Towards Defeasible Mappings for Tractable Description Logics

now as Software Development Engineer at Amazon

Kunal Sengupta

Pascal Hitzler

Data Semantics Laboratory

Wright State University

http://daselab.org

Semantic Heterogeneity

$$Veg \sqcap NonVeg \sqsubseteq \bot \quad (5.1)$$
$$\exists consumes.EggFood \sqsubseteq NonVeg \quad (5.2)$$
$$consumes \circ contains \sqsubseteq consumes \quad (5.3)$$
$$\{juliet\} \sqsubseteq Veg \quad (5.4)$$

$$\{romeo\} \sqsubseteq Eggetarian \quad (5.5)$$
$$Eggetarian \sqsubseteq Vegetarian \quad (5.6)$$
$$Eggetarian \sqsubseteq \exists eats.Egg \quad (5.7)$$
$$Eggetarian \sqcap NonVegetarian \sqsubseteq \bot \quad (5.8)$$
$$\{caesar\} \sqsubseteq Vegetarian \quad (5.9)$$
$$\{caesar\} \sqsubseteq NotEggetarian \quad (5.10)$$
$$NotEggetarian \sqcap Eggetarian \sqsubseteq \bot \quad (5.11)$$

Differences in terminology:

Left:
Vegetarians don’t eat eggs.

Right:
Some Vegetarians eat eggs.

Requires a complex mapping,
i.e. manual mapping
Which is very costly.

Semantic Heterogeneity

$$Veg \sqcap NonVeg \sqsubseteq \bot \quad (5.1)$$
$$\exists consumes.EggFood \sqsubseteq NonVeg \quad (5.2)$$
$$consumes \circ contains \sqsubseteq consumes \quad (5.3)$$
$$\{juliet\} \sqsubseteq Veg \quad (5.4)$$

$$\{romeo\} \sqsubseteq Eggetarian \quad (5.5)$$
$$Eggetarian \sqsubseteq Vegetarian \quad (5.6)$$
$$Eggetarian \sqsubseteq \exists eats.Egg \quad (5.7)$$
$$Eggetarian \sqcap NonVegetarian \sqsubseteq \bot \quad (5.8)$$
$$\{caesar\} \sqsubseteq Vegetarian \quad (5.9)$$
$$\{caesar\} \sqsubseteq NotEggetarian \quad (5.10)$$
$$NotEggetarian \sqcap Eggetarian \sqsubseteq \bot \quad (5.11)$$

Automated mapping leads to inconsistencies:

$$Vegetarian \equiv Veg$$
$$NonVeg \equiv NonVegetarian$$
$$EggFood \equiv Egg$$
$$eats \sqsubseteq consumes$$

Can we do better via auto-repair?

Semantic Heterogeneity

$$Veg \sqcap NonVeg \sqsubseteq \bot \quad (5.1)$$
$$\exists consumes.EggFood \sqsubseteq NonVeg \quad (5.2)$$
$$consumes \circ contains \sqsubseteq consumes \quad (5.3)$$
$$\{juliet\} \sqsubseteq Veg \quad (5.4)$$

$$\{romeo\} \sqsubseteq Eggetarian \quad (5.5)$$
$$Eggetarian \sqsubseteq Vegetarian \quad (5.6)$$
$$Eggetarian \sqsubseteq \exists eats.Egg \quad (5.7)$$
$$Eggetarian \sqcap NonVegetarian \sqsubseteq \bot \quad (5.8)$$
$$\{caesar\} \sqsubseteq Vegetarian \quad (5.9)$$
$$\{caesar\} \sqsubseteq NotEggetarian \quad (5.10)$$
$$NotEggetarian \sqcap Eggetarian \sqsubseteq \bot \quad (5.11)$$

Axiom removal is not fine-grained enough:

~~$Vegetarian \equiv Veg$~~

$$NonVeg \equiv NonVegetarian$$
$$EggFood \equiv Egg$$
$$eats \sqsubseteq consumes$$

Can't distinguish between romeo and caesar, resulting in loss of desired consequences.

Suggested approach

Idea:

> **A subclass axiom should only propagate an instance if this does not lead to a logical inconsistency.**

This is essentially the driving idea behind Reiter's Default Logic and its variants.

Sounds straightforward, but the details are tricky.

E.g. it was shown in 1995 by Baader and Hollunder that a specific extension of ALC with Reiter defaults is undecidable.

We don't even know yet whether or not ALC with Reiter defaults (without the extension) is decidable.

What is the problem?

- **In description logics, you want to be able to reason with unknowns.**

First ontology:

$$\{john\} \sqsubseteq USCitizen \qquad \{john\} \sqsubseteq Traveler$$

$$USCitizen \sqsubseteq \exists hasPassport.USPassport,$$

Second ontology:

$$Tourist \sqsubseteq \exists hasPP.Passport$$

$$\exists hasPP.AmericanPassport \sqsubseteq EuVisaNotRequired,$$

John has a passport but we don't have the instance.
We still infer he does not need a visa.

What is the problem?

We do not know how many unknowns (in the example, passports) we need deal with for reasoning. Potentially, we may need infinitely many.

Description logics are designed such that, if infinitely many are needed, they become repetitive, such that it's enough to look at a finite number.

However, with defaults, we get unknowns which are mapped, and unknowns which are not mapped, and we don't know which are which.

It is currently not known if (for ALC) it suffices to look at a finite number.

Solution

For OWL EL, it can be shown that very few unknowns suffice, and we know how to create them up front.

So we have only a finite set of unknowns for which to decide which need to be mapped and which need not be mapped.

That this actually works, needs of course formal definitions and proofs.

But they cannot be presented in a 15-minute talk.

[Side condition: we assume unidirectional mapping.]

Evaluation

We qualitatively evaluated our approach, in comparison to

1. Repair (by mapping removal, using Protégé explanations)

2. Paraconsistent (i.e. inconsistency-tolerant) reasoning as in [Maier et al. 2013]

Re. 1., as expected, we are loosing desired consequences (see vegetarian example above).

Paraconsistent reasoning

$\mathcal{O}_1$

$$Male(david)$$
$$Male(jacob)$$
$$Male(mark)$$
$$Male(mike)$$
$$Female(jane)$$
$$Female(julie)$$
$$hasSpouse(david, mike)$$
$$hasSpouse(jacob, jane)$$

$\mathcal{O}_2$

$$Famale \sqcap Male \sqsubseteq \bot$$
$$\exists hasSpouse.Female \sqsubseteq Male$$
$$\exists hasSpouse.Male \sqsubseteq Female$$
$$Male(john)$$
$$Female(mary)$$
$$hasSpouse(john, mary)$$

With the obvious mappings of Male to Male, Female to Female, hasSpouse to hasSpouse.

Ontology/System	Query	Instances
Default	$\mathcal{O}_2$:*hasSpouse(?,?)*	{(*john, mary*), (*jacob,jane*), (*mark, julie*)}
Default	$\mathcal{O}_2$:*Male(?)*	{*mark, jacob, john, mike*}
Default	$\mathcal{O}_2$:*Female(?)*	{*mary, julie, jane*}
Paraconsistent	$\mathcal{O}_2$:*hasSpouse(?,?)*	{(*john, mary*), (*jacob,jane*), (*mark, julie*), (*david, mike*)}
Paraconsistent	$\mathcal{O}_2$:*Male(?)*	{*mark, jacob, john, mike*}
Paraconsistent	$\mathcal{O}_2$:*Female(?)*	{*mary, julie, jane*}

Conclusions

DaSe Lab

- Within the limited scenario (OWL EL, unidirectional mappings), the approach works better than others, without manual intervention.
- However, runtime performance is an issue. We did not attempt an efficient implementation, as a naïve algorithm would be exponential, and we don't have a better one at this stage.
- It is not clear whether the approach can be carried over to description logics outside the EL family.

References

- Kunal Sengupta, Pascal Hitzler, Towards Defeasible Mappings for Tractable Description Logics. In: Proceedings ISWC2015.
- Kunal Sengupta, Pascal Hitzler, Krzysztof Janowicz, Revisiting default description logics – and their role in aligning ontologies. In: T. Supnithi, T. Yamaguchi, J.Z. Pan, V. Wuwongse, M. Buranarach (eds.), Semantic Technology, 4th Joint International Conference, JIST 2014, Chiang Mai, Thailand, November 9-11, 2014. Revised Selected Papers. Lecture Notes in Computer Science, Vol. 8943, Springer, Heidelberg, 2015, pp. 3-18.
- Kunal Sengupta, Adila Krisnadhi, Pascal Hitzler, Local Closed World Reasoning: Grounded Circumscription for OWL. In: L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. F. Noy, E. Blomqvist (Eds.): The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I. Lecture Notes in Computer Science Vol. 7031, Springer, Heidelberg, 2011, pp. 617-632.

References

- Kunal Sengupta, A Language for Inconsistency-Tolerant Ontology Mapping. Dissertation, Department of Computer Science and Engineering, Wright State University, 2015.
- Frederick Maier, Yue Ma, Pascal Hitzler, Paraconsistent OWL and Related Logics. Semantic Web 4 (4), 2013, 395-427.
- Franz Baader, Bernhard Hollunder, Embedding Defaults into Terminological Knowledge Representation Formalisms. Journal of Automated Reasoning 14(1): 149-180, 1995.
- Tania Tudorache, Csongor Nyulas, Natalya Fridman Noy, Mark A. Musen, WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the Web. Semantic Web 4(1): 89-99, 2013.
- Raymond Reiter, A Logic for Default Reasoning. Artificial Intelligence 13(1-2): 81-132, 1980.