Published in the Proceedings of the Fifth International Conference on Conceptual Structures (ICCS'97)

Contexts: A Formal Definition of Worlds of Assertions

Guy W. Mineau
Department of Computer Science
Université Laval
Quebec City, Canada
G1K 7P4
tel.: (418) 656-5189
fax: (418) 656-2324
email: mineau@ift.ulaval.ca

Olivier Gerbé
Groupe DMR Inc.
1200 McGill College
Montreal, Canada
H3B 4G7
tel.: (514) 877-3301
fax: (514) 866-0423
email: olivier.gerbe@dmr.ca

Abstract. For many years now on-going discussions, not to say endless discussions, about the intrinsic definition of contexts have been at the center stage of every single meeting of the conceptual graph community. It is our opinion that this lack of consensus about contexts is in direct relation to its lack of a formal definition. As a matter of fact, no formal definition of contexts was given up to this moment, not even in John Sowa's original book. Being a vital issue in conceptual graph theory, this paper addresses the problem of providing formal semantics to the definition of contexts, when used for information packaging. It proposes a definitional framework for contexts, based on formal concept analysis [17], bridging these two research areas. It also presents how querying a knowledge base structured as a lattice of contexts can be done.

Keywords: contexts, conceptual graph theory, semantics, concept formation, knowledge representation

1 Introduction

Contexts are a vital notion of the conceptual graph theory. Inferences are based on the notion of contexts; contexts are useful for packaging information. Over the years, they were extensively used in many areas of conceptual graph related research, particularly in knowledge definition [2, 3, 4], in knowledge structuring [13], in natural language processing [1, 9, 12, 11], in the representation of modalities, intentional verbs and temporal relations [9], in multi-agent systems [10] and in object systems [16]. Despite this obvious need for contexts, there is still no consensus on their definition and usage in the community. This is mainly due to the fact that they are ill-defined. As a matter of fact, as surprising as this may be, there exists no formal definition of contexts so far, not even in John Sowa's original book. The only intuitive definition given in Sowa's book is that a context exists whenever some proposition is asserted: a proposition is said to be true in a particular context. No way of defining this context is provided, resulting in different uses for different types of applications. To our opinion, this lack of semantics when defining contexts is its source of problems.

For inferential purposes, contexts are useful to delimit the scope of negation, pretty much like parentheses in Peano-Russel's notation. They introduce a notion of subsumed (dominated) contexts that is not formally defined in the cg literature so far, but which is useful for the mechanics of the inference process. One realizes that this scope delimitation mechanism could be represented otherwise. However, the information packaging capabilities that these contexts provide proved to be useful for other application areas such as modal logic, linguistics and multi-agent applications. The implied notion of subsumption between contexts is useful, though it needs to be well defined so that different *worlds of assertion* may exist in relation to one another, as is required with these types of applications. When the truth value of a set of assertions is conditional to the same set of conditions, then we say that a *world of assertion* is created, and that it is defined by both the assertions which describe it and the inherent conditions which allow its existence.

This paper presents a formal definition of contexts (when used for packaging information[1]), based on the notion of *intention* and *extension*. We define a context in terms of what propositions it *allows* and *represents*. The propositions that it represents (a set of conceptual graphs) is said to be its extension. These graphs are conjunctively true (asserted) in that context. Its intention will be the conditions upon which the context exists. These conditions are the premises under which the extension of the context is said to be true. These conditions are expressed as a set of conceptual graphs conjunctively true in that context[2]. Together, the intention and extension of a context provide a formal definition which explicitly states what is (the extension) and what can be (according to the intention) represented in this context. They are each represented as a set of conceptual graphs. The association of an intention set and an extension set provides a formal definition for a context since the graphs of the extension set have to abide by the conditions of those of the intention set; and the graphs of the intention set are the conditions that must be enforced to allow the extension set to exist. As will be seen below, not all such associations are possible; not all such associations are necessary. A minimal representation of these associations is sufficient to represent the whole knowledge base. This representation is a *concept lattice*.

Concept lattices were introduced in [17]; this paper adapts these definitions to the conceptual graph terminology and presents how they can be used to properly structure a knowledge base defined in terms of a set of contexts. This structure, which we named *context lattice*, can be used to optimize query mechanisms which can address their queries to particular contexts, or which can now inquire about the links between intention and extension sets, i.e., about the structure of the knowledge base itself, which represents the relations among different worlds of assertions.

Section 2 below presents a brief literature survey on contexts in the conceptual graph theory. Section 3 states our proposition. Section 4 presents different

---

[1] This paper will not present contexts for inferential purposes, though we feel that the work presented here will eventually permit such extensions to be defined; it only presents its information packaging capability, used to create a lattice of subsumed worlds of assertions. However, section 6 presents how certain characteristics of contexts used in an inference engine for existential graphs, are also present with the definitional framework that we propose in this paper.

[2] In this paper, the intention of a context will be a unique conceptual graph, called *intention graph*, stating under what condition the context exists. Section 5 shows how this graph is produced. Other constraints on the extension of a context could be added to intention of a context, as will be done in a forthcoming paper.

functions which can be used to query a knowledge base structured as a context lattice. Section 5 describes the algorithm that produces the intention graphs used in section 3 to build the context lattice. Section 6 shows how some characteristics of contexts used for inferential purposes are present within the framework that we propose in this paper. Section 7 concludes.

2 Literature Survey

Contexts and concepts have been largely studied in the theory of conceptual graphs [2, 4, 14, 15]. It has been proven that contexts are fundamental to existential graphs and that the structuring of a knowledge base may benefit from using contexts for packaging information. Even though all authors agree on C.S. Peirce's preliminary definition of contexts as sheets of assertions, there is no common understanding of their underlying semantics.

[14] introduced the notion of context in the conceptual graph theory through the formal definition of propositions. He defined a proposition p as a concept where its concept type is PROPOSITION, and its referent, a set of conceptual graphs. These graphs are said to be true in the context of p. In [15], Sowa analyzed the semantic foundations of contexts through four theories. In this analysis, Sowa distinguished three syntactic aspects of contexts: a mechanism for packaging information, the contents of the package (a set of assertions), and the permissible operations on the package, including importing and exporting information to and from a package. In this study, Sowa himself points out that much of the controversy about the notion of context results from a lack of definitional semantics.

Traditionally, conceptual graphs specialists have used contexts to represent and partition (structure) the universe of discourse, as illustrated by [10, 11] and [1]. In the first case, Bernard Moulin gives a pragmatic interpretation of contexts. He defines the notion of discourse space that extends the notion of contexts by adding information grouped in an envelope describing a context. By analyzing the cognitive operations that human agents use when producing or understanding a discourse, Moulin identifies different kinds of discourse spaces. Among them, temporal discourse spaces support the representation of temporal structures, including situations; definitional discourse spaces support the definition of concept and extends the notion of white box proposed by Esch [2]).

In the second case, Judith Dick uses contexts to represent texts, especially legal texts, and notes that there is little clarity about the nature of contexts and how they ought to be used to help represent knowledge effectively. Contexts are definitely needed to represent natural language sentences and to package information. In order to fulfill all the requirements to successfully support this usage, it is obvious that contexts need to be semantically well defined.

In this paper, we propose to contribute to the semantic definition of contexts by adding a semantic dimension to the three syntactic aspects identified in [15]. This semantic dimension completely specifies the *raison d'être* of contexts when information packaging is concerned, thus bridging the gap between different points of view concerning the theory of contexts in the conceptual graph community.

3 Contexts: a Formal Definition

When a graph g represents a true statement, it is said to be an assertion. Except for universal truths, the truth value of a graph often depends on other assertions; it is conditional to these assertions which define the proper conditions under which it applies. These other assertions define situations where the graph g can be considered an assertion. The set of all situations in which g is an assertion is called its *scope*. Consequently, no assertion can be made without reference to the situations where it applies, i.e., to its scope; graphs are always asserted in relation to a situation. By default, in a conceptual graph system, all assertions are done on the *universal sheet of assertion* which represents universal truths, i.e., graphs whose truth value does not depend on some conditions, which in itself, is a particular situation. Consequently, since assertions are made in relation to some situations, there is an obvious need to define a situation where graphs which comply to the same conditions may be asserted. Such a situation will be called a *context*.

A context is a sheet of assertion whose existence depends on some graphs which describe the premises of its existence. All graphs in a context comply to the same premises. For example, modalities may be represented using such contexts. The assertion: “Mary thinks that Peter loves her” is universally true[3]: “Mary does think that Peter loves her”, but does Peter really love Mary or not? We could not assert anything about that, except that “Peter loves Mary” in Mary’s thoughts. So Mary’s thoughts form a context under which the statement: “Peter loves Mary” becomes true. The graph of Figure 1 shows how this assertion would be represented as a conceptual graph[4].

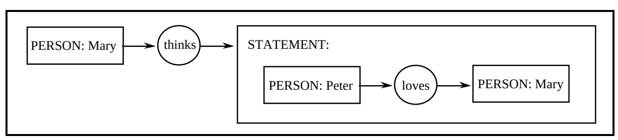

Fig. 1. The statement: “Mary thinks that Peter loves her”.

As another example, we could have the following statement: “Mary thinks that Peter is handsome” (see Figure 2). Also, we could have embedded contexts as shown in Figure 3, where the following statement is represented: “Mary thinks that Peter thinks that he is handsome”. As a shorter notation, Figure 4 may be used to represent all previous statements. In this figure, the set of statements instantiating the concept STATEMENT represents a sheet of assertion created by Mary’s thoughts.

Analyzing the graphs reveals that the three statements of Figure 4 are all related to what Mary thinks. So the following context (see Figure 5) exists de facto. A context will be defined by both the conditions from which it is created, and the set of graphs asserted under it. Similarly with embedded contexts, the context of Figure 6 is implied by the graph of Figure 3; it represents what Mary thinks that Peter thinks. This latter

[3] Provided that the vocabulary is well defined and the instances Peter and Mary do exist.

[4] In this article, proper names such as Mary and Peter are used as unique identifiers. Consequently, they can be used where referents normally appear.

context is obviously part of the former one, as represented by the embodiment structure of the graph of Figure 3.

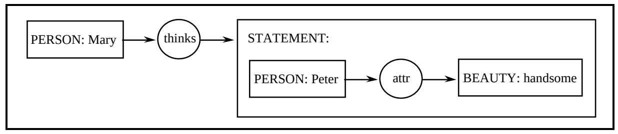

Fig. 2. The statement: “Mary thinks that Peter is handsome”.

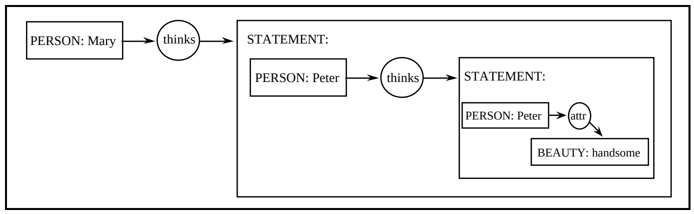

Fig. 3. The statement: “Mary thinks that Peter thinks that he is handsome”.

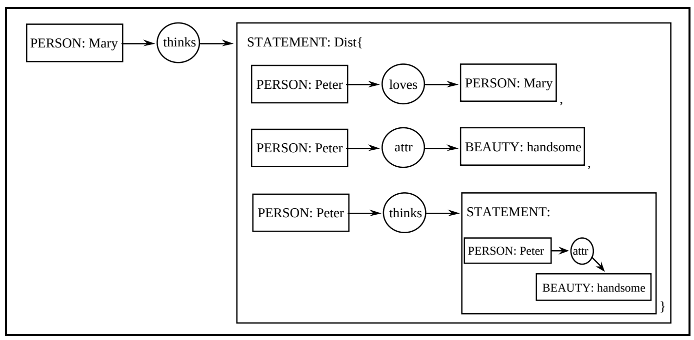

Fig. 4. A shorter notation for representing sets of statements in the same context.

As illustrated in Figures 5 and 6 below, a context is defined as two parts: an *intention*, a set of conceptual graphs which describe the conditions which make the asserted graphs true, and an *extension*, which is composed of all the graphs true under these conditions. Of course, a graph can appear in the extension of more than one context. For example, if Peter thinks that he loves Mary, this would imply the existence of the context of Figure 7, where the graph in the extension of this context already appears in some other context (in the extension set of the context shown in Figure 5).

Formally, a context Ci can be described as a tuple of two sets of conceptual graphs, Ti and Gi. Ti defines the conditions under which Ci exists, represented for now by a single intention graph; Gi is the set of conceptual graphs true in that context. So, for a context Ci, Ci = <Ti,Gi> = <I(Ci), E(Ci)>, where I(Ci) is a single conceptual graph, the intention graph of Ci, and E(Ci), the set of graphs conjunctively true in Ci, called the extension graphs. At this point, the reader should notice that a graph g may be true in E(Ci) without having been explicitly asserted in that context. For instance, if g is asserted in some other context but is a generalization of a graph asserted in Ci, than g is considered to be part of E(Ci). We define E(Ci) as the transitive closure of the asserted graphs of Ci under the subsumption relation that exists among all graphs of the knowledge base.

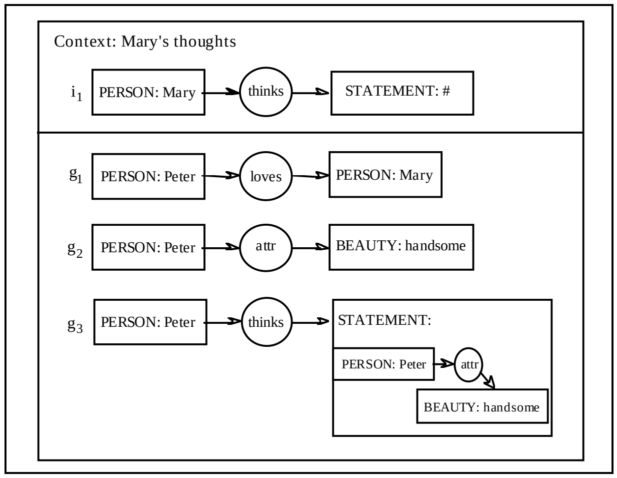

Fig. 5. The context of what Mary thinks.

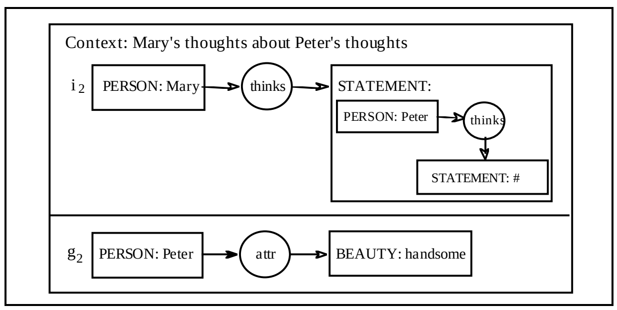

Fig. 6. The context of what Mary thinks that Peter thinks.

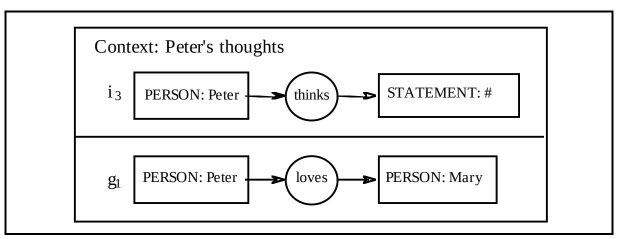

Fig. 7. The context of what Peter thinks.

Also, since a graph may appear in more than one context, if two contexts C_i and C_j are such that their respective intention graphs are related in a subsumption relation such that $I(C_j) < I(C_i)$, then the graphs true in C_j, i.e., the graphs of $E(C_j)$, are true in C_i as well, that is, $E(C_j) \subseteq E(C_i)$. So, we will compute $E(C_i)$ as the union of the graphs originally asserted in C_i and in C_j for all j such that $I(C_j) < I(C_i)$. After $E(C_i)$ is computed for all contexts, we say that g is *asserted* in C_i if g belongs to $E(C_i)$, no matter if it was originally asserted in that context or not. Knowing that $E(C_i)$ is computed using the subsumption relation that exists between the intentions graphs of all contexts, and using the subsumption relation that exists between the asserted graphs of the whole knowledge base, we now can give the following definitions using the intersection and union operations as done in [17].

First, let us define the *scope* S of a set of graphs G, i.e., the set of all contexts where the elements of G are conjunctively asserted. The scope of G is formally defined as: $S(G) = \{C_i \mid G \subseteq E(C_i)\}$. The scope of a single graph g could be computed as S({g}). I*(G) will be defined as the set of the intention graphs of S(G) (see equation 1 below).

$$\text{Equation 1: } I^*(G) = \cup_i I(C_i) \mid G \subseteq E(C_i)$$

Conversely, one could compute the set of graphs G conjunctively asserted in all contexts designated by T, a set of intention graphs. Equation 2 below introduces the function E* which computes the graphs conjunctively asserted in all contexts identified by the intention graphs of T.

$$\text{Equation 2: } E^*(T) = \cap_i E(C_i) \mid I(C_i) \subseteq T$$

Queries about graphs asserted in different contexts may be as interesting as queries about graphs conjunctively asserted in the same context; links between different contexts may be explored through these queries. Consequently, it is useful to relate sets of contexts to sets of asserted graphs, providing a structure for the whole knowledge base. In order to achieve that, we now introduce the notion of a *formal context*, named C_i^*, defined as a tuple $<T_i,G_i>$, where $G_i = E^*(T_i)$ and $T_i = I^*(G_i)$ at the same time (see equation 3 below). Because two partial orders of inclusion defined respectively over the sets $I^*(G_i)$ and $E^*(T_i)$ exist, the set of all formal contexts form a lattice structure L, i.e., all formal contexts are part of a partial order ≤. In effect, $C_2^* < C_1^*$ iff $C_2^* \neq C_1^*$ and $E(C_2^*) \subset E(C_1^*)$, or conversely, iff $C_2^* \neq C_1^*$ and $I(C_1^*) \subset I(C_2^*)$. Assuming the completion of $E(C_i)$ for all contexts C_i under both subsumption relations (of the intention graphs and of the extension graphs), we can adapt the notions presented here from those presented in [17]. Equations 3 and 4 below remind the reader of essential definitions used to develop the query mechanism presented in the next section.

$$\text{Equation 3: } C_i^* = <T_i,G_i> \text{ where } G_i = E^*(T_i) = E(C_i^*) \text{ and } T_i = I^*(G_i) = I(C_i^*)$$

$$\text{Equation 4: } L = < \{ C_i^*\}, \leq >$$

The context lattice L can be computed automatically without needing the knowledge engineer to intervene, providing an explanation and access structure to the knowledge base, and relating different worlds of assertions to one another. As explained earlier, all $E(C_i)$ sets are automatically computed from the sets of graphs originally asserted in each C_i. As will be shown in section 5, the intention graphs of each context C_i, i.e, $I(C_i)$, can also be automatically extracted from the graphs asserted in the system. Since all of this is automatic, the context lattice is built, i.e., the knowledge base is structured according to the semantics of the graphs it contains, without involving the knowledge engineer. With our example, the context lattice L of Figure 8 below would be produced.

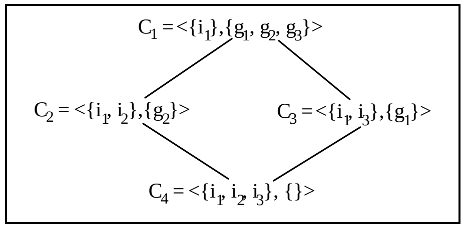

Fig. 8. The Hasse diagram of the context lattice produced from our example[5].

4 Querying a Knowledge Domain Structured as a Context Lattice

As mentioned earlier, queries about the truth value of any graph must always be computed according to a designated sheet of assertion, a context. The query mechanism then must include the information about the scope of its query graph. For that purpose, we now define a query mechanism which is based on formal contexts. Queries will be required to specify two parameters, one designating the appropriate context C* of the query, the other being the query graph itself. Since contexts are built from two types of information: an intention and an extension, two queries mechanisms will be devised. The first one will query the extension of a context; while the second one will query the intention of a context. The corresponding functions are introduced as Equations 5 and 6.

$$\text{Equation 5: } \delta_{E*}(C^*,q) = \{g \in E(C^*) \mid g \leq q\}$$

$$\text{Equation 6: } \delta_{I*}(C^*,q) = \{g \in I(C^*) \mid g \leq q\}$$

Because queries can be embedded, i.e., the result of a query being the object of the next query, and because we may want to use the structure of L to navigate, explore, and query the structure of the knowledge base, we now introduce two functions which identify a context based on a set of graphs G representing a subset of some extension set (Equation 7), or based on a set of graphs T representing a subset of some intention set (Equation 8).

$$\text{Equation 7: } C_E(G) = \langle I^*(G), E^*(I^*(G))\rangle$$

$$\text{Equation 8: } C_I(T) = \langle I^*(E^*(T)), E^*(T)\rangle$$

Combined together, the functions of Equations 5 and 7, and of Equations 6 and 8, are useful to change contexts based on the result of some previous query. When querying a knowledge base, going from one context to the next (navigating) can be done as illustrated in Figure 9, where we show how a change of context can be done based on the result of query q_1, prior to the evaluation of query q_2 (sent to a potentially different context)[6].

---

[5] Please notice that the graph identifiers in this figure, are those found in Figures 5, 6 and 7.

[6] Embedded queries can involve intention sets as well. We only show one simple example here to shorten the presentation.

$$\delta_{E*}(\ C_E(\delta_{E*}(C^*,q_1))\ ,\ q_2)$$

Fig. 9. Embedded queries.

As a shorter notation, we could define two functions ctx_{E*} and ctx_{I*}, shown in Equations 9 and 10, which produce a new context based on a previous context and a query. The appropriate subscript indicates to which part of the context the query q is sent to. Please notice that since $ctx_{E*}(C^*,q)$ and $ctx_{I*}(C^*,q)$ each identify a single formal context, because the knowledge base is structured as a lattice, one could use either one of them as parameters for the functions I and E in order to extract the intention set or the extension set of the new context, respectively.

Equation 9: $ctx_{E*}(C^*,q) = C_E(\delta_{E*}(C^*,q))$

Equation 10: $ctx_{I*}(C^*,q) = C_I(\delta_{I*}(C^*,q))$

Also, it is often the case that the result Q of a query q sent to the extension set of a context E(C*) is a subset of the extension set of the new context, i.e., $Q \subset E(ctx_{E*}(C^*,q)) \subseteq E(C^*)$. In that case, subsequent queries may need to consider only this subset Q and not the whole extension set of the new context, for obvious efficiency reasons. To accommodate that need, we now introduce variations on previous functions, where the set to which a query is sent may be smaller than the one to which the query would normally be sent to.

Equation 11: $\delta_E(C^*,G,q) = \{g \in G \mid G \subseteq E(C^*) \text{ and } g \leq q\}$

Equation 12: $\delta_I(C^*,T,q) = \{g \in T \mid T \subseteq I(C^*) \text{ and } g \leq q\}$

These definitions are equivalent to their previous counter-parts if G = E(C*) and T = I(C*) (see Equations 13 and 14). The user now has the possibility of restraining the set to which the query is sent to, even though the context may change as the result of the query, by using either δ_E or δ_{E*}, or similarly, either δ_I or δ_{I*}.

Equation 13: $\delta_E(C^*,E(C^*),q) = \delta_{E*}(C^*,q)$

Equation 14: $\delta_I(C^*,I(C^*),q) = \delta_{I*}(C^*,q)$

Embedded queries could then be expressed using variables representing sets of graphs obtained by previous queries. This way, long algebraic formulas can be broken down into shorter and more comprehensible ones. As such, these queries are much easier to write and read. An example is given in Figure 10. With such a framework, at any moment i in time, the set of graphs retrieved so far is s_i, and the current context is given by $C_E(s_i)$ (or by $C_I(s_i)$).

$$s_1 = \delta_{E*}(C^*,\ q_1)$$
$$s_2 = \delta_E(\ C_E(s_1)\ ,\ s_1\ ,\ q_2)$$
$$s_3 = \delta_E(\ C_E(s_2)\ ,\ s_2\ ,\ q_3)$$
$$s_4 = \delta_{I*}(C_E(s_3),\ q_4)$$

Fig. 10. An example of embedded and consecutive queries.

Given the context lattice of Figure 8, the graph of Figure 1 as the first query q_1 and the graphs of Figures 11 and 12 as subsequent queries q_2 and q_3, the embedded formulation of Figure 13 (explained below) would result in the navigation illustrated in Figure 14. This embedded query aims at determining if Mary thinks that Peter is aware of his love for her.

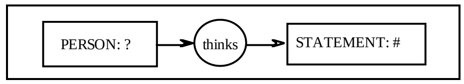

Fig. 11. Who thinks (the same thing)? (q_2)

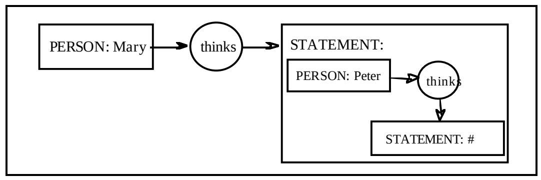

Fig. 12. Does Mary think that Peter shares her thought that he loves her? (q_3)

$$s_1 = \delta_{E*}(C_1, q_1)$$
$$s_2 = \delta_{I*}(C_E(s_1), q_2)$$
$$q_3 \in I(C_I(s_2))?$$

Fig. 13. The three consecutive queries q_1, q_2 and q_3.

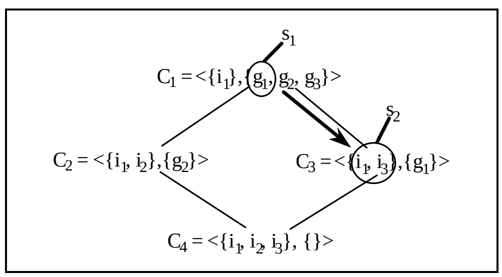

Fig. 14. The navigation resulting from the queries of Figure 13.

The first query, the graph of Figure 1 (q_1), could be sent as such to the universal sheet of assertion[7], or the context to which the most embedded assertion is made can be automatically determined (in our example, it is context C_1) before a simple graph (the most embedded graph in q_1, being the graph g_1 in Figure 5) can be sent for evaluation to the extension set of context C_1. This simplified query graph matches graph g_1 in the extension of C_1. So the answer to this query graph would be *true*, the set $s_1 = \{g_1\}$ being non-empty: Mary does think that Peter loves her. That gives us a starting point (a current context) in L, to navigate from. The computation of the most specific

[7] The universal sheet of assertion has not been modeled in our paper in order to simplify our example. However, it could be described using an intention and an extension set, as any other context, and could be part of the context lattice representing the knowledge base. However, it would introduce redundancy for graphs containing STATEMENT concepts, as the asserted information is then represented in some other context.

context containing the elements of s_1 as part of their extension set results in a downward move of the current context from C_1 to C_3 (when $C_E(s_1)$ is evaluated, see Figure 14). The second query q_2 is then sent to the intention set of C_3: "who shares this thought?". The answer would be *Peter* and *Mary*, the two referents instantiating the ? symbol in both graphs of s_2. Finally, we update the current context using the C_I function before testing if q_3 belongs to the intention set of the potentially new context. In our example, the current content does not change and the final answer is *no*, since i_2 is not part of the intention set of the current context. So, under a closed world assumption, Mary does not think that Peter is aware of his love for her.

This example should emphasize two advantages of our approach: 1) queries can be simplified (in some cases even flattened) by identifying a context to which a simpler query can be sent to, 2) the structure of the knowledge base itself can be queried by allowing queries to be sent to either the intention or extension set of a context. The process used to automatically identify a context when a graph is acquired (see section 5 below) is used once more when a query graph is sent to the knowledge base: the context where the query graph should be sent to is identified; the query is thus simplified; its evaluation is much simpler than it would normally be if sent to the universal sheet of assertion where a graph matching procedure must be applied recursively on embedded parts (which are graphs).

5 The Extraction of an Intension Graph

The intension graphs associated with the existence of contexts can be automatically extracted from the conceptual graphs acquired by the system. In fact, the extraction procedure given below is adaptable to different applications; the only requirement is that it must be used when the knowledge base is both acquired and queried. What is important is that intension graphs must be produced in a uniform manner. The example of Figure 15 below is used to show how this procedure works.

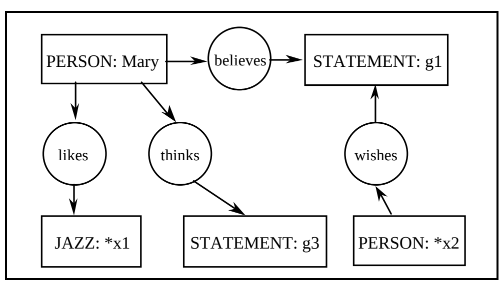

Fig. 15. An asserted graph on the universal sheet of assertion.

Step 1: Make sure that each generic concept is identified by a unique referent variable with regard to any other concept in the knowledge base, unless the concept should be in coreference with an already existing concept. In that case, it should be identified using the referent variable of this existing concept[8]. With our example, we suppose

[8] In this paper, we assume *global* coreference. As a matter of fact, all cg system developers know that global coreference is mandatory to distinguish concepts belonging to different graphs and to allow the coreference of concepts belonging to

that this step was already carried out, and that g1 and g3 refer to the graphs of Figure 5.

Step 2: Duplicate (copy) every STATEMENT concept so that they are used with only one relation, either as input or output parameter. Notice that a set of graphs may result from this step. With our example, we get the two graphs shown in Figure 16.

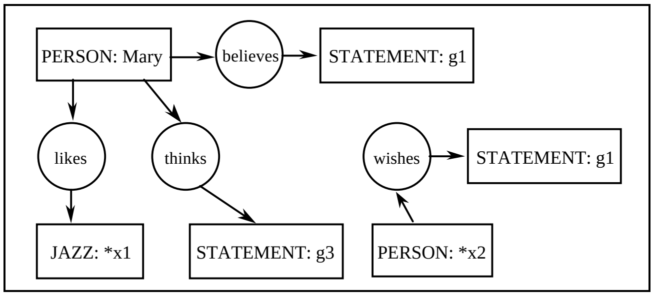

Fig. 16. The graph of Figure 15 after step 2.

Step 3: From the set of graphs produced by step 2, extract the relations which use a STATEMENT concept, either as input or output parameter. With our example, we would get the graphs of Figure 17.

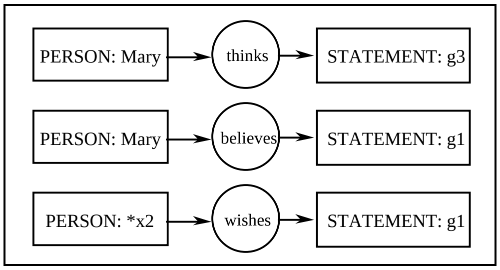

Fig. 17. The relations extracted by step 3.

Step 4: For each graph g produced by step 3, replace the referent r of g by the # symbol. With our example, we would get the graphs of Figure 18.

Step 5: For each graph produced by step 4, called g', if the corresponding graph in step 3 g was asserted in a context, use the intention graph of this context and replace its # symbol by g', producing the intension graph designating the context where the referent of g, r, should be asserted. Since the graph of Figure 15 was asserted on the universal sheet of assertion, this step does not apply to the first iteration of our example, and the intention graphs produced with our example are the ones shown in Figure 18. However, this step would apply to the second iteration of the procedure.

Step 6: Select the graphs of step 3 where the referent of the STATEMENT concept is an individual referent, that is, a conceptual graph, where a STATEMENT concept appears. Reapply steps 1 thru 6 to these graphs. With our example, only the first graph

---

graphs asserted independently. This may seem to be a philosophical difference from what is currently done in any cg system, but in fact, considering the iteration rule, it is not (see section 6 for more details).

of Figure 17 would be kept. From it, the procedure would produce the intention graph of the context shown in Figure 6.

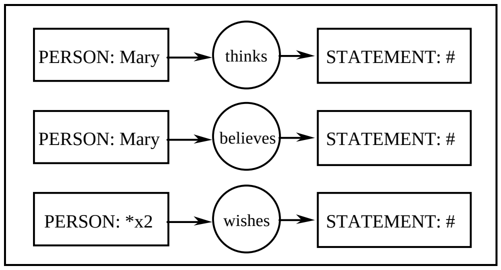

Fig. 18. The three intention graphs produced from the graph of Figure 15.

6 Using a Context Lattice for Inference Purposes

This paper proposes a way to automatically structure a knowledge base using a *context lattice*. Each context can be seen as a world of assertion. The extension of this work will lead to the definition of inference capabilities within each context. For that purpose, contexts used to delimit the scope of negation with regard to sets of propositions, called *basic contexts*, could be used within each of these worlds. In that case, their use is almost purely syntactical, and it is hard to find a mapping between them and the contexts described in this article. However, there exist similarities in the functionnalities that both types of contexts provide, which may lead to a mapping from one to the other. For instance, the "lifting" and "pushing" rules allowing the propagation of information between subsumed contexts could probably be defined using basic contexts. The iteration rule associated with basic contexts would be represented by a propagation mechanism based on the subsumption relation between contexts. The deiteration rule would be just the inverse process; it would imply that inheritance would be used in order to conduct inferences and avoid redundancy. The double negation rule would be an assertion rule with regard to some context. The insertion and erasure rules, however, need negation to be represented within these contexts in order to have their counter-part in what we propose in this article. Our work remains to be extended in that direction. However, one may think that an information propagation mechanism between contexts, coupled with inferential capabilities, could erase the need for basic contexts as used so far in the cg literature. Of course, that remains to be seen, as the burden of proof is still ours.

7 Conclusion and Future Developments

This paper proposes a definitional framework for defining contexts formally; we hope that with regard to the information packaging facilities that it offers, that it makes a consensus among the CG community. We explained how contexts could be automatically abstracted from a set of graphs; we presented query mechanisms useful to explore not only the asserted graphs, but also the structure of the contexts themselves, i.e., the way they relate to one another. The framework that we propose in this paper is based on formal concept analysis [17]; it links the intention and extension of a knowledge base in a single unique representation paradigm since both the intention and extension of a context are described using conceptual graphs. The knowledge base, structured as a *context lattice*, represents how the different sets of assertions are partitioned and shared among different agents who believe them. To our opinion, this structuring will prove to be useful for many application areas, particularly for natural language processing, modal logic and multi-agent systems. However, even though section 6 identified some trails as to how a context lattice may be used to fully implement an inferential mechanism based on these contexts, more research in that area is definitely needed. Still, we believe that the contexts used in an inference system for existential graphs should be defined with sufficient semantics to allow their full mapping to some other packaging constructs whose existence could be formally defined. We hope that this paper starts a discussion in that direction.

Implementation issues concerning lattices and lattice-based data structures have been on our research agenda for many years now. We have developed very efficient batch and incremental algorithms that create and maintain a lattice structure [6, 8]. We have been particularly concerned with the feasibility of our approach for large knowledge bases. As a result, we proposed in [5] a wide spectrum of lattice-based data structures useful for organizing and structuring large knowledge bases. We will soon address the problem of computing the $E(C_i)$ sets effectively. As mentioned before, there are a lot of interesting work about computing subsumption relations at the lowest cost possible, that could be adapted for our needs. All algorithmic and implementation aspects of that task will be a major concern of ours at least for the next year.

Meanwhile, we are currently extending the work presented in this paper to allow extension sets to be infinite. That way, contexts will not only represent sheets of assertions, but complete worlds of inferences. Agents in multi-agent systems need the capability to infer more than what they were originally told. It is our opinion that this possibility of distributing and partitioning inference capabilities will allow the conceptual graph formalism to be considered for distributed applications. Furthermore, we feel that there are application domains for which this distribution of inference capabilities will result in a gain of efficiency for the whole system, as mentioned in [7]. At this point in time, this remains to be seen; and we certainly plan to explore in that direction in the near future.

8 Bibliography

1. Dick, J. P. Using Contexts to Represent Text. In W.M. Tepfenhart, J.P. Dick & J.F. Sowa (Eds.), *Conceptual Structures: Current Practices,* (pp. 196-213). Springer-Verlag, 1994.
2. Esch, J. Contexts and Concepts, Abstraction Duals. In W.M. Tepfenhart, J.P. Dick & J.F. Sowa (Eds.), *Conceptual Structures: Current Practices,* (pp. 175-184). Springer-Verlag, 1994.
3. Esch, J. Contexts, Canons and Coreferent Types. In W.M. Tepfenhart, J.P. Dick & J.F. Sowa (Eds.), *Conceptual Structures: Current Practices,* (pp. 185-195). Springer-Verlag, 1994.
4. Esch, J. W. Contexts as white box concepts. In *Proceedings of the 1st Int. Conf. on Conceptual Structures*, G.W. Mineau, B. Moulin & J.F. Sowa (Ed.), Université Laval, pp. 17-29, 1993.

5. Godin, R., Mineau, G., Missaoui, R. & Mili, H. Méthodes de classification conceptuelle basées sur le treillis de Galois et applications. *Revue d'Intelligence Artificielle*, **9**(2), 105-137, 1995.

6. Godin, R., Mineau, G. W. & Missaoui, R. Incremental Structuring of Knowledge Bases. In *Proceedings of the 1st Int. Symposium on Knowledge Retrieval, Use, and Storage for Efficiency (KRUSE)*, G. Ellis, R. Levinson, A. Fall & V. Dahl (Ed.), Santa Cruz, CA.: Dept. of Computer Science, UCSC, CA., pp. 179-193, 1995.

7. Lenat, D. B. & Guha, R. V. *Building Large Knowledge-Based Systems, Representation and Inference in the Cyc Project*. Addison-Wesley, 1990.

8. Mineau, G. W. & Godin, R. Automatic Structuring of Knowledge Bases by Conceptual Clustering. *IEEE Transaction on Knowledge and Data Engineering*, **7**(5), 824-829, 1995.

9. Moulin, B. The representation of linguistic information in an approach used for modeling temporal knowledge in discourses. In G.W. Mineau, B. Moulin & J.F. Sowa (Eds.), *Conceptual Graphs for Knowledge Representation*, (pp. 182-204). Springer-Verlag, 1993.

10. Moulin, B. Discourse Spaces: A Pragmatic Interpretation of Contexts. In G. Ellis, R. Levinson, W. Rich & J.F. Sowa (Eds.), *Conceptual Structures: Applications, Implementation and Theory*, (pp. 89-104). Springer-Verlag, 1995.

11. Moulin, B. A Pragmatic Representational Approach of Context and Reference in Discourses. In G. Ellis, R. Levinson, W. Rich & J.F. Sowa (Eds.), *Conceptual Structures: Applications, Implementation and Theory*, (pp. 105-114). Springer-Verlag, 1995.

12. Moulin, B. & Dumas, S. The Temporal Structure of a Discourse and Verb Tense Determination. In W. M. Tepfenhart,J. P. Dick, & J. F. Sowa (Eds.), *Conceptual Structures: Current Practices*, (pp. 45-68). Springer-Verlag, 1994.

13. Moulin, B. & Mineau, G. W. Factoring Knowledge into Worlds. In *Proceedings of the 7th Ann. Workshop on Conceptual Graphs*, H. Pfeiffer (Ed.), New Mexico State University, Las Cruces, New Mexico: pp. 119-128, 1992.

14. Sowa, J. F. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

15. Sowa, J. F. Syntax, Semantics and Pragmatics of Contexts. In G. Ellis, R. Levinson, W. Rich & J.F. Sowa (Eds.), *Conceptual Structures: Applications, Implementation and Theory*, (pp. 1-15). Springer-Verlag, 1995.

16. Sowa, J. F. Processes and Participants. In P.W. Eklund, G. Ellis & G. Mann (Eds.), *Conceptual Structures: Knowledge Representation as Interlingua*, (pp. 1-22). Springer-Verlag, 1996.

17. Wille, R. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. In I. Rival (Eds.), *Ordered Sets*, (pp. 445-470). Dordrecht-Boston: Reidel, 1982.