathbb{0}, \mathbb{1}\rangle$ a *$\mathcal{K}$-relation* is a function $R : U\text{-}\mathsf{Tup} \to K$ such that its *support*, i.e., the set $\mathsf{Supp} = \{t \mid R(t) \neq \mathbb{0}\}$, is finite. A *positive relational algebra on $\mathcal{K}$-relations* [18] (denoted by $\mathcal{RA}^+_\mathcal{K}$) contains operators:

empty relation for any set of attributes U there exists $\emptyset : U\text{-}\mathsf{Tup} \to K$ such that $\emptyset(t) = \mathbb{0}$;

union if $R_1, R_2 : U\text{-}\mathsf{Tup} \to K$ then $R_1 \cup R_2 : U\text{-}\mathsf{Tup} \to K$ such that

$$(R_1 \cup R_2)(t) = R_1(t) \oplus R_2(t);$$

projection if $R : U\text{-}\mathsf{Tup} \to K$ and $V \subseteq U$ then $\pi_V(R) : U\text{-}\mathsf{Tup} \to K$ such that

$$(\pi_V(R))(t) = \bigoplus_{t=t' \text{ on } V} R(t');$$

selection if $R : U\text{-}\mathsf{Tup} \to K$ and $\mathbf{P} : U\text{-}\mathsf{Tup} \to \{\mathbb{0}, \mathbb{1}\}$ is a *selection predicate* then $\sigma_\mathbf{P}(R) : U\text{-}\mathsf{Tup} \to K$ such that

$$(\sigma_\mathbf{P}(R))(t) = R(t) \otimes \mathbf{P}(t);$$

natural join if $R_i : U_i\text{-}\mathsf{Tup} \to K$, $i = 1, 2$, then $R_1 \bowtie R_2 : (U_1 \cup U_2)\text{-}\mathsf{Tup} \to K$ such that

$$(R_1 \bowtie R_2)(t) = R_1(t) \otimes R_2(t);$$

renaming if $R : U\text{-}\mathsf{Tup} \to K$ and $\beta : U \to U'$ is a bijection then $\rho_\beta(R) : U'\text{-}\mathsf{Tup} \to K$ such that

$$(\rho_\beta(R))(t) = R(t \circ \beta).$$

A fundamental property of semirings for the positive relational algebra is a *homomorphism property*, i.e., for semirings $\mathcal{K} = \langle K, \oplus, \otimes, \mathbb{0}, \mathbb{1}\rangle$ and $\mathcal{K}' = \langle K', \oplus', \otimes', \mathbb{0}', \mathbb{1}'\rangle$ and a mapping $h : K \to K'$, extended to the transformation from $\mathcal{K}$-relations to $\mathcal{K}'$-relations, the equality $Q(h(R)) = h(Q(R))$ holds for every positive query $Q \in \mathcal{RA}^+_\mathcal{K}$ iff h is a semiring homomorphism. The provenance semiring $\mathcal{P}_X$ from Ex. 3 is the most general in the sense that all other semirings are homomorphic images of it. We refer the reader to [18] for further details.

3. DEPENDENCIES BETWEEN ANNOTATION DOMAINS

In Ex. 1 and Ex. 2 we saw that two annotation domains can be either independent or dependent. In this section we will look how several domains can interact with each other.

Coming back to Ex. 2, one can assume that $\mathsf{Customer}$ annotations pertain to the tuple for a specific Time period. Another assumption would be that the sets of customers are placed on the tuples and that the time intervals are placed on those sets to indicate the periods for which the customers are valid. This situations are different *semantically*. However, if we abstract away from the meanings of these annotations, it is clear that *syntactically* these cases are the same. As another example of difference between semantics and syntax of annotations, we can look at the following.

Example 4. Consider a belief database where tuples are annotated with sets of believers, which in turn are annotated with other sets of believers, i.e., we want to record information like "Alice believes that Bob believes in a tuple t". To do it we can use two copies $\mathcal{A}'_B$ and $\mathcal{A}''_B$ of the same annotation domain $\mathcal{A}_B$. Here the semantics of $\mathcal{A}'_B$ is "who believes in a tuple" and of $\mathcal{A}''_B$ – "who believes in someone's belief in a tuple", i.e., there is a clear semantic order on these domains. However, if we have a database annotated like this, nothing prevents us from interpreting elements from $\mathcal{A}''_B$ as believers in tuples and elements from $\mathcal{A}'_B$ – as believers in believers of these tuples. Again, these situations are semantically different, but syntactically the same.

For our mathematical development of combined annotations we shall therefore consider the dependency relation on annotations domains to be undirected (i.e., symmetric). We will use $\mathcal{K}' \rightleftharpoons \mathcal{K}''$ to indicate that the domains $\mathcal{K}'$ and $\mathcal{K}''$ are mutually dependent.

So far we have looked on situations when we have only two annotation domains. But it is possible to have more than two, and the situation is quite subtle. It can happen in curated databases that different people are responsible for annotating other annotations of the data, and the interaction between these requires some care. We give a contrived example, but one which reflects the practice in databases that are constructed by some community.

Example 5. Suppose we want to construct a database of composers with details of their lives and what they wrote by a process of annotation. To start with we create a "base" table **Composers**($\underline{\mathsf{CompName}}$, BDate) which contains names $\mathsf{CompName}$ and dates of birth BDate of composers. We now want to describe when each of them was active and annotate the tuples with a set of years, i.e., use the annotation domain $\mathcal{A}_Y$.

Now a further annotation is added to describe what works a composers wrote. For this we can use an annotation domain $\mathcal{A}_W$ there W is the set of all works. (We can assume, if we want, that a work can have more than one composer.) Since we want to record the years in which the composer wrote the work, the domains $\mathcal{A}_Y$ and $\mathcal{A}_W$ are dependent. That is $\mathcal{A}_Y \rightleftharpoons \mathcal{A}_W$.

There are several further annotations one might associate with the previous structure: transaction time (the period for which the information is in the database), believers (the set of people who believe the data) or, as is common in curated databases, set of validators V, the people who have checked that the data is correct. Taking validators as an example, we use $\mathcal{A}_V$ as the third domain. However, we have to ask what is being validated. There are several different possibilities. Here are three of them:

1. The validators check that the composer wrote the work at the time specified by $\mathcal{A}_W$ and $\mathcal{A}_Y$. In this case all three annotations are mutually dependent: $\mathcal{A}_Y \rightleftharpoons \mathcal{A}_W$, $\mathcal{A}_W \rightleftharpoons \mathcal{A}_V$, and $\mathcal{A}_Y \rightleftharpoons \mathcal{A}_V$.

2. The validators check that the composer wrote the work, but do not verify the time. here we have $\mathcal{A}_W \rightleftharpoons \mathcal{A}_Y$ and $\mathcal{A}_W \rightleftharpoons \mathcal{A}_V$, but $\mathcal{A}_Y \not\rightleftharpoons \mathcal{A}_V$.

3. The validators check only that a composer tuple is valid. Here we have the annotation domain $\mathcal{A}_V$ independent of $\mathcal{A}_Y$ and $\mathcal{A}_W$, i.e., $\mathcal{A}_W \rightleftharpoons \mathcal{A}_Y$, but $\mathcal{A}_W \not\rightleftharpoons \mathcal{A}_V$ and $\mathcal{A}_Y \not\rightleftharpoons \mathcal{A}_V$.

We shall consider a general case in which we are given fixed n annotation domains. (In general, some of them can coincide, but to simplify notation we consider all of them to be different). Let $\mathcal{K}_i = \langle K_i, \oplus_i, \otimes_i, \mathbb{0}_i, \mathbb{1}_i \rangle$, $1 \leq i \leq n$, be these domains and denote the set of them by $\mathbf{K}$. To summarize the discussion above, we have the following definition which can express all possible interactions between annotations domains.

Definition 1. *Given a set of semirings* $\mathbf{K}$, *we call* dependency graph *a binary irreflexive symmetric relation* $\rightleftharpoons$ *on* $\mathbf{K}$. *We say a set* $\mathbf{I} \subseteq \mathbf{K}$ *is* maximal independent over $\rightleftharpoons$ *(or simply* independent, *for short) and write* $\mathbf{I} \not\subseteq \rightleftharpoons$ *iff for each* $\mathcal{K}_i$ *and* $\mathcal{K}_j$ *from* $\mathbf{I}$ *it holds that* $\mathcal{K}_i \not\rightleftharpoons \mathcal{K}_j$. *For convenience, we write* $i \in \mathbf{I}$ *iff* $\mathcal{K}_i \in \mathbf{I}$.

Example 6. In Ex. 5 we have $\mathbf{K} = \{\mathcal{A}_Y, \mathcal{A}_W, \mathcal{A}_V\}$ and for the corresponding cases of $\rightleftharpoons$ the independent sets are

1. $\{\mathcal{A}_Y\}, \{\mathcal{A}_W\}$, and $\{\mathcal{A}_V\}$;
2. $\{\mathcal{A}_W\}$ and $\{\mathcal{A}_Y, \mathcal{A}_V\}$;
3. $\{\mathcal{A}_W, \mathcal{A}_V\}$ and $\{\mathcal{A}_Y, \mathcal{A}_V\}$.

One more issue which should be mentioned here is the possibility of "loops" in a dependency graph, i.e., the cases when we have four (or more) domains connected by $\rightleftharpoons$ in a circle and no other dependencies hold between them. It is not obvious whether such situations should be allowed, or the structure of $\rightleftharpoons$ should be restricted to being "tree-like".[3] However, whether or not we have this restriction does not change the development in this paper, so for convenience we follow the general case.

4. COMBINED ANNOTATIONS

Having established the notion of dependency graph, in this section we will look at combined annotations over such graphs. We first consider the structure of a combined annotation and give its formal definition. After this we introduce some additional assumptions about the semirings $\mathbf{K}$, which, on the one hand, allow us to obtain desirable properties of combined annotations and, on the other, are minor enough to be satisfied by all semirings considered in the literature for annotations of which we are aware. Then we consider a containment preorder relation on combined annotations and some of its properties first for the simple case of two dependent domains and second for an arbitrary dependency graph. Finally, we look at the induced equivalence which formalize an intuition that different annotations can represent essentially the same information.

[3]Strictly speaking the assumption is that $\rightleftharpoons$ is chordal.

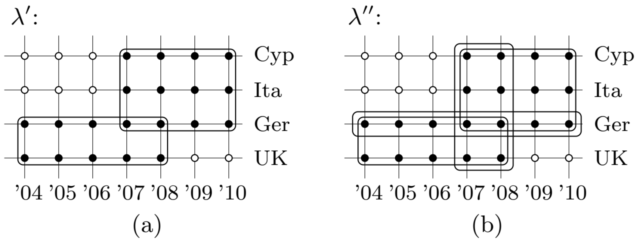

Figure 2: Graphical representation of annotations

An annotation to a tuple is value from a specific annotation domain. If we need to annotate a relation with several domains, one could expect annotations to be vectors with components from these domains. But returning to Ex. 2, it is easy to see that it is in general impossible to represent annotation information for a tuple correctly by a single vector of annotations. The following definition states that a sufficient structure for elements of combined annotations is a set of such vectors.

Definition 2. *A* combined annotation *over a set of domains* $\mathbf{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}$ *is a finite non-empty set of vectors from* $K_1 \times \ldots \times K_n$.

We use Greek letters to denote combined annotations. To justify this definition, consider the following example.

Example 7. As we saw in Ex. 2 the annotation for the tuple in the query $\mathbf{Q}$ in Fig. 1(b) is an undesirable over-approximation of the correct information. Possible representations for this information are the following combined annotations:

$$\lambda' = \begin{array}{lll} \{ & (\text{2004-2008}, & \{\text{UK, Germany}\}), \\ & (\text{2007-2010}, & \{\text{Germany, Italy, Cyprus}\}) \quad \}; \end{array}$$

$$\lambda'' = \begin{array}{lll} \{ & (\text{2004-2008}, & \{\text{UK, Germany}\}), \\ & (\text{2007-2010}, & \{\text{Germany, Italy, Cyprus}\}), \\ & (\text{2007-2008}, & \{\text{UK, Germany, Italy, Cyprus}\}), \\ & (\text{2004-2010}, & \{\text{Germany}\}) \quad \}. \end{array}$$

The first of these, λ' is an annotation derived from annotations in Fig. 1(a) by simple merging. The second, λ'' is a different but equivalent annotation. Note, that in this example elements of the original domains $\mathcal{A}_Y$ and $\mathcal{A}_E$ are respectively *sets* of years and countries, so the annotations are not just sets of pairs of elements from the combining domains $\mathcal{A}_Y$ and $\mathcal{A}_E$, but sets of pairs of subsets of these domains. Graphical representations of these annotations are shown in Fig. 2(a) and (b) where the solid dots represent the information which should be covered by the annotation pairs, and the hollow dots are the undesirable ones resulting from over-approximation. The round-corner rectangles bound the elements of λ' and λ'', each of which is a product of two sets.

As we have seen in this example, one set of rectangles can cover all the dots covered by another one, i.e., represent more information. Moreover, different sets of rectangles can cover the same set of dots indicating that different combined annotations can represent the same information. Hence, our next step is to define containment and equivalence relations on combined annotations. The first one will be a preorder. It will induce the second one which forms a quotient set of equivalent combined annotations. However, we need to take into account that in the example both of the domains are finite boolean algebras, i.e., semirings with strong additional requirements. Many annotation domains are not finite (e.g., the semiring of continuous time) and even not boolean algebras (e.g., the bag semiring), so equivalence cannot – as Fig. 2 might suggest – in general be expressed in terms of subsets of a cartesian product. Next we introduce restrictions to semirings which allow us to define the containment and equivalence relations but weak enough to be satisfied by all known annotation domains.

Definition 3. *Given a semiring $\mathcal{K} = \langle K, \oplus, \otimes, \mathbb{0}, \mathbb{1}\rangle$ let $\trianglelefteq$ be the preorder of K, defined as $u \trianglelefteq v$ iff there exists $w \in K$ such that $u \oplus w = v$. The semiring $\mathcal{K}$ is* naturally lattice ordered *(*l-semiring *for short) iff $\trianglelefteq$ forms a lattice on K, i.e., for every pair $v, u \in K$ there exist the least upper bound $v \sqcup u$ and the greatest lower bound $v \sqcap u$ in K (w.r.t. $\trianglelefteq$).*

A trivial property of l-semirings is that the order $\trianglelefteq$ is *compatible* with $\oplus$ and $\otimes$, i.e., for each u_1, u_2 and u_3 if $u_1 \trianglelefteq u_2$ then $u_1 \oplus u_3 \trianglelefteq u_2 \oplus u_3$ and $u_1 \otimes u_3 \trianglelefteq u_2 \otimes u_3$. The natural lattice order will be important in Def. 4 of containment of combined annotations. Hence, from now on we assume that all domains in $\mathbf{K}$ are l-semirings. We will usually omit indexes on their operation and predicate symbols when the instantiation is clear. Also, we will use $\trianglelefteq$ to denote their lattice orders as well as the pairwise order induced on vectors from $K_1 \times \ldots \times K_n$: $(u_1, \ldots, u_n) \trianglelefteq (v_1, \ldots, v_n)$ iff $u_i \trianglelefteq v_i$, $i = 1, \ldots, n$.

Example 8. All the semirings in the list given in Ex. 3 are l-semirings. All the substantiations of $\oplus$ except for the bag and provenance semirings are idempotent which means that $\sqcup$ coincides with $\oplus$. The dual operation $\sqcap$ is realized as $\otimes$ in boolean algebras. For the bag semiring $\sqcup$ and $\sqcap$ are realized as max and min correspondingly.

Note, that every l-semiring is bounded below and the minimal element is always $\mathbb{0}$. However, it is not necessary for an l-semiring to be bounded above. For example, the bag and provenance semirings do not have maximal elements.

Now we are ready to define a containment for combined annotations. For readability first we start with the case of two dependent domains.

Definition 4. *Let $\mathbf{K}$ consist of two dependent over $\rightleftharpoons$ l-semirings $\mathcal{K}_1$ and $\mathcal{K}_2$. Then $\lambda \preceq^* \mu$ holds for combined annotations λ and μ over $\mathbf{K}$ iff there exist*

- *combined annotations λ' and μ' such that $\lambda' \subseteq \mu'$ and*
- *vectors $(u_1, u_2), (v_1, v_2)$ and (w_1, w_2) such that*
    - *either $u_1 \sqcup v_1 \trianglelefteq w_1$ and $u_2 = v_2 = w_2$,*
    - *or $u_2 \sqcup v_2 \trianglelefteq w_2$ and $u_1 = v_1 = w_1$;*

such that

$$\lambda = \{(u_1, u_2), (v_1, v_2)\} \cup \lambda',$$
$$\mu = \{(w_1, w_2)\} \cup \mu'.$$

A combined annotation λ is contained in μ iff $\lambda \preceq \mu$, where $\preceq$ is the transitive closure of $\preceq^*$.

Since $\preceq^*$ is clearly reflexive, the containment relation $\preceq$ is a preorder. Trivially, for each λ and μ it holds that $\lambda \preceq \lambda \cup \mu$. The definition of containment as a transitive closure is intuitive, but not constructive. Next we give an algebraic characterization of this relation for a special case when the underlying lattices of the domains are distributive.[4] (We will discuss such domains in more detail in Sec. 5.) Again, we characterize the containment for the special case of two dependent domains $\mathcal{K}_1$ and $\mathcal{K}_2$. To do it we need an auxiliary notion: for a combined annotation λ over $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ and $w \in K_1$ we use $Cov(\lambda, w)$ for the set of subsets of λ *covering w by the first component*, i.e., the set of all $\kappa \subseteq \lambda$ such that

$$w \trianglelefteq \bigsqcup_{(u_1,u_2)\in\kappa} u_1.$$

Proposition 1. *For combined annotations λ and μ over $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ such that $\mathcal{K}_1$ and $\mathcal{K}_2$ are dependent l-semirings and the underlying lattices are distributive, the following are equivalent:*
(1) $\lambda \preceq \mu$,
(2) for every value $w \in K_1$ it holds that

$$\bigsqcup_{\kappa'\in Cov(\lambda,w)} \left(\bigsqcap_{(u_1,u_2)\in\kappa'} u_2 \right) \trianglelefteq \bigsqcup_{\kappa''\in Cov(\mu,w)} \left(\bigsqcap_{(v_1,v_2)\in\kappa''} v_2 \right).$$

Note, that in the left hand side of the latter equation the outer $\sqcup$ is given just for symmetry and can be safely replaced by a universal quantification over all κ' from $Cov(\lambda, w)$. Also, as opposed to Def. 4 this characterization is asymmetric w.r.t. the order of the domains. It means that we can formulate a "mirror" analog of Prop. 1 based on the notion of covering by the second component.

Example 9. Consider combined annotations λ' and λ'' in Ex. 7 with their graphical representations in Fig. 2. For the element $w =$ 2004-2006 from $\mathcal{A}_Y$ the set $Cov(\lambda', w)$ consists of all κ' such that

$$\{(\text{2004-2008}, \{UK, Germany\})\} \subseteq \kappa' \subseteq \lambda',$$

and the set $Cov(\lambda'', w)$ consists of all κ'' such that

$$\{(\text{2004-2008}, \{UK, Germany\})\} \subseteq \kappa'' \subseteq \lambda'' \text{or}$$
$$\{(\text{2004-2010}, \{Germany\})\} \subseteq \kappa'' \subseteq \lambda''.$$

Which means that on both sides of the inequality from the part (2) in Prop. 1 we have $\{UK, Germany\}$, i.e this inequality holds. By the same reasons for every other $w \in \mathcal{A}_Y$ the corresponding inequality also holds, which shows that by Prop. 1 we have $\lambda' \preceq \lambda''$ as desired. Similarly we can check, that $\mu \preceq \lambda''$ holds for any set of rectangles μ which cover some solid dots from Fig. 2, but none of the hollow dots.

[4]A lattice is *distributive* iff $(u \sqcup v) \sqcap w = (u \sqcap w) \sqcup (v \sqcap w)$ holds for every elements u, v and w.

Now we give the definition of the containment relation for the general case of an arbitrary dependency graph.

Definition 4′. *Let $\mathbf{K}$ be a set of l-semirings and $\rightleftharpoons$ be a dependency graph over it. Then $\lambda \preceq^* \mu$ holds for combined annotations λ and μ over $\mathbf{K}$ iff there exist*

- *an independent set $\mathbf{I} \not\subseteq \rightleftharpoons$,*
- *combined annotations λ' and μ' such that $\lambda' \subseteq \mu'$, and*
- *vectors $(u_1, \ldots, u_n), (v_1, \ldots, v_n)$ and $(w_1, \ldots, w_n)$ such that $u_i \sqcup v_i \trianglelefteq w_i$ for each $i \in \mathbf{I}$, and $u_i = v_i = w_i$ otherwise*

for which

$$\lambda = \{(u_1, \ldots, u_n), (v_1, \ldots, v_n)\} \cup \lambda', \\ \mu = \{(w_1, \ldots, w_n)\} \cup \mu'.$$

A combined annotation λ is contained in *μ iff $\lambda \preceq \mu$, where $\preceq$ is a transitive closure of $\preceq^*$.*

This definition is a generalization of Def. 4. Indeed, if $\mathbf{K}$ consists of two dependent domains $\mathcal{K}_1$ and $\mathcal{K}_2$, there are two independent sets $\mathbf{I}_1 = \{\mathcal{K}_1\}$ and $\mathbf{I}_2 = \{\mathcal{K}_2\}$. Hence, each of the cases in the second condition in Def. 4 is just an instantiation of the third condition in Def. 4′.

The containment relation for the general case is a preorder as well. It is also possible to generalize the characterization from Prop. 1. However, it would look rather unreadable, since a domain may be in different independent sets of a dependency graph. Hence, we leave it out of the presentation of this work.

Note, that the definitions of combined annotations use only the lattice operations and does not assume any other structure on the annotation domains, i.e., the containment preorder is well-defined for any set of l-semirings and any dependency graph.

Next we introduce the equivalence on combined annotations.

Definition 5. *Given a set of l-semirings and a dependency graph $\rightleftharpoons$ over $\mathbf{K}$, combined annotations λ and μ are* equivalent over *$\rightleftharpoons$ (written $\lambda \sim \mu$) iff $\lambda \preceq \mu$ and $\mu \preceq \lambda$, i.e., $\sim$ is the equivalence relation induced by $\preceq$.*

We use $\lambda^{\sim}$ for the equivalence class which contains λ.

The following example shows that the equivalent combined annotations really formalize the intuition that they represent the same information.

Example 10. Coming back to Fig. 2, in Ex. 9 we saw that $\lambda' \preceq \lambda''$. Similarly, we can show that $\lambda'' \preceq \lambda'$, which means that $\lambda' \sim \lambda''$ as expected.

A trivial property of the equivalence is that if a combined annotation λ contains a vector $(u_1, \ldots, u_n)$ which is greater (by $\trianglelefteq$) than a vector $(v_1, \ldots, v_n)$ then $\lambda \sim \lambda \cup \{(v_1, \ldots, v_2)\}$. It means, that the equivalence of combined annotations is stable against adding and removing vectors which are less than other vectors in the annotation.

Hence, our object of interest is the quotient set of the combined annotations over the equivalence relation $\sim$. We write this set $C^{\sim}[\mathbf{K}]$. Also, we will sometimes refer to combined annotations by their equivalence classes if it does not introduce an ambiguity. In the next section we will construct a semiring of combined annotations which will allow to use it as a proper annotation domain.

5. SEMIRINGS OF COMBINED ANNOTATIONS

Having defined the set of combined annotations $C^{\sim}[\mathbf{K}]$, the next step is to introduce basic operations on it. First we will define the operations $\oplus$ and $\otimes$ for combined annotations, ignoring the equivalence $\sim$. After it we will show that, with a reasonable restriction for the semirings, $\sim$ is in fact a congruence relation w.r.t. $\oplus$ and $\otimes$, i.e., it can be safely applied to any element of an equivalence class. Finally, we will prove that $C^{\sim}[\mathbf{K}]$ with those operations forms a semiring (again, under some assumptions about the dependency graph $\rightleftharpoons$). This will allow us to use combined annotations as a domain for $\mathcal{RA}^+_{\mathcal{K}}$.

As for the definition of the containment relation in the previous Sec. 4, for the readability reasons we will first define $\oplus$ and $\otimes$ for the simplest interesting case of two dependent domains.

Definition 6. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be dependent l-semirings. Then the* sum *of combined annotations λ and μ over $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ is defined as follows:*

$$\begin{aligned} \lambda \oplus \mu \;=\;& \lambda \cup \mu \cup \\ & \{(u_1 \oplus v_1, u_2 \sqcap v_2) \mid (u_1, u_2) \in \lambda, (v_1, v_2) \in \mu\} \cup \\ & \{(u_1 \sqcap v_1, u_2 \oplus v_2) \mid (u_1, u_2) \in \lambda, (v_1, v_2) \in \mu\}. \end{aligned}$$

The idea behind this definition is the following. For every pair of vectors $(u_1, u_2) \in \lambda$ and $(v_1, v_2) \in \mu$ the value $u_2 \sqcap v_2$ is their "common part" by the second component. Hence, both of the first components u_1 and v_1 "hold" on this common part, and their sum should be included to $\lambda \oplus \mu$ on $u_2 \sqcap v_2$. It means that $\lambda \oplus \mu$ should contain the vector $(u_1 \oplus v_1, u_2 \sqcap v_2)$. By the reasons of symmetry it should contain also the vector $(u_1 \sqcap v_1, u_2 \oplus v_2)$. Finally, $\lambda \oplus \mu$ should be always greater than both λ and μ, so it includes them as subsets.

Example 11. As we have already observed, the semirings $\mathcal{A}_Y$ and $\mathcal{A}_E$ are boolean algebras which means that their sums coincide with $\sqcup$ (as set unions). Hence, for the combined annotations from Ex. 7 we have

$$\{(\text{2004-2008}, \{UK, Germany\})\} \oplus \\ \{(\text{2007-2010}, \{Germany, Italy, Cyprus\})\} = \lambda''.$$

In this simple example the domains are boolean algebras and, hence, their $\oplus$ are dual for $\sqcap$. However it is powerful enough to handle also any l-semirings where $\oplus$ and $\sqcap$ are not dual.

Definition 7. *The* product *of combined annotations λ and μ over $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ with dependent components is defined as follows:*

$$\begin{aligned} \lambda \otimes \mu \;=\;& \{(u_1 \otimes v_1, u_2 \sqcap v_2) \mid (u_1, u_2) \in \lambda, (v_1, v_2) \in \mu\} \cup \\ & \{(u_1 \sqcap v_1, u_2 \otimes v_2) \mid (u_1, u_2) \in \lambda, (v_1, v_2) \in \mu\}. \end{aligned}$$

The intuition behind the definition of the product is almost the same as for the definition of the sum, but the difference is that it does not contain the multipliers λ and μ as subsets, which appears to be reasonable for many examples. One of them is the following.

Example 12. If $\mathcal{K}_1$ and $\mathcal{K}_2$ are dependent boolean algebras, then their products coincide with corresponding $\sqcap$. It means that the definition of the product over $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ collapses just to pairwise intersection of components.

However, we have more freedom in the choice of the product than we have for the sum. For example, if both $\mathcal{K}_1$ and $\mathcal{K}_2$ are lineage semirings, since the sum and product behave similarly for each of them, it might be expected for the sum and the product of their combined annotations also to be similar. In this case it is reasonable to include to the product $\lambda \otimes \mu$ the sources λ and μ, as it is for the sum $\lambda \oplus \mu$. We leave this choice open but note that it is important to check the dependency graph $\rightleftharpoons$ to be an *ad*-graph w.r.t. $\oplus$ and this new $\otimes$, as it is needed in Theor. 1 (see further in this section).

Next we define $\oplus$ and $\otimes$ for the general case of arbitrary set $\mathbf{K}$ and dependency graph $\rightleftharpoons$. To do it we need auxiliary *extension* operations $\oplus^{\mathbf{I}}$ and $\otimes^{\mathbf{I}}$ over an independent set $\mathbf{I} \not\subseteq \rightleftharpoons$ defined for any vectors $(u_1, \ldots, u_n), (v_1, \ldots, v_n)$ in the following way:

$$(u_1, \ldots, u_n) \oplus^{\mathbf{I}} (v_1, \ldots, v_n) = (u_1 \circ_1 v_1, \ldots, u_n \circ_n v_n),$$
$$\text{where } \circ_i = \begin{cases} \oplus \text{ if } i \in \mathbf{I}, \\ \sqcap \text{ if } i \notin \mathbf{I}; \end{cases}$$

$$(u_1, \ldots, u_n) \otimes^{\mathbf{I}} (v_1, \ldots, v_n) = (u_1 \diamond_1 v_1, \ldots, u_n \diamond_n v_n),$$
$$\text{where } \diamond_i = \begin{cases} \otimes \text{ if } i \in \mathbf{I}, \\ \sqcap \text{ if } i \notin \mathbf{I}. \end{cases}$$

The meaning of these extensions is the following: for a pair of vectors it constructs a vector which is the sum (or the product) of them on some pairwise independent dimensions but the intersection on all other dimensions. It means that we sum (or multiply) the elements of the vectors only for the "common" part of the values of the dependent domains. The extensions generalize vectors like $(u_1 \oplus v_1, u_2 \sqcap v_2)$ in Defs. 6 and 7.

Example 13. In the settings of Ex. 7, for $\mathbf{I} = \{\mathcal{A}_Y\}$ we have

$$\begin{aligned}&(2004-2008, \{UK, Germany\}) \oplus^{\mathbf{I}}\\ &(2007-2010, \{Germany, Italy, Cyprus\}) \;=\\ &(2004-2010, \{Germany\}).\end{aligned}$$

Having the extensions we are ready to define the $\oplus$ and $\otimes$ operations for the general case of combined annotations.

Definition 6′. *Given a set of l-semirings* $\mathbf{K}$*, for any pair* λ, μ *of combined annotations over* $\rightleftharpoons$*:*

$$\begin{aligned}\lambda \oplus \mu \;=\; & \lambda \cup \mu \cup \\ & \{(u_1, \ldots, u_n) \oplus^{\mathbf{I}} (v_1, \ldots, v_n) \mid \\ & (u_1, \ldots, u_n) \in \lambda, (v_1, \ldots, v_n) \in \mu, \mathbf{I} \not\subseteq \rightleftharpoons\}.\end{aligned}$$

Definition 7′. *Similarly,*

$$\begin{aligned}\lambda \otimes \mu \;=\; & \{(u_1, \ldots, u_n) \otimes^{\mathbf{I}} (v_1, \ldots, v_n) \mid \\ & (u_1, \ldots, u_n) \in \lambda, (v_1, \ldots, v_n) \in \mu, \mathbf{I} \not\subseteq \rightleftharpoons\}.\end{aligned}$$

These definitions are clearly generalizations of Defs. 6 and 7.

The next step is to show that the equivalence $\sim$ is a congruence relation w.r.t. $\oplus$ and $\otimes$. However, it is not always the case and we need one more restriction on semirings.

Definition 8. *A l-semiring* $\mathcal{K}$ *is* lattice-distributive *(dl-*semiring*, for short) if* $\sqcap, \oplus,$ *and* $\otimes$ *distribute over* $\sqcup$*, i.e., for each* u, v *and* w *the following equations of distributivity hold*

$$\begin{aligned}(u \sqcup v) \sqcap w &= (u \sqcap w) \sqcup (v \sqcap w),\\ (u \sqcup v) \oplus w &= (u \oplus w) \sqcup (v \oplus w),\\ (u \sqcup v) \otimes w &= (u \otimes w) \sqcup (v \otimes w).\end{aligned}$$

Note, that the first law says, that the underlying lattice is distributive, and $(v \sqcap u) \sqcup w = (v \sqcup w) \sqcap (u \sqcup w)$ also holds.

Again, all of annotation domains are *dl*-semirings, as far as we are aware. For them we can state the following proposition.

Proposition 2. *Let all the annotation domains in* $\mathbf{K}$ *be dl-semirings. Then the equivalence* $\sim$ *is congruent over the* $\oplus$ *and* $\otimes$ *operations, i.e., if* $\lambda \sim \lambda'$ *and* $\mu \sim \mu'$ *over* $\rightleftharpoons$ *then* $\lambda \oplus \mu \sim \lambda' \oplus \mu'$ *and* $\lambda \otimes \mu \sim \lambda' \otimes \mu'$*.*

From this proposition we see that the operations are compatible with the equivalence and can be extended to elements of $C^{\sim}[\mathbf{K}]$.

Also, we need to define the $\mathbb{0}$ and $\mathbb{1}$ elements for combined annotations. This is straightforward:

$$\begin{aligned}\mathbb{0} &= \{(\mathbb{0}_1, \ldots, \mathbb{0}_n)\}^{\sim},\\ \mathbb{1} &= \{(\mathbb{1}_1, \ldots, \mathbb{1}_n)\}^{\sim}.\end{aligned}$$

Next we want to show, that the set of combined annotations $C^{\sim}[\mathbf{K}]$ together with $\oplus$ and $\otimes$ operations forms a semiring. However, it is not always the case. Particularly, consider the following example.

Example 14. Let $\mathbf{K} = \{\mathcal{N}', \mathcal{N}''\}$ where $\mathcal{N}''$ and $\mathcal{N}''$ are two dependent bag semirings. Consider $\lambda = \{(1,1)\}, \mu = \{(1,1)\}$ and $\nu = \{(1,2)\}$. Then $(\lambda \oplus \mu) \oplus \nu$ contains the vector $(2,2)$, but $\lambda \oplus (\mu \oplus \nu)$ does not contain any vector greater or equal to it (w.r.t. $\trianglelefteq$). Hence these combined annotations are not equivalent and the associativity does not hold.

Similar things can be shown about the associativity of $\otimes$ and the distributivity of $\otimes$ over $\oplus$. Hence, we have to restrict ourselves only with dependency graphs for which these laws hold.

Definition 9. *A dependency graph* $\rightleftharpoons$ *is* associative-distributive *(ad-*graph*, for short) if the corresponding operations* $\oplus$ *and* $\otimes$ *are associative, and* $\otimes$ *is distributive over* $\oplus$*.*

This restriction is quite strong. As we have seen above in Ex. 14, if $\rightleftharpoons$ contains two dependent bag semirings, it is not an *ad*-graph. However, we could not find a reasonable practical example of dependent bag semirings. Next we give some examples when $\rightleftharpoons$ is an *ad*-graph and when it is not.

Example 15. If all domains in $\mathbf{K}$ are distributive lattices then $\rightleftharpoons$ is an *ad*-graph. If all the domains in $\mathbf{K}$ but one are distributive lattices and that one is either bag, fuzzy, or lineage semirings, then $\rightleftharpoons$ is still an *ad*-graph. However, if $\mathbf{K}$ contains bag and fuzzy semirings which both are dependent to a distributive lattice, then $\rightleftharpoons$ is not an *ad*-graph.

Finally, we have the main theorem.

THEOREM 1. *Let $\mathbf{K}$ be a set of dl-semirings and $\rightleftharpoons$ be an ad-graph. Then the algebra*

$$\mathcal{C}^{\sim}[\mathbf{K}] = \langle C^{\sim}[\mathbf{K}], \oplus, \otimes, \mathbb{0}, \mathbb{1}\rangle$$

forms a semiring.[5]

This theorem enables us to combine several annotation domains with a complex dependency relation into a single annotation domain for the positive relational algebra and be sure that the annotations "pass" as expected over queries in $\mathcal{RA}^{+}_{\mathcal{C}^{\sim}[\mathbf{K}]}$. This is one of the main results of the paper. However, the definition of the combined annotations as a quotient set gives us a large and impractical representation and leaves open the question of their practical comparison. To fill this gap, in the next section we introduce a normal form for combined annotations and develop a normalization algorithm.

6. NORMAL FORM FOR COMBINED ANNOTATIONS

The semiring $\mathcal{C}^{\sim}[\mathbf{K}]$ gives us a mechanism to work with combined annotations, i.e., annotations which are dependent on each other. However, the definitions of the containment $\preceq$ and the equivalence $\sim$ are not practical. It is even unclear if the containment and equivalence are decidable. Hence, it would be useful to have a normal form for combined annotations and a normalization algorithm which would provide the possibility of storing combined annotations in a uniform way in order to facilitate comparison. In this section we introduce such a normal form together with a normalization procedure. Rather than starting with a pair of annotation domains, we do this directly for an arbitrary dependency graph.

From now on in this section, for any independent set $\mathbf{I} \notin \rightleftharpoons$, we define

$$\odot_i^{\mathbf{I}} = \begin{cases} \sqcup \text{ if } i \in \mathbf{I}, \\ \sqcap \text{ if } i \notin \mathbf{I}. \end{cases}$$

DEFINITION 10. *A combined annotation λ over a set of l-semirings $\mathbf{K}$ is in* extensional normal form *over a dependency graph $\rightleftharpoons$ (*e.n.f.*, for short) iff*

(1) for every $(u_1, \dots, u_n), (v_1, \dots, v_n) \in \lambda$ and every $\mathbf{I} \not\subseteq \rightleftharpoons$ there exists $(w_1, \dots, w_n) \in \lambda$ such that

$$(u_1 \odot_1^{\mathbf{I}} v_1, \dots, u_n \odot_n^{\mathbf{I}} v_n) \trianglelefteq (w_1, \dots, w_n),$$

(2) it is an antichain[6] *(w.r.t. $\trianglelefteq$).*

Example 16. The combined annotation λ'' given in Ex. 7 is in e.n.f.

To justify the claim that this is a normal form, we need to prove that in every equivalence class from $C^{\sim}[\mathbf{K}]$ there exists one and only one combined annotation in the extensional normal form. Unfortunately, it is not true in the general case. Particularly, if for two l-semirings their underlying lattices are free, then their sublattices can be infinite even for finite number of generators. Hence, we can find a combined annotation such that sets with all the properties from Def. 10 of equivalence exist, but all of them are infinite, i.e., not combined annotations by themselves. Hence, we have to restrict ourselves once more to a model where e.n.f. is always finite. A reasonable constraint is again to use semirings which the lattice order is distributive, i.e., dl-semirings from the previous Sec. 5. Distributive lattices have a property that they are always finite if they have a finite number of generators. Using this fact we can prove that every equivalence class of combined annotations over distributive lattices contain an element in e.n.f. (and it is finite).

[5]It is possible to show that it is actually a dl-semiring, such that the partial order induced on $C^{\sim}[\mathbf{K}]$ by the preorder $\preceq$ is the natural lattice order.

[6]A subset of a partial ordered set is an *antichain* iff its elements are pairwise incomparable.

```
function Expand ($\lambda$)
    while exists $(u_1,\dots,u_n),(v_1,\dots,v_n) \in \lambda$ and $\mathbf{I} \not\subseteq \rightleftharpoons$,
            such that $(u_1 \odot_1^{\mathbf{I}} v_1,\dots,u_n \odot_n^{\mathbf{I}} v_n) \notin \lambda$ do
        set $\lambda := \lambda \cup \{(u_1 \odot_1^{\mathbf{I}} v_1,\dots,u_n \odot_n^{\mathbf{I}} v_n)\}$;
    od;
    return $\lambda$
end of Expand;

function Red ($\lambda$)
    while exists $(u_1,\dots,u_n),(v_1,\dots,v_n) \in \lambda$
            such that $(u_1,\dots,u_n) \lhd (v_1,\dots,v_n)$ do
        set $\lambda := \lambda \backslash \{(u_1,\dots,u_n)\}$;
    od;
    return $\lambda$
end of Red;

function Norm ($\lambda$)
    return Red (Expand ($\lambda$))
end of Norm;
```

Figure 3: Normalization algorithm

THEOREM 2. *If all the semirings in $\mathbf{K}$ are dl-semirings, then any combined annotation over a dependency graph $\rightleftharpoons$ has a unique equivalent one in e.n.f.*

Hence, for dl-semirings e.n.f. always exists. However, the following proposition says, that it can be exponentially large even for boolean algebras.

PROPOSITION 3. *If $\mathbf{K}$ consists of two dependent infinite boolean algebras, then for every number N there exists a combined annotation λ over $\mathbf{K}$ with a polynomial in N number of elements, such that its e.n.f. contains $O(2^N)$ elements.*

The next step is to design a normalization procedure for combined annotations. Consider the algorithm in Fig. 3. To compute e.n.f. for a combined annotation λ one can evaluate the function `Norm`, which successively calls `Expand` and `Red`. In the function `Expand`, λ is enriched iteratively by all possible pairs of the form $(u_1 \odot_1^{\mathbf{I}} v_1, \dots, u_n \odot_n^{\mathbf{I}} v_n)$, $\mathbf{I} \not\subseteq \rightleftharpoons$. Intuitively, it "union" the values on the independent set $\mathbf{I}$ on the intersection by other dimensions. In the following Prop. 4 it will be formally proved that it does not change the equivalence class of λ. After its evaluation the first requirement of Def. 10 is satisfied. In the function `Red`(λ),[7] all

[7]In this function the notation $\mu \lhd \nu$ is used as a shorthand for $(\mu \trianglelefteq \nu) \wedge (\mu \neq \nu)$.

redundant vectors are removed, i.e., pairs which are strictly less than others in λ, so that second requirement of Def. 10 is also satisfied. This function also does not change the equivalence class of λ. Note that the exact order of choice of pairs in these functions is not defined, but the result is deterministic. Formally we have the following proposition.

Proposition 4. *Let* $\mathbf{K}$ *be a set of dl-semirings. Then for any combined annotation* λ *over* $\mathbf{K}$
(1) the normalization algorithm stops,
(2) `Norm(`λ`)` *is a combined annotation in the e.n.f.,*
(3) $\lambda \sim$ `Norm(`λ`)`.

Next we give some examples of combined annotations in e.n.f.

Example 17. As we have seen, the combined annotations λ' and λ'' from Ex. 7 are equivalent and the second one is in e.n.f. If we apply `Norm` to λ' a run of `Expand` takes just two iterations of the loop and the run of `Red` is empty. As expected, we have `Norm(`λ'`)` $= \lambda''$.

An obvious optimization of the algorithm is to apply `Red` function after every step of `Expand`. It is based on the observation that vectors, which may be removed by an intermediate call of `Red`, will give rise only to vectors which will be removed by the final run of `Red` anyway.

The following proposition says, that e.n.f. offers an efficient method for checking equality and containment for annotations, provided that the normal forms are small. However, this is not always the case, as we have seen in the Prop. 3.

Proposition 5. *Given two combined annotations* λ *and* μ *over a dependency graph* $\rightleftharpoons$ *in e.n.f. we have* $\lambda \preceq \mu$ *iff for each vector* $(u_1, \ldots, u_n)$ *from* λ *there exists a vector* $(v_1, \ldots, v_n)$ *from* μ *such that* $(u_1, \ldots, u_n) \trianglelefteq (v_1, \ldots, v_n)$.

To summarize, under an assumption that $\mathbf{K}$ consists of *dl*-semirings and $\rightleftharpoons$ is an *ad*-graph, in last two sections we have developed machinery to work with combined annotations some of which are dependent and some of them are not.

7. PRACTICALITIES

We have seen that properly combining annotations is a non-trivial matter. What can we do in practice? How should we store combined annotations in such a form that we can use them in queries? As we have seen in Prop. 3, in the general case, for annotations arising from the combination of two *dl*-semirings the extensional normal form can be exponential in the size of the given annotations and checking equivalence is intractable. By a size of a combined annotation we mean the number of pairs in it; such a choice is justified by the fact, that an exact representation model can be different for different annotation domains. Even when, as in Ex. 2 each combined annotation is a pair of sets, we have seen that it is possible to find a set of annotations whose e.n.f. is exponential in the number of annotations. Of course, in this special case, it is much more economical to represent the combined annotation as the set of all pairs that are in some annotation (the solid dots in Fig. 2).

On the other hand, the good news is that there are special cases in which we can expect this normal form to be an efficient representation for combined annotations. For example, one annotation is often time – transaction or valid – and the other annotation is monotone or almost monotone, in time. This is the case, for example, with sets of keywords or comments. Deletion of a comment is rare, and sometimes not even allowed. In this case the representation of the combined annotation in existential normal form is quite efficient.

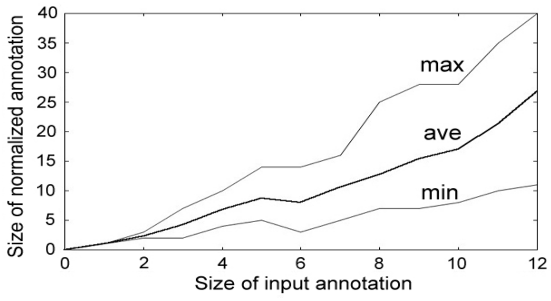

Figure 4: Dependency of the e.n.f. size of the size of the original annotations

Because the interdependence of annotations is not understood, it is unsurprising that it is difficult to find good examples. As a crude experiment we took the information from a suitably transformed version of the history of the World Factbook [9, 8] which provides sets of export partners and the set of versions which gives that set, roughly as described in Ex. 7. For example the entry for Greece is of the form:

$[v_0, v_1)$	{Italy, Germany, US, UK, France}
$[v_1, v_2)$	{EU, Italy, US, UK, France}
$[v_2, v_3)$	{EU, Italy, US, UK}
...	

in which the $[v_i, v_j)$ indicates that the entry was of this form in contiguous versions $v_i, v_i + 1, \ldots, v_j - 1$. A plot is shown in Fig. 4 of the size of the normal form against the size of the input annotation for the 258 countries that have a non-empty annotation, grouped by the size of their input annotations.

The average increase in size of e.n.f. over the size of the input annotation is about 1.6. Very little should be inferred from this example. We should reiterate that in this case, there are more efficient methods of storing the annotation for the purpose of evaluating algebraic expressions over them, but it gives some indication that there are cases in which the extensional normal form is practical.

8. RELATED AND FUTURE WORK

In this paper we have attempted to provide a basis for the study of combined annotations. While we are unaware of any other work that attempts a general approach at this problem, there are certainly cases in which the combination of specific annotation domains has been addressed in some detail. Most notably the interaction of transaction time, $\mathcal{A}_{Tt}$, valid time $\mathcal{A}_{Tv}$ and the bag semiring $\mathcal{N}$ has been thoroughly worked out [21] in order to deal with the practical issues of having temporal annotations on SQL, which is already equipped with bag semantics. Interestingly the dependency graph adopted appears to be $\{\mathcal{A}_{Tt} \rightleftharpoons \mathcal{N}, \mathcal{A}_{Tv} \rightleftharpoons \mathcal{N}\}$. Another place in which combined annotations figure is the study of belief databases [15]. Here the authors study chains of beliefs such as "a believes that b believes that c believes ...". The chains have varying lengths, but this can easily be simulated in our model by padding shorter chains with the top element of the domain – meaning "everybody believes…". Recently, in [2], semiring provenance has been extended for aggregate queries; this approach is, on the face of it, quite different with ours, but there may be a useful connection.

While this paper highlight the problem of combined annotations, the next step is to develop a more comprehensive theory of this topic. Some possible extensions are listed below.

- We have concentrated on the principles of combined annotations and their normalization. It would be useful to have general complexity bounds for containment and equivalence. While the normalization process is exponential in the general case, there are numerous special cases in which the existential normal form is itself small, or can be compactly represented.
- Recently several extensions and applications of the original work [18] have been proposed, including [16], where the algebra $\mathcal{RA}^+_\mathcal{K}$ has been extended with a difference operator, and [22, 7], where a similar model has been applied to inference rules of Semantic Web language RDF(S). In these cases the semiring model has been augmented with additional axioms. Can the same ideas be applied to these models?
- Consider a pair in a combined annotation, such that one of its components is $\mathbb{0}$. One can argue that such a pair has no real meaning and may be removed from the annotation safely. In this paper we turned a blind eye to this case. It may be appropriate to give this case special treatment.

9. ACKNOWLEDGEMENTS

We are grateful to Rick Snodgrass for his help in our understanding of TSQL and to Heiko Müller for extracting temporal data from the CIA World Factbook.

10. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] Y. Amsterdamer, D. Deutch, and V. Tannen. Provenance for aggregate queries. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS'11, pages 153–164, New York, NY, USA, 2011. ACM.

[3] G. Bella and S. Bistarelli. Soft constraints for security protocol analysis: Confidentiality. In I. Ramakrishnan, editor, *Practical Aspects of Declarative Languages*, volume 1990 of *Lecture Notes in Computer Science*, pages 108–122. Springer Berlin / Heidelberg, 2001.

[4] T. Berners-Lee. Linked data. `http://www.w3.org/DesignIssues/LinkedData.html`, June 2009.

[5] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research*, 31(1):365–370, 2003.

[6] P. Buneman, S. Khanna, and W. Tan. Why and Where: A Characterization of Data Provenance. In *ICDT 2001*, volume 1973 of *LNCS*, pages 316–330. Springer, 2001.

[7] P. Buneman and E. V. Kostylev. Annotation Algebras for RDFS. In *SWPM 2010*. CEUR Workshop Proceedings, 2010.

[8] P. Buneman, H. Müller, and C. Rusbridge. Curating the CIA World Factbook. *International Journal of Digital Curation,*, 4(3), 2009.

[9] Central Intelligence Agency. The World Factbook. `https://www.cia.gov/library/publications/the-world-factbook/`.

[10] L. Chiticariu, W.-C. Tan, and G. Vijayvargiya. DBNotes: a post-it system for relational databases based on provenance. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 942–944, New York, NY, USA, 2005. ACM.

[11] Y. Cui, J. Widom, and J. L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25(2):179–227, 2000.

[12] Y. Cui, J. Widom, and J. L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25:179–227, June 2000.

[13] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems*, 15:32–66, 1994.

[14] M. Y. Galperin and G. R. Cochrane. The 2011 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nuclear Acids Research*, 39:1–6, 2011.

[15] W. Gatterbauer, M. Balazinska, N. Khoussainova, and D. Suciu. Believe it or not: Adding belief annotations to databases. *CoRR*, abs/0912.5241, 2009.

[16] F. Geerts and A. Poggi. On database query languages for K-relations. *J. Applied Logic*, 8(2):173–185, 2010.

[17] T. Green. Containment of conjunctive queries on annotated relations. *Theory of Computing Systems*, 49:429–459, 2011. 10.1007/s00224-011-9327-6.

[18] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40, New York, NY, USA, 2007. ACM.

[19] T. Imielinski and W. Lipski Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.

[20] Z. M. Ma and L. Yan. A literature overview of fuzzy database models. *J. Inf. Sci. Eng.*, 24(1):189–202, 2008.

[21] R. T. Snodgrass, editor. *The TSQL2 Temporal Query Language.* Kluwer, 1995.

[22] U. Straccia, N. Lopes, G. Lukacsy, and A. Polleres. A General Framework for Representing and Reasoning with Annotated Semantic Web Data. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10).* AAAI Press, 2010.

[23] E. Zimányi. Query evaluation in probabilistic relational databases. *Theor. Comput. Sci.*, 171:179–219, January 1997.

APPENDIX

In the appendix we give proofs of the propositions and theorems of the paper. Many of them are based on the results of Sec. 6, which are however proved further in this appendix.

LEMMA 1. *Let all the semirings in* $\mathbf{K}$ *be dl-semirings. Let* λ *and* μ *be combined annotations such that* μ *is in e.n.f. and* $\lambda \preceq \mu$*. Then for each vector* $(u_1, \ldots, u_n) \in \lambda$ *there exists a vector* $(v_1, \ldots, v_n) \in \mu$ *such that* $(u_1, \ldots, u_n) \trianglelefteq (v_1, \ldots, v_n)$*.*

PROOF. Since μ is in e.n.f. it is clear that $\lambda \preceq^* \mu$. Hence by the definition of e.n.f we conclude that $(u_1, \ldots, u_n) \trianglelefteq (v_1, \ldots, v_n)$. $\square$

PROPOSITION 1. *For combined annotations* λ *and* μ *over* $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ *such that* $\mathcal{K}_1$ *and* $\mathcal{K}_2$ *are dependent l-semirings and the underlying lattices are distributive, the following are equivalent:*
(1) $\lambda \preceq \mu$*,*
(2) for every value $w \in K_1$ *it holds that*

$$\bigsqcup_{\kappa' \in Cov(\lambda, w)} \left(\bigsqcap_{(u_1,u_2) \in \kappa'} u_2 \right) \trianglelefteq \bigsqcup_{\kappa'' \in Cov(\mu, w)} \left(\bigsqcap_{(v_1,v_2) \in \kappa''} v_2 \right).$$

PROOF. (1) $\Rightarrow$ (2). Since $\trianglelefteq$ is reflexive and transitive, it is valid to consider only the case when $\lambda \preceq^* \mu$. However, for any w replacing vectors $(u_1, u_2), (v_1, v_2)$ in λ with a vector (w_1, w_2) with properties given in Def. 4 clearly may only increase (w.r.t. $\trianglelefteq$) the value of $\bigsqcup_{\kappa' \in Cov(\lambda, w)} \left(\bigsqcap_{(u_1,u_2) \in \kappa'} u_2 \right)$. All the more it can be concluded about adding there the vectors from μ' (which are not in λ').
(2) $\Rightarrow$ (1). Note that in the following proofs of Theor. 2 and Prop. 4 (as well as Lemma 1) the requirement for the l-semirings $\mathcal{K}_1, \ldots, \mathcal{K}_2$ to be dl-semirings is redundant and only the destributivity of $\sqcap$ over $\sqcup$ is used. Hence, the results of Theor. 2 and Prop. 4 are valid also for the set $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ of this proposition. Thereby, having the first part proved, without lost of generality we can assume that λ and μ are in e.n.f. But then the statement is immediately following from Lemma 1. $\square$

PROPOSITION 2. *Let all the annotation domains in* $\mathbf{K}$ *be dl-semirings. Then the equivalence* $\sim$ *is congruent over the* $\oplus$ *and* $\otimes$ *operations, i.e., if* $\lambda \sim \lambda'$ *and* $\mu \sim \mu'$ *over* $\rightleftharpoons$ *then* $\lambda \oplus \mu \sim \lambda' \oplus \mu'$ *and* $\lambda \otimes \mu \sim \lambda' \otimes \mu'$*.*

PROOF. Note, that all the conditions of Theor. 2 and Prop. 4 holds. It means that it is enough to show that if λ' is obtained from λ by one step of the normalization algorithm, then

$$\lambda \oplus \mu \sim \lambda' \oplus \mu \text{ and } \lambda \otimes \mu \sim \lambda' \otimes \mu. \tag{1}$$

As it was already noted, in l-semirings the natural order $\trianglelefteq$ is compatible with $\oplus$ and $\otimes$. It means that (1) clearly holds for every step of `Red`. Hence, we only need to check it for a step of `Expand`, i.e., for the case when $\lambda' = \lambda \cup \{(u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n)\}$ for some independent set $\mathbf{I}$ and vectors $(u_1, \ldots, u_n), (v_1, \ldots, v_n)$ from λ (the operation $\odot_i^{\mathbf{I}}$ is defined in Sec. 6).

We will check only the first expression of (1), and the proof of the second one is absolutely the same. The difference between the left and the right parts of the first expression is only that the right one for any $(w_1, \ldots, w_n) \in \mu$ and $\mathbf{J} \not\subseteq \rightleftharpoons$ contains vector

$$\begin{array}{l} (u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n) \oplus^{\mathbf{J}} (w_1, \ldots, w_n) \quad = \\ ((u_1 \odot_1^{\mathbf{I}} v_1) \circ_1 w_1, \ldots, (u_n \odot_n^{\mathbf{I}} v_n) \circ_n w_n) \end{array} \tag{2}$$

but the left one not. (Here all $\circ_i$ correspond to $\mathbf{J}$.) Hence it is enough to show that every such vector is smaller (w.r.t. $\trianglelefteq$) than a vector obtained by one step of `Expand` running on $\lambda \oplus \mu$. Indeed, consider the vectors

$$(u_1, \ldots, u_n) \oplus^{\mathbf{I}} (w_1, \ldots, w_n) \text{ and } (v_1, \ldots, v_n) \oplus^{\mathbf{I}} (w_1, \ldots, w_n)$$

from $\lambda \oplus \mu$. Applying the step of `Expand` for the independent set $\mathbf{J}$ to them we got the vector

$$((u_1 \circ_1 w_1) \odot_1^{\mathbf{I}} (v_1 \circ_1 w_1), \ldots, ((u_n \circ w_n) \odot_n^{\mathbf{I}} (v_n \circ_n w_n)).$$

We claim, that this vector is greater than the vector in (2), i.e., for every i it holds that

$$(u_i \odot_i^{\mathbf{I}} v_i) \circ_i w_i \trianglelefteq (u_i \circ_1 w_i) \odot_1^{\mathbf{I}} (v_i \circ_i w_i). \tag{3}$$

Indeed, we have 4 options:

1. if $i \in \mathbf{I}$ and $i \in \mathbf{J}$ then $\odot_i^{\mathbf{I}} = \sqcup$, $\circ_i = \oplus$ and (3) holds by the distributivity of $\oplus$ over $\sqcup$;
2. if $i \in \mathbf{I}$ but $i \notin \mathbf{J}$ then $\odot_i^{\mathbf{I}} = \sqcup$, $\circ_i = \sqcap$ and (3) holds by the distributivity of $\sqcap$ over $\sqcup$;
3. if $i \notin \mathbf{I}$ but $i \in \mathbf{J}$ then $\odot_i^{\mathbf{I}} = \sqcap$, $\circ_i = \oplus$ and (3) holds by the compatibility of $\oplus$ with $\trianglelefteq$;
4. if $i \notin \mathbf{I}$ and $i \notin \mathbf{J}$ then $\odot_i^{\mathbf{I}} = \sqcap$, $\circ_i = \sqcap$ and (3) holds by the idempotence of $\sqcap$.

$\square$

THEOREM 1. *Let* $\mathbf{K}$ *be a set of dl-semirings and* $\rightleftharpoons$ *be an ad-graph. Then the algebra*

$$\mathcal{C}^{\sim}[\mathbf{K}] = \langle C^{\sim}[\mathbf{K}], \oplus, \otimes, \mathbb{0}, \mathbb{1} \rangle$$

forms a semiring.

Proof. Straightforward: the commutativity of $\oplus$ and $\otimes$ follows from commutativity of operations in the source semirings; the associativity of them and the distributivity are the properties of *ad*-graphs; the neutrality of $\mathbb{0}$ and $\mathbb{1}$ and the annihilation of $\mathbb{0}$ are trivial. $\square$

Theorem 2. *Let all the semirings in* $\mathbf{K}$ *be dl-semirings. Then any combined annotation over a dependency graph* $\rightleftharpoons$ *has a unique equivalent one in e.n.f.*

Proof. By the following Prop. 4 we have that for every combined annotation λ there exists an equivalent one $\texttt{Norm}(\lambda)$ which is in e.n.f. Hence, we only need to prove that if combined annotations λ and μ are equivalent and in e.n.f. then $\lambda = \mu$. Applying Lemma 1 twice, we have that for each vector $(u_1, \ldots, u_n) \in \lambda$ there exists a vector $(v_1, \ldots, v_n) \in \mu$ and a vector $(u'_1, \ldots, u'_n) \in \lambda$ such that $(u_1, \ldots, u_n) \trianglelefteq (v_1, \ldots, v_n) \trianglelefteq (u'_1, \ldots, u'_n)$. Since λ is an antichain, we can conclude that $(u_1, \ldots, u_n) = (v_1, \ldots, v_n) = (u'_1, \ldots, u'_n)$, i.e., $\lambda \subseteq \mu$. The same way we can prove that $\lambda \supseteq \mu$. It means that $\lambda = \mu$. $\square$

Proposition 3. *If* $\mathbf{K}$ *consists of two dependent infinite boolean algebras, then for every number* N *there exists a combined annotation* λ *over* $\mathbf{K}$ *with a polynomial in* N *number of elements, such that its e.n.f. contains* $O(2^N)$ *elements.*

Proof. For convenience, let the elements of $\mathbf{K}$ be two copies $\mathcal{B}'$ and $\mathcal{B}''$ of the same boolean algebra $\mathcal{B}$. Let S be a set of N pairwise nonintersecting elements from $\mathcal{B}$. Consider the annotation $\{(\{x\}, \{y\}) \mid x, y \in S \text{ and} x \neq y\}$. Clearly, its e.n.f. contain $O(2^N)$ elements. $\square$

Proposition 4. *Let* $\mathbf{K}$ *be a set of dl-semirings. Then for any combined annotation* λ *over* $\mathbf{K}$
(1) the normalization algorithm stops;
(2) $\texttt{Norm}(\lambda)$ *is a combined annotation in the e.n.f.;*
(3) $\lambda \sim \texttt{Norm}(\lambda)$.

Proof. To prove that the algorithm stops notice, that for every vector $(u_1, \ldots, u_n)$ which is added to λ during a run of the $\texttt{Expand}$ function, all the annotations v_i are $\sqcup, \sqcap$-combinations of annotations in the input combined annotation. By the distributivity laws of the underlying lattices of the semirings, every such $\sqcup, \sqcap$-combination can be converted to a disjunction of conjuncts such that all the conjuncts are different and all elements of a conjunct are different as well. The input combined annotation is finite by the definition, which means that there are only finitely many different vectors which can be added to λ during a run of the algorithm. It means that only finitely many iterations of the loop in $\texttt{Expand}$ are possible. The function $\texttt{Red}$ always stops by its form. Hence the normalization algorithm also always stops.

The proof that $\texttt{Norm}(\lambda)$ is a combined annotation in normal form is strait forward. It is a finite set by previous statement of this theorem, which means that it is a combined annotation. Also, every pair of vectors $(u_1, \ldots, u_n), (v_1, \ldots, v_n)$ in $\texttt{Norm}(\lambda)$ must be processed in $\texttt{Expand}(\lambda)$ during a run of the algorithm. Hence, for every independent set $\mathbf{I}$ the combined annotation $\texttt{Norm}(\lambda)$ contains the vector $(w_1, \ldots, w_n)$ such that $(u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n) \trianglelefteq (w_1, \ldots, w_n)$, i.e., the first requirement of the definition of e.n.f. holds. Finally, $\texttt{Norm}(\lambda)$ is an antichain, since a result of $\texttt{Red}$ is always an antichain, i.e., the second requirement also holds.

To proof that $\lambda \sim \texttt{Norm}(\lambda)$ we need to show that every modification of λ during a run of the algorithm does not change its equivalence class. First, $\texttt{Red}(\lambda) \sim \lambda$ by the property of the equivalence mentioned just after its definition. Hence, we need to show that $\texttt{Expand}(\lambda) \sim \lambda$, i.e., for every vectors $(u_1, \ldots, u_n)$ and $(v_1, \ldots, v_n)$, combined annotation λ' and every independent set $\mathbf{I} \not\subseteq \rightleftharpoons$ it holds that

$$(\{(u_1, \ldots, u_n), (v_1, \ldots, v_n), (u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n)\} \cup \lambda') \sim (\{(u_1, \ldots, u_n), (v_1, \ldots, v_n)\} \cup \lambda').$$

The inequality

$$(\{(u_1, \ldots, u_n), (v_1, \ldots, v_n), (u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n)\} \cup \lambda') \succeq (\{(u_1, \ldots, u_n), (v_1, \ldots, v_n)\} \cup \lambda'),$$

holds by the above-mentioned fact that for every μ and ν we have $\mu \cup \nu \succeq \mu$. Hence, we only need to prove that

$$(\{(u_1, \ldots, u_n), (v_1, \ldots, v_n), (u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n)\} \cup \lambda') \preceq (\{(u_1, \ldots, u_n), (v_1, \ldots, v_n)\} \cup \lambda'). \tag{4}$$

We will need some extra notation:

- let $(p_1, \ldots, p_n)$ and $(q_1, \ldots, q_n)$ be vectors defined as $p_i = u_i$ and $q_i = v_i$ if $i \in \mathbf{I}$, and $p_i = q_i = u_i \sqcap v_i$ otherwise;
- let $\lambda'' = \lambda' \cup \{(u_1, \ldots, u_n), (v_1, \ldots, v_n)\}$.

We will prove (4) by the following steps:

$$\begin{array}{lll}
\{(u_1, \ldots, u_n), (v_1, \ldots, v_n), (u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n)\} \cup \lambda' & \sim & \\
\{(p_1, \ldots, p_n), (q_1, \ldots, q_n), (u_1 \odot_1^{\mathbf{I}} v_1, \ldots, u_n \odot_n^{\mathbf{I}} v_n)\} \cup \lambda'' & \preceq^* & (*) \\
\{(p_1, \ldots, p_n), (q_1, \ldots, q_n)\} \cup \lambda'' & \sim & \\
\{(u_1, \ldots, u_n), (v_1, \ldots, v_n)\} \cup \lambda'. & &
\end{array}$$

The first and the last equivalences hold since $(p_1, \ldots, p_n) \trianglelefteq (u_1, \ldots, u_n)$ and $(q_1, \ldots, q_n) \trianglelefteq (v_1, \ldots, v_n)$, which means that we can safely add and remove these vectors by the properties of the equivalence. The inequality (*) holds by the definition of $\preceq^*$. $\square$

Proposition 5. *Given two combined annotations* λ *and* μ *over a dependency graph* $\rightleftharpoons$ *in e.n.f. we have* $\lambda \preceq \mu$ *iff for each vector* $(u_1, \ldots, u_n)$ *from* λ *there exists a vector* $(v_1, \ldots, v_n)$ *from* μ *such that* $(u_1, \ldots, u_n) \trianglelefteq (v_1, \ldots, v_n)$.

Proof. This proposition immediately follows from Lemma 1. $\square$