HAL
open science

Algebras of ontology alignment relations

Jérôme Euzenat

To cite this version:

Jérôme Euzenat. Algebras of ontology alignment relations. Proc. 7th international semantic web conference (ISWC), Oct 2008, Karlsruhe, Germany. pp.387-402, 10.1007/978-3-540-88564-1_25 . hal-00793543

HAL Id: hal-00793543

https://inria.hal.science/hal-00793543v1

Submitted on 22 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algebras of ontology alignment relations

Jérôme Euzenat

INRIA & LIG
Grenoble, France
Jerome.Euzenat@inrialpes.fr

Abstract. Correspondences in ontology alignments relate two ontology entities with a relation. Typical relations are equivalence or subsumption. However, different systems may need different kinds of relations. We propose to use the concepts of algebra of relations in order to express the relations between ontology entities in a general way. We show the benefits in doing so in expressing disjunctive relations, merging alignments in different ways, amalgamating alignments with relations of different granularity, and composing alignments.

1 Motivations

The heterogeneity of ontologies on the semantic web requires finding the correspondences between them in order to interoperate. The operation of finding correspondences is called ontology matching and its result is a set of correspondences called an alignment [8]. Alignments are used for importing data from one ontology to another or for translating queries.

In general, a correspondence relates an entity, e.g., a class, a property, an instance, of a first ontology to an entity of the second ontology by a specific relation. This relation can be the equivalence or subsumption between these entities or more complex relations, e.g., mereologic relations such as partOf.

Within an alignment, correspondences are interpreted conjunctively, but it may sometimes be necessary to express disjunctions of relations, e.g., when one is only able to establish a subset of the possibly holding relations. Moreover, in the wider context of sharing alignments on the web and composing ontology matchers, it is necessary to manipulate alignments: combining alignments either conjunctively or disjunctively, composing alignments when a direct alignment between two ontologies does not exist or converting alignments using a different set of relations. Current support for alignment is not adapted to this: correspondences are usually expressed with respect to simple relations and the connection between relations is not explicit.

Example 1 (Background). We consider the example of three geographic ontologies designed for statistical purposes. They are loosely built on the Eurostat Nomenclature for Territorial Units for Statistics (or NUTS). We deal with three ontologies adapted to the German (o), English (o') and French (o'') territory. In order to be able to aggregate information from the German and the British sources, engineers need an alignment between o and o'. They will take advantage of alignments provided by various sources.

We propose to solve this problem by expressing alignment relations within the formalism of algebras of relations. At first sight, algebras of relations may seem like just one possible solution to express disjunctions. However, we show that, in addition to allowing disjunctive relations, this formalism provides many advantages in the manipulation of alignments.

We first present in more detail the notion of ontology alignment and relations between ontology entities (§2), as well as algebra of relations (§3). We then show how, in addition to expressing disjunction, algebras of relations can support several types of relation aggregation (§4), alignment composition (§5), algebraic reasoning on alignments (§6), and weakening of representations (§7). We finally show that this kind of relations can still be manipulated coherently with confidence measures (§8).

2 Alignments and relations

Alignments express the correspondences between entities belonging to different ontologies (we restrict ourselves to two ontologies here). We provide the definition of the alignment following the work in [6, 3].

Definition 1 (Correspondence). *Given two ontologies o and o' with associated entity languages Q_L and $Q_{L'}$, a set of alignment relations Θ, and a confidence structure over Ξ, a* correspondence *is a quadruple:*

$$\langle e, e', r, n \rangle,$$

such that

- $e \in Q_L(o)$ *and* $e' \in Q_{L'}(o')$*;*
- $r \in \Theta$*;*
- $n \in \Xi$*.*

The correspondence $\langle e, e', r, n \rangle$ asserts that the relation r holds between the ontology entities e and e' with confidence n.

The entities can be simply made of all the formulas of the ontology language based on the ontology vocabulary. They can be restricted to particular kinds of formulas from the language, such as atomic formulas, or even to terms of the language, like class expressions. It can also restrict the entities to be only named entities. The entity language can be an extension of the ontology language. For instance, it can be a query language, such as SPARQL [14], adding operations for manipulating ontology entities that are not available in the ontology language itself, like concatenating strings or joining relations.

In some tradition, e.g., schema matching [16, 15], some authors tend to consider that a correspondence like:

$$\mathsf{address} = \mathsf{street} + \text{“ ”} + \mathsf{number}$$

is some kind of ternary complex relation ($\cdot = \cdot + \text{“ ”} + \cdot$) between three entities address, street and number. In our setting, this is simply a normal correspondence in which the binary relation is equivalence ($=$) and the ontology entities are address and

street+“ ”+number. This is the main reason why we consider ontology entities, the latter entity is a term built on strings and operations on strings (here concatenation $+$).

The next important component of the alignment is the relation that holds between the entities. We identify a set of relations Θ that is used for expressing the relations between the entities. Matching algorithms primarily use the equivalence relation ($=$) meaning that the matched objects are the same or are equivalent if these are formulas. It is possible to use relations from the ontology language within Θ. For instance, using OWL, it is possible to take advantage of the `owl:equivalentClass`, `owl:disjointWith` or `rdfs:subClassOf` relations in order to relate classes of two ontologies. These relations correspond to set-theoretic relations between classes: *equivalence* ($=$), *disjointness* ($\perp$), *less general* ($\sqsubseteq$). They can be used without reference to any ontology language.

For pragmatic reasons, the relationship between two entities is assigned a degree of confidence which can be viewed as a measure of trust in the fact that the correspondence holds – ‘I trust 70% the fact that the correspondence is correct or reliable’ – and can be compared with the certainty measures provided with meteorological forecasts. These values are taken from a bounded ordered set Ξ that we call a confidence structure. We will come back on this in Section 8 and ignore it until then.

Finally, an alignment is defined as a set of correspondences.

Definition 2 (Alignment). *Given two ontologies o and o', an* alignment *is made up of a set of correspondences between pairs of entities belonging to $Q_L(o)$ and $Q_{L'}(o')$ respectively.*

Example 2 (Alignment). Consider two alignments A_1 and A_2, relating respectively the German to the French ontology and the French ontology to the British one, containing the following correspondences (A_1 is on the left, A_2 on the right):

$$\mathsf{Konstruktion} \perp \mathsf{Commune} \qquad \mathsf{Commune} \geq \mathsf{Municipality}$$
$$\mathsf{Stadtgebiet} > \mathsf{Ville} \qquad \mathsf{Ville} \between \mathsf{Municipality}$$

This means that A_1 considers that a Konstruction, i.e., a Building, is disjoint from a Commune, i.e., a Ward, and a Stadtgebiet, i.e., a Urban area, is more general than a Ville, i.e., a Town. A_2 expresses that a Commune is more general or equivalent to a Municipality and Ville overlaps withMunicipality, i.e., that both concepts have common instances but none is more general than the other.

This definition does not tell how to interpret this set of correspondences. However, it is clear from usage that it has to be interpreted in a conjunctive manner: all the correspondences are asserted to hold when asserting an alignment.

Hence, the problem of expressing disjunctions of correspondences can be raised. This can be because it is necessary to aggregate the result of methods which address the ontology matching problem from different dimensions, this can be because the person or the program generating the alignment is unsure about the exact relation but knows that this relation is constrained to a specific set of alternative relations.

Example 3 (Disjunctive relations). For instance, an engineer may know that a Stadt, i.e., Town, and a Town are similar things but may not know exactly the nature of the

overlaps. She can express that they are not disjoint by the disjunction of relations $<, >$, $\between$ and $=$, thus prohibiting $\perp$. This can also be because the alignment has been generated by composing two alignments. This operation does not usually return a simple relation but a disjunction of such relations, e.g., if Stadtgebiet, i.e., Urban area, is more general than Ville, i.e., Town, and Ville overlaps Municipality, then Stadtgebiet either is more general or overlaps Municipality, it cannot be disjoint with it. Hence, the result is a disjunction of relations.

This is also the case of the $\geq$ (more-general-or-equal) and $\leq$ (more-specific-or-equal) relations used by some systems. These are typically the disjunction of $<$ and $=$ or $>$ and $=$. In fact, practice which considers that if both $\leq$ and $\geq$ hold (conjunction), then $=$ holds, only reflects the set operation: $\{<,=\} \cap \{>,=\} \ = \ \{=\}$ or the logical interpretation that:

$$\forall a, b, (a < b \vee a = b) \wedge (a > b \vee a = b) \models a = b \text{ if } <, > \text{ and } = \text{ are exclusive}$$

The first goal of this paper is to consider a systematic treatment for disjunctive alignment relations. For that purpose, we use algebra of relations and we show that this has many advantages.

3 Relation algebra

An algebra of binary relations (hereafter referred to as relation algebra) [19] is a structure $\langle \Theta, \wedge, \vee, *, 1, 0, 1', \neg \rangle$ such that $\langle \Theta, \wedge, \vee, 1, 0 \rangle$ is a Boolean algebra; $*$ is an associative internal composition law with (left and right) unity element $1'$, that distributes over $\vee$; $\neg$ is an internal involutive unary operator, that distributes over $\vee$, $\wedge$ and *.

We consider a particular type of relation algebras[1] in which Θ is the powerset of a generating set Γ closed under $\neg$ (hereafter $^{-1}$) and $\wedge/\vee$ are set intersection/union ($\cap/\cup$). Such an algebra of (binary) relations is defined by $\langle 2^{\Gamma}, \cap, \cup, \cdot, \Gamma, \varnothing, \{=\}, ^{-1} \rangle$ such that:

- Γ is a set of jointly exhaustive and pairwise disjoint (JEPD) relations between two entities. This means that, in any situation, the actual relation between two objects is one and only one of these relations. Sets of relations allow to express uncertainty: the full Γ set is the "I do not know" relation since it is satisfied by any pair of entities;
- $\cap$ and $\cup$ are set operations used to meet and join two sets of base relations, hence if xry or $xr'y$, then $xr \cup r'y$;
- $\cdot$ is the composition operator such that if xry and $y\ r'\ z$, then $x\ r \cdot r'\ z$; "=" is such that $\forall r \in \Gamma, (r \cdot =) = (= \cdot r) = r$;
- $^{-1}$ is the converse operator, i.e., such that $\forall e, e' \in \Gamma, ere' \Leftrightarrow e'r^{-1}e$.

These operations are applied to sets of base relations by distributing them on each element, e.g., $R \cdot R' = \bigcup_{r \in R, r' \in R'} r \cdot r'$.

[1] [12] shows that a weaker structure than algebras of relations, non associative, can be used in most of the purposes of qualitative calculi. However, we will need associativity later.

A typical example of such an algebra is the Allen algebra of temporal interval relations [1]. Here, we will consider, as an example, a simpler algebra, called $A5$, isomorphic to that applying to sets in which the base (JEPD) relations between two sets are equivalent ($=$), includes ($>$), is-included-in ($<$), overlaps ($\between$) and disjoint ($\perp$). In this algebra, all base relations but $<$ and $>$ are there own converse while $>^{-1}=<$ and $<^{-1}=>$.

The complete set of $2^5 - 1 = 31$ valid relations that can be made out of these 5 base relations is depicted in Figure 1. Among these relations, Γ means "I do not know" as it contains all the base relations. $\neq$ is equivalent to $\{<, >, \between, \perp\}$, $\leq$ is equivalent to $\{=, <\}$, $\geq$ to $\{=, >\}$, $\not\approx$ to $\{<, >, \between\}$ and $\not\perp$ to $\{=, <, >, \between\}$. The composition table is given in Table 1.

Relation algebras can still be used when the ontology entities are formulas (or queries) and the base relations are logical connectives between formulas ($\Rightarrow$, $\equiv$). Indeed, it is sufficient to split the disjunctive relations into a disjunction of formulas:

$$\phi\{\Rightarrow, \equiv\}\psi \text{ would be equivalent to } \phi \equiv \psi \vee \phi \Rightarrow \psi$$

(these relations are not exclusive anymore).

4 Aggregating matcher results

The set Θ is closed for $\vee$, $\wedge$ and $\neg$. This means that any combination of these operations yields an element of Θ. This is very powerful if one wants to combine relations, e.g., for combining correspondences and alignments.

When matching methods bring new evidences for a correspondence from a different perspective, they are though of as bringing new arguments in favor of a correspondence. Hence, its results must be aggregated with union ($R \cup R'$). When the matching methods, instead, are competing algorithms providing all the possible base relations, i.e., providing arguments against the non selected correspondences, then intersection ($R \cap R'$) should be used. Because, we now have two distinguished operations, they can be used together in the same application.

These operations can be used for describing two cases of matching process: an expanding matching process which starts with the empty relation ($\varnothing$) between each pair of entities and which finds evidences for more base relations between these entities aggregating them with $\cup$ and a contracting process which starts with Γ between each pair of entities and which discards support for some base relations between entities aggregating the result with $\cap$. In the first case, the more matching methods are used, the less precise the alignment becomes: this can be balanced by confidence measures as we will see below. In the second case, the more methods are used, the more precise the alignment becomes.

These operations on Θ are used in correspondence aggregation: an alignment is interpreted as all the correspondences it contains hold, and a distributed system is interpreted as all alignments hold. Hence, the disjunctive aggregation of alignments is based on the combination of their set of correspondences with the union of relations; the conjunctive aggregation of alignments is based on the combination of their set of

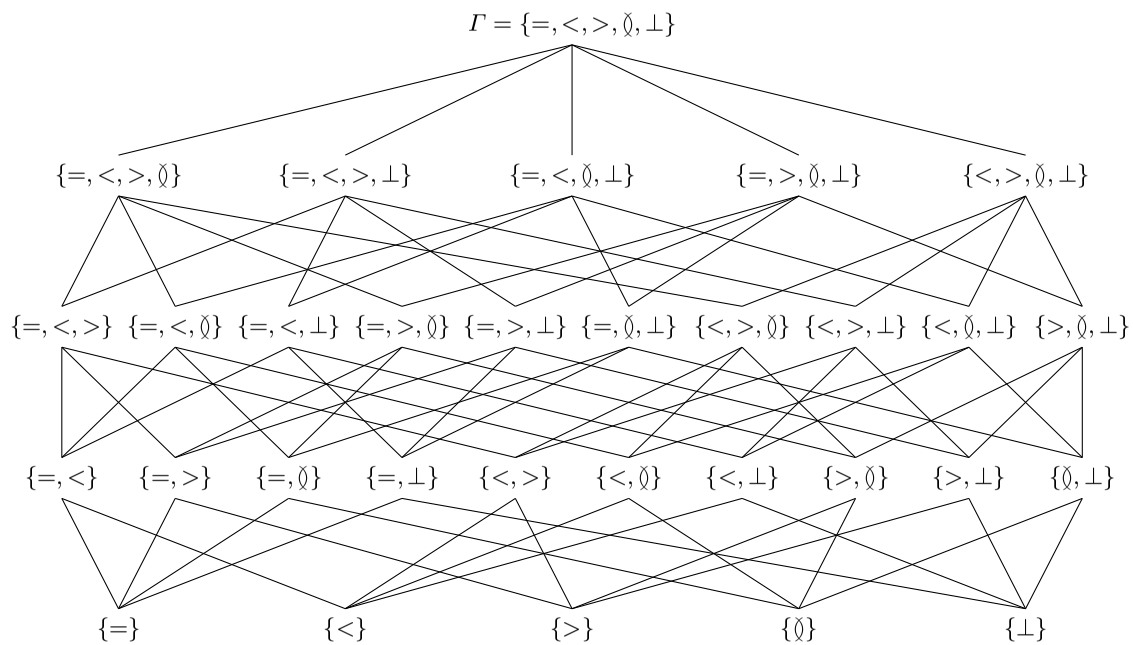

Fig. 1. The lattice of 31 disjunctive relations in $A5$.

correspondences with the intersection of relations. Hence, we define a normalisation operation $\bar{A}$, which implements the conjunctive interpretation of alignments. It provides exactly one correspondence per pair of entities and makes explicit all the relations between entities in A:

$$A^0 = \{\langle e, e', \Gamma\rangle | e \in Q_L(o), e' \in Q_{L'}(o')\}$$
$$\bar{A} = \{\langle e, e', \cap_{\langle e,e',r\rangle \in A \cup A^0} r\rangle\}$$

It is then easy to define intersection:

$$A \wedge A' = \{\langle e, e', r \cap r'\rangle | \langle e, e', r\rangle \in \bar{A}, \langle e, e', r'\rangle \in \bar{A}'\}$$

as well as additional operators such as disjunction and converse of alignments:

$$A \vee A' = \{\langle e, e', r \cup r'\rangle | \langle e, e', r\rangle \in \bar{A}, \langle e, e', r'\rangle \in \bar{A}'\}$$
$$A^{-1} = \{\langle e', e, r^{-1}\rangle | \langle e, e', r\rangle \in \bar{A}\}$$

Within this paper, alignments are always presented in a reduced way, i.e., without trivial $\langle e, e', \Gamma\rangle$ correspondences added by normalisation.

Example 4 (Alignment aggregation). Consider two alignments A_3 and A_5, resulting from two different matchers which match ontologies using different features for ruling out correspondences (A_3 is on the left, A_5 on the right):

Konstruktion$\{\perp\}$Municipality	Stadt$\{<\}$Town
Stadtgebiet$\{>, \between\}$Municipality	Stadtgebiet$\{\perp, \between\}$Municipality

Since these matchers provide competing alignments between ontology o and o', their result can be aggregated conjunctively. The result $A_6 = A_3 \wedge A_5$, is given below as the left-hand side alignment:

Konstruktion$\{\perp\}$Municipality	
Stadt$\{<\}$Town	Stadt$\{=, <, >, \between\}$Town
Stadtgebiet$\{\between\}$Municipality	Stadtgebiet$\{\perp\}$Municipality

This alignment is aggregated with the right-hand side alignment A_4. Since they provide evidence for alignments from different perspective, they are aggregated disjunctively, yielding $A_8 = A_4 \vee A_6$:

Stadt$\{=, <, >, \between\}$Town
Stadtgebiet$\{\between, \perp\}$Municipality

Algebras of relations are useful because they can account for these two behaviours. However, there are other benefits brought by algebra of relations.

5 Composing alignments

Another way of reusing alignments is to deduce new alignments from existing ones. One way to do so, is to compose alignments. If there exists an alignment between ontology o and ontology o'', and another alignment between o'' and a third ontology o', we would like to find which correspondences hold between o and o'. The operation that returns this set of correspondences is called composition.

Alignment composition has already considered [21] with the idea that, in an open system like the Alignment API, the rules for composing alignment relations, e.g., $\mathsf{instanceOf} \cdot \mathsf{subClassOf} = \mathsf{instanceOf}$, should be given by a composition table. Composition tables come directly from algebra of relations and they naturaly extend from base relations to disjunctions of base relations.

Alignment composition can thus be reduced to combining correspondences with regard to their relations and the structure of related entities and computing the confidence degree of the result. The composition table between the base relations of $A5$ is given in Table 1.

	$=$	$>$	$<$	$\between$	$\perp$
$=$	$=$	$>$	$<$	$\between$	$\perp$
$>$	$>$	$>$	$>\,<\,=\between$	$>\between$	$>\between\perp$
$<$	$<$	Γ	$<$	$<\between\perp$	$\perp$
$\between$	$\between$	$>\between\perp$	$<\between$	Γ	$>\between\perp$
$\perp$	$\perp$	$\perp$	$<\between\perp$	$<\between\perp$	Γ

Table 1. Composition table for the $A5$ relation algebra.

The composition of two alignments A and A' is defined by:

$$A \cdot A' = \overline{\{\langle e, e'', r \cdot r' \rangle | \langle e, e', r \rangle \in A, \langle e', e'', r' \rangle \in A'\}}$$

One can compose an alignment with itself (self-composition) through: $A^2 = A \cdot A^{-1} \cdot A$. This operation may provide new correspondences.

Example 5 (Composing alignments). The alignment A_3 of Example 4, is the result of the composition of alignments A_1 and A_2 of Example 2: $A_3 = A_1 \cdot A_2$. The first simple application of Table 1 occurs when composing Konstruktion $\{\perp\}$ Commune and Commune $\{>, =\}$ Municipality, then it can be deduced that Konstruktion $\{\perp\}$ Municipality because $\{\perp\} \cdot \{>, =\} = (\perp \cdot >) \cup (\perp \cdot =) = \{\perp\}$. Things can be more complex, when composing Stadtgebiet $\{>\}$ Ville and Ville $\{\between\}$ Municipality, then Table 1 allows to deduce that Stadtgebiet $\{>, \between\}$ Municipality because $\{>\} \cdot \{\between\} = > \cdot \between = \{<, \between\}$. The result provided by the table in this case is a disjunction of relations because it is not possible to obtain more precise information from the alignments alone.

Very often the composition of two base relations is not a base relation but a disjunction of relations. Hence, if we were not dealing with sets of base relations, it would not be possible to represent the composition of two alignments by an alignment.

Moreover, defining composition by an algebra of relations automatically satisfies all the constraints on the categorical characterisation of alignments defined in [21]: it must be associative and have an identity element. This is true from the definition of algebra of relations.

6 Algebraic reasoning with alignments

α-consequences are correspondences which are entailed by two aligned ontologies [7]; they can be extended as the correspondences entailed by a system of many ontologies and many alignments between them. [20] introduced the notion of quasi-consequences as the set of formulas entailed by a set of alignments alone (without considering ontologies). This notion can be straightforwardly extended to correspondences as quasi-α-consequences: the correspondences which are entailed by the set of alignments, when considering the ontologies as void of axioms. Quasi-α-consequences are also α-consequences.

Reasoning on alignments aims at using existing alignments in order to deduce more and more complete alignments. Such a reasoning procedure can be considered, for soundness and completeness, with respect to α-consequences.

Algebraic reasoning (using combination of composition, converse, and intersection) can be used as a practical and efficient way to reason with alignments. The algebraic closure of a set of alignments S is the set of normalised alignments, containing $\bar{S}$, closed under composition, converse and intersection.

This procedure is correct (the algebraic operations can be transformed into their logical equivalent). However, since it does not consider ontologies, it can only deduce quasi-α-consequences. We have no guarantee that it is complete even for finding quasi-α-consequences.

However, this can already be used for two purposes: (1) improving the existing alignments by deducing new correspondences coming either from the alignment itself or from other alignments, and (2) checking the consistency of a set of alignments (or one alignment). Indeed, if we can deduce $x\{\}y$, e.g., because $x\{<\}y$ and $x\{>\}y$ for two competing matchers, since the intersection is empty we know that the alignment itself is inconsistent. This kind of reasoning can be more complex, involving several alignments as well as composition operations, i.e., checking a whole distributed system. Then, the set of alignments as a whole would be inconsistent, hence the distributed system would have no model.

Example 6 (Algebraic reasoning). The simplest instance of an inconsistent alignment is to have two contradictory statements like Konstruktion$\{\bot\}$Town and Konstruktion$\{<\}$Town in the same alignment. The conjunction of these two relations, obtained by normalisation, is empty. Such an inconsistent alignment can also be obtained by combining consistent alignments. For instance, aggregating conjunctively alignments A_3 and A_4 of Example 4, will generate the inconsistent Stadtgebiet $\{\}$ Municipality correspondence.

Algebraic reasoning allows to expand alignments. For instance, Stadt$\{>\}$Town, Stadtgebiet$\{<\}$Town, Stadtgebiet$\{\bot\}$Municipality entails Stadt$\{>, \between, \bot\}$Municipality. Computing the compositional closure of this alignment will find this correspondence.

Moreover, if the initial alignment also contain $\mathsf{Stadt}\{=,<\}\mathsf{Municipality}$, the compositional closure will bring the inconsistence to light.

Once again, it is possible to use disjunctive relations to better evaluate alignments. Indeed, the problem is that if a matcher returns a correspondence between two entities with relation $\leq$ while the expected (and exact) relation was $<$, then the use of syntactic precision and recall measures would count this relation as incorrect. Hence, if the expected alignment was made of this correspondence alone, both precision and recall would be 0. This is unfair because it cannot be said that this correspondence is both incorrect and incomplete. In fact, it is incomplete, because it does not provide the exact relation, but not incorrect, because the relation is more general than the correct one.

This is indeed what happens with semantic precision and recall [7]: since the relation $\{<,=\}$, which is the disjunction of $<$ and $=$, can be deduced from $\{<\}$ alone, it would count as correct for semantic precision and still as incorrect for semantic recall because $\{<,=\}$ does not entail $\{<\}$.

We could introduce an algebraic precision and recall for evaluating ontology alignments as an intermediary step between classic precision and recall and semantic precision and recall. It would simply use the inclusion between the relations as suggested above instead of the entailment between correspondences of [7] and would be far easier to compute. The resulting measure would be a relaxation of precision and recall in the sense of [4].

7 Algebra granularity

In order to investigate granularity within algebras of relations, we introduced the notion of weakening [5]. Weakening an algebra of relations simply consists of grouping together several base relations and taking the result as the base relations of the, less precise, weaker algebra.

In fact, taking any maximal antichain[2] that preserves converse in the lattice of Figure 1 yields a base for an algebra of relations. Other constraints can be put on weakening, such as requiring that they preserve a neighbourhood structure. Neighbourhood structures for algebras of relations have been introduced in [10]. They are based on a connectivity relation between relations that is used for defining neighbourhood. This connectivity relation can be based on different properties of the domain the relations apply to. We have shown that granularity operators, at least in time and space algebras, can be automatically built on such neighbourhoods [5].

In terms of alignment, the interesting aspect of this weakening operation is that it helps considering that alignments using different sets of relations are compatible and can still be used together. Coming back to the example of set-relations, there can be systems that provides only the $=$ base relation leaving implicitly all the others as Γ, there are other systems like the one considered previously which consider $=$, $<$, $>$, $\between$ and $\perp$. The set of base relations is different and thus it is not easy to combine two such alignments. However, if we consider that the first one is a weakening of the second one (grouping $<$, $>$, $\between$ and $\perp$ into $\neq$), then it is possible to import one alignment into

[2] An antichain is a set of relations such that no one is comparable to the other.

another formalism and vice-versa (at the expense of completeness when we export to the weaker algebra).

Example 7 (Algebra granularity). Different sources of alignments may provide alignments with different kinds of relations between objects. For instance, the following A_7 alignment (left-hand side) is expressed in the simple $\{\bot, \not\perp\}$ algebra, i.e., identifying only incompatible elements.

Stadt$\{\not\perp\}$Town	Stadt$\{\not\perp\}$Town
Stadtgebiet$\{\bot\}$Municipality	Stadtgebiet$\{\bot, \not\perp\}$Municipality

Thanks to the use of compatible algebras, A_7 can be expressed in the more expressive algebra. In fact, the alignment A_4 of Example 4 is the transcription of A_7 in the $A5$ algebra. On the other hand, it is possible to degrade an alignment into the coarser algebra at the expense of precision. The alignment on the right-hand side is the result of converting the alignment A_5 of Example 4 to the $\{\bot, \not\perp\}$ algebra.

Figure 2 shows the "interesting" weakened algebras of relations from the initial algebra. It features the $\{=, \neq\}$ algebra but also shows that the usually considered $\{=, \leq, \geq, \bot\}$ is not a correct base for such an algebra because it is neither jointly exhaustive ($\leq$ and $=$ can occur at the same time), nor pairwise disjoint ($\between$ is missing).

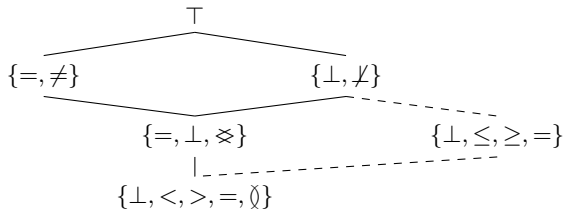

Fig. 2. The reasonable weakenings of $A5$ ($\not\perp = \{<, >, =, \between\}$ and $\not\bowtie = \{<, >, \between\}$).

8 Compatibility with confidence measures

Most matchers assign confidences to the correspondences they produce. They express to what extent they trust the correspondence. This confidence is expressed in a confidence structure:

Definition 3 (Confidence structure). *A confidence structure is an ordered set of degrees $\langle \Xi, \leq \rangle$ for which there exists a greatest element $\top$ and a smallest element $\bot$.*

The usage of confidence degrees is that the higher the degree with regard to $\leq$, the most likely the relation holds. This means that a particular confidence degree entails all the inferior confidence degrees.

With algebras of relations, the confidence assigned to a relation applies to the disjunction as a whole. Hence, when a confidence is assigned of some relation, this confidence can be assigned to all its supersets, but not to its subsets. The larger the relation, the stronger the confidence can be: typically, Γ should be given full confidence ($\top$) and $\varnothing$ the lowest one ($\bot$).

A correspondence $\langle e, e', r, n\rangle$ entails another correspondence $\langle e, e', r', n'\rangle$ if and only if $r \subseteq r'$ and $n \geq n'$. The normalisation operation will then only retain maximal elements for the induced order:

$$\begin{aligned} A^0 &= \{\langle e, e', \Gamma, \top\rangle | e \in Q_L(o), e' \in Q_{L'}(o')\} \\ \bar{A} &= max^{\subseteq,\geq} A \cup A^0 \\ A^{-1} &= \{\langle e', e, r^{-1}, n\rangle | \langle e, e', r, n\rangle \in \bar{A}\} \end{aligned}$$

The previous definition of $\bar{A}$ satisfies this definition, either with $\subseteq$ as an order or with $\top$ as the only confidence grade.

The most widely used structure is based on the real number unit interval $[0\ 1]$, but some systems simply use the boolean lattice. It is convenient to interpret the greatest element as the boolean true and the smallest element as the boolean false. Some other possible structures are fuzzy degrees, probabilities or other lattices. [11] has investigated the structure of fuzzy confidence relations. Below, we simply consider the unit interval with the usual arithmetic operations.

Example 8 (Normalisation with confidences). For instance,

$$\mathsf{Stadt}\{<\}_{.8}\mathsf{Town}, \mathsf{Stadt}\{<,=\}_{.9}\mathsf{Town}$$

is a normalised alignment (if it is added $\mathsf{Stadt}\Gamma_{\top}\mathsf{Town}$), which is interpreted (still conjunctively) as: I am confident with .8 that Stadt is strictly subsumed by Town, and confident with .9 that Stadt subsumed or equivalent to Town. If the confidences were stripped down, this conjunctive statement would be reduced to $\mathsf{Stadt}\{<\}\mathsf{Town}$. This corresponds to having the confidences set to 1. Similarly, the alignment:

$$\mathsf{Stadt}\{<\}_{.9}\mathsf{Town}, \mathsf{Stadt}\{<,=\}_{.8}\mathsf{Town}$$

is not minimal and can be reduced to $\mathsf{Stadt}\{<\}_{.9}\mathsf{Town}$. This is because $\{<\} \subseteq \{<,=\}$ and $.9 \geq .8$, hence $\mathsf{Stadt}\{<\}_{.9}\mathsf{Town}$ entails $\mathsf{Stadt}\{<,=\}_{.8}\mathsf{Town}$.

Designing operators for merging two alignments with disjunctive relations and confidence, is more open than considering the disjunctive relations alone. The implementation of the aggregation operators can differ as well as the measure for aggregating confidence. Instead of applying a union or intersection, each pair of entities will be assigned for each disjunction of relation (returned by a matcher) a confidence measure which will result in the aggregation of the confidence of all other matchers. There are many different ways to aggregate matcher results depending on confidence [8]:

– Triangular norms (min, weighted products) are useful for selecting only the best results in case of competing alignments;

– Multidimentional distances (Euclidean distance, weighted sum) are useful for taking into account all dimensions in case of complementary alignments;
– Fuzzy aggregation (min, weighted average) is useful for aggregating competing algorithms and averaging their results;
– Other specific measures, e.g., ordered weighted average, may also be used.

The natural aggregation measure is the conjunctive one induced by the normalisation procedure:

$$A \wedge A' = \overline{\bar{A} \cup \bar{A}'}$$

However, other aggregation operations can be designed from a confidence aggregation function f and a relation combination $\times$ as:

$$A \times^f A' = \overline{\{\langle e, e', r \times r', f(n, n')\rangle | \langle e, e', r, n\rangle \in \bar{A}, \langle e, e', r', n'\rangle \in \bar{A}'\}}$$

The $\times$ operation can be, for instance, $\cap$, $\cup$ or id (which applies only if both sets of relations are equal). We have presented it with two alignments, but this extends straightforwardly to n alignments.

Example 9 (Aggregation with confidence). Consider two alignments A_9 and A_{10} containing the following correspondences (A_9 is on the left, A_{10} on the right):

$$\begin{array}{cc} \mathsf{Stadt}\{=, <\}_{.5}\mathsf{Town} & \mathsf{Stadt}\{=\}_{1.0}\mathsf{Town} \\ \mathsf{Stadt}\{=, \bot\}_{.2}\mathsf{Town} & \\ \mathsf{Stadtgebiet}\{\bot\}_{.1}\mathsf{Municipality} & \mathsf{Stadtgebiet}\{\bot, \between\}_{.2}\mathsf{Municipality} \end{array}$$

Applying $\cap^{\min}(A_9, A_{10})$ yields, once normalised, the following correspondences:

$$\begin{array}{c} \mathsf{Stadt}\{=\}_{1.}\mathsf{Town} \\ \mathsf{Stadtgebiet}\{\bot, \between\}_{.2}\mathsf{Municipality} \\ \mathsf{Stadtgebiet}\{\bot\}_{.1}\mathsf{Municipality} \end{array}$$

The same operation with $\cup$ and weighted sum (weight being $1/3$ and $2/3$), would yield:

$$\begin{array}{c} \mathsf{Stadt}\{<, =\}_{.83}\mathsf{Town} \\ \mathsf{Stadt}\{\bot, =\}_{.4}\mathsf{Town} \\ \mathsf{Stadtgebiet}\{\bot, \between\}_{.16}\mathsf{Municipality} \end{array}$$

Which cannot be reduced.

Ideally, we would like that these aggregation functions preserve the opportunity given in Section 4 to have different operations usable in different situations.

This works well for $\min$ and weighted product functions together with intersection.

Property 1 (Reduction to intersection). The (normalised) weighted products and min functions, if applied to correspondences with confidences in $\{0, 1\}$ with $\cap$ computes $\cap$.

This works well for intersection because the methods used have 0 as an absorbing element. Since the other operations do not have a (upper) absorbing element, the confidence that they return are not necessary equal to 1 (or $\top$).

Property 2 (Reduction to union). The weighted sum, Euclidean distance and other instance of the Minlowski distance, if applied to correspondences with confidences in $\{0, 1\}$ and $\cup$ computes $\cup$. However, the confidence value assigned to the resulting correspondences may not be $\top$.

9 Related work

There are very few papers about alignment relations in the literature. [2] uses a spatial relation algebra for the purpose of expressing correspondences between spatio-temporal ontologies. However, the relations are only used in the ontology language and the alignment relations are still the classical $=$ and $\leq$ relations. [18] went one step further by explicitly considering a set of base relations similar to $A5$, but they do not consider using disjunction of relations as alignment relations (hence most of what is in this paper does not apply).

In database schema matching, the notion of mapping composition is prominent and has been thoroughly investigated [13]. The problem there is to design a composition operator that guarantee that the successive application of two mappings yields the same results as the application of their composition [9]. The approach is relatively different since relations in this context are always subsumption and their applications involve manipulating the ontology language instead of the alignment language (here the alignment relations). It is even shown that in general the result of composition may require a stronger language than the alignment language (hence the actual result of the composition is not an alignment). The approach taken with composition in algebra of relations is weaker – results are always in the alignment language – but is not complete.

[17] suggests to transform the XML schema matching problem into a constraint optimisation problem. Constraint optimisation problems are constraint satisfaction problems whose solutions maximise a quantity. Though the authors do not consider algebras of relations, this work suggests to use them, since Allen's constraint propagation algorithm [1] is an instance of a constraint propagation algorithm (which can be generalised as arc-consistency).

10 Conclusions

Starting from the need to express disjunctions of relations between ontology entities, we have introduced algebras of relations as a tool for expressing alignment relations. We have shown that this tool can easily express the most common relations used in ontology matching. However, algebras of relations are very flexible tools and new ones can be created for specific applications, e.g., mixing set and mereological relations.

What makes algebras of relations particularly attractive, besides expressing disjunctions is their ability to support genuinely other needed operations, and, in particular:

– conjunction and disjunction operators can be used as a more flexible means of combining alignments;
– composition had already been identified as the perfect tool for composing alignments: disjunction in alignments enables the expression of composition results as alignments;
– it can also be used for reasoning directly at the algebraic level and detecting valid consequences and constraints on alignments, before even considering the deeper (onto)logical level;
– weakening algebras can be used for combining alignments expressed in different (but compatible) algebras of relations.

This paper is a theoretical paper. We have no experimental setting to demonstrate the superiority of the approach over an eventual previous one. However, we claim that it conveniently demonstrate the benefits brought by the use of algebra of relations within ontology alignments. Algebra of relations is a well-studied domain and the fact that it can apply straight away to ontology alignment is very precious in a context when we want to freely share alignments and combine matching methods on the web.

No implementation is available yet: the full support for algebra of relations remains to be implemented in our Alignment API [6]. Implemented operations would effectively compute composition so as to be able to provide new alignments to clients.

Acknowledgements

This work started from a discussion with José Ángel Ramos and Asunción Gómez Pérez at Universidad Politécnica de Madrid. José Ángel provided further comments on the previous version of this paper. It also received comments from Jérôme David and Pavel Shvaiko.

References

1. James Allen. Maintaining knowledge about temporal intervals. *Communication of the ACM*, 26(11):832–843, 1983.
2. Nacéra Bennacer. Formalizing mappings for OWL spatiotemporal ontologies. In *Proc. 17th International conference on Database and Expert Systems Applications (DEXA), Kraków (PL)*, pages 368–378, 2006.
3. Paolo Bouquet, Marc Ehrig, Jérôme Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krötzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris. Specification of a common framework for characterizing alignment. Deliverable D2.2.1, Knowledge web NoE, 2004.
4. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Proc. K-CAP Workshop on Integrating Ontologies*, pages 25–32, Banff (CA), 2005.
5. Jérôme Euzenat. Granularity in relational formalisms with application to time and space representation. *Computational intelligence*, 17(4):703–737, 2001.
6. Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd International Semantic Web Conference (ISWC)*, volume 3298 of *Lecture notes in computer science*, pages 698–712, Hiroshima (JP), 2004.

7. Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, Hyderabad (IN), 2007.
8. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2007.
9. Ronald Fagin, Phokion Kolaitis, Lucian Popa, and Wang-Chiew Tan. Composing schema mappings: Second-order dependencies to the rescue. *ACM Transactions on Database Systems*, 30(4):994–1005, 2005.
10. Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial intelligence*, 54(1):199–227, 1992.
11. Avigdor Gal, Ateret Anaby-Tavor, Alberto Trombetta, and Danilo Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *The VLDB Journal*, 14(1):50–67, 2005.
12. Gérard Ligozat and Jochen Renz. What is a qualitative calculus? a general framework. In *Proc. 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI), Auckland (NZ)*, number 3157 in Lecture notes in computer science, pages 53–64, 2004.
13. Jayant Madhavan and Alon Halevy. Composing mappings among data sources. In *Proc. 29th International Conference on Very Large Data Bases, Berlin (DE)*, pages 572–583, 2003.
14. Eric Prud'hommeaux and Andrew Seaborne (ed.). SPARQL query language for RDF. Working draft, W3C, 2007.
15. Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
16. Amit Sheth and James Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.
17. Marko Smiljanić, Maurice van Keulen, and Willem Jonker. Formalizing the XML schema matching problem as a constraint optimization problem. In *Proc. 16th International conference on Database and Expert Systems Applications (DEXA), Copenhagen (DK)*, pages 333–342, 2005.
18. Anastasiya Sotnykova, Christelle Vangenot, Nadine Cullot, Nacéra Bennacer, and Marie-Aude Aufaure. Semantic mappings in description logics for spatio-temporal database schema integration. *Journal on Data Semantics*, III:143–167, 2005.
19. Alfred Tarski. On the calculus of relations. *Journal of symbolic logic*, 6(3):73–89, 1941.
20. Antoine Zimmermann. *Sémantique des connaissances distribuées*. PhD thesis, Université Joseph-Fourier, Grenoble (FR), 2008.
21. Antoine Zimmermann, Markus Krötzsch, Jérôme Euzenat, and Pascal Hitzler. Formalizing ontology alignment and its operations with category theory. In *Proc. 4th International Conference on Formal Ontology in Information Systems (FOIS)*, pages 277–288, Baltimore (MD US), 2006.