Coherence Approach to Logic Program Revision *

Li-Yan Yuan
Computing Science
University of Alberta
Edmonton Canada T6G 2H1
yuan@cs.ualberta.ca

Jia-Huai You
Computing Science
University of Alberta
Edmonton Canada T6G 2H1
you@cs.ualberta.ca

Abstract

We present a new approach to the problem of revising extended programs, based on the *coherence theory* initially advocated by Gardenfors for belief revision. Our approach resolves contradiction by removing only conflicting information, not the believed source of it, and therefore, keeps information loss minimal. Furthermore, since there is no need of searching for problematic assumptions, as is done in the traditional assumption-removal approach, our approach provides a skeptical revision semantics that is tractable.

We define the skeptical and credulous coherence semantics and show that both semantics can be characterized in terms of the fixpoint semantics of a revised program using a simple program revision technique. These semantics provide a suitable framework for knowledge and belief revision in the context of logic programs. Semantical properties and advantages of the proposed revision semantics are also analyzed.

1 Introduction

The extension of logic programs with classical negation significantly increases the expressive power of logic programs, but also presents new challenges [7]. Among other things, the contradiction problem brought up by classical negation has to be addressed.

Unlike normal programs, an extended program with classical negation may not be consistent. For example, $\Pi = \{\neg a \leftarrow;\ \ a \leftarrow;\ \ b \leftarrow\}$ is considered contradictory since both a and $\neg a$ can be derived from it.

Many attempts have been made to resolve the contradiction problem. However, almost all proposals are based on one approach: Resolve the contradiction problem by removing some problematic assumptions [1] , though different mechanisms may be used for the removal. Three notable examples are the contradiction removal semantics [10], the argumentation semantics [5], and the assumption denial semantics [14]. The assumption removal approach, though avoiding contradictory conclusions caused by negation as failure, is not suitable for many applications, such as knowledge and belief revision.

An analysis of semantical properties shows that the assumption removal approach violates two important semantical principles in logic programming and knowledge revision: the principles of *conservatism* and *relevance*.

One of the basic postulates in knowledge revision is that of *conservatism* which requires that any conclusions derived from a revised system be derivable from the system prior to the revision [6]. The assumption-removal approach, though avoiding contradiction, does not address the consequences of assumption removal. As such, conservatism may not hold.

Example 1.1 Consider Π *below*

$$\neg a \leftarrow \mathbf{not} b;\quad a \leftarrow;\quad c \leftarrow \mathbf{not} \neg a,$$

and the credulous argumentation semantics [5].

With each negative literal $\neg \alpha$ *viewed as a named proposition* $\bar{\alpha}$, *the credulous argumentation semantics of* Π *does not imply* c, *which demonstrates that* c *is not true with respect to the credulous argumentation semantics without involving any revision mechanism. Since* Π *has a unique preferred extension* $N = \{\mathbf{not} \neg a\}$, *the credulous argumentation semantics of* Π *does imply* a *and* c *which violates conservatism (see Section 5 for details).*

The principle of *relevance* requires that the values of the literals in a set be determined by only those clauses that define them [3, 8]. Relevance has been considered essential in any goal-directed query evaluation [1] while the goal-directed theoretical characterization forms suitable bases for logic programming [9]. The following example demonstrates that the assumption removal approach may violate relevance.

Example 1.2 Assume Π *is defined by*

$$a \leftarrow \mathbf{not} b;\quad \neg a \leftarrow \mathbf{not} b;\quad d \leftarrow \mathbf{not} b.$$

To resolve the conflict between a *and* $\neg a$, *the assumption-removal semantics has to remove* $\mathbf{not} b$. *Consequently,* d *cannot be derived. Note that the relevant program of* d , *i.e., the set of all clauses that define* d, *is* $\{d \leftarrow \mathbf{not} b\}$ *which deduces* d.

*This paper is based on the technical report *Coherence Approach to Logic Program Revision*, ISIS-RR-94-19E, Fujitsu Laboratories, 1994. A preliminary version of this paper appears in the Proc. of the 12th International Conference on Logic Programming, Page 167-181, June 1995. The work of the first author was performed while visiting the ISIS, Fujitsu Labs, Numazu, Japan.

[1] The assumptions here refer to the values of the default negations. The term is used simply because such values are usually determined by first assuming a value and then justifying the value.

Furthermore, the assumption removal approach suffers computationally from finding a minimal set of problematic assumptions. This is the cause of intractability in the contradiction removal semantics, as shown in [12], and in the grounded version of the *ideal skeptical semantics*, as shown in [13]. Although Dung's grounded argumentation semantics [5] does not attempt to compute a minimal removal set, it is also NP-hard since it is NP-complete to determine if an atom is derivable from a sound argument [13], an essential property in attack and counterattack relationships.

In this paper, we present an alternative approach, based on the coherence theory [6], to removing contradiction from extended programs. This approach provides a suitable semantics for knowledge and belief revision in the framework of logic programs. The idea is simple but very effective:

> Given an extended logic program Π and a set N of assumptions, the coherence revision semantics is determined by the disjunction of maximal consistent sets of literals derivable from Π^N [2]

Note that the assumption-removal approach is based on the assumption that inconsistency is caused by problematic assumptions while the coherence approach removes contradiction regardless the source of it.

We investigate the coherence approach mainly because (1) the coherence approach resolves contradiction by removing only conflicting information, not the *believed source* of it, and therefore, keeps information loss minimal; and (2) in this approach there is no need of searching for minimal sets of problematic assumptions, and thus it provides a skeptical revision semantics that is tractable.

In the paper, we first define the notion of consistent-and-justified (CJ) partial models, based on a negation-as-failure rule and the coherence theory, and then present the skeptical and credulous coherence semantics in terms of CJ partial models.

A remarkable fact about CJ partial models of a program Π is that they are fixpoints of a suitably defined operator for program Π_R, which is obtained from Π by a simple program revision technique: semi-normalizing the *doubling* program [11]. This program revision technique is simple but not trivial, and it gives a natural revision semantics based on the coherence theory.

We also analyze the semantical properties of the proposed revision semantics and show that these semantics satisfy conservatism and relevance, among a number of properties along the line similar to that of Dix [4].

2 Conclusions

The coherence approach to logic program revision is based on the idea that contradiction may be resolved by removing only conflicting information, without searching for believed source of conflict. We have demonstrated that this approach satisfies many desirable properties, inclosing the conservatism and relevance. Furthermore, simply because of the fact that the revision can be achieved regardless of the source of conflict, the coherence approach provides a tractable skeptical revision semantics.

Various coherence semantics provides an alternative approach to logic program revision, and its applications are currently under investigation.

[2] Π^N is the program obtained by GL-transformation

Acknowledgements

The authors would like to thank Jüergen Dix and anonymous reviewers for providing many constructive comments. This work is partially supported by grants from the National Science and Engineering Research Council of Canada and by the ISIS, Fujitsu Labs, Numazu, Japan.

References

[1] J.J. Alferes, C. V. Damasio, and L. M. Pereira. Top-down query evaluation for well-founded semantics with explicit negation. In *Proc. ECAI '94*, pages 140–144, 1994.

[2] J.J. Alferes, P. Dung, and L. Pereira. Scenario semantics of extended logic programs.

[3] J. Dix. A framework for representing and characterizing semantics of logic programs. In *Proceedings of the KR'92*, pages 591–602, 1992.

[4] J. Dix. A classification theory of semantics of normal logic programs: II. weak properties. *to appear*, 1994.

[5] P. M. Dung. An argumentation semantics for logic programming with explicit negation. In *Proc. the 10th ICLP Programming*, pages 615–630. The MIT Press, 1993.

[6] Peter Gardenfors, editor. *Belief Revision.* Cambridge University Press, 1992.

[7] M. Gelfond and V. Lifschitz. Logical programs with classical negation. In *Proceedings of the 7th International Conference on Logic Programming*, pages 579–597. The MIT Press, 1990.

[8] Y. Hu and L. Y. Yuan. Extended well founded semantics for logic programs with negations. In *Proceedings of the 8th International Conference on Logic Programming*, pages 412–425. The MIT Press, 1991.

[9] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[10] L.M. Pereira, J.J. Alferes, and J.N. Aparicio. Contradiction removal within well founded semantics. In *Proc. of the 1st International Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 106–119, 1991.

[11] M. Wallace. Unrestricted logic programs or if stratification is the cure, what is the malady? In *Proc. of European Conference on AI*, pages 682–687, 1990.

[12] C. Witteveen and C. M. Jonker. Revision by expansion in logic programs.

[13] J.H. You and R. Cartwright. Tractable argumentation semantics via iterative belief revision. In *Proc. of ILPS'94*, pages 239–254, 1994.

[14] J.H. You and L.Y. Yuan. Logic programming with assumption denials. In *Proc. of the Workshop 5, ICLP'94*, 1994.