Some approaches of ontology Decomposition in Description Logics

Thi Anh Le PHAM *, Nhan LE-THANH and Peter SANDER

Laboratoire I3S, Université de Nice Sophia-Antipolis, France
{tpham,nhan.le-thanh,peter.sander}@unice.fr

Abstract. In this paper, we investigate the problem of decomposing an ontology in Description Logics (DLs) based on graph partitioning algorithms. Also, we focus on syntax features of axioms in a given ontology. Our approach aims at decomposing the ontology into many sub-ontologies that are as distinct as possible. We analyze the algorithms and exploit parameters of partitioning that influence the efficiency of computation and reasoning. These parameters are the number of concepts and roles shared by a pair of sub-ontologies, the size (the number of axioms) of each sub-ontology, and the topology of decomposition. We provide two concrete approaches for automatically decomposing the ontology, one is called partitioning based on minimal separator, and the other is segmenting based on eigenvectors and eigenvalues.

Keywords. Graph partitioning, ontology decomposition, image segmentation.

Introduction

Previous studies about DL-based ontologies focus on such tasks as ontology design, ontology integration, ontology deployment,... Starting from the fact that one wants to effectively solve and reason with a large ontology, instead of integrating multiple ontologies we examine the decomposition of an ontology into several sub-ontologies that overlapping content. Some reasoning algorithms on the system of decomposed ontologies and the criteria for decomposition have been proposed in the previous paper [5]. In this paper, we resolve the problem of decomposing, more concretely, we delve into the techniques of decomposing an ontology into several sub-ontologies. Our principal goal is how select a "good" one? A decomposition is called "good" if it improves the efficiency of reasoning and guarantees the properties proposed in [5]. Our computational analysis for reasoning algorithms guides us to suggest the parameters of such a decomposition: the number of concepts and roles included in the semantic mappings between partitions, the size of each component ontology (the number of axioms in each component) and the topology of the decomposition

* Laboratoire I3S, UMR 6070, CNRS, Les Algorithmes, Bât. Euclide 2000, Route des Lucioles, BP121, 06903 Sophia-Antipolis, France; Email: tpham@i3s.unice.fr

graph. There are two approaches to be considered here. They are based on two presentations of the ontology. First, we present the ontology by a symbols graph, and implement decomposition based on minimal separator method. Second, the ontology is presented by an axiom graph, corresponding to the image segmentation method.

The paper is organized as follows. Section 1 describes two ways for transforming an ontology by an undirected graph (symbol graph or weighted graph). Section 2 defines the overlap decomposition of a TBox and the criteria for a good decomposition. In sections 3 and 4, we discus the methods and two partitioning algorithms of the graph representing an ontology corresponding to the above graphs. Section 5 presents some evaluations of the efects of the decomposition algorithms. Section 6 provides some conclusions and future work.

1 Representing ontology as the graph

A knowledge base (ontology) built by a concept language is generally composed of two component levels: intensional (called TBox) and extensional (called ABox). We are interested in the intensional level, the reasoning problem in a large TBox. A TBox is a finite set of general concept inclusions (axioms). Therefore decomposing an ontology is reduced to examining the set of axioms - we split the axioms into small groups following the proposed criteria in [5].

In this section, we present an ontology as an undirected adjacency graph. In particular, we examine two approaches for representing a TBox by graphs that deal with the problem of decomposition. The axioms are coded as follows.

Given an original TBox $\mathcal{T}$ with N axioms. Unfolding (expanding) all the axioms in the original TBox $\mathcal{T}$ into the expressions of primitive concepts and primitive roles (i.e. computing $Ex(A_i)$ as in [5], we recall that $Ex(A_i)$ is a set of primitive concepts and roles which occurs in A_i).

Let $|Ex(A_i)| = n_i$ (the number of concepts and roles in A_i), $|Ex(A_i) \cap Ex(A_j)| = n_{ij}$ (the number of common concepts and roles in two axioms A_i and A_j), for $i, j = \overline{1, N}, i \neq j$

Example 1. Given a TBox $\mathcal{T}$ presented in Figure 1, then $N = 10$ (there is ten axioms in $\mathcal{T}$) and the axioms are unfolded as follows:

$Ex(\mathcal{T}) = \{C_1, C_2, C_3, C_4, C_5, C_6, X, Y, H, T\}$; $|Ex(\mathcal{T})| = 10$

(A_1) : $C_1 \sqcap C_2 \sqsubseteq X$	(A_6) : $Y \sqcap T \sqsubseteq H$
(A_2) : $C_3 \sqsubseteq \neg C_2$	(A_7) : $C_3 \sqsubseteq X$
(A_3) : $X \sqcap C_4 \sqcap C_5 \sqsubseteq Y$	(A_8) : $\neg C_3 \sqsubseteq C_2$
(A_4) : $\neg C_4 \sqsubseteq \neg Y$	(A_9) : $\neg X \sqsubseteq \neg Y$
(A_5) : $Y \sqcap C_6 \sqsubseteq H$	(A_{10}) : $\neg C_5 \sqsubseteq \neg Y$

Fig. 1. TBox $\mathcal{T}$

To simplify, we use the notation *symbol* instead of primitive concept (role) name, i.e., a symbol is either a primitive concept (role) name in a TBox. A *symbol graph* will be introduced in the following section.

1.1 Symbol graph

Definition 1. *(Symbol graph) A graph $G = (V, E)$, where V is a set of nodes and E is a set of edges, is called a symbol graph if each node $v \in V$ is a symbol in TBox $\mathcal{T}$, each edge $e = (u, v) \in E$ if $u, v \in V$ and u, v occur in the same axiom.*

Construct a symbol graph $G = (V, E)$ from a TBox $\mathcal{T}$ by taking each symbol in every axiom in $\mathcal{T}$ as a node and connecting each pair of symbols by an edge if this pair co-occurs in the same axiom. According to this method, each axiom is presented as a "clique" (see definition 6) in the graph.

Example 2. The above TBox $\mathcal{T}$ can be presented by a symbol graph as Figure 2.

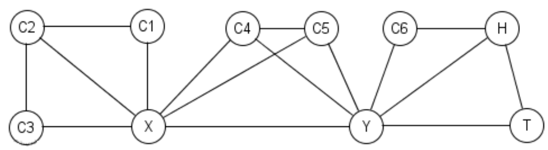

Fig. 2. Symbol graph of the TBox $\mathcal{T}$

1.2 Axiom graph

Definition 2. *(Axiom graph) A graph $G = (V, E)$, where V is a set of nodes and E is a set of edges, is called an axiom graph if each node $v \in V$ is an axiom in TBox $\mathcal{T}$, each edge $e = (u, v) \in E$ if $u, v \in V$ and there is at least a shared symbol between u and v.*

Example 3. The above TBox $\mathcal{T}$ (Figure 1) is represented by an axiom graph as in Figure 3.

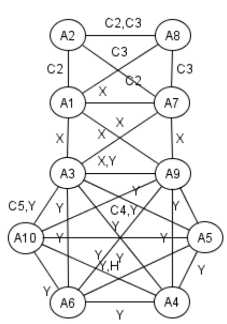

Fig. 3. Axiom graph of the TBox $\mathcal{T}$

From a point of view of graph, we present a TBox $\mathcal{T}$ by a weighted graph $G = (V, E)$, denoted the "connected axiom graph": each vertex v of V corresponds to an axiom in $\mathcal{T}$, and each pair of vertices in V is connected by an edge if they have some shared symbols. A weight is then assigned to each edge in E. This weight should be chosen such that it reflects the association degree between the linked vertices of that edge. Using just the common symbols between every pair of axioms, we can define the graph edge weight w_{ij} connecting the two nodes i and j that correspond to two axioms A_i and A_j as: $w_{ij} = \frac{n_{ij}}{n_i + n_j}$, where $i, j = \overline{1, N}, i \neq j$.

2 Ontology Decomposition

2.1 Definition

In this paper, the ontology is examined as a graph, therefore the notion of ontology decomposition is based on graph partitioning. Assume that a graph G is partitioned into m sub-graphs $G_{i,i \leq m}$, then partitioning G is defined as follows:

Definition 3. *(Graph partitioning) A graph partitioning $G_p = (V_p, E_p)$ is a connected graph (see definition 7) of the individual partitions G_i. Each node $v \in V_p$ is a partition (sub-graph) G_i, each edge $e_{ij} = (v_i, v_j) \in E_p$ is labeled by the set of shared symbols between G_i and G_j, where $i, j \leq m$ and $i \neq j$.*

This definition is equivalent to the overlap decomposing in [5], where G_i corresponds to component TBox $\mathcal{T}_i$, the edge e_{ij} corresponds to the set of identical bridge rules $\mathcal{B}_{ij}$.

2.2 What is a good decomposition

Our goal is to eliminate general concept inclusions (GCIs) as much as possible from a general ontology (presented by a TBox) by decomposing the set of GCIs of this ontology into several subsets of GCIs (presented by a distributed TBox). In this paper, we only consider the syntax approach based on the structures of GCIs, the decomposition only implements the set of axioms (GCIs). All the concepts and roles of the original ontology will be kept through the decomposition. As a result, we propose two techniques for decomposing based on graphs.

We defined the decomposition of a TBox as a distributed TBox consisting of sub-TBoxes and the semantic mappings between each pair of sub-TBoxes. Our decomposition approach is most related to the graph formulation and partitioning. A delicate aspect of decomposition-based logical reasoning is in the selection of a good partitioning of the ontology based on the following questions:

1. What is the precise criterion for a good decomposition ?
2. How can such a decomposition be computed efficiently ?

These questions have been also proposed for image segmentation and data clustering. In the general DL case, our purpose is not to reduce computational complexity, but the results for reasoning in decomposition suggest similar relationship between the decomposed TBoxes and the original TBox. The computation analysis of our decomposition-based reasoning algorithms provides a metric for identifying parameters of decomposition that influence our computation: the size of the communication part between component TBoxes, the size of each component, and the topology of the decomposition graph. Our goal is minimizing the disassociation between the component TBoxes and maximizing the association within the components. Moreover these component TBoxes must preserve the properties of decomposition that we proposed in [5]. These parameters guide us to propose two greedy algorithms for decomposing an ontology into sub-ontologies by trying to optimize these parameters. The decomposition algorithms depend on the representation graphs of ontology that will be presented in the following sections.

3 Partitioning based on the minimal separator

Starting from the symbol graph, we examine a decomposition method based on the minimal separators. Before describing details of the algorithm, we recall some notions in graph theory that can be found in [1, 2, 4].

3.1 Definitions

Definition 4. *(Neighbor) Let $G = (V, E)$ be an undirected graph, with $|V| = n, |E| = m$. For $x, y \in V$, the set of neighbors of x is $N(x) = \{y \neq x | (x, y) \in E\}$. For $X \subseteq V$, $N(X) = \bigcup_{x \in X} N(x)$.*

Definition 5. *(Minimum Vertex Separators) A set of vertices S is called an (a, b)-vertex-separator if $\{a, b\} \subset V \setminus S$ and every path connecting a and b in G passes through at least one vertex contained in S. If S is an (a, b)-vertex-separator and no proper subset of S is an (a, b)-vertex-separator, then S is called a minimal (a, b)-separator.*

Two vertices a, b are said to be adjacent if there is an edge connecting a and b in G.

Definition 6. *(Cliques) A clique (complete graph) is a graph with edge set consisting of all possible edges between the vertices of the graph. We denote by $G[S]$ the clique built on vertices from S of the graph G.*

Definition 7. *(Connected graph) A graph G is connected if for each pair u, v of its vertices there exists a path from u to v. If G is not connected, then a connected subgraph C of G is called a (maximal) connected component if there exists no connected subgraph C' of G such that C is a proper subgraph of C'.*

3.2 Algorithm

In this section, we introduce a recursive algorithm using Even's algorithm [2], which takes a symbol graph $G = (V, E)$ (is transformed from a TBox $\mathcal{T}$) as input and returns the set of separate sub-graphs of G. We briefly describe the algorithm as follows:

Input: $G = (V, E)$
Output: connected graph $G_p = (V_p, E_p)$

- Find the set of minimal vertex separators of graph G:
  + Chose an arbitrary pair a, b of non-adjacent vertices, and compute the set of minimal ab-separators.
  + Repeat this process on every possible pair $\{x, y\}$ of non-adjacent vertices.
- Find the overall minimum vertex separator S^* between all vertices in G.
- Split G into the two subgraphs G_1, G_2 that are separated by S^*, with S^* included in both subgraphs.
- Create an undirected graph $G_p = (V_p, E_p)$ with $V_p = \{G_1, G_2\}$ and $E_p = S^*$

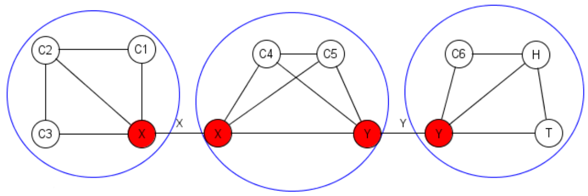

Fig. 4. Decomposing $\mathcal{T}$'s symbol graph

Figure 4 illustrates the connected graph of TBox $\mathcal{T}$ in Figure 2.

In this example, we have two minimum vertex separators $\{X\}$ and $\{Y\}$. If we chose $\{X\}$ to split $\mathcal{T}$, then we collect two groups of symbols $\{C_1, C_2, C_3, X\}$ and $\{X, C_4, C_5, C_6, Y, H, T\}$. Hence, we obtain two sub-TBoxes respectively : $\mathcal{T}_1 = \{A_1, A_2, A_7, A_8\}$ and $\mathcal{T}_2 = \{A_3, A_4, A_5, A_6, A_9, A_{10}\}$. Similarly, if we chose $\{Y\}$, we obtain two sub-TBoxes $\mathcal{T}_1 = \{A_1, A_2, A_3, A_4, A_7, A_8, A_9, A_{10}\}$ and $\mathcal{T}_2 = \{A_5, A_6\}$.

We will take the first result, because there is a better balance between the number of axioms of the two sub-TBoxes. Thus, the axioms in the original TBox is distributed into $\mathcal{T}_1$ and $\mathcal{T}_2$ with the number of axioms respectively $N_1 = 4$, $N_2 = 6$.

4 Decomposition based on image segmentation

This section describe a grouping method as image segmentation using eigenvectors. First of all, we assume that a graph $G = (V, E)$ has been partitioned into two disjoint parts with the two sets of nodes A, B respectively. The degree of association between these two parts can be computed as total weight of edges that have been removed from the original graph. The optimal partitioning of a graph not only minimizes this degree but also maximizes the degree of association within parts. In graph theory, these degrees are selected as [3]:

$Ncut(A, B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$;

$Nassoc(A, B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$

where $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$, the total connection from nodes in A to nodes in B; and $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$, the total connection from nodes in A to all nodes in the graph.

We can simply see that, $Ncut(A, B) = 2 - Nassoc(A, B)$.

We recall some notations from [3]: Let x be an $N = |V|$ dimensional indicator vector with $x_i = 1$ if node i is in A, and $x_i = -1$, otherwise. Let $d(i) = \sum_j w(i, j)$ be the total connection from node i to all other nodes. Let D be an NxN diagonal matrix with d on its diagonal, W be an NxN symmetrical matrix with $W(i, j) = w_{ij}$.

They are used for computing the optimal partition by solving $(D - W)x = \lambda Dx$.

The algorithm is implemented by the following principal steps [3]:

1. Transform a given ontology into an undirected graph (as the above section).
2. Find the minimum value of NCut by solving the eigenvectors with the smallest eigenvalues of the system: $(D - W)x = \lambda Dx$
3. Use the eigenvector with the second smallest eigenvector for decomposing the graph into two partitions. In the ideal case, the eigenvector should only take on two discrete values and the signs of the values can tell us exactly how to decompose the graph.
4. After the graph is broken into two partitions, we can recursively implement our algorithm on these two decomposed partitions.

Figure 5 illustrates the NCut obtained (denoted by the blue line) in the example of Figure 2, and result graph of decomposition is shown in Figure 6. This result is the same in the method based on minimal separator.

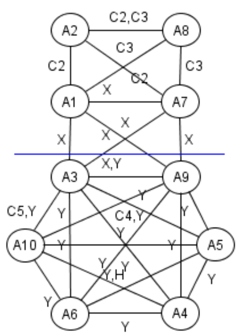

Fig. 5. Axiom graph of $\mathcal{T}$ and its NCut

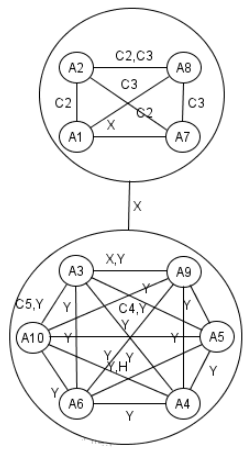

Fig. 6. Decomposing $\mathcal{T}$'s axiom graph

5 Effect of the decomposition algorithms

We have been applied two graph partition algorithms based on minimal separator and image segmentation for decomposing an ontology. These two methods return the same result that satisfies the proposed decomposition properties. All the concepts, roles and axioms are preserved over the decomposition implementation. The relations between them are expressed by the edges in symbol, and connected axiom graphs. The set of axioms in the original TBox is then reduced by regular distribution into sub-TBoxes. Decomposition techniques focus on finding a good decomposition. The approach based on minimal separator minimizes the number of symbols shared between partitions and tries balancing the number of axioms in the partitions. However, we have to regroup the axioms based on the cliques in the symbol graph after decomposition. However, in reality there exists some cliques in symbol graph which do not represent axioms.

The possible advantage of an approach based on image segmentation is that it conserves the axioms. Furthermore, the NCut measure is normalized, it expresses the disassociation between partitions and the association within partitions. However, to implement this technique effectively, we must propose a proper weight function for the edges between nodes of the axiom graph.

6 Conclusion and future work

In this paper, we have presented two algorithms for ontology decomposition in DLs using graph partitioning. By treating the grouping problem in graph theory, we proposed two methods for representing the ontology by a symbol graph and a weighted graph. Each graph type is applied corresponding to a decomposition algorithm. Two graph partitioning algorithms are independently implemented. We see that two approaches are sufficient for our purpose in requirements of decomposed ontologies. However, the choice of decomposition method depends on the structure of the original ontology. For example, with an ontology that consists of a large number of symbols then one can chose the weighted graph method. And for an ontology contains many axioms, one can use the symbol graph method. These proposals are only suggestions for future work. We also propose some essential properties of a good decomposition that influence the reasoning performance as provided in [5]. The graph transformations of an ontology again depends on having an effective (and cheap) method for analyzing the likely characteristics of a given test ontology. We are also performing more experiments with very large KBs (as UMLS,...) for decomposing. Preliminary results in the decomposed ontologies suggest similarity between both methods. We are embarking on optimizing these decomposition algorithms and effectively treating large ontologies.

References

1. Dieter Jungnickel, Graphs,Networks and Algorithms. Springer 1999.
2. Eyal Amir and Sheila McIlraith, Partition-Based Logical Reasoning for First-Order and Propositional Theories. *Artificial Intelligence, Volume 162, February 2005, pp. 49-88.*
3. Jianbo Shi and Jitendra Malik, Normalized cuts and Image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 888-905, August 2000.*
4. Kirill Shoikhet and Dan Geiger, Finding optimal triangulations via minimal vertex separators. *In Proceedings of the 3rd International Conference, p. 270-281, Cambridge, MA, October 1992.*
5. Thi Anh Le Pham and Nhan Le Thanh, Decomposition-Based Reasoning for Large Knowledge Bases in Description Logics. *In Proceedings of the 13th ISPE International Conference on Concurrent Engineering: Research and Applications, p. 288-295, Antibles, France, September 2006*
6. T.Kloks and D.Kratsch, Listing all minimal separators of a graph. *In Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science, Spinger, Lecture Notes in Computer Science, 775, pp.759-768.*