A different perspective on canonicity

Conference or Workshop Item

A Different Perspective on Canonicity

Michel Wermelinger

Departamento de Informática, Universidade Nova de Lisboa
2825 Monte da Caparica, Portugal
E-mail: mw@di.fct.unl.pt

Abstract. One of the most interesting aspects of Conceptual Structures Theory is the notion of canonicity. It is also one of the most neglected: Sowa seems to have abandoned it in the new version of the theory, and most of what has been written on canonicity focuses on the generalization hierarchy of conceptual graphs induced by the canonical formation rules. Although there is a common intuition that a graph is canonical if it is "meaningful", the original theory is somewhat unclear about what that actually means, in particular how canonicity is related to logic.
This paper argues that canonicity should be kept a first-class notion of Conceptual Structures Theory, provides a detailed analysis of work done so far, and proposes new definitions of the conformity relation and the canonical formation rules that allow a clear separation between canonicity and truth.
Topics: Conceptual Graph Theory, Knowledge Representation, Ontologies

1 Introduction

The development of Conceptual Structures Theory (CST) has been driven to great extent by natural language and its meaningfulness levels [8, p. 94]: gibberish; ungrammatical sequence; violation of selectional constraints; logically inconsistent; possibly false; empirically true. Syntax distinguishes the first two levels from the other ones, canonicity handles level 3, and logic the rest. Thus canonicity provides the ontological level that draws the borderline between meaningless and meaningful expressions (which are graphs in CST).

One could argue that syntax and logic are enough because any conceptual graph that obeys the arity of relations can be translated into a syntactically well-formed first-order formula and as such can be given a truth value. We feel however that logic alone does not distinguish the different "degrees" of falsehood: "Pigs fly" and "Portugal is a monarchy" are both false statements but not in the same way. Any knowledge representation theory should provide a way to capture our intuitions about such statements. In this paper, that role will be played by the ontology. Besides its conceptual importance, it has practical advantages: it can be shared by many knowledge bases; the knowledge representation system becomes more flexible and robust regarding arbitrary user input; processing an expression becomes more efficient.

Ideally, the ontology should be accompanied by two mechanisms: one to derive all the meaningful expressions, and the other to check whether an expression is meaningful. In CST the ontology is called *canon*, the derivation mechanism is given by the *canonical formation rules*, the checking mechanism is *projection*, and the meaningful expressions are the *canonical graphs*. A canon is specified by its types, markers, *conformity relation* and *canonical basis*. The conformity relation indicates for each marker all the types to which it is compatible, while the canonical basis is the initial set of graphs to which the canonical formation rules are applied.

In spite of its conceptual and practical importance, canonicity has been seldom a central theme of investigation. Most of the time only two notions "derived" from the canonical formation rules have received attention: projection and the generalization hierarchy. Only seldom have the knowledge representation aspects of canonicity been investigated: what are meaningful graphs? how is canonicity related to logic? This paper analyzes the work known to us, which can be summarized as follows. The original theory regards conformant concepts (i.e. those that obey the conformity relation) as true ones, while the canonical formation rules preserve falsehood: hence the relationship to logic is unclear since there are both true canonical graphs and true non-canonical graphs (and the same for false graphs). Kocura [5] considers that truth implies canonicity and therefore all non-canonical graphs must be false. Wermelinger [13] and Sexton [7] consider a graph to be canonical if it obeys the relation signatures given in the canonical basis. Finally, Sowa's new version of the theory [9] has no notion of canonicity: the definitions of conformity relation, canon, and canonical graphs have disappeared, and the canonical formation rules are just an auxiliary definition used by the inference rules.

This paper presents a different perspective on canonicity which fits better into the three meaningfulness levels (syntax, ontology, logic). A level characterizes a set of graphs, each level being a superset of the next one: canonical graphs are conceptual graphs (i.e., syntactically well-formed graphs), and true and false graphs are canonical. Seeing it the other way round, it does not make sense to speak about the canonicity of a non-conceptual graph or about the truth value of a non-canonical graph. To adhere to this principle, the conformity relation and the canonical formation rules (and thus the definition of projection) will be changed.

The structure of the paper is as follows. The next section is dedicated to the conformity relation. In the original theory, if marker m conforms to type t then "m is a t", i.e. m belongs to t's denotation. As will be seen, this interpretation of conformity has several problems. To allow false graphs to be canonical (and therefore conformant), the conformity relation will be relaxed.

The third section deals with the canonical basis. The main issue is to define what kinds of graphs the knowledge engineer may put into the canonical basis in order to specify useful *selectional constraints*. There have been several proposals, ranging from simple relation signatures to an arbitrary collection of graphs. We analyze those proposals and conclude that Sowa's original definition is still the most satisfactory one.

Section 4 handles the canonical formation rules. The original ones [8] only specialize the graphs they are applied to, but in [9] they may also generalize them. We follow the same approach because due to our principle true graphs (handled in [11]) should be canonical. Therefore the inference rules should be a special case of the formation rules. However, the formal definition of the latter will be changed to allow them to be applied directly instead of through the inference rules (as in [9]).

The last section characterizes canonical graphs in the usual two ways: using the canonical formation rules or projection. The definition of the latter will be extended to cope with the new definition of the former. An algorithm that decides whether a graph is canonical or not will be given. It is almost identical to the one for the original theory [6] and has the same complexity.

We use the following notational conventions: t and t' are concept types, i is an individual marker, m is an individual marker or the generic marker `*`, $\leq$ is the subtype relation, and $t \wedge t'$ denotes the maximal common subtype of both types. All references to pages (p. X) and to the original theory (Assumption/Theorem/Definition $x.y.z$) are to be understood in the context of [8]. The definitions given in the paper will be rather informal, since the formal details depend on the exact formalization of the basic notions (marker, type, concept, relation, etc.). Due to lack of space we will only deal with CST in its simplest form, assuming however that relation types may form a hierarchy [13]. The complete formalization of canonicity [12] also deals with higher-order types, a marker hierarchy including the absurd marker, contexts and coreference links.

2 The conformity relation

Sowa introduced the conformity relation :: as a test to be done when changing a concept's type: "if `#98077` is a cat then `CAT` :: `#98077` is true; otherwise, it is false"; in the second case "$\boxed{\text{ANIMAL: \#98077}}$ could not be restricted to $\boxed{\text{CAT: \#98077}}$" (p. 87). The sentence quoted first makes it very clear that an individual conforms to a type if and only if it belongs to the set denoted by the type. This is further stressed by the formal definition (Assumption 3.3.3) which imposes the following conditions:

1. for any concept $\boxed{t\colon m}$, $t :: m$;
2. if $t \leq t'$ and $t :: i$, then $t' :: i$;
3. if $t :: i$ and $t' :: i$, then $t \wedge t' :: i$;
4. for any i, $\top :: i$, but not $\bot :: i$;
5. for any t, $t :: *$.

In fact, if we define the conformity relation as the expression of the denotation (formally, $t :: m \Leftrightarrow m = * \vee m \in \delta t$) then we easily get condition 5 from the definition, and conditions 2 and 4 from the properties of δ: $\delta\top$ is the universal set, $\delta\bot$ is the empty set, and $t \leq t'$ implies $\delta t \subseteq \delta t'$.

As noted in [2], an individual marker cannot conform to two incompatible types (i.e., $t \wedge t' = \bot$). Otherwise conditions 3 and 4 would contradict each other. But the problem roots deeper. In fact, condition 3 implies that $\delta(t \wedge t') =$

$\delta(t) \cap \delta(t')$ which is called the lattice-theoretic interpretation of the type hierarchy in [1]. This contradicts the order-theoretic interpretation of Theorem 3.2.6: $\delta(t \wedge t') \subseteq \delta t \cap \delta t'$. Also, the lattice-theoretic approach has conceptual and practical drawbacks: a maximal common subtype must be interpreted as the "implication" of its supertypes, and the intersection of each pair of compatible types must be represented by an explicit type, leading to an explosion of conceptually irrelevant types.

Example 1. Consider a knowledge base about people, containing types for jobs and family relationships. According to the lattice-theoretic interpretation, `UNCLE` $=$ `SON` $\wedge$ `BROTHER` means that every person which is a son and a brother is also an uncle, clearly an undesired meaning. If `TEACHER` :: `#Michael` and `FATHER` :: `#Michael` then the type `FATHER-TEACHER` must exist for condition 3 to be satisfied, even if there is no other teacher with children in the knowledge base.

Even if condition 3 is abandoned other problems remain. Condition 1 forces every concept to be conformant. We will not impose this constraint as it is too strong in our opinion. Furthermore, notice that $\perp :: *$ but not $\perp :: i$. Logically, both are false statements since they correspond to $\exists x \in \delta\perp$ and $i \in \delta\perp$, contradicting $\delta\perp = \emptyset$. There is thus no valid reason to allow one but not the other.

Furthermore, if the conformity relation is just the indication of the types each individual marker is an instance of, then it is theoretically useless, because the same effect can be obtained by axioms in the knowledge base: assert in the outer context $\boxed{\perp : *}$ and $\boxed{t : i}$ whenever $t :: i$. Applying the generalization inference rule one gets $\boxed{t' : i}$ for any $t' \geq t$ (including $t' = \top$) and $\boxed{t : *}$ for any t. Also, since the first-order rules of inference are consistent, the graph $\boxed{\perp : i}$ can never be obtained. Thus conditions 2, 4, and 5 are satisfied.

The real problem however is not of formal but of conceptual nature. If $t :: i$ means that "i is a t", then all false concepts will not be conformant and as such cannot be generated by the canonical formation rules. As Sexton [7] noted, this is not consistent with the statement "The formation rules enforce selectional constraints, but they make no guarantee of truth or falsity" (p. 94). As a concrete example, take the one on page 92: if `BEAGLE` :: `#Snoopy` then $\boxed{\text{DOG: \#Snoopy}}$ cannot be restricted to $\boxed{\text{COLLIE: \#Snoopy}}$ since Snoopy is a beagle, not a collie. Sexton remarks that the latter concept is a meaningful one and therefore should be allowed in a canonical graph, although the individual marker does not belong to the denotation of (i.e., conform to) the type.

Therefore, in this paper a new notion of conformity relation is proposed: a marker m will conform to a type t if $\boxed{t : i}$ should be part of a true or false graph. Thus, conformity (as part of the broader notion of canonicity) does not imply truth any longer. Conversely, it does not make sense to speak about the truth or falsehood of a graph with non-conformant concepts.

Before presenting the formal definition, some observations are in order. First, as $\boxed{\perp : m}$ is false for any m, it will be considered a conformant concept. Second, if $\boxed{t : i}$ is true, so is $\boxed{t' : i}$ for any supertype t'; and if it is false, it is for any subtype t', too. To put it simply,

if $t :: i$ then $t' :: i$ for any $t' \leq t$ or $t' \geq t$.

Formally, however, we cannot state it this way, because $t :: i$ would imply $\top :: i$ (and $\perp :: i$) and therefore $t' :: i$ for any type t'. In other words, any individual marker would conform to any type, thus making the conformity relation meaningless. Even if we impose the restriction $\perp < t < \top$ in the above rule, an individual marker would still conform to concept types that are "zig-zag"-related in the type hierarchy. To circumvent this, we split the conformity relation into two relations: the base relation is given by the knowledge engineer and states for each individual marker what are the most relevant types it should conform to; the actual conformity relation is basically just the closure of the base relation over subtypes and supertypes.

Assumption 1. Given a relation R between concept types and markers, the *conformity relation* :: is the smallest superset of R such that

- for any m, $\top :: m$ and $\perp :: m$;
- for any t, $t :: *$;
- for any t and m, if tRm and $t \leq t'$ or $t' \leq t$, then $t' :: m$.

Relaxing the definition of conformity is not only theoretically more elegant, it has also practical advantages: a conceptual graph processor can be made more robust and it can indicate the source of errors precisely. Consider for example a natural language processor that has to join the concepts $\boxed{\text{MAN: \#Lou}}$ and $\boxed{\text{WOMAN: \#Lou}}$. According to the original definition, the resulting concept $\boxed{\perp\text{: \#Lou}}$ does not obey the conformity relation and as such the join would fail (i.e. the text would not be parsed). An implementation could provide some *ad-hoc* way to indicate the source of error to the user, but it is always better to have a clean theoretical framework, as is the case with the new definition: the concept is meaningful, although false, and the absurd type clearly shows where the parsing has produced an inconsistency.

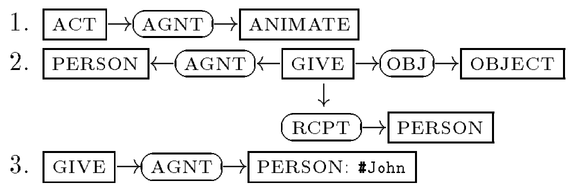
3 The Canon

The conformity relation is only a small part of the overall definition of an ontology to be used by one or more knowledge bases. In Conceptual Structures Theory the ontology is called *canon* and contains the types, the markers, the conformity relation, and an initial set of well-formed graphs, the *canonical basis* (Assumption 3.4.5). By applying the *canonical formation rules* to those graphs one obtains all *canonical graphs*, i.e. all graphs that "are meaningful" (p. 91).

However, the Conceptual Catalog [8, Appendix B] assigns a canonical graph to each concept or relation type in order to specify the selectional constraints to be observed by each type. Besides not being part of the formal definition of canonical basis, this association lead the Conceptual Structures community to use the term “canonical graph” in two different senses: (1) a graph that is derivable from the canonical basis, and (2) *the* graph in the canonical basis that is associated to a given type. Of course, these two senses are not incompatible, since (2) implies (1). To make the distinction clear, the elements of the canonical basis will be called *base graphs*.

The existence or not of associations between types and base graphs influences greatly the notion of canonical graph, because in the former case the base graph of type t must project on any graph using t. As Willems pointed out[1], this leads to another dual view of the canonical basis: whether it represents selectional constraints on the links between relations and concepts, or mandatory “arguments” of types.

Example 2 (adapted from Willems). Consider these graphs, the first two being base graphs:



If base graphs are not assigned to types, then graph 3 is canonical because it can be derived from the first one. But if graph 2 is associated to concept type `GIVE`, then graph 3 is no longer canonical as it is missing two arguments of the verb.

Contrary to what Willems seems to imply, the problem is not the existence of associations *per se*, but the *kind* of associations done. As seen, assigning base graphs to concept types rules out meaningful graphs, that we would like to consider canonical, on the ground of having only partial information. This is not acceptable for a knowledge representation formalism. Moreover, we feel that the “arguments” view of the canonical basis is more appropriate of a lexicon [10].

It is however possible to retain the “selectional constraint” view as long as associations are restricted to relation types and if base graphs consist only of a single relation. Thus each base graph states the “signature” of the associated relation type, i.e. its arity and the maximal concept types of its arguments. This kind of base graph is called *star graph* and was introduced in [2]. Graph 1 of Example 2 could be the star graph of `AGNT`. This approach has been adopted by [13] and extended to handle relation type hierarchies: if $t \leq t'$ then the star graph of relation type t' must project into the star graph of t. Although not apparent at first sight, Sexton [7, Section VIII] also advocates the use of star graphs: “In order for a graph to be canonical, the type of each arc of each conceptual relation must be predicable [i.e., a supertype] of the type of the concept the arc points to”. In other words, each relation must state the maximal type of each of its

[1] In a message sent to the CG mailing list on July 31, 1992.

arcs (i.e., arguments). To sum up, both [13, 7] consider a graph to be canonical if and only if all relations are used according to their signatures.

This is obviously a very weak notion of canonicity because the set of canonical graphs is too large. Star graphs are also too restrictive: by imposing the form and the number of base graphs (one for each relation type), the user can only specify very simple selectional constraints that take no contextual information into account. However, star graphs have computational advantages. As the recognition of a canonical graph is based on graph projection (see Section 5), if the elements of the canonical basis have a single relational vertex the complexity becomes polynomial.

The approach of Chein and Mugnier [2, 6] is better. They distinguish between the canonical basis and the *basis of the support*. The latter is the set of star graphs, and the graphs generated by the canonical formation rules from the star graphs are called *well-formed*. As in the original theory, the authors consider the canonical basis to be the generator set of the canonical graphs, but as expected they require the graphs of the canonical basis to be well-formed. This means that every canonical graph is well-formed. Therefore in this approach there is a new meaningfulness level between arbitrary conceptual graphs and canonical graphs. Notice that in the approaches mentioned above [13, 7] the canonical basis corresponds to the basis of the support and hence there is no distinction between well-formed and canonical graphs.

Chein and Mugnier's "mixed" approach is not as restrictive, but still there are conditions imposed on the elements of the canonical basis. This limits the knowledge engineer's flexibility to specify an ontology. Moreover, if the type of an argument of a relation depends on the type of another of its arguments, more than one star graph is necessary for that relation type. Also, it seems to us that the selectional constraints specified by star graphs are just a special case of the selectional constraints that base graphs are supposed to express. In fact, it is easy to provide a tool that checks whether the relations occurring in the canonical basis are used consistently.

To sum up, Sowa's original definition of a canonical basis as a set of conceptual graphs is still the most satisfactory one, as it provides all the flexibility required by a knowledge engineer, who is free to adhere to the specification discipline imposed by star graphs if he wishes. The formal definition of canon remains hence similar to Sowa's, but as expected it uses the new definition of conformity relation, which must be obeyed by every base graph. This is not explicitly stated in the original Assumption 3.4.5 since condition 1 of the original definition of conformity relation already required every concept to be conformant. As we have abandoned condition 1 in general, we must impose it for the base graphs.

Notice that the canonical basis may be redundant: it might be possible to derive exactly the same canonical graphs just from a proper subset of the canonical basis. A Conceptual Structures system can detect that case using Theorem 7 (Section 5) to verify for each base graph if it can be derived from the other ones.

4 The canonical formation rules

The rules proposed by Assumption 3.4.3 have several advantages: they are simple, they do not contain redundancies (i.e. they are independent from each other), and they are specialization rules. This means that their application (called a *canonical derivation*) establishes a relationship between the initial graphs and the resulting one that can be analyzed both from the logical (implication) as from the graph-theoretical viewpoint (projection).

The drawback of using just specialization rules is that not every true graph is a canonical one, clearly an undesirable state of affairs. Considering Example 2 again, if graph 3 is true and `ACT` $\leq$ `EVENT` then $\boxed{\text{EVENT}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{PERSON: \#John}}$ is also true but as it cannot be obtained by specialization from the other three graphs, it is not a canonical graph. This is contrary to the idea that canonical graphs are "meaningful graphs that represent real or possible situations in the external world" (p. 91). In other words, the true graphs must be a subset of the canonical graphs. Hence, given a set of true graphs, the graphs derived from them using the inference rules must be canonical and as such should be obtained by applying the canonical formation rules to the same set of graphs. Put differently, the inference rules must be a particular case of the canonical formation rules.

This is the approach followed in the new version of the theory [9]. Sowa has made the canonical formation rules more general, and the inference rules limit the applicability of the formation rules. He made two kinds of changes. First, some rules do not apply to a single vertex or to a complete graph any more but to a subgraph. Second, rules have been divided into three groups: those that generate a logically equivalent graph, those that specialize the graph to which they are applied, and those that generalize it.

As the next example shows, the rules are no longer independent from each other. It is possible to get the same result from the same graph(s) applying different rules (to different subgraphs in some cases). Although [9] does not state what a subgraph is, from the rules we interpret it in the graph-theoretical way as a subset of vertices and edges. A subgraph therefore does not have to be a conceptual graph. In particular it may be just a single relation node. This allows the new rules to include the original simplification rule.

Example 3. From the graph $\boxed{\text{ACT}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{ANIMATE}}$ one can derive

$$\begin{tikzcd} \boxed{\text{ACT}} \arrow[r] \arrow[dr] & (\text{AGNT}) \arrow[r] & \boxed{\text{ANIMATE}} \\ & (\text{AGNT}) \arrow[ur] & \end{tikzcd}$$

in two distinct ways. The first is to make a copy of the subgraph (AGNT), which shows that the two graphs are equivalent. The second one starts with a copy of the whole graph and then joins pairwise equal concepts.

The notion of canonicity does not exist in [9]. As such, the canonical formation rules are not used autonomously but by the inference rules. There are however good reasons to want to use the canonical formation rules directly:

- in many applications (like natural language understanding [10]) it is useful to process graphs whose truth value is unknown;
- it is desirable to have an "operational" characterization of canonical graphs;
- the canonical formation rules can be a starting point for the definition of other operations.

Since the new canonical formation rules may specialize part of a graph and generalize some other part, the notion of projection must be relaxed to allow the "declarative" characterization of the canonical graphs thus obtained.

Definition 2. Let g and g' be two conceptual graphs. A *semi-projection* $\pi : g \to g'$ is a function that maps g to a subgraph of g' such that

- for any vertex v of g, either $\pi(v) \leq v$ or $\pi(v) \geq v$[2];
- for any relation r of g, if its i-th arc a links r to concept c, then $\pi(a)$ is the i-th arc between $\pi(r)$ and $\pi(c)$.

If the function is a bijection then g' is called an *semi-instance* of g.

The new canonical formation rules can now be presented. They are similar to Sowa's. The changes made arose from the need to generate only ontologically meaningful graphs. Therefore some rules had to be restricted. Others had to be added to make sure that the first-order rules of inference are indeed a particular case of the canonical formation rules[3].

Assumption 3. Given a canon and zero or more conceptual graphs, the *canonical formation rules* generate new graphs. Some rules are defined in terms of subgraph duplication, removal and substitution. The definition of subgraph depends on the rule to be applied. In any case the operations also duplicate, remove, or substitute the arcs between subgraph vertices and external vertices. If the graphs to which the rules are applied obey the conformity relation then so must the resulting graph. In the following c is a context, either empty or containing the conceptual graphs g_1 and g_2 which might be the same one. Graphs g'_1 and g'_2 are subgraphs of g_1 and g_2, respectively.

- *Equivalence Rules*. In these rules, if a subgraph contains a concept, it also contains all relations linked to the concept.
  Copy Make a copy of g'_1.
  Simplify Remove g'_1 if g'_1 and g'_2 are identical and are linked to the same external vertices but have no vertices in common.
- *Specialization Rules*. In these rules, if a subgraph contains a relation, it also contains all concepts linked to the relation.
  Join Overlay g'_1 and g'_2 if they are identical.

[2] This is an extension of the partial order over types to concepts and relations.

[3] That does not happen in [9] and neither in this paper since we restrict ourselves to graphs without contexts or coreference links. The full version of the canonical formation rules [12] is however a generalization of the inference rules [11].

Restrict Substitute a vertex v of g_1 by a specialization if v has not been unrestricted before.

Insertion Insert a base graph in c.

– *Generalization Rules.* In these rules, if a subgraph contains a relation, it also contains all concepts linked to the relation.

Detach Substitute g_1 by g'_1 if g'_1 is a semi-instance of some base graph.

Unrestrict Substitute a vertex v of g_1 by a generalization if v has not been restricted previously.

Remove Remove g_1.

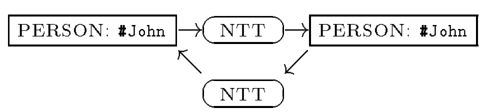
The detailed explanation of the rules is given in [12]. The next subsections will just highlight the most important issues. First some general remarks. As in Sowa's original rules, the conformity relation must be checked before restricting or relaxing a concept. The formulation is also much more concise and simpler than those in [9, 4]. It is also clearer as it gives precise definitions of subgraphs. As Sowa's new rules, these are not independent from each other but they are so within each of the three groups.

4.1 Copy and Simplify

To see the reason for the given definition of subgraph, consider a concept c linked to a relation r such that c is part of the subgraph but r is not. Then the copy or simplification (i.e., removal) of c adds or removes the arc to r. In other words, the arity of r increments or decrements by one, and the resulting graph is not canonical. Therefore r must also be part of the subgraph as required by Assumption 3. Now, if r is duplicated or removed, its links to some external concept c' will be duplicated or removed, too, but that only changes the number of arcs attached to c', not the arity of r or its copy r' (see Example 3).

As for the simplify rule, two subgraphs g'_1 and g'_2 are duplicates only if they are connected to the same external vertices. There are two cases. If there are no such vertices then $g'_1 = g_1$ and $g'_2 = g_2$, which means that we are considering two copies of a complete graph. Hence one of them can be eliminated. In the second case, if the external vertices are the same, the two subgraphs must be part of the same graph: $g_1 = g_2$. In both cases the two subgraphs may not overlap. As for the first case that would amount to $g_1 = g_2$ and the simplify rule would become the remove rule. The problem in the second case is similar.

Example 4. Consider the relation `NTT` (not taller than) between persons. The subgraph (NTT)→[PERSON: #John]→(NTT) occurs twice in



Eliminating one of the copies one gets just [PERSON: #John] which is not equivalent to the original.

Notice that the simplify rule stated in [9] does not impose any restriction on the duplicate subgraphs. As such, it is not an equivalence rule.

4.2 Restrict and Unrestrict

The restrict and unrestrict rules now allow to generalize *or* specialize *any* vertex (including relations), but they prevent the generalization *and* specialization of the same vertex. If that would be possible, any type t could be substituted by any other type t', even an incompatible one. This has been noted independently by [12, 5]. It means that the canonical formation rules could derive almost any non-canonical graph from the canonical basis, thus making the selectional constraints imposed by the base graphs useless.

Example 5. [ACT]→(AGNT)→[ANIMATE] can be restricted to [ACT]→(AGNT)→[$\perp$] and then unrestricted to [ACT]→(AGNT)→[IDEA]. The intermediate step could also be a generalization to [ACT]→(AGNT)→[$\top$] followed by a specialization to the final graph.

In Sowa's approach, the canonical formation rules are only used by the inference rules. Since a graph cannot be simultaneously in an even context (where it can be generalized) and in an odd one (where it could be specialized), the restrict and unrestrict rules are never mixed. In our approach there were two possibilities: generalizations and specializations are forbidden for the same graph or just for the same vertex. The second option is more flexible and has been adopted.

The interplay between specialization and generalization is subtle. Certain vertices cannot be changed at all, namely those that were obtained by joining a vertex that has been generalized with one that was specialized.

Example 6. Let `PERSON` < `ANIMAL` < `ANIMATE` and let `NTT` be the relation of Example 4. From the conceptual graphs

[ACT]→(AGNT)→[ANIMATE]   [PERSON]→(NTT)→[PERSON]

it is possible to obtain [ACT]→(AGNT)→[ANIMAL]→(NTT)→[PERSON] by restricting [ANIMATE] and unrestricting [PERSON] followed by a join on both. The [ANIMAL] concept can be no longer changed. Otherwise it would be possible to obtain e.g. [ACT]→(AGNT)→[DOG]→(NTT)→[PERSON] through specialization and [ACT]→(AGNT)→[PHYSOBJ]→(NTT)→[PERSON] through generalization, which violate `NTT`'s and `AGNT`'s selectional constraints, respectively.

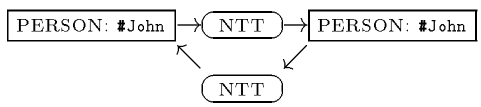
To correctly implement these rules it is necessary to keep the history of each vertex. That can be done using two boolean variables, one indicating if the vertex has been generalized, the other whether it was specialized. An operation can be performed only if the variable corresponding to the *other* operation is set to false. When two vertices are joined, the variables of the resulting vertex are the conjunction of the corresponding variables of the original vertices.

4.3 Join and Detach

Both of Sowa's join rules [8, 9] only handle two (identical) concepts at a time. The rule of Assumption 3 allows one to join identical subgraphs. Notice that the rule allows them to belong to different graphs. That case is called an *external join* in [6]. As the join of two relations also involves the join of their arguments, the definition of subgraph is exactly the opposite of the one used by the equivalence rules.

One should also point out that the simplification can be simulated by an *internal join*, i.e. when the two subgraphs belong to the same graph. Indeed, overlaying two subgraphs is equivalent to eliminating one of them while keeping its arcs to the rest of the graph. As in the simplify rule both subgraphs have the same external links, the overlapping effect is obtained. Let us see an example using the graph of Example 4.

Example 7. Subgraph [PERSON: #John] → (NTT) → [PERSON: #John] occurs twice in



Overlaying the two of them one gets [PERSON: #John] → (NTT) → [PERSON: #John]. The same result can be obtained by applying the simplify rule to the (NTT) subgraph.

Sowa's detach rule allows one to erase any subgraph. It is obvious that the remaining subgraph may not be canonical. Up to this part of the work, the only graphs that are guaranteed to be canonical are the base graphs. Therefore our rule must check that the remaining subgraph must be a base graph up to some generalizations or specializations. By repeated application of the copy and the detach rules it is possible to separate a graph into a cover of base components (compare with Theorem 7).

5 Canonical Graphs

Now that we have a generator set (the canonical basis) and the generation rules (the canonical formation rules) we can finally define the notion of canonical graph, which will be equivalent to the one of Assumption 3.4.5. Only the formulation differs. The original canonical formation rules (Assumption 3.4.3) use just the conformity relation and therefore the initial set of graphs to which the rules are applied must be explicitly stated. In our formulation that set (the canonical basis) is already part of the definition of the formation rules. Thus a canonical graph is a graph that can be generated from a canon and an "empty sheet".

Definition 4. A conceptual graph is called *canonical* regarding a given canon C if it is possible to derive it from the empty set of graphs through application of canonical formation rules using C.

In particular, applying the insertion rule one gets, as expected,

Proposition 5. *A base graph is canonical (regarding the canon it belongs to).*

Although the definition of canonical graph is the same as the original one, due to the differences in the definitions of the conformity relation and the canonical formation rules, given the same canonical basis both frameworks generate quite different sets of canonical graphs. The sets are incomparable (i.e. neither is a subset of the other) because Sowa's rules are not a subset of ours or vice-versa. However, the presented rules guarantee as wished that every true or false graph is canonical. Notice also that a graph can be considered canonical regarding a canon, and non-canonical regarding another one.

Besides forming new canonical graphs it is also convenient to be able to recognize them without explicitly constructing the derivation (the sequence of rules) that leads to their formation. The original canonical formation rules just specialize a graph. Hence the derivation process corresponds to a projection (Theorem 3.5.4). In addition, Mugnier and Chein [6] have shown that if there is a projection between two canonical graphs then there is a derivation. From this and other results they obtained the following characterization: a conceptual graph g is canonical if and only if there are projections of base graphs into g that cover the whole of g. As the new canonical formation rules also allow vertices to be generalized, projection is substituted by semi-projection (Definition 2) but the main idea remains.

Definition 6. A *cover* of a conceptual graph g is a finite set of conceptual graphs $\{g_1, \ldots, g_n\}$ such that each g_i is a subgraph of g and each vertex and arc of g occurs in at least one graph of the cover.

Theorem 7. *A conceptual graph is canonical regarding canon C if and only if it obeys the conformity relation and has a cover G such that each graph $g \in G$ is an semi-instance of a base graph of C.*

The importance of the theorem (proven in [12]) stems from the fact that it provides an algorithm to check whether a conceptual graph g (e.g. given by an user) is canonical or not. The method consists basically in finding base graphs whose semi-projections into g cover g completely. Since the semi-projection of a relation also includes the concepts it is linked to, the algorithm can be simply as follows. Choose a relation r of g and go through the canonical basis until finding a base graph whose semi-projection into g contains r. All relations covered by that semi-projection are marked and the process is repeated with an unmarked relation. This algorithm is identical to the one presented in [6] except that the projection is substituted by the semi-projection. Hence the algorithm still is polynomial in relationship to the complexity of the base operation (in this case semi-projection). Most of the time base graphs are trees. Therefore, in those cases semi-projection has polynomial complexity and recognizing a canonical graph takes polynomial time in the size of the graph and of the canonical basis.

6 Conclusions

This paper has argued that canonicity is fundamental to Conceptual Structures Theory since it corresponds to the intermediate ontological level between syntax and logic. Therefore it has both conceptual as practical advantages. We have analyzed some of the literature on canonicity and found out that the (implicit) meaning of canonicity is either too weak (as in the case of relation signatures) or its relationship to logic is vague or dubious. We have therefore explicitly adopted a very precise guideline: a graph should be canonical if we would like to make a definite statement about its truth or falsehood. Therefore canonical graphs are a superset of true and false graphs: it is meaningless to speak about the truth value of a non-canonical graph and, for a given knowledge base, the truth value of some canonical graphs may be unknown.

Based on this new perspective of canonicity, which provides a simple yet clear relationship between the ontological and logical levels, we have substantially changed the definitions of the conformity relation and the canonical formation rules and improved their formulation. To stress the relation between canonical and true graphs, the first-order rules of inference have become a special case of the canonical formation rules. Furthermore, the latter can now be used independently of the former without the risk of generating non-canonical graphs.

In order to keep a "declarative" definition of canonical graphs projection was generalized to semi-projection, but without computational impact since the recognition of canonical graphs still has the same complexity as for the original theory, in many cases being polynomial.

References

1. C. Beierle, U. Hedstück, U. Pletat, P. H. Schmitt, and J. Siekmann. An order-sorted logic for knowledge representation systems. Technical Report 113, IWBS, April 1990.
2. Michel Chein and Marie-Laure Mugnier. Conceptual graphs: fundamental notions. *Révue d'Intelligence Artificielle*, 6(4):365–406, 1992.
3. Gerard Ellis, Robert Levinson, William Rich, and John F. Sowa, editors. *Conceptual Structures: Applications, Implementation and Theory — Proceedings of the Third International Conference on Conceptual Structures*, volume 954 of *Lecture Notes in Artificial Intelligence*, Santa Cruz, USA, August 1995. Springer-Verlag.
4. John Esch and Robert Levinson. An implementation model for contexts and negation in conceptual graphs. In Ellis et al. [3], pages 247–262.
5. Pavel Kocura. Conceptual graph canonicity and semantic constraints. In Peter W. Eklund, Gerard Ellis, and Graham Mann, editors, *Conceptual Structures: Knowledge Representation as Interlingua — Auxiliary Proceedings of the Fourth International Conference on Conceptual Structures*, pages 133–145, Sydney, Australia, August 1996. University of New South Wales.
6. M. L. Mugnier and M. Chein. Characterization and algorithmic recognition of canonical conceptual graphs. In Guy W. Mineau, Bernard Moulin, and John F. Sowa, editors, *Conceptual Graphs for Knowledge Representation — Proceedings of the First International Conference on Conceptual Structures*, volume 699 of *Lecture*

Notes in Artificial Intelligence, pages 294–311, Québec City, Canada, August 1993. Springer-Verlag.
7. Clark A. Sexton. Types, type hierarchies, and canonical graphs. In Gerard Ellis, Robert A. Levinson, William Rich, and John F. Sowa, editors, *Conceptual Structures: Applications, Implementation and Theory — Supplementary Proceedings of the Third International Conference on Conceptual Structures*, pages 213–221, Santa Cruz, USA, August 1995.
8. John F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.
9. John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. August 1994. Draft of a book to be published by PWS Publishing Company, Boston.
10. John F. Sowa and Eileen C. Way. Implementing a semantic interpreter using conceptual graphs. *IBM Journal of Research and Development*, 30(1):57–69, January 1986.
11. Michel Wermelinger. Conceptual graphs and first-order logic. In Ellis et al. [3], pages 323–337.
12. Michel Wermelinger. Teoria básica das estruturas conceptuais. Master's thesis, Universidade Nova de Lisboa, Lisbon, Portugal, January 1995.
13. Michel Wermelinger and José Gabriel Lopes. Basic conceptual structures theory. In William M. Tepfenhart, Judith P. Dick, and John F. Sowa, editors, *Conceptual Structures: Current Practices — Proceedings of the Second International Conference on Conceptual Structures*, volume 835 of *Lecture Notes in Artificial Intelligence*, pages 144–159, College Park, USA, August 1994. Springer-Verlag.

This article was processed using the LaTeX macro package with LLNCS style