

Revision of stratified OWL ontologies based on integer linear programming

Ji Qiu¹ Qi Guilin^{2,3} Boutouhami Khaoula^{2,3}

(¹School of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

(²School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

(³Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing 211189, China)

Abstract: To revise stratified web ontology language (OWL) ontologies, the kernel revision operator is extended by defining novel conflict stratification and the incision function based on integer linear programming (ILP). The ILP-based model considers an optimization problem of minimizing a linear objective function which is suitable for selecting the minimal number of axioms to remove when revising ontologies. Based on the incision function, a revision algorithm is proposed to apply ILP to all minimal incoherence-preserving subsets (MIPS). Although this algorithm can often find a minimal number of axioms to remove, it is very time-consuming to compute MIPS. Thus, an adapted revision algorithm to deal with unsatisfiable concepts individually is also given. Experimental results reveal that the proposed ILP-based revision algorithm is much more efficient than the commonly used algorithm based on the hitting set tree. In addition, the adapted algorithm can achieve higher efficiency, while it may delete more axioms.

Key words: ontology revision; inconsistency handling; semantic web; integer linear programming

DOI: 10.3969/j.issn.1003-7985.2020.01.001

With the standardization of web ontology language (OWL)^[1] and rapid development of knowledge graphs^[2], a large number of ontologies are available online and often evolve over time in many real-world applications^[3]. For instance, in the fields of ontology-driven knowledge management and e-commerce, ontologies may keep growing or changing with the development of a knowledge base and the modification of productions. When new information is received, an original ontology needs to be revised for incorporating newly received in-

formation consistently. Generally, logical inconsistency consists of inconsistency and incoherence. An ontology is inconsistent if and only if it has no model; i. e., it is inconsistent in the first-order sense. An ontology is incoherent if and only if there is some unsatisfiable concept which is interpreted as an empty set. As reasoning with inconsistent ontologies produces meaningless results, handling inconsistencies in OWL ontologies is an important task. In this paper, the attention is focused on revising incoherent OWL ontologies since inconsistency is always caused by adding instances of unsatisfiable concepts or relations to an ontology.

So far, researchers have proposed various approaches for revising OWL ontologies by deleting some axioms. Qi et al.^[4] proposed a kernel revision operator based on the well-known AGM framework^[5] to deal with incoherence in description logic (DL) ontologies, where DL is the underlying semantics of OWL. They provided practical algorithms based on the hitting set tree algorithm by considering the weights or frequency of axioms. Similarly, Golbeck et al.^[6] also adopted the kernel revision operator and incision functions, but their goal is to deal with inconsistency by considering trust information. Due to the issue of computational efficiency^[7], such works often focus on theoretical work and lack efficient algorithms. To improve efficiency, Fu et al.^[8] proposed a graph-based approach to revising DL-Lite ontologies by transforming an ontology to graphs and then locating incoherence. Besides, many researchers also defined fine-grained revision operators to only remove or modify part of an axiom. Qi and Du^[9] proposed to revise DL terminologies through adapting the Dalal revision operator defined in propositional logic and using the notion of concept forgetting. Zhuang et al.^[7,10] defined a new semantics, called type semantics, instead of standard DL semantics to revise DL-Lite TBoxes. One recent work defined a typicality-based revision of an ALC knowledge base to deal with exceptions by adding explicit exceptions to weaken inconsistencies^[11].

From the claims above, it can be seen that many approaches are designed for less expressive DLs like DL-Lite and ALC, especially those fine-grained ones, while they seldom consider (partly) priority information. How-

Received 2019-09-22, **Revised** 2019-12-20.

Biography: Ji Qiu (1980—), female, doctor, lecturer, qiuji@njupt.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 61602259, U1736204), Research Foundation for Advanced Talents of Nanjing University of Posts and Telecommunications (No. NY216022), the National Key Research and Development Program of China (No. 2018YFC0830200).

Citation: Ji Qiu, Qi Guilin, Boutouhami Khaoula. Revision of stratified OWL ontologies based on integer linear programming[J]. Journal of Southeast University (English Edition), 2020, 36(1): 1–7. DOI: 10.3969/j.issn.1003-7985.2020.01.001.

ever, in many cases like ontology merging or user-defined priority, a partial order is usually available and these approaches cannot be applied directly^[12]. For example, two source ontologies and their mappings may be generated by different tools and thus weights associated with axioms in different ontologies or mappings are incomparable. In addition, many knowledge bases have been constructed and their fusion may also face the problem of partial order^[13]. More importantly, a partial order provides an easy and useful way for users to manually assign different priorities to axioms.

In this paper, a kernel revision operator for partially ordered OWL ontologies is defined and efficient algorithms based on integer linear programming (ILP) are proposed. Note that, ILP is an optimization problem of maximizing or minimizing a linear objective function. Due to their high efficiency, ILP-based models have been widely used in many tasks in the semantic web such as in fuzzy ontology reasoning^[14], ontology matching^[15] and knowledge base embeddings^[16]. In particular, our recent work^[17] applied ILP to resolve logical contradictions in single ontologies, which is different from the current work because the latter is more general and considers stratified and multiple ontologies. It is challenging to redefine ILP-based models for revising stratified ontologies. In the experiments, a comprehensive evaluation is conducted over various ontologies to show the high efficiency of applying our ILP-based model. The efficiency and effectiveness of our adapted algorithm are also analyzed for stratified ontologies.

1 Preliminaries

In this section, key notions of logical inconsistency and integer linear programming are provided and they are used to revise ontologies. The readers are assumed to be familiar with description logics (DLs)^[18].

1.1 Logical inconsistency

To deal with incoherence in DL ontologies, minimal sets of axioms for explaining logical contradictions are often required to be computed first and then a solution of axioms can be generated such that removing these axioms can resolve the contradictions. In the following, the key concepts of such minimal sets are provided.

Definition 1 (MUPS)^[12] Let U be an unsatisfiable concept in an ontology O . A sub-ontology O' of O is a minimal unsatisfiability-preserving sub-ontology (MUPS) w. r. t. U if U is unsatisfiable in O' and satisfiable in every sub-ontology $O'' \subset O'$.

A MUPS of O w. r. t. U is a minimal sub-ontology of O in which U is unsatisfiable. MUPS are useful to relate sets of axioms with unsatisfiability of specific concepts. We use MUPS to indicate a MUPS or multiple ones for conciseness. To relate sets of axioms to the incoherence

of an ontology in general, a minimal incoherence-preserving sub-ontology is defined.

Definition 2 (MIPS) Let O be an incoherent ontology. An ontology $O' \subseteq O$ is a minimal incoherence-preserving sub-ontology (MIPS) of O if O' is incoherent and every sub-ontology $O'' \subset O'$ is coherent.

A MIPS of O is a minimal incoherent sub-ontology of O . Note that a MIPS must be a MUPS, but not vice versa. In the following, both the original ontology and the newly arrived one are assumed to be individually coherent and consistent but their merging produces incoherence.

1.2 Integer linear programming

Linear programming formulates our real-life problem into a mathematical model. A linear program is an optimization problem of maximizing or minimizing a linear objective function of variables that are subject to a set of constraints expressed as linear equations or inequations^[17]. An integer program forces the variables in a linear program to be integers and is referred to integer linear programming (ILP)^[19].

To resolve inconsistency by deleting a set of selected axioms, pure 0-1 ILP is used and its variables can only have a value of 0 or 1. If an axiom should be deleted, its associated variable should have a value of 1, and 0 otherwise. Specifically, given n variables and m linear constraints, the problem of 0-1 ILP is to find an assignment of either 0 or 1 for the variables such that all constraints are satisfied. A 0-1 ILP optimization problem can be stated in the standard form^[19]:

$$\min/\max z = \sum_{j=1}^n w_j x_j \quad (1a)$$

$$\text{s. t. } c_i: \sum_{j=1}^n a_{ij} x_j \geq b \quad (1b)$$

$$x_j \in \{0, 1\} \quad (1c)$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. w_j and a_{ij} are coefficients and b is an integer. The values of w_j , a_{ij} and b are determined by specific application scenarios.

The objective function (1a) describes a criterion (or a measure) that we wish to minimize (e. g., cost) or maximize (e. g., profit). The limitations that restrict our choices for decision variables are described using mathematical constraints (1b). In many practical problems, people's decisions are often restricted by the limitations like resource limitations and physical, strategic or economical constraints. It is noted that the complexity of solving an ILP problem is NP-complete and it depends on the number of variables^[20].

2 Kernel Revision Operator

Inspired by our previous work^[12], the general case of an original ontology O is considered as $O = O_s \cup O_r$, where O_s and O_r indicate the stable part and removable

part, respectively. The main difference between their stratification and ours is that axioms in O_s are assumed to be more reliable than those in O_T and both of them are removable. We try to remove those less reliable axioms first. In this way, users can freely assign different priorities to axioms by available or user-defined priorities.

Partially ordered ontologies can be formally defined as below.

Definition 3 (partial ordered ontologies)^[12] A partially ordered ontology is a pair (O, \leq) , where \leq is a partial order over axioms in O and satisfies the condition: For any axiom $\phi \in O_s$, there exists $\psi < \phi$ for all $\psi \in O_T$, where $\psi < \phi$ denotes $\psi \leq \phi$ but $\phi \not\leq \psi$.

That is, any axiom in O_s should be preferred to all axioms in O_T . When $\psi < \phi$, this means that ϕ is preferred to ψ w. r. t. \leq . In addition, $\psi \approx \phi$ is used to denote $\psi \leq \phi$ and $\phi \leq \psi$, which indicates that ψ is equal to ϕ w. r. t. \leq .

Definition 4 (conflict)^[4] Let O and O_n be two coherent ontologies. A conflict M of O w. r. t. O_n is a sub-ontology of O and satisfies that $M \cup O_n$ is incoherent; and $\forall M' \subset M$, $M' \cup O_n$ is coherent.

This definition is similar to the notion of a minimal axiom set consisting of a static part and a rebuttal part^[21], which can be considered as a kernel. If O_n is empty, a conflict becomes a MUPS or MIPS as defined in Section 1.1.

Definition 5 (conflict stratification) Assume that there is a partially ordered OWL ontology (O, \leq) , where $O = O_s \cup O_T$ is incoherent w. r. t. a newcomer ontology O_n . Given a conflict M of O w. r. t. O_n , a simple stratification of M is a partition $(\underline{M}, \overline{M})$, where $\underline{M} = \{\phi \in M \cap O_T : \nexists \psi \in M, \psi < \phi\}$ if $\overline{M} \cap O_T \neq \emptyset$ (\emptyset is an empty set) and $\underline{M} = \{\phi \in M : \nexists \psi \in M, \psi < \phi\}$ otherwise. Besides, $\overline{M} = M \setminus \underline{M}$.

In this definition, \underline{M} and \overline{M} are called the lower and upper stratum of M , respectively. If some axioms in M belong to O_T , all axioms in \underline{M} should belong to O_T and have the least priority w. r. t. \leq . Otherwise, \underline{M} consists of those axioms in M that have the least priority w. r. t. \leq . Different from the previous definition of stratification^[12], \underline{M} is allowed to contain axioms from O_s as a general consideration. Namely, all axioms in O are removable but have different priorities. According to Definition 5, $\underline{M} \neq \emptyset$ can be obtained.

Inspired by the work of kernel revision^[4], an incision function is defined below to select axioms from each conflict for removal.

Definition 6 (incision function) Assume that there is a partially ordered ontology (O, \leq) , where $O = O_s \cup O_T$ is incoherent w. r. t. a newly coming ontology O_n . Given a set of conflicts $F_{O_n}(O)$ of O w. r. t. O_n , an incision function σ for O can be defined as a function $(\sigma: 2^{2^O} \rightarrow 2^O)$ such that

Condition 1 $\sigma(F_{O_n}(O)) \subseteq \bigcup_{M \in F} M$, $F = F_{O_n}(O)$;

Condition 2 If $M \in F_{O_n}(O)$, then $M \cap \sigma(F_{O_n}(O)) \neq \emptyset$.

This function means only axioms from a conflict can be selected (see Condition 1) and at least one axiom should be selected from each conflict (see Condition 2). Obviously, this function is an incision function defined by Qi et al^[4]. Note that, if $F_{O_n}(O)$ contains all the MIPS of O , removing the axioms in $\sigma(F_{O_n}(O)(O))$ from O can make O coherent.

Based on an incision function, a revision operator can be defined for incorporating a new ontology consistently-based on all the MIPS.

Definition 7 (kernel revision operator)^[4] Let O be an ontology to be revised and σ be an incision function for O . The kernel revision operator \otimes_σ for O is defined as follows. For each newly received ontology O_n ,

$$O \otimes_\sigma O_n = (O \setminus \sigma(\text{MIPS}_{O_n}(O))) \cup O_n$$

According to the operator, the resulting ontology becomes coherent after removing those axioms selected by the incision function.

Algorithm 1 Incision function based on ILP

Input: $O = O_s \cup O_T$, O_n and $F_{O_n}(O)$ w. r. t. O_n .

Output: A set of axioms to remove

```

1   $S_{\text{union}} = \bigcup_{M \in F} M$ ,  $F = F_{O_n}(O)$ 
2   $X = \{x_j \mid \phi_j \in S_{\text{union}}, j = 1, 2, \dots, |S_{\text{union}}|\}$ 
3   $z = \sum_{x \in X} x$ 
4  for  $(M \in F_{O_n}(O))$ 
5      if  $(M \cap O_T \neq \emptyset)$  then  $\underline{M} = \{\phi \in M \cap O_T : \nexists \psi \in M, \psi < \phi\}$ 
6      else  $\underline{M} = \{\phi \in M : \nexists \psi \in M, \psi < \phi\}$ 
7   $X_m = \{x_j \mid \phi_j \in \underline{M}, x_j \in X\}$ 
8   $c_i = (\sum_{x \in X_m} x) \geq 1$ 
9   $C = C \cup \{c_i\}$ 
10 }
11  $S_{\text{assi}} = \text{ILP\_Solver}(z, C, \min)$ 
12  $\sigma_1(F_{O_n}(O)) = \{\phi_j \mid (x_j = 1) \in S_{\text{assi}}\}$ 
13 return  $\sigma_1(F_{O_n}(O))$ 
```

To define a specific incision function, the ILP model is applied instead of traditional methods using the hitting set tree algorithm^[4]. We use the ILP model due to its high efficiency and flexible framework to select axioms and minimize the number of selected axioms. Algorithm 1 defines such an incision function based on ILP. It removes as few axioms as possible and chooses axioms from the lower stratum first. This algorithm consists of the following three parts:

First, an object function can be constructed. In Algorithm 1, a decision variable x_j is first associated with each axiom ϕ_j in the union of all given conflicts (i. e., S_{union}) since only axioms appearing in conflicts can be removed (see Lines 1 and 2), where j indicates the position of an

axiom in S_{union} . To define an objective function z (see Eq. (1a)), the coefficient of each variable is set to be 1 (see Line 3) and all axioms are assumed to be equally important. In this way, minimizing z is actually to minimize the number of axioms to be removed.

Secondly, constraints can be constructed. A constraint is constructed for each conflict (see Lines 4 to 10). Specifically, for each conflict M , \underline{M} is first computed (see Lines 5 and 6) according to Definition 5. Then, the variables corresponding to axioms in \underline{M} are obtained and their sum is set to be no less than 1 (see Lines 7 and 8). That is, all variables associated to axioms in M are assigned to be 0 and at least one variable in X_M will be assigned to be 1. This is to ensure that at least one axiom from \underline{M} can be included in the final solution.

Thirdly, a solution can be computed. Based on objective function z and constraints C , the function ILP_Solver (z, C, \min) is utilized to apply a traditional ILP solver like Cplex (<https://www.ibm.com/analytics/cplex-optimizer>) for finding an optimal assignment to all variables under constructed constraints (see Line 11). The parameter \min indicates minimizing objective function z . Finally, a solution of axioms is found and each variable of such an axiom has a value of 1 in the found assignment (see Line 12).

Proposition 1 Algorithm 1 computes an incision function σ_1 .

Proof We prove that σ_1 is an incision function according to the conditions given in Definition 6.

From Algorithm 1, it can be seen that $\sigma_1(F_{O_s}(O))$ is obtained according to the assignment S_{assi} , each of whose elements corresponds to an axiom in the union of $F_{O_s}(O)$. Clearly, $\sigma_1(F_{O_s}(O)) \subseteq \bigcup_{M \in F} M$, $F = F_{O_s}(O)$.

According to the definition of \underline{M} (see lines 5 and 6), we know that \underline{M} is not empty if M is not empty. Besides, the constructed constraint c_i requires that at least one of these variables should be assigned to be 1. If such an assignment can be found by applying an ILP solver, it means that each constraint c_i has been satisfied and at least one axiom from \underline{M} is selected. Namely, $\sigma_1(F_{O_s}(O)) \cap M \neq \emptyset$.

Based on a specific incision function, a straightforward revision algorithm (see Algorithm 2) can be proposed to apply the function directly on all MIPS as shown in Definition 7. In our work, the incision function defined in Algorithm 1 is used. Besides, the commonly used method is adopted to compute MIPS based on all MUPS of all unsatisfiable concepts (see Lines 3 to 11).

Algorithm 2 Algorithm to revise a stratified ontology

Input: $O = O_s \cup O_T$ and O_n .

Output: A revised coherent ontology.

- 1 $M = \emptyset$
- 2 $M_{\text{MIPS}} = \emptyset$
- 3 $U_{\text{all}} = \text{All unsatisfiable concepts in } O \text{ w. r. t. } O_n$

- 4 for ($U \in U_{\text{all}}$) {
- 5 $M_U = \text{All MUPS of } O \text{ w. r. t. } O_n \text{ and } U$
- 6 $M = M \cup M_U$
- 7 }
- 8 for ($M_1 \in M$) {
- 9 if ($\nexists M_2 \in M$ such that $M_2 \subset M_1$) then
- 10 $M_{\text{MIPS}} = M_{\text{MIPS}} \cup \{M_1\}$
- 11 }
- 12 $O \otimes_{\sigma}^1 O_n = (O \setminus \sigma(M_{\text{MIPS}})) \cup O_n$
- 13 return $O \otimes_{\sigma}^1 O_n$

Since computing MIPS is often very expensive, especially for those large ontologies or ontologies containing many conflicts, an adapted revision algorithm (see Algorithm 3) is provided to handle unsatisfiable concepts one by one. For each unsatisfiable concept U , it is necessary to check whether it is still unsatisfiable in the current ontology O which may have been changed (see Line 4). If U is still unsatisfiable, all MUPS in O w. r. t. O_n and U need to be computed (see Line 5) and then an incision function is used to select axioms from these MUPS. Finally, ontology O becomes coherent w. r. t. O_n after removing all selected axioms.

Algorithm 3 Adapted algorithm to revise a stratified ontology

Input: $O = O_s \cup O_T$ and O_n .

Output: A revised coherent ontology.

- 1 $S = \emptyset$
- 2 $U_{\text{all}} = \text{All unsatisfiable concepts in } O \text{ w. r. t. } O_n$
- 3 for ($U \in U_{\text{all}}$) {
- 4 if ($(O \cup O_n \models U \subseteq \perp)$) then {
- 5 $M_U = \text{All MUPS of } O \text{ w. r. t. } O_n \text{ and } U$
- 6 $S = S \cup \sigma(M_U)$
- 7 $O = O \setminus S$
- 8 }
- 9 }
- 10 $O \otimes_{\sigma}^2 O_n = O \cup O_n$
- 11 return $O \otimes_{\sigma}^2 O_n$

3 Experimental Results

Our evaluation is performed on a laptop with 2.4 GHz Intel(R) Core(TM)2 Duo CPU and 8.0 GB of RAM using 64 bit operating system Windows 7. The maximum heap space is set to be 6 GB. The relevance-based ontology debugging algorithm^[22] is adopted to compute conflicts.

3.1 ILP-based algorithm vs. HST-based one

First of all, to see the efficiency of applying ILP, our ILP-based algorithm (namely Algorithm 2 with Algorithm 1 as an incision function, marked as Alg2) is compared with a commonly used algorithm based on the hitting set tree (HST)^[4] (marked as Hst). Hst applies the HST algorithm directly on the conflicts to be resolved and satisfies the minimal change principle. We do not compare with some other efficient algorithms for revising

OWL ontologies, such as graph-based or new semantics-based ones, since they are often designed for a specific DL ontology.

The data set consists of 6 ontologies with different numbers (i. e., from 10 to 15) of MUPS that do not share any axioms and each MUPS includes 7 axioms. Such a construction generating MUPS without overlapping axioms is more challenging to handle than those with overlapping axioms. In this way, each MUPS is also a MIPS. To concentrate on showing the efficiency of using ILP or HST, repairing single ontologies is also considered by setting $O_n = \emptyset$ and all axioms in the ontologies are removable.

Fig. 1 presents the experimental results, where the number in x -axis indicates the number of MIPS. In this figure, only the revision time is shown without the time to compute MIPS since both algorithms worked on the same sets of MIPS. From the figure, it can be observed that, Alg2 performs much better than Hst and often spends no more than 300 ms to finish each revision process. As for Hst, its revision time increases quickly when the number of MIPS increases. For example, it spends 13 and 919 s for the ontologies including 10 and 12 MIPS, respectively. For those ontologies containing more than 12 MIPS, Hst failed to finish each revision process within the limited time (i. e., 1 000 s). This is because constructing a HST is very time-consuming, especially when the MIPS are uncorrelated. ILP provides a mathematical method and can achieve very high efficiency.

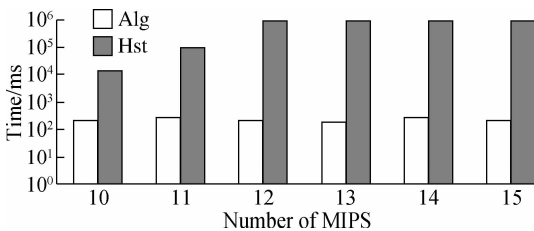


Fig. 1 Time to find a solution to revise an artificial ontology

3.2 ILP-based algorithm vs. adapted one

In this experiment, our two revision algorithms are evaluated based on the well-known incoherent ontologies km1, km2, km3 and km4. These ontologies were originally generated by using ontology learning algorithms. Both algorithms (namely Algorithm 2 and Algorithm 3) take Algorithm 1 as an incision function and are marked as Alg2 and Alg3, respectively. Since this ontology contains too many unsatisfiable concepts (i. e., 1 375), several sub-ontologies are extracted for testing.

Tab. 1 provides the details about the extracted ontologies and the experimental results, where UC indicates unsatisfiable concepts and Time out means that a process cannot be finished within 70 min. Time in the last two

columns consists of the time to compute the MUPS and the time to revise an ontology.

From Tab. 1, it can be clearly observed that Alg3 is much more efficient than Alg2 since computing MIPS is very time-consuming. At the same time, it can be seen that Alg3 may remove more axioms for resolving inconsistency. Take ontology km3 as an example. It originally contained 1 800 axioms and the newly arrived ontology contained 300 axioms. The merging of the two ontologies generates 49 unsatisfiable concepts. To revise the ontology, Alg2 spent 3 679 s and removed 2 axioms while Alg3 spent 314 s and removed 8 axioms.

Tab. 1 Experimental results of finding a solution for each km ontology

O	$ O $	$ O_n $	#UC	#Removed axioms		Time/ms	
				Alg2	Alg3	Alg2	Alg3
km1	800	100	1	1	1	5 154	4 360
km2	1 800	200	46	2	8	2 230 675	289 768
km3	1 800	300	49	2	8	3 679 010	314 057
km4	1 800	400	243	Unknown	33	Time out	600 621

3.3 Results over stratified ontologies

The third experiment tests the efficiency and effectiveness of our algorithm to revise stratified ontologies, where the effectiveness takes into consideration of the number of removed axioms to revise ontologies. We did this experiment by using Alg3 due to its higher efficiency and choosing km2 as a test ontology. Similar observations can be obtained by using Alg2 or choosing other ontologies. Specifically, the original ontology of km2 is stratified by randomly choosing a given number of axioms as a stable part and the remaining axioms are considered as the removable part. Tab. 2 shows the number of the stable and removable parts for each test data set, where time represents the time to compute MUPS and do revision.

Tab. 2 Experimental results for each stratified ontology by applying Alg3

Number of UC	$ O_n $	$ O_s $	$ O_T $	Number of removed axioms	Time/ms
46	200	200	1 600	8	277 719
		400	1 400	15	598 596
		600	1 200	10	245 875
		800	1 000	3	74 219
		1 000	800	3	71 940

From the table, it can be seen that, when the size of stable axioms is 800 or 1 000, Alg3 only spent about 70 s to finish the whole process and removed very few axioms (i. e., 3 axioms). In other cases, more time was spent and more axioms needed to be removed. This is caused by the axioms in the stable part (i. e., the axioms in O_s). For conflict M , as our goal is to remove axioms in its removable part first, some of the axioms in the stable

part will be selected for removing only if it has no intersections with the removable part. Thus, it may occur that an axiom with a high frequency in MUPS belongs to the stable part and more axioms in the removable part need to be removed for resolving inconsistency. The cases of $|O_s| < 800$ belong to such examples.

4 Conclusions

1) A novel definition of conflict stratification is provided to introduce a more general model for ontology revision, and thus users can freely assign different priorities to axioms.

2) A specific incision function based on ILP is defined for stratified ontologies and two algorithms are proposed to revise stratified ontologies.

3) To evaluate the performance of our algorithms, comprehensive experiments have been conducted to measure the efficiency and the number of removed axioms.

4) The experimental results show that the proposed ILP-based model is much more efficient than the HST-based one; and the adapted algorithm outperforms the one based on MIPS w. r. t. efficiency, while it may delete more axioms. It can also be observed that the performance of the adapted algorithm to revise stratified ontologies is influenced largely by the stratification of axioms.

References

- [1] Grau C B, Horrocks I, Motik B, et al. OWL 2: The next step for OWL [J]. *Journal of Web Semantics*, 2008, **6** (4): 309 – 322. DOI: 10. 1016/j. websem. 2008.05.001.
- [2] Ji Q, Qi G L, Gao H, et al. Survey on schema induction from knowledge graphs [C]//*China Conference of Knowledge Graph and Semantic Computing*. Tianjin, China, 2018:136 – 142.
- [3] Zablith F, Antoniou G, d’Aquin M, et al. Ontology evolution: A process-centric survey[J]. *The Knowledge Engineering Review*, 2015, **30** (1): 45 – 75. DOI: 10. 1017/s0269888913000349.
- [4] Qi G L, Haase P, Huang Z S, et al. A kernel revision operator for terminologies—algorithms and evaluation [C]// *7th International Semantic Web Conference*. Karlsruhe, Germany, 2008: 419 – 434.
- [5] Ribeiro M M, Wassermann R, Flouris G, et al. Minimal change: Relevance and recovery revisited[J]. *Artificial Intelligence*, 2013, **201**: 59 – 80. DOI: 10. 1016/j. artint. 2013.06.001.
- [6] Golbeck J, Halaschek-Wiener C. Trust-based revision for expressive web syndication [J]. *Journal of Logic and Computation*, 2009, **19** (5): 771 – 790. DOI:10. 1093/ logcom/ exn045.
- [7] Zhuang Z Q, Wang Z, Wang K W, et al. DL-Lite contraction and revision[J]. *Journal of Artificial Intelligence Research*, 2016, **56**: 329 – 378. DOI: 10. 1613/jair. 5050.
- [8] Fu X F, Qi G L, Zhang Y, et al. Graph-based approaches to debugging and revision of terminologies in DL-Lite[J]. *Knowledge-Based Systems*, 2016, **100**: 1 – 12. DOI:10. 1016/j. knosys. 2016. 01. 039.
- [9] Qi G L, Du J F. Model-based revision operators for terminologies in description logics [C]// *21st International Joint Conference on Artificial Intelligence*. Pasadena, CA, USA, 2009: 891 – 897.
- [10] Wang Z, Wang K W, Zhuang Z Q, et al. Instance-driven ontology evolution in DL-Lite [C]// *29th AAAI Conference on Artificial Intelligence*. Austin, TX, USA, 2015: 1656 – 1662.
- [11] Micalizio R, Pozzato G L. Revision of ontologies to accommodate exceptions: A typicality-based approach[J]. *Fundamenta Informaticae*, 2018, **161** (1/2): 163 – 189. DOI:10. 3233/fi-2018-1699.
- [12] Ji Q, Gao Z Q, Huang Z S. Conflict resolution in partially ordered OWL DL ontologies [C]// *21st European Conference on Artificial Intelligence*. Prague, Czech Republic, 2014: 471 – 476.
- [13] Dong X L, Gabrilovich E, Heitz G, et al. From data fusion to knowledge fusion[J]. *Proceedings of the VLDB Endowment*, 2014, **7** (10): 881 – 892. DOI:10. 14778/ 2732951. 2732962.
- [14] Bobillo F, Straccia U. Reducing the size of the optimization problems in fuzzy ontology reasoning [C]//*11th International Workshop on Uncertainty Reasoning for the Semantic Web*. Bethlehem, USA, 2015: 54 – 59.
- [15] Niepert M, Meilicke C, Stuckenschmidt H. A probabilistic-logical framework for ontology matching [C]// *24th AAAI Conference on Artificial Intelligence*. Atlanta, GA, USA, 2010: 1413 – 1418.
- [16] Wang Q, Wang B, Guo L, et al. Knowledge base completion using embeddings and rules [C]// *24th International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina, 2015: 1859 – 1866.
- [17] Ji Q, Boutouhami K, Qi G L. Resolving logical contradictions in description logic ontologies based on integer linear programming[J]. *IEEE Access*, 2019, **7**: 71500 – 71510. DOI:10. 1109/access. 2019. 2919498.
- [18] Baader F, Calvanese D, McGuinness D L, et al. *The description logic handbook* [M]. Cambridge: Cambridge University Press, 2007. DOI: 10. 1017/ cbo97805117111787.
- [19] Li H Y, Shao C P, Wang Z. Detecting fault injection attacks based on compressed sensing and integer linear programming[J]. *IEEE Transactions on Dependable and Secure Computing*, 2019, **16** (3): 476 – 483. DOI:10. 1109/tdsc. 2018. 2821140.
- [20] Karp R M. Reducibility among combinatorial problems [C]// *A symposium on the Complexity of Computer Computations*. Yorktown Heights, NY, USA, 1972: 85 – 103.
- [21] Baader F, Penaloza R, Suntisrivaraporn B. Pinpointing in the description logic EL + [C]// *30th Annual German Conference on AI*. Osnabrück, Germany, 2007: 52 – 67.
- [22] Ji Q, Qi G L, Haase P. A relevance-directed algorithm for finding justifications of DL entailments [C]// *4th Annual Asian Semantic Web Conference*. Shanghai, China, 2009: 306 – 320.

基于整数线性规划的分层 OWL 本体修正

季 秋¹ 漆桂林^{2,3} Boutouhami Khaoula^{2,3}

(¹南京邮电大学现代邮政学院, 南京 210023)

(²东南大学计算机科学与工程学院, 南京 211189)

(³东南大学教育部计算机网络与信息整合重点实验室, 南京 211189)

摘要:为了修正分层 OWL(web ontology language)本体,通过定义新的冲突分层和基于整数线性规划(ILP)的切割函数扩展了核修正算子. 基于 ILP 的模型考虑了最小化线性目标函数的优化问题,适合于修正本体时移除最少数量的公理. 基于该切割函数,提出了一个修正算法,将 ILP 应用到所有最小的不协调保持子集(MIPS)上. 该算法虽然能够经常找到用于移除的最少公理,但计算 MIPS 非常耗时. 因此,又提出另一个改进的修正算法用于逐个处理不可满足概念. 实验结果表明:提出的基于 ILP 的修正算法比经常使用的基于碰集树的算法更加高效;改进的修正算法能够达到更高的效率,但可能会删除更多的公理.

关键词:本体修正;不一致性处理;语义网;整数线性规划

中图分类号:TP181