

Finding Explanations of Inconsistency in Multi-Context Systems*

Thomas Eiter and Michael Fink and Peter Schüller and Antonius Weinzierl

Institute of Information Systems
Vienna University of Technology
Favoritenstrasse 11, A-1040 Vienna, Austria
{eiter,fink,schueller,weinzierl}@kr.tuwien.ac.at

Abstract

We provide two approaches for explaining inconsistency in multi-context systems, where decentralized and heterogeneous system parts interact via nonmonotonic bridge rules. Inconsistencies arise easily in such scenarios, and nonmonotonicity calls for specific methods of inconsistency analysis. Both our approaches characterize inconsistency in terms of involved bridge rules: either by pointing out rules which need to be altered for restoring consistency, or by finding combinations of rules which cause inconsistency. We show duality and modularity properties, give precise complexity characterizations, and provide algorithms for computation using HEX-programs. Our results form a basis for inconsistency management in heterogeneous knowledge integration systems.

Introduction

In recent years, there has been increasing interest in interlinking knowledge bases, possibly expressed in different formalisms, to obtain richer knowledge systems. Multi-context systems (MCSs) as introduced by Brewka and Eiter (2007) are an expressive framework for this purpose. MCSs consist of knowledge bases (in possibly heterogeneous and/or nonmonotonic logics) at nodes (called contexts) that exchange information via bridge rules such as

$$(c_1 : h) \leftarrow (c_2 : a), \text{not } (c_3 : d).$$

which informally says that h is believed at context c_1 , if a is believed at context c_2 and d is not believed at context c_3 . MCSs are based on MultiLanguage systems by Giunchiglia and Serafini (1994). They are a powerful knowledge representation formalism for many scenarios where heterogeneity and pointwise, inter-contextual information exchange are essential properties. MCSs enable knowledge integration at a general level, like, e.g., interlinking ontologies, databases, and logic programs. However, due to their decentralized nature, information exchange can have unforeseen effects, and in particular cause an MCS to be inconsistent.

For example, consider a system for supporting health care decisions in a hospital, which comprises several components: a database of laboratory test results, a patient

record database, an ontology for disease classification, and an expert system suggesting suitable treatments for patients. Modeled as an MCS, each component is a context and information flow is specified by bridge rules (cf. Example 3 for details). Suppose the expert system concludes that a patient must be given a special drug, but the patient record states that she is allergic to that drug, thus counter-indicating its use. The whole system gets inconsistent if such special cases were not anticipated when contexts and bridge rules were modeled, rendering the system useless.

In real world applications, system complexity tends to increase, both in terms of contexts and in terms of interconnectivity. Anticipating all possible states of a system is unfeasible, therefore inconsistency handling methods are necessary.

In our approach, we aim to *analyze* inconsistencies in MCSs, in order to understand where and why such inconsistencies occur, and how they can be removed. This will allow to specify how to handle inconsistencies and to extend systems with inconsistency management mechanisms.

While the task reminds of a traditional data integration problem, an important point is that we focus on the exchange of information, i.e., adjusting bridge rules instead of modifying data in the contexts; in loose integrations (e.g., if companies link their business logics), changing contexts or their data to restore consistency may not be an option.

Therefore, we identify bridge rules as the source of inconsistency, and their modification as a possibility of counteracting. We assume every context to be consistent if no bridge rules apply, therefore we can fully capture reasons of inconsistency in terms of bridge rules.

Based on this, we make the following contributions.

- Inspired by debugging approaches used in the nonmonotonic reasoning community, especially in answer set programming (Syrjänen 2006; Brain et al. 2007), we introduce two notions of explaining inconsistency in MCSs: a *consistency-based* notion, which characterizes inconsistency in terms of altered sets of bridge rules that are consistent, and an *entailment-based* notion which derives inconsistency in a given system. Possible nonmonotonicity makes intuitive and sound notions challenging; that our notions have appealing properties may serve as some evidence for their suitability.

- We establish useful properties of our notions. First, a du-

*This work was supported by the Vienna Science and Technology Fund (WWTF) under grant ICT 08-020.
Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ality result between the consistency- and entailment-based notion shows that they identify the same bridge rules as relevant for inconsistency. This result in fact generalizes a similar result by Reiter (1987). Second, modularity properties in the spirit of Splitting Sets (Lifschitz and Turner 1994) allow an incremental computation of explanations, taking the MCS topology into account.

- We sharply characterize the computational complexity of identifying explanations, under varying assumptions for the complexity of contexts (note that explanations always do exist). It turns out that this problem has for a range of context complexities no (or only mildly) higher complexity than the contexts themselves. As a consequence, computing explanations is in some cases not harder than consistency checking.
- Finally, we show how consistency-based explanations can be computed by means of HEX-programs, which are a variant of answer set programs with access to external sources.

Our results provide a basis for building enhanced MCS systems which are capable of analyzing and reasoning about emerging inconsistencies. Rather than automatically resolving inconsistency, as e.g. in (Bikakis and Antoniou 2008), we envisage a (semi-)automatic approach with user support for locating and tracking parts that cause inconsistency. Indeed, user invention may be indispensable as often no automatic solution is suitable (like in our healthcare example).

For space reasons some proofs have been omitted, selected proof sketches are provided in Appendix Proofs.

Preliminaries

A heterogeneous nonmonotonic MCS (Brewka and Eiter 2007). consists of *contexts*, each composed of a knowledge base with an underlying *logic*, and a set of *bridge rules* which control the information flow between contexts.

A logic $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$ consists, in an abstract view, of the following components:

- \mathbf{KB}_L is the set of well-formed knowledge bases of L . We assume each element of \mathbf{KB}_L is a set (of “formulas”).
- \mathbf{BS}_L is the set of possible belief sets, where the elements of a belief set are “formulas”.
- $\mathbf{ACC}_L : \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ is a function describing the “semantics” of the logic by assigning to each knowledge base a set of acceptable belief sets.

This concept of a *logic* captures many monotonic and non-monotonic logics, e.g., classical logic, description logics, modal, default, and autoepistemic logics, circumscription, and logic programs under the answer set semantics.

For an intuition how this abstraction captures some well-known KR formalisms, consider the following Examples.

Example 1. For propositional logic under the closed world assumption over signature Σ , \mathbf{KB} is the set of propositional formulas over Σ ; \mathbf{BS} is the set of deductively closed sets of propositional Σ -literals; and $\mathbf{ACC}(kb)$ returns for each kb a singleton set, containing the set of literal consequences of kb under the closed world assumption. \square

Example 2. For normal disjunctive logic programs under answer set semantics over a signature Σ (Przymusinski 1991), \mathbf{KB} is the set of normal disjunctive logic programs over Σ ; \mathbf{BS} is the set of sets of atoms over Σ ; and $\mathbf{ACC}(kb)$ returns the set of kb ’s answer sets. \square

A bridge rule can add information to a context, depending on the belief sets which are accepted at other contexts. Let $L = (L_1, \dots, L_n)$ be a sequence of logics. An L_k -bridge rule r over L is of the form

$$(k : s) \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \\ \text{not } (c_{j+1} : p_{j+1}), \dots, \text{not } (c_m : p_m). \quad (1)$$

where $1 \leq c_i \leq n$, p_i is an element of some belief set of L_{c_i} , k refers to the context receiving information s . We denote by $h_b(r)$ the belief formula s in the head of r .

Definition 1. A multi-context system $M = (C_1, \dots, C_n)$ is a collection of contexts $C_i = (L_i, kb_i, br_i)$, $1 \leq i \leq n$, where $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ is a logic, $kb_i \in \mathbf{KB}_i$ a knowledge base, and br_i is a set of L_i -bridge rules over (L_1, \dots, L_n) . For each $H \subseteq \{h_b(r) \mid r \in br_i\}$ it holds that $kb_i \cup H \in \mathbf{KB}_{L_i}$, i.e., bridge rule heads are compatible with knowledge bases.

A belief state of an MCS $M = (C_1, \dots, C_n)$ is a sequence $S = (S_1, \dots, S_n)$ such that $S_i \in \mathbf{BS}_i$. A bridge rule (1) is applicable in a belief state S iff for $1 \leq i \leq j$: $p_i \in S_{c_i}$ and for $j < l \leq m$: $p_l \notin S_{c_l}$. By $br_M = \bigcup_{i=1}^n br_i$ we denote the set of all bridge rules of M .

Example 3. Consider an MCS M which is a health care decision support system and contains the following contexts: a patient history database (C_1), a blood and X-Ray analysis database (C_2), a disease ontology (C_3), and an expert system (C_4) which suggests proper treatments.

The knowledge bases for these contexts are as follows:

$$kb_1 = \{\text{allergy_strong_ab}\}, \\ kb_2 = \{\neg \text{blood_marker}, \text{xray_pneumonia}\}, \\ kb_3 = \{\text{Pneumonia} \sqcap \text{Marker} \sqsubseteq \text{AtypPneumonia}\}, \\ kb_4 = \{\text{give_strong} \vee \text{give_weak} \leftarrow \text{need_ab}, \\ \text{give_strong} \leftarrow \text{need_strong}, \\ \perp \leftarrow \text{give_strong}, \text{not allow_strong_ab}, \\ \text{give_nothing} \leftarrow \text{not need_ab}, \text{not need_strong}\}.$$

Contexts C_1 and C_2 use propositional logic (see Example 1 for the corresponding definition of L_1 and L_2). They provide information that the patient is allergic to strong antibiotics, that a certain blood marker is not present, and that pneumonia was detected in an X-ray examination.

C_3 specifies that presence of a blood marker in combination with pneumonia indicates atypical pneumonia. This context is based on a basic \mathcal{AL} description logic (Baader et al. 2003): \mathbf{KB}_3 is the set of all well-formed theories within that description logic, \mathbf{BS}_3 is the powerset of the set of all assertions $C(o)$ where C is a concept name and o an individual name, and \mathbf{ACC}_3 returns the set of all concept assertions entailed by a given theory.

C_4 suggests a treatment which is either a strong antibiotic, a weak antibiotic, or no medication at all. C_4 is built

from a normal disjunctive logic program, see Example 2 for a suitable definition of L_4 .

The bridge rules of M are given as follows:

$$\begin{aligned} r_1 &= (3 : \text{Pneumonia}(p)) \leftarrow (2 : \text{xray_pneumonia}). \\ r_2 &= (3 : \text{Marker}(p)) \leftarrow (2 : \text{blood_marker}). \\ r_3 &= (4 : \text{need_ab}) \leftarrow (3 : \text{Pneumonia}(p)). \\ r_4 &= (4 : \text{need_strong}) \leftarrow (3 : \text{AtypPneumonia}(p)). \\ r_5 &= (4 : \text{allow_strong_ab}) \leftarrow \text{not } (1 : \text{allergy_strong_ab}). \end{aligned}$$

Rules r_1 and r_2 provide input for disease classification to the ontology, they assert facts about a new individual ‘ p ’ corresponding to the patient. Rules r_3 and r_4 link disease information with medication requirements, and r_5 relates acceptance of strong antibiotics with an allergy check on the patient database. \square

Equilibrium semantics selects certain belief states of an MCS M as acceptable. Intuitively, an equilibrium is a belief state S , where each context C_i takes the heads of all bridge rules that are applicable in S into account, and accepts S_i .

We denote by $\text{app}(R, S)$ the set of all bridge rules $r \in R$ that are applicable in belief state S .

Definition 2. A belief state $S = (S_1, \dots, S_n)$ of M is an equilibrium iff the following holds: for all $1 \leq i \leq n$,

$$S_i \in \mathbf{ACC}_i(\text{kb}_i \cup \{h_b(r) \mid r \in \text{app}(br_i, S)\}).$$

Example 4. In our example, one equilibrium S exists:

$$S = (\{\text{allergy_strong_ab}\}, \{\neg\text{blood_marker}, \text{xray_pneumonia}\}, \{\text{Pneumonia}(p)\}, \{\text{need_ab}, \text{give_weak}\}).$$

Rules r_1 and r_3 are applicable in S . A belief state S_3 consists of all ABox (Baader et al. 2003) concept assertions.

Note that if we replace S_4 with $\{\text{need_ab}, \text{give_strong}, \text{allow_strong_ab}\}$, the resulting belief state is not an equilibrium: C_4 uses answer set semantics, therefore allow_strong_ab cannot be part of S_4 unless it is added by a bridge rule. The only bridge rule with this head is r_5 , and its applicability is blocked by the presence of allergy_strong_ab in kb_1 and in S_1 . \square

Inconsistency in an MCS is the lack of an equilibrium. As the combination and interaction of heterogeneous systems can easily have unforeseen and intricate effects, inconsistency is a major, to our knowledge unaddressed problem in MCSs. To provide support for restoring consistency, we seek to understand and give reasons for inconsistency.

Example 5. As a running example, we consider a slightly modified version of Example 3, where the blood serum analysis shows presence of the blood marker:

$$\text{kb}_2 = \{\text{blood_marker}, \text{xray_pneumonia}\}.$$

This MCS is inconsistent since r_2 and r_4 become applicable, which in turn requires the strong antibiotic. This is in conflict with the patient’s allergy.

Note that applicability of r_5 would resolve this inconsistency by activating allow_strong_ab . However, presence of allergy_strong_ab in S_1 together with body atom ‘not (1 : allergy_strong_ab)’ in r_5 prevents the applicability of r_5 (due to negation as failure). \square

We will use the following notation. Given an MCS M and a set R of bridge rules (compatible with M), by $M[R]$ we denote the MCS obtained from M by replacing its set of bridge rules br_M with R (e.g., $M[br_M] = M$ and $M[\emptyset]$ is M with no bridge rules). By $M \models \perp$ we denote that M has no equilibrium, i.e., is inconsistent, and by $M \not\models \perp$ the opposite. For any set of bridge rules A , $\text{heads}(A) = \{\alpha \leftarrow \mid \alpha \leftarrow \beta \in A\}$ are the rules in A in unconditional form.

Diagnoses and Explanations

In the following, we consider two possibilities for explaining inconsistency in MCSs: first, a consistency-based formulation, which identifies a part of the bridge rules which need to be changed to restore consistency. Second, an entailment-based formulation, which identifies a part of the bridge rules which is required to make the MCS inconsistent. Following common terminology, we call the first formulation *diagnosis* (cf. Reiter 1987) and the second *inconsistency explanation*.

Diagnoses. As well-known, in nonmonotonic reasoning, adding knowledge can both cause and prevent inconsistency; the same is true for removing knowledge.

For our consistency-based explanation of inconsistency, we therefore consider pairs of sets of bridge rules, s.t. if we deactivate the rules in the first set, and add the rules in the second set in unconditional form, the MCS becomes consistent (i.e., admits an equilibrium).

Definition 3. Given an MCS M , a diagnosis of M is a pair (D_1, D_2) , $D_1, D_2 \subseteq br_M$, s.t. $M[br_M \setminus D_1 \cup \text{heads}(D_2)] \not\models \perp$. $D^\pm(M)$ is the set of all such diagnoses.

To obtain a more relevant set of diagnoses, we prefer pointwise subset-minimal diagnoses. For pairs $A = (A_1, A_2)$ and $B = (B_1, B_2)$ of sets, the pointwise subset relation $A \subseteq B$ holds iff $A_1 \subseteq B_1$ and $A_2 \subseteq B_2$.

Definition 4. $D_m^\pm(M)$ is the set of all pointwise subset-minimal diagnoses of an MCS M .

Example 6. In our running example,

$$D_m^\pm(M) = \{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_4\}, \emptyset), (\emptyset, \{r_5\})\}.$$

Accordingly, deactivating r_1 , or r_2 , or r_4 , or adding r_5 unconditionally, will result in a consistent MCS. \square

We note that one could generalize Definition 3 to more fine-grained changes of rules, such that only some body atoms are removed instead of all. However, while this significantly increases the search space for diagnoses, there is little information gain: every diagnosis (D_1, D_2) as above, together with a witnessing equilibrium S , can be refined to a generalized diagnosis (D_1, D'_2) , where $D'_2 \subseteq \{\alpha \leftarrow \beta \mid \alpha \leftarrow \beta, \gamma \in br_M\}$ contains for each $\alpha \leftarrow \beta, \gamma$ in D_2 some $\alpha \leftarrow \beta$ that is applicable in S . Conversely, every generalized diagnosis (D_1, D'_2) , together with a witnessing equilibrium S , induces a diagnosis (D_1, D_2) as above (D_2 contains all heads of rules in D'_2 that are applicable in S).

Explanations. In the spirit of abductive reasoning, we also propose an entailment-based notion of explaining inconsistency: an *inconsistency explanation* (in the following also called *explanation*) is a pair of sets of bridge rules, s.t. their presence or absence entails a relevant (cf. below) inconsistency in the given MCS.

Definition 5. Given an MCS M , an inconsistency explanation of M is a pair (E_1, E_2) of sets $E_1, E_2 \subseteq br_M$ of bridge rules s.t. for all (R_1, R_2) where $E_1 \subseteq R_1 \subseteq br_M$ and $R_2 \subseteq br_M \setminus E_2$, it holds that $M[R_1 \cup heads(R_2)] \models \perp$. By $E^\pm(M)$ we denote the set of all inconsistency explanations of M , and by $E_m^\pm(M)$ the set of all pointwise subset-minimal ones.

The intuition about E_1 is as follows: bridge rules in E_1 create an inconsistency in M ($M[E_1] \models \perp$), and this inconsistency is relevant for M . By relevance we mean that adding some bridge rules from br_M (the set of original bridge rules) to $M[E_1]$ never yields a consistent system.

This condition is necessary, for example the program $P = \{a \leftarrow not\ a.\}$, is inconsistent under the answer set semantics, but its superset $P' = \{a \leftarrow not\ a.\ a.\}$ is consistent. The inconsistency of P does not matter for P' . In terms of MCSs, a set of bridge rules may create an inconsistency in M , but this inconsistency is irrelevant, as it does not occur if more or all bridge rules are present.

Intuition about E_2 concerns inconsistency wrt. the addition of unconditional bridge rules: $M[E_1]$ cannot be made consistent by adding bridge rules unconditionally, unless we use at least one bridge rule from E_2 .

In summary, bridge rules E_1 create a relevant inconsistency, and at least one bridge rule in E_2 must be added unconditionally to repair that inconsistency.

Example 7. In our running example, we have one minimal inconsistency explanation, namely $(\{r_1, r_2, r_4\}, \{r_5\})$. To trigger the only possible inconsistency, which is in C_4 , we need to import `need_strong` (using r_4) and we must not import `allow_strong_ab` (using r_5). Furthermore, r_4 can only fire if C_3 accepts `AtypPneumonia(p)`, which is only possible if r_1 and r_2 fire. Therefore, r_1, r_2 , and r_4 must be present to get inconsistency, and the head of r_5 must not be present.

Similar to diagnoses, it is possible to consider more fine-grained modifications of rules (rather than $heads(R_2)$) in Definition 5. Note however, that this would not alter the notion of inconsistency explanation. Thus, in contrast to diagnoses, we cannot infer from an explanation whether the addition of a more fine-grained version of a rule in E_2 would yield consistency. However, this could be achieved considering a transformed MCS M' : roughly, every bridge rule in M is split into a core rule and a supplementary rule for each body atom; e.g., $(c_1:h) \leftarrow (c_2:a), not(c_3:b)$ is rewritten to $(c_1:h) \leftarrow (c_\alpha:a'), (c_\alpha:b')$, $(c_\alpha:a') \leftarrow (c_2:a)$, and $(c_\alpha:b') \leftarrow not(c_3:b)$. Rules in E_2 are replaced by a subset of their supplementary rules, thus E_2 indicates which body atoms to remove from the original rules to avoid the explained inconsistency.

Duality. Adding explanation rules E_1 to contexts causes inconsistency, removing diagnosis rules D_1 from an MCS can cause consistency; hence they represent dual aspects. Similarly, D_2 and E_2 have dual intuitions, as D_2 requires to add rules unconditionally, while E_2 forbids to do so.

Notation: for any set X of tuples (A, B) of sets A and B (e.g., for some set of diagnoses), we denote by $\bigcup X$ the pair $(\bigcup\{A \mid (A, B) \in X\}, \bigcup\{B \mid (A, B) \in X\})$.

Theorem 1. Given an inconsistent MCS M , $\bigcup D_m^\pm(M) = \bigcup E_m^\pm(M)$, i.e., the unions of all minimal diagnoses and all minimal inconsistency explanations coincide.

Hence, the duality between both components of minimal diagnoses and explanations extends beyond our example, and our definitions are closely related. This strengthens our view that both notions capture exactly those parts of an MCS that are relevant for inconsistency as duality shows that, in total, two very different perspectives on inconsistency state exactly the same parts of the MCS as erroneous.

In practice this allows one to compute the set of all bridge rules which are relevant for making a MCS consistent (i.e., appear in at least one diagnosis) in two ways: either compute all minimal explanations, or compute all minimal diagnoses. Conversely stated, the duality result allows to exclude all bridge rules that are not part of any diagnosis (or explanation) from further investigation as those parts are known to be irrelevant.

Special Diagnoses/Explanations. For domains where removal of bridge rules is preferred to unconditional addition of rules, we specialize D^\pm to obtain diagnoses of the form (D_1, \emptyset) only. We again prefer subset-minimal diagnoses.

Definition 6. Given an MCS M , an s -diagnosis of M is a set $D \subseteq br_M$ s.t. $M[br_M \setminus D] \not\models \perp$. The set of all s -diagnoses (resp., \subseteq -minimal s -diagnoses) is $D^-(M)$ (resp., $D_m^-(M)$).

Example 8. In our example, $D_m^-(M) = \{\{r_1\}, \{r_2\}, \{r_4\}\}$.

We also specialize the inconsistency explanation to the first component, i.e., we do not consider adding rules unconditionally, so all explanations are of the form (E_1, br_M) .

Definition 7. Given an MCS M , an s -inconsistency explanation of M is a set $E \subseteq br_M$ s.t. each R where $E \subseteq R \subseteq br_M$, satisfies $M[R] \models \perp$. The set of s -inconsistency explanations is denoted by $E^+(M)$, and the set of \subseteq -minimal s -inconsistency explanations of M is denoted by $E_m^+(M)$.

Example 9. The only minimal s -inconsistency explanation in our running example is $\{r_1, r_2, r_4\}$. \square

Our running example suggests, that duality also holds for simplified diagnoses and explanations, which indeed is true:

Theorem 2. Given an inconsistent MCS M , $\bigcup D_m^-(M) = \bigcup E_m^+(M)$, i.e., the unions of all minimal s -diagnoses and all minimal s -inconsistency explanations coincide.

The proof of this theorem is similar to the proof of Theorem 4.4 in Reiter's seminal paper (1987), which states that diagnoses are minimal hitting sets on the set of conflict sets, where a conflict set is similar to what we call s -inconsistency explanation. The main difference is that Reiter's conflict sets are defined on monotonic (first-order) logic, while our explanations are defined on nonmonotonic logics. However, the condition that an explanation must not be repairable by adding bridge rules of the original system, effectively ensures that explanations become monotonic.

Properties

We first consider a simple property of minimal diagnoses. According to Definition 3, given (D_1, D_2) with $r \in D_2$, it is irrelevant for being a diagnosis whether $r \in D_1$ or not.

Proposition 1. *In a minimal diagnosis (D_1, D_2) of an MCS M , $D_1 \cap D_2 = \emptyset$, i.e., no rule occurs in both components.*

This is not true for inconsistency explanations. For the MCS M consisting of bridge rules $r_1 = (1 : a) \leftarrow (2 : b)$ and $r_2 = (2 : b) \leftarrow \text{not } (1 : a)$, and empty contexts under a minimal model semantics, we get the following minimal explanations: $E_m^\pm(M) = \{(\{r_1, r_2\}, \{r_1\}), (\{r_1, r_2\}, \{r_2\})\}$.

Modularity of Explanations. We next give a syntactic criterion which allows for breaking up the computation of explanations for an MCS M into computing explanations for parts of it. Minimal explanations of M are then just combinations of the minimal explanations of the smaller parts. This can be exploited for computing minimal explanations more efficiently for certain classes of MCSs.

A criterion for modularization is that some part is independent of the rest of the system. For formalizing such a criterion, we adapt the notion of *splitting sets* as introduced by Lifschitz and Turner (1994) in the context of logic programming. A splitting set characterizes a subset of a logic program which is independent of other rules in the program by a syntactic property.

Since an MCS may include contexts with arbitrary logics, a purely syntactical criterion can only be obtained resorting to beliefs occurring in bridge rules, implicitly assuming that every output belief of a context depends on any input belief of the context. Hence, we split at the level of contexts, i.e., a splitting set is a set of contexts rather than a set of literals.

Let $c(M)$ denote the set of contexts of an MCS $M = M[br_M]$, and for a bridge rule r , let $h_c(r)$ be the context in its head and $b_c(r)$ the set of contexts referenced in its body.

Definition 8. *A set of contexts $U \subseteq c(M)$ is a splitting set of an MCS M , if for every rule $r \in br_M$ the following holds: if $h_c(r) \in U$ then $b_c(r) \subseteq U$. The set $b_U \subseteq br_M$ of rules s.t. $r \in b_U$ iff $h_c(r) \in U$, is called the bottom relative to U .*

In our running example, we have $c(M) = \{C_1, \dots, C_4\}$, $h_c(r_1) = C_3$, and $b_c(r_1) = \{C_2\}$. The set $U = \{C_2, C_3\}$ is a splitting set of M , with $b_U = \{r_1, r_2\}$.

Intuitively, if U is a splitting set of M , then the consistency or inconsistency of contexts in U does not depend on contexts in $c(M) \setminus U$. Thus if $M[b_U]$ is inconsistent, M stays inconsistent.

Proposition 2. *Let U be a splitting set of an MCS M . Then each (minimal) explanation of $M[b_U]$ is a (minimal) explanation of M , and each (minimal) diagnosis of $M[b_U]$ is a pointwise subset of a (minimal) diagnosis of M .*

Proposition 3. *Suppose that both U and $U' = c(M) \setminus U$ are splitting sets of an MCS M . Then for every $(E_1, E_2) \in E_m^\pm(M)$, either $X = U$ or $X = U'$ satisfies $\{h_c(r)\} \cup b_c(r) \subseteq X$, for every $r \in E_1 \cup E_2$.*

Thus, a respective M can be partitioned into two parts where minimal explanations can be computed independently.

Ceteris Paribus preferences. In the previous sections, subset-minimality is used to select intuitively preferred diagnoses and explanations. We can generalize subset-based preference orders to cover a certain class of Ceteris Paribus (CP) preference orders. The latter encode CP statements like

“I prefer A over B, all else being equal”. We consider CP orders \preceq on the set of bridge rules br_M of M which are global orders naturally built from local orders on the elements of a partitioning of br_M , each given by some CP statement.

To combine a global order on br_M from local orders, we need to define the product of orders.

Definition 9. *The order product \prec of n orders \prec_1, \dots, \prec_n over disjoint sets P_1, \dots, P_n is a subset of $(P_1 \times \dots \times P_n)^2$ with $\langle p_1, \dots, p_n \rangle \prec \langle p'_1, \dots, p'_n \rangle$ iff $p_i \prec_i p'_i$ for some $1 \leq i \leq n$.*

As the P_i are disjoint we identify each tuple of the product with the union of its elements, i.e., $\langle p_1, \dots, p_n \rangle = \bigcup_i \{p_i\}$ to express an order \prec defined on $\bigcup_i P_i$.

It is easy to show that the order product is a Boolean lattice if all orders it is composed of are Boolean lattices. Furthermore, if the lattice-complement coincides with set-complement on orders \prec_1, \dots, \prec_n , then the complements also coincide on the order product of \prec_1, \dots, \prec_n .

Example 10. *Let $br_M = \{a, b\}$ and the local order be expressed by $P_1 :=$ “I prefer a over \emptyset ” and $P_2 :=$ “I prefer \emptyset over b, i.e., I prefer b to be absent”. The respective local orders are $\emptyset \prec_{P_1} a$ and $b \prec_{P_2} \emptyset$. The global order \prec resulting from the lattice product of both local orders is induced by $\{b\} \prec \{a, b\}$, $\{b\} \prec \{\emptyset\}$, $\{a, b\} \prec \{a\}$, and $\{\emptyset\} \prec \{a\}$.*

For the following theorem, observe that diagnoses and explanations can be described purely by structural conditions over Boolean lattices where elements of the lattice carry a flag for inconsistency. For example, an s-diagnosis is an element whose lattice-complement is not flagged inconsistent and a minimal s-diagnosis is an s-diagnosis that is minimal wrt. the order relation of the lattice. As such a diagnosis intuitively is a set of bridge rules that must be removed to gain consistency and this intuition is expressed by using the complement, it is necessary that the lattice-complement and the set-complement are the same.

Proposition 4. *Suppose each local order of a CP-order \preceq of an MCS $M[br_M]$ is a Boolean lattice such that its lattice-complement coincides with set-complement. Then, Theorem 1 holds for respective notions of explanations and diagnoses.*

Given the same conditions, then diagnoses induce repairs which change least preferred sets of rules only, and we get similar complexity results. Moreover, an algorithm for computing subset-minimal explanations resp. diagnoses can be adapted to compute CP-preferred ones.

The technique of order embedding can be employed to cover also CP-orderings which are not composed of Boolean lattices. We briefly illustrate this in the following example.

Example 11. *Let \prec_e be an ordering induced by $\emptyset \prec_e \{a\} \prec_e \{b\} \prec_e \{a, b\}$. We can embed \prec_e into a Boolean lattice over $\{a, b, c\}$ by adding an additional bridge rule c to the system. The resulting Boolean lattice is given by:*

$$\begin{array}{lll} \{b\} \prec \{a, b\} & \{a, b, c\} \prec \{a, b\} & \{b, c\} \prec \{a, b\} \\ \{a\} \prec \{b\} & \{a\} \prec \{a, b, c\} & \{c\} \prec \{b\} \\ & \{a, c\} \prec \{a, b, c\} & \{a, c\} \prec \{b, c\} \\ \emptyset \prec \{a\} & \emptyset \prec \{c\} & \emptyset \prec \{a, c\} \end{array}$$

If the newly introduced bridge rule is such that it causes inconsistency by its own, then all elements of the lattice except those from $prec_e$ are guaranteed to be inconsistent. By that, only the elements from $prec_e$ may lead to a (minimal) s -diagnosis. Also note that set-complement and lattice-complement coincide modulo c in the above lattice, i.e., whenever D is a minimal s -diagnosis on \prec then $D \cap \{c\}$ is a \prec_e -minimal s -diagnosis.

Computational Complexity

Calculating equilibria by guessing so-called “kernels of context belief sets” has been outlined in (Eiter et al. 2009). Note that, by our basic assumption that M is inconsistent but $M[\emptyset]$ is consistent, the existence of a diagnosis resp. explanation is trivial. For the purpose of recognizing diagnoses and explanations, it suffices to check for consistency, i.e., for existence of some equilibrium in a (modified) MCS.

Consistency checking can be done by limiting the equilibrium calculation to *output beliefs*, which are the beliefs used in the bodies of bridge rules.

Definition 10. Given an MCS M and one of its contexts C_i , we denote by OUT_i the set of output beliefs of C_i ; those are beliefs p of context C_i which are contained in the body of some bridge rule $r \in br_M$ in the form of a literal ‘ $(i : p)$ ’ or ‘**not** $(i : p)$ ’.

Using the notion of output beliefs, we project each belief set S_i of a belief state S to that context’s output beliefs OUT_i . Given belief state $S = (S_1, \dots, S_n)$ in MCS M , the *output-projected belief state* $S' = (S'_1, \dots, S'_n)$ is the projection of S to output beliefs of M : $S'_i = S_i \cap OUT_i$.

An output-projected belief state provides sufficient information to evaluate bridge rule applicability, furthermore we can define witnesses for equilibria using this projection.

Definition 11. A *output-projected belief state* $S' = (S'_1, \dots, S'_n)$ of an MCS M is an *output-projected equilibrium* iff the following holds: for all $1 \leq i \leq n$,

$$S'_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) \mid r \in \text{app}(br_i, S')\})|_{OUT_i}.$$

S' contains information about all output beliefs, which are the beliefs that determine bridge rule applicability, therefore $\text{app}(R, S) = \text{app}(R, S')$ and we obtain the following.

Lemma 1. For each equilibrium S of an MCS M , S' is an output-projected equilibrium. Conversely, for each output-projected equilibrium S' of M there exists at least one equilibrium T of M such that $T' = S'$.

For consistency checking (equilibrium existence), it is therefore sufficient to consider output-projected equilibria.

Consistency of M can be decided by a Turing machine with input M which (a) guesses an output-projected belief state $S' \in OUT_1 \times \dots \times OUT_n$, (b) evaluates the bridge rules on S' , yielding for each context C_i a set of active bridge rule heads H_i wrt. S' , and (c) checks whether each context accepts the guessed S'_i wrt. H_i . Formally, (c) decides whether there exists an $S_i \in \mathbf{ACC}_i(kb_i \cup H_i)$ s.t. $S'_i = S_i \cap OUT_i$. We call the complexity of this check *context complexity*; the system’s context complexity $\mathcal{CC}(M)$ is a (smallest) upper bound for the context complexity classes of all C_i .

Context complexity	$(A, B) \stackrel{?}{\in}$			
	$D^\pm(M)$	$D_m^\pm(M)$	$E^\pm(M)$	$E_m^\pm(M)$
P	NP	$\mathbf{D}_1^{\mathbf{P}}$	coNP	$\mathbf{D}_1^{\mathbf{P}}$
NP	NP	$\mathbf{D}_1^{\mathbf{P}}$	coNP	$\mathbf{D}_1^{\mathbf{P}}$
$\Sigma_2^{\mathbf{P}}$	$\Sigma_2^{\mathbf{P}}$	$\mathbf{D}_2^{\mathbf{P}}$	$\Pi_2^{\mathbf{P}}$	$\mathbf{D}_2^{\mathbf{P}}$

Table 1: Complexity of recognizing (minimal) diagnoses / explanations, given (A, B) and an MCS M (completeness results; hardness holds for major ASP context classes).

Note, that for complexity considerations each context C_i is explicitly represented by kb_i and br_i , and the logic is implicitly given, which is taken care of by an oracle that decides (c) in $\mathcal{CC}(M)$.

Given $\mathcal{CC}(M)$ is in **P**, consistency checking is in **NP**. The same is true if $\mathcal{CC}(M)$ is in **NP**, as output belief and context guesses can be combined into one guess. Similarly, we can obtain that for $\mathcal{CC}(M)$ in $\Sigma_i^{\mathbf{P}}$, consistency checking of M is in $\Sigma_i^{\mathbf{P}}$, $i \geq 2$; for $\mathcal{CC}(M)$ in **PSPACE** (resp., **EXPTIME**), it is in **PSPACE** (resp., **EXPTIME**).

Inconsistency Analysis. Using these results, we obtain complexity results for deciding, given an MCS M of certain context complexity and a pair (A, B) of sets of bridge rules in M , whether (A, B) is a diagnosis resp. a \subseteq -minimal diagnosis, an explanation, or a \subseteq -minimal explanation for M .

Table 1 shows the results for $\mathcal{CC}(M)$ in **P**, **NP**, and $\Sigma_2^{\mathbf{P}}$, which are typical complexities for Answer Set Programs (ASPs)—a premier KR formalism for modeling non-monotonic behavior, cf. (Dantsin et al. 2001). Here, the class $\mathbf{D}_1^{\mathbf{P}}$ contains decision problems which are the “conjunction” of a $\Sigma_i^{\mathbf{P}}$ and an independent $\Pi_i^{\mathbf{P}}$ decision problem (for $i = 1$, e.g. a SAT and an independent UNSAT instance). The results for $\Sigma_2^{\mathbf{P}}$ easily generalize to $\Sigma_i^{\mathbf{P}}$ for $i \geq 2$.

In the following, we give a brief intuition of the proofs. Diagnosis recognition can be done by calculating $M[br_M \setminus A \cup \text{heads}(B)]$ and checking this system for consistency as outlined above. Therefore, diagnosis recognition has the same complexity as consistency checking.

Checking for minimality, one additionally guesses all pairs $(A', B') \subset (A, B)$ and checks for consistency of $M[br_M \setminus A' \cup \text{heads}(B')]$. The candidate is a minimal diagnosis iff the diagnosis check is successful, and all subset-diagnosis checks are not, leading to a complexity of $\mathbf{D}_1^{\mathbf{P}}$ for **P** and **NP** contexts ($\mathbf{D}_i^{\mathbf{P}}$ for $\Sigma_i^{\mathbf{P}}$ contexts, $i \geq 2$).

Explanation recognition can be done by guessing all sets (R_1, R_2) of Definition 5 and checking whether $M[R_1 \cup \text{heads}(R_2)]$ is consistent. The instance is a yes instance iff all such checks fail, leading to **coNP** ($\Pi_1^{\mathbf{P}}$) complexity.

For checking explanation minimality, we use a Lemma.

Lemma 2. An explanation $Q = (Q_1, Q_2)$ is \subseteq -minimal iff no pair $(Q_1, Q_2 \setminus \{r\})$ with $r \in Q_2$ or $(Q_1 \setminus \{r\}, Q_2)$ with $r \in Q_1$ is an explanation.

Hence, we can check subset-minimality of explanations by deciding, whether for linearly many subsets of the candidate (A, B) , none is an explanation, i.e., whether for each subset,

some (R_1, R_2) exists s.t. $M[R_1 \cup \text{heads}(R_2)]$ is consistent. As NP (resp., Σ_1^P) is closed under conjunction, this check is in NP (Σ_1^P). Additionally checking whether the candidate is an explanation leads to a complexity of D_1^P (D_1^P).

For $\mathcal{CC}(M)$ in PSPACE (EXPTIME), all tests can be done within PSPACE (EXPTIME). The matching hardness results are established via the context complexity.

For the other cases, matching hardness is shown by reductions from 3-SAT, 3-QSAT $_i$, $i \geq 2$, and suitable Boolean combinations thereof. Specifically for the practically relevant setting of ASP contexts, we sketch a reduction from 3-SAT to D^\pm recognition.¹

A 3-SAT instance $F = c_1 \wedge \dots \wedge c_n$ on variables x_1, \dots, x_k with $c_i = c_{i,1} \vee c_{i,2} \vee c_{i,3}$ is reduced to an MCS consisting of one context C_1 . This context is an acyclic ASP and can be evaluated in P, it contains bridge rules (2) to (5) and context-internal rules (6) to (9):

$$(1 : x_i) \leftarrow \text{not } (1 : \bar{x}_i). \quad \forall 1 \leq i \leq k \quad (2)$$

$$(1 : \bar{x}_i) \leftarrow \text{not } (1 : x_i). \quad \forall 1 \leq i \leq k \quad (3)$$

$$(1 : en) \leftarrow \top. \quad (4)$$

$$(1 : inc) \leftarrow \top. \quad (br_{inc}) \quad (5)$$

$$sat_i \leftarrow l_{i,1}. \quad sat_i \leftarrow l_{i,2}. \quad sat_i \leftarrow l_{i,3}. \quad \forall 1 \leq i \leq n \quad (6)$$

$$\text{where } l_{i,j} = \begin{cases} x_v & \text{if } c_{i,j} = x_v \\ \bar{x}_v & \text{if } c_{i,j} = \neg x_v \end{cases}$$

$$sat \leftarrow sat_1, \dots, sat_n. \quad (7)$$

$$\perp \leftarrow en, \text{not } sat. \quad (8)$$

$$\perp \leftarrow en, inc. \quad (9)$$

Intuitively, (4) enables constraints within C_1 , (5) in combination with (9) makes the MCS inconsistent. (2) and (3) guess a satisfying assignment for F , which is evaluated by (6) and (7). Satisfaction is required by (8). The diagnosis candidate is $(\{br_{inc}\}, \emptyset)$, i.e., removing br_{inc} allows an equilibrium iff there exists a satisfying assignment for F . Therefore the candidate is a diagnosis iff F is satisfiable.

The reduction for Σ_2^P ASP contexts encodes QSAT $_2$ in a similar context. For minimal diagnoses (D_1^P), we reduce SAT-UNSAT to an MCS (C_1, C_2) containing C_1 above and another context C_2 which is inconsistent iff a further 3-SAT instance is unsatisfiable. The reduction for Σ_2^P contexts is similar, using QSAT $_2$ -QUNSAT $_2$ and disjunctive ASP contexts. For explanation and minimal explanation recognition, reductions reuse contexts from the previous reductions.

In summary, diagnosis recognition has the same computational complexity as equilibrium existence. This is used in the following section to compute diagnoses using a logic programming formalism with external oracle calls which is capable of handling NP problems.

In general the results indicate that, if it is possible to calculate MCS equilibria with a particular solver, it is likewise possible to calculate diagnoses using this solver using a polynomial transformation.

¹Note that ASP is NP-complete, therefore, although this reduction is specifically shown for ASP contexts, it can be easily adapted to other NP-complete KR formalisms.

Minimal diagnosis and minimal explanation recognition are harder than checking consistency (under usual complexity assumptions), while they are polynomially reducible to each other.

Computation

In the following, we show how to calculate MCS diagnoses using HEX-programs (Eiter et al. 2005) which can be evaluated using the dlhex system.² HEX-programs extend disjunctive logic programs by allowing for access to external information with *external atoms*, and by *predicate variables* (which we disregard here). We consider only ground (variable-free) HEX-programs and simplify definitions.

Syntax of HEX-Programs. Let \mathcal{C} and \mathcal{G} be mutually disjoint sets of *constant names* and *external predicate names*, respectively. We note that constant names serve both as individual and predicate names.

An *ordinary atom* is a predicate $p(c_1, \dots, c_n)$ where p , and c_1, \dots, c_n are constants. An *external atom* is of the form

$$\&g[\vec{v}](\vec{w}),$$

where \vec{v} , and \vec{w} are fixed length lists of constants, and $\&g \in \mathcal{G}$ is an external predicate name. Intuitively, an external atom provides a way for deciding the truth value of the output tuple \vec{w} depending on the extension of input predicates \vec{v} .

A HEX rule r is of the form

$$\alpha_1 \vee \dots \vee \alpha_k \leftarrow \beta_1, \dots, \beta_m, \text{not } \beta_{m+1}, \dots, \text{not } \beta_n \quad (10)$$

$m, k \geq 0$, where all α_i are ordinary atoms and all β_j are ordinary or external atoms. Rule r is a *constraint*, if $k = 0$.

A HEX-program (or program) is a finite set of HEX rules.

Semantics of HEX-Programs. The ordinary *Herbrand base* HB_P^o of a HEX-program P contains all atoms $p(c_1, \dots, c_n)$ with a predicate p occurring in P and constants c_i from \mathcal{C} .

An *interpretation* of P is any subset $I \subseteq HB_P^o$; it is a *model* of

- an ordinary atom α , denoted $I \models \alpha$, if $\alpha \in I$.
- an external atom $\alpha = \&g[\vec{v}](\vec{w})$ (denoted $I \models \alpha$), if $f_{\&g}(I, \vec{v}, \vec{w}) = 1$, where $\vec{v} \in \mathcal{C}^n$, $\vec{w} \in \mathcal{C}^m$, and $f_{\&g}$ is a (fixed) function $f_{\&g} : 2^{HB_P^o} \times \mathcal{C}^{n+m} \rightarrow \{0, 1\}$.
- a rule r of form (10) ($I \models r$), if either $I \models \alpha_i$ for some α_i , or $I \models \beta_j$ for some $j \in \{m+1, \dots, n\}$, or $I \not\models \beta_i$ for some $i \in \{1, \dots, m\}$.
- a program P ($I \models P$), iff $I \models r$ for all $r \in P$.

The *FLP-reduct* (Faber et al. 2004) of P wrt. I is the set $fP^I \subseteq P$ of all rules r of form (10) in P such that $I \models \beta_i$, for all $i \in \{1, \dots, m\}$ and $I \not\models \beta_j$ for all $j \in \{m+1, \dots, n\}$. Then, I is an *answer set* of P , if I is a \subseteq -minimal model of fP^I . For P without external atoms, this amounts to answer sets as in (Gelfond and Lifschitz 1991).

More background information about HEX and how it relates to MCS is given in (Eiter et al. 2009).

Calculating Diagnoses. We use HEX-programs to describe a generic approach for calculating diagnoses, and a way for

²<http://www.kr.tuwien.ac.at/research/systems/dlhex/>

checking consistency of MCS. Then, we combine both approaches in order to calculate diagnoses more efficiently.

Generic Approach. We can calculate diagnoses for some MCS M by guessing a candidate diagnosis and checking whether it yields a consistent system. Due to Proposition 1, we only consider diagnoses (D_1, D_2) where $D_1 \cap D_2 = \emptyset$.

Program $P_D(M)$ for calculating diagnoses is as follows: for each bridge rule $r \in br_M$, we add a guess:

$$norm(r) \vee d_1(r) \vee d_2(r). \quad (11)$$

We “outsource” the diagnosis check into an external atom $\&eq_M[d_1, d_2]()$: the function $f_{\&eq_M}(I, d_1, d_2)$ returns 1 iff $M[br_M \setminus \{r \mid d_1(r) \in I\} \cup heads(\{r \mid d_2(r) \in I\})]$ has an equilibrium. The following constraint enforces that an answer set induces a diagnosis.

$$\leftarrow not \&eq_M[d_1, d_2](). \quad (12)$$

The resulting program properly captures MCS diagnoses.

Theorem 3. *The answer sets I of $P_D(M)$ correspond 1-1 to the diagnoses $(D_{I,1}, D_{I,2})$ of an MCS M s.t. $D_{I,1} \cap D_{I,2} = \emptyset$, where $I \Leftrightarrow D_I = (\{r \mid d_1(r) \in I\}, \{r \mid d_2(r) \in I\})$.*

Checking Consistency. Consistency of an MCS can be checked by calculating output-projected equilibria.

We assemble a program $P_p(M)$ as follows: we guess presence or absence of each output belief:

$$a_i(p) \vee \bar{a}_i(p). \quad \forall p \in OUT_i \quad (13)$$

We then evaluate each bridge rule (1) by two corresponding HEX rules, depending on previously guessed output beliefs:

$$b_i(s) \leftarrow not d_1(r), a_{c_1}(p_1), \dots, a_{c_j}(p_j), \\ not a_{c_{j+1}}(p_{j+1}), \dots, not a_{c_m}(p_m). \quad (14)$$

$$b_i(s) \leftarrow d_2(r). \quad (15)$$

Atoms $d_2(r)$ and $d_1(r)$ will become useful when integrating $P_p(M)$ with $P_D(M)$. For now, they do not occur in any rule head in the program, so (14) will not be deactivated by $d_1(r)$ and (15) will not become applicable.

Given an interpretation I , we use $A_i^I = \{p \mid a_i(p) \in I\}$, $1 \leq i \leq n$, to denote the set of output beliefs at context C_i (corresponding to the guess in (13)), and $B_i^I = \{s \mid b_i(s) \in I\}$ to denote the set of bridge rule heads at context C_i , activated by these output beliefs.

Finally, we ensure that answer sets of the program correspond to output-projected equilibria, by adding constraints

$$\leftarrow not \&con_out_i[a_i, b_i](). \quad (16)$$

for all $i = 1, \dots, n$. Each external atom in (16) represents \mathbf{ACC}_i ; it returns true iff context C_i accepts a belief set upon input of B_i^I , which corresponds to the guessed A_i^I after projection to OUT_i . Formally, $f_{\&con_out_i}(I, a_i, b_i) = 1$ iff there exists an $S \in \mathbf{ACC}_i(kb_i \cup B_i^I)$ s.t. $S \cap OUT_i = A_i^I$.

Proposition 5. *The answer sets I of $P_p(M)$ correspond 1-1 to the output-projected equilibria S' of M , where $I \Leftrightarrow S'_I = (S'_{1I}, \dots, S'_{nI})$ and $S'_{iI} = \{p \mid a_i(p) \in I\}$, $i = 1, \dots, n$.*

Combining both approaches. In order to calculate diagnoses using P_D , we need an algorithm for $f_{\&eq_M}$. Note that P_p can be used for this purpose. However, it is more efficient to directly integrate $P_p(M)$ and $P_D(M)$.

Let $P_p^D(M)$ be the program obtained by adding (11) to $P_p(M)$. We can show the following result.

Theorem 4. *For each diagnosis $(D_1, D_2) \in D^\pm(M)$ where $D_1 \cap D_2 = \emptyset$, at least one corresponding answer set of $P_p^D(M)$ exists. Each answer set I of $P_p^D(M)$ corresponds to a unique diagnosis D_I , as given in Theorem 3.*

This optimization removes one level of HEX indirection (compared to using P_D and within its function $f_{\&eq_M}$ using P_p), and allows for optimization of the guess by the solver.

Computing entailment-based explanations is a subject of future work; checking all pairs (R_1, R_2) is more involved and suggests to use saturation techniques.

Discussion and Conclusion

Related Work. Nonmonotonicity in MCSs was introduced by Roelofsen and Serafini (2005) and then further developed (Brewka, Roelofsen, and Serafini 2007, Brewka and Eiter 2007) to eventually allow heterogeneous as well as nonmonotonic systems.

Bikakis and Antoniou (2008) address inconsistency in MCSs by making bridge rules defeasible for inconsistency removal, i.e., a rule is applicable only if its conclusion does not cause inconsistency. The decision which bridge rules to ignore is based, for every context, on a *strict total order* of all contexts. This yields a unique diagnosis, whose declarative description is more involved than our notion, but which is polynomially computable. However, the strict total order forces the user to make (perhaps unwanted) decisions at design time; alternative orders require redesigns and separate runs. Our approach avoids this, but can be refined to respect also orders.

Debugging answer set programs, i.e., finding out why some program has no answer set, has been studied by Syrjänen (2006) and Brain et al. (2007), who developed notions of removal-based diagnoses. Their results can be used to compute (possibly constrained) diagnoses of an MCS, given that it has ASP contexts and uses the more restrictive grounded equilibria semantics (cf. Brewka and Eiter 2007).

Inoue and Sakama (1995) used abduction to repair theories in (nonmonotonic) logic using their notions of ‘explanation’ and ‘anti-explanation’. Given a theory K and abducibles Γ , they remove $O \subseteq \Gamma$ and add $I \subseteq \Gamma$ to entail (resp. not entail) an observation F ; i.e., $(K \cup I) \setminus O \models F$ (explanation), resp. $(K \cup I) \setminus O \not\models F$ (anti-explanation). A repair of an inconsistent K is given by an anti-explanation of $F = \perp$. Our notion of diagnosis amounts to a 2-sorted variant of such anti-explanations, where $O \subseteq \Gamma_O$ and $I \subseteq \Gamma_I$; under suitable conditions, it is reducible to ordinary anti-explanations. However, our notion of explanation has no counterpart.

Different from peer-to-peer data integration (Calvanese et al. 2008), besides disregarding that contexts may enter or leave the system, we do not repair inconsistency by ignoring inconsistent components or minorities in the system.

Conclusion. We presented two notions for explaining inconsistencies in multi-context systems, and showed a duality aspect of these notions. Moreover, we derived useful modularity and complexity results, and described the computation of one of the notions using HEX-programs.

Future work aims at scenarios with information hiding, the exploitation of modularity of explanations to reach more efficient means of calculation, and implementation.

Appendix: Proofs

Proof of Theorem 1. Let M be an MCS with bridge rules br_M . The complement w.r.t. br_M is denoted as $\mathbb{C}R := br_M \setminus R$, $(X, Y) \models \perp$ is shorthand for $M[X \cup heads(Y)] \models \perp$. For a pair X of sets of bridge rules, X_1 is the first component and X_2 the second. Let $D \in D_m^\pm$ and $E \in E_m^\pm$. By definition, (17) and (18) hold for D , and (19) holds for E :

$$(\mathbb{C}D_1, D_2) \not\models \perp \quad (17)$$

$$(\mathbb{C}D'_1, D'_2) \models \perp \quad \forall (D'_1, D'_2) \subset D \quad (18)$$

$$(E'_1, E'_2) \models \perp \quad \forall (\mathbb{C}E'_1, E'_2) \subseteq (\mathbb{C}E_1, \mathbb{C}E_2) \quad (19)$$

(\Rightarrow) We show that there exists $E^* \in E_m^\pm$ with $x \in E_i^*$, for $x \in D_i$ and $i \in \{1, 2\}$.

Case $x \in D_1$: consider $E = (\mathbb{C}(D_1 \setminus \{x\}), \mathbb{C}D_2)$. We observe $(\mathbb{C}(D_1 \setminus \{x\}), D_2) \models \perp$ from $D_1 \setminus \{x\} \subset D_1$ and (18). Similarly, for all (E'_1, E'_2) with $\mathbb{C}(D_1 \setminus \{x\}) \subseteq E'_1 \subseteq br_M$ and $E'_2 \subseteq E_2$, it follows that $(E'_1, E'_2) \models \perp$, and therefore $E \in E^\pm$ with $x \in E_1$ is a candidate for E^* .

It remains to show that there exists $E^* \subseteq E$ with $x \in E_1^*$ and $E^* \in E_m^\pm$. Assume there exists $E' \subseteq E$ with $x \notin E'_1$ and $E' \in E_m^\pm$. From $E'_1 \subset \mathbb{C}(D_1 \setminus \{x\})$ and $x \notin E'_1$ conclude $E'_1 \subseteq \mathbb{C}D_1$, thus $E' \subseteq D$, so $E' \notin E_m^\pm$ by (17).

Case $x \in D_2$: consider $P = (\mathbb{C}D_1, D_2)$. By (18) every (E'_1, E'_2) with $P_1 \subseteq E'_1 \subseteq br_M$, $E'_2 \subseteq P_2$, and $E' \neq P$ is an explanation. It remains to show that there exists $E^* \subset E'$ with $x \in E_2^*$ and $E^* \in E_m^\pm$.

Assume for contradiction, that all $E'' \subseteq E'$ with $E'' \in E_m^\pm$ are such that $x \notin E_2''$. From (18) follows for $P' = (\mathbb{C}D_1, D_2 \setminus \{x\})$ that $P' \models \perp$ as $P' \subset D$, but from (19) and $x \notin E_2''$ follows: if $(P'_1, P'_2 \cap \{x\}) \models \perp$ then $(P'_1, P'_2 \cup \{x\}) \models \perp$ given $P'_1 \subseteq P'_1$ and $P'_2 \subseteq P'_2 \subseteq br_M$. Specifically, $P' \models \perp$ implies $D \models \perp$, hence $D \models \perp$. This contradicts with $D \in D_m^\pm$.

(\Leftarrow) We show that there exists $D^* \in D_m^\pm$ with $x \in D_i^*$, for $x \in E_i$ and $i \in \{1, 2\}$.

Case $x \in E_1$: consider $S = \{A \setminus \{x\} \mid E_1 \subseteq A \subseteq br_M\}$. Let $S' = \{A \in S \mid \exists A_2 \subseteq \mathbb{C}E_2 : (A, A_2) \not\models \perp\}$, which completes (wrt. inconsistency) the lattice beginning at $(E_1 \setminus \{x\}, \mathbb{C}E_2)$ upwards in the first component and downwards in the second component. Assume $S' = \emptyset$. Then for all $A \in S$, $E'_2 \subseteq \mathbb{C}E_2$, it holds that $(A, E'_2) \models \perp$ and $E' = (E_1 \setminus \{x\}, E_2) \in E^\pm$, which contradicts $E \in E_m^\pm$ as $E' \subset E$. So $S' \neq \emptyset$. Select from S' a \subseteq -maximal set A_1 together with a \subseteq -minimal $A_2 \subseteq br_M$ s.t. $(A_1, A_2) \not\models \perp$. Then $(\mathbb{C}A_1, A_2) \in D_m^\pm$ as $\mathbb{C}A_1$ and A_2 are minimal, i.e., for all R', R'' with $A_1 \subset R' \subseteq br_M$ and $R'' \subseteq A_2$, it holds that

$(R', R'') \models \perp$. Furthermore, from $x \notin A_1$ it follows that $x \in \mathbb{C}A_1$.

Case $x \in E_2$: consider $D = (\mathbb{C}E_1, P)$ with $P \subseteq br_M$, $x \in P$, and $D \not\models \perp$. From $E \in E_m^\pm$ follows the existence of such D as otherwise E would not be minimal.

It remains to show that there exists $D^* \subseteq D$ with $x \in D_2^*$ and $D^* \in D_m^\pm$. Assume no such D^* exists, then making x unconditional does not remove any inconsistency. For $E' = (E_1, E_2 \setminus \{x\})$ and any R with $R \subseteq \mathbb{C}E'_2$ then follows $(E_1, R) \models \perp$. Hence $E' \in E^\pm$, which is contradicting $E \in E_m^\pm$ as $E' \subset E$. \square

Proof of Theorem 2 (sketch). This is a direct consequence of Theorem 1. Fix in its proof the second components of diagnoses and explanations to be \emptyset . \square

Proof of Proposition 1 (sketch). If the unconditional variant r' of a bridge rule r is present in M , it does not matter whether r is still contained in M , or not. Using Definition 3, we can therefore infer that, given $(D_1, D_2) \in D^\pm(M)$ and $\emptyset \neq X = D_1 \cap D_2$, it is true that $(D_1 \setminus X, D_2) \in D^\pm(M)$. Therefore every (D_1, D_2) with $D_1 \cap D_2 \neq \emptyset$ is not pointwise subset-minimal, and thus no minimal diagnosis. \square

Proof of Proposition 2 (sketch). Let U be a splitting set of $M[br_M]$, and let $E = (E_1, E_2) \in E_m^\pm(M[b_U])$ be a minimal explanation of an inconsistency in U . As U is a splitting set, no bridge rule of $br_M \setminus b_U$ can influence contexts in U . Therefore, $M[b_U] \models \perp$ implies $M[br] \models \perp$ for $b_U \subseteq br \subseteq br_M$. So $E \in E_m^\pm(M[br_M])$.

The proof is analogous for non-minimal explanations.

For (minimal) diagnoses we have to show that every (minimal) diagnosis $D \in D_{(m)}^\pm(M[b_U])$ can be extended to a (minimal) diagnosis $D' \in D_{(m)}^\pm(M[br_M])$. This is obvious for inconsistencies which only depend on rules of b_U , or on $br' = br_M \setminus b_U$. For those depending on both, only rules in br' can depend on rules in b_U as U is a splitting set. This dependency then is established via rules $br_d \subseteq br'$, and therefore can be removed by modifying rules of br_d . Hence an extension D' of D exists. \square

Proof of Proposition 3 (sketch). As U and U' are a partitioning of br_M such that b_U and $b_{U'}$ do not influence each other, it follows that every $E \in E_m^\pm(M[br_M])$ completely resides in either U or U' . \square

Proof of Proposition 4 (sketch). For a Boolean lattice \prec over X , we denote complement by \neg , and the pointwise extension of \prec to $(X \times X)$, in slight abuse of notation, is also denoted by \prec .

Definition 12. Given an MCS M and a lattice \prec over $P = 2^{br_M}$, a Lattice Diagnosis of M is a pair (D_1, D_2) , $D_1, D_2 \in P$, s.t. $M[\neg D_1 \cup heads(D_2)] \not\models \perp$. A lattice diagnosis (D_1, D_2) is minimal iff it is minimal wrt. \prec , i.e., for all $(D'_1, D'_2) \prec (D_1, D_2)$, holds $M[\neg D'_1 \cup heads(D'_2)] \models \perp$.

The proposition then follows from Theorem 1, Stone's Representation Theorem (by which all finite Boolean lattices are structurally the same), and the fact that the proof

of Theorem 1 relies on set-theoretic properties only, i.e., this proof can be rewritten to rely on \prec instead of \subset . \square

Proof of Lemma 1 (sketch). (\Rightarrow) Given an equilibrium $S = (S_1, \dots, S_n)$, the set H of active bridge rule heads at each context is determined by $app(br_i, S)$: $S_i \in \mathbf{ACC}(kb_i \cup H)$.

As bridge rule applicability only depends on output beliefs, we have that $app(br_i, (S_1|_{OUT_1}, \dots, S_n|_{OUT_n})) = app(br_i, (S_1, \dots, S_n))$. Therefore the projection S' of equilibrium S to output beliefs yields the same bridge rule heads as inputs for each context, so the same belief sets are accepted, and S' is an output-projected equilibrium.

(\Leftarrow) Given an output-projected equilibrium S' we have $S'_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) \mid r \in app(br_i, S')\})|_{OUT_i}$, so there exist at least one $S = (S_1, \dots, S_n)$ with $S_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) \mid r \in app(br_i, S')\})$. From the above insights about bridge rule applicability, we get that there exists a belief state S , s.t. $S_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) \mid r \in app(br_i, S)\})$. \square

Proof of Lemma 2 (sketch). We write $(A_1, A_2) \subset (B_1, B_2)$ iff \subseteq and \neq holds between the tuples.

We obtain the following Corollary from Definition 5.

Corollary 1. *Given an explanation $E = (E_1, E_2)$, all E' s.t. $E \subseteq E' \subseteq br_M \times br_M$ are explanations as well.*

(\Rightarrow) Assume $Q = (Q_1, Q_2)$ is a minimal explanation. Contrary to the Lemma, assume there exists another explanation Q' , which is either $(Q_1, Q_2 \setminus \{r\})$ with $r \in Q_2$ or $(Q_1 \setminus \{r\}, Q_2)$ with $r \in Q_1$. Then $Q' \subset Q$, therefore Q is not minimal, contradicting the assumption.

(\Leftarrow) Assume an explanation $Q = (Q_1, Q_2)$, and no pair $(Q_1, Q_2 \setminus \{r\})$ with $r \in Q_2$ or $(Q_1 \setminus \{r\}, Q_2)$ with $r \in Q_1$ is an explanation. Contrary to the Lemma, assume another explanation $P = (P_1, P_2)$ with $P \subset Q$.

By $P \subset Q$, either a) $P_1 \subset Q_1$ and $P_2 \subseteq Q_2$ or b) $P_1 \subseteq Q_1$ and $P_2 \subset Q_2$. For a) we create $T' = (Q_1 \setminus \{r\}, Q_2)$ for some $r \in Q_1 \setminus P_1$. Then $P \subseteq T' \subset Q$. Due to Corollary 1, T' is an explanation, contradicting the initial assumption. Similarly for b). \square

Proof of E_m^\pm recognition complexity result from Table 1. Given (E_1, E_2, M) with $\mathcal{CC}(M) = \mathbf{P}$, we show, that deciding $(E_1, E_2) \in E_m^\pm(M)$ is $\mathbf{D}_1^{\mathbf{P}}$ -complete.

Membership is shown by a Turing machine argument, where one machine decides whether a given input (E_1, E_2) is an explanation, while another one decides minimality.

$(E_1, E_2) \in E^\pm(M)$ can be decided in \mathbf{coNP} by a non-deterministic Turing machine: it guesses $R_1, R_2 \subseteq br_M$ and checks if $E_1 \subseteq R_2$ and $R_2 \subseteq br_M \setminus E_2$. If not, it immediately rejects, otherwise it decides consistency of $M[R_1 \cup heads(R_2)]$. All execution paths reject iff (E_1, E_2) is an explanation. Therefore explanation recognition is in \mathbf{coNP} .

Minimality of a given pair $(E_1, E_2) \in E^\pm(M)$ can be decided in \mathbf{coNP} using Lemma 2: a Turing machine checks whether all of $(E_1, E_2 \setminus \{r \mid r \in E_2\}) \notin E^\pm(M)$ and $(E_1 \setminus \{r \mid r \in E_1\}, E_2) \notin E^\pm(M)$ are true. Each check is in \mathbf{NP} , and \mathbf{NP} is closed under conjunction. The number of these E^\pm -checks is linear in the size of the instance. Therefore checking explanation minimality is in \mathbf{NP} .

Combining the above results, we get that minimal explanation recognition is in $\mathbf{D}_1^{\mathbf{P}}$ for \mathbf{P} contexts.

Hardness is shown by reducing the joint decision of a SAT instance $\phi(\vec{x})$, and an independent UNSAT instance $\psi(\vec{y})$, to minimal explanation existence as follows. Let M be an MCS with bridge rules

$$(1 : x_i) \leftarrow \top. \quad 1 \leq i \leq n \quad (20)$$

$$(1 : y_j) \leftarrow \top. \quad 1 \leq j \leq m \quad (21)$$

$$(1 : en) \leftarrow \top. \quad (22)$$

$$(1 : inc_phi) \leftarrow \top. \quad (23)$$

$$(1 : inc_psi) \leftarrow \top. \quad (24)$$

and let C_1 be a propositional logic context which is consistent iff $\kappa = \neg en \vee (\neg inc_phi \wedge \phi) \vee (\neg inc_psi \wedge \psi)$ evaluates to true.

M is inconsistent because (22), (23), and (24) together ensure that κ evaluates to false, $M[\emptyset]$ is consistent because $\neg en$ is true. For the same reason, $M[\{(23)\} \cup R]$ is consistent for $R = \emptyset$. Therefore $\omega = (\{(23)\}, \emptyset)$ is no explanation.

$M[\{(22)\} \cup R]$ is inconsistent for all $R \subseteq br_M$ iff both ϕ and ψ are unsatisfiable, because (a) if ϕ is satisfiable, we can set R exactly to those rules in (20) which correspond to the set of x_i which makes ϕ evaluate to true, thus making κ true, and (b) if ϕ is satisfiable, we can do the same wrt. y_i, ψ and (21). Therefore $\tau = (\{(22)\}, \emptyset)$ is an explanation of M iff ϕ and ψ are unsatisfiable. $M[\{(22), (23)\} \cup R]$ is inconsistent for all $R \subseteq br_M$ iff ψ is unsatisfiable, because (a) the presence of en and inc_phi ensure, that the first and second disjuncts in κ evaluate to false, respectively, and (b) the absence of inc_psi requires, that no set of bridge rules from (21) makes the third disjunct true, meaning that ψ is unsatisfiable. Therefore $\mu = (\{(22), (23)\}, \emptyset)$ is an explanation of M iff ψ is unsatisfiable. Furthermore, μ is a minimal explanation, if neither τ nor ω are explanations. If μ is an explanation and therefore ψ is unsatisfiable, τ is no explanation iff ϕ is satisfiable, therefore μ is a minimal explanation of M iff ϕ is satisfiable and ψ is unsatisfiable. Therefore minimal explanation recognition for $\mathcal{CC}(M) = \mathbf{P}$ is $\mathbf{D}_1^{\mathbf{P}}$ complete. \square

For proofs regarding HEX-programs, we rely on the Global Splitting Theorem for HEX (Eiter et al. 2006). In short, this theorem allows to decompose a HEX-program into an ordered set of program components which can be evaluated using the following principles: answer sets of ‘lower’ components do not depend on ‘higher’ components; answer sets of ‘lower’ components can be extended by ‘higher’ components to form answer sets of the whole program; answer sets of ‘lower’ components can be invalidated (by constraints or inconsistencies) in ‘higher’ components.

Proof of Theorem 3 (sketch). (\Rightarrow) The set of atoms in (11) is a Global Splitting Set of $P_D(M)$. Therefore the answer sets of (11) do not depend on (12), $D_{I,1} \cap D_{I,2} = \emptyset$ is true for all answer sets $I \in \mathcal{AS}(P_D(M))$, and for each $r \in br_M$, I contains exactly one of $d_2(r)$, $d_1(r)$, or $norm(r)$.

The program part (12) consists of constraints, so it can only eliminate certain answer sets created by (11). The definition of $f_{\&eq_M}$ essentially states, that all answer sets I

where D_I is not a diagnosis of M are eliminated. Therefore each answer set of $P_D(M)$ corresponds to a diagnosis.

(\Leftarrow) Given $(D_1, D_2) \in D^\pm(M)$ with $D_1 \cap D_2 = \emptyset$, the corresponding $I = \{norm(r) \mid r \in br_M, r \notin (D_1 \cup D_2)\} \cup \{d_1(r) \mid r \in D_1\} \cup \{d_2(r) \mid r \in D_2\}$ is an answer set of rules (11), and $f_{\&eq_M}(I, d_1, d_2)$ returns 1, as $(D_1, D_2) \in D^\pm(M)$. Therefore I is an answer set of $P_D(M)$. \square

Proof of Proposition 5 (sketch). (\Rightarrow) We show that each answer set $I \in \mathcal{AS}(P_p(M))$ uniquely corresponds to an output-projected equilibrium S' of M . The Global Splitting Theorem splits $P_p(M)$ into P_1 consisting of (13), P_2 consisting of (14) and (15), and P_3 consisting of constraints (16). P_1 has as answer sets I_1 all output-projected belief states $S'_I = (S'_{I1}, \dots, S'_{In})$ of M . P_2 extends each answer set I_1 by the set of bridge rule heads B_i^I that are active at context C_i given S' . P_3 eliminates all answer sets generated by P_1 and P_2 , where for a certain i , $1 \leq i \leq n$, there exists no $S_i \in \mathbf{ACC}_i(kb_i \cup B_i^I)$ s.t. $S_i \cap OUT_i = S'_{iI}$. Therefore each S'_I corresponding to an answer set $I \in P_p(M)$ is an output-projected equilibrium of M .

(\Leftarrow) We construct a unique answer set $I \in \mathcal{AS}(P_p(M))$ from an output-projected equilibrium S' : given an output-projected equilibrium $S' = (S'_1, \dots, S'_n)$, we calculate bridge rule applicability, yielding the set of active bridge rule heads H_i at each context C_i . We then assemble I as follows: $I = \{a_i(p) \mid p \in S'_i\} \cup \{\bar{a}_i(p) \mid p \in OUT_i \setminus S'_i\} \cup \{b_i(s) \mid s \in H_i\}$. By construction and by the fact that S' is an output-projected equilibrium, I satisfies all rules in $P_p(M)$, in particular the constraints (16). Therefore I also satisfies the reduct $f_{P_p(M)}^I$. As $P_p(M)$ contains no positive cycles, I is a minimal model of the reduct and thus an answer set. \square

Proof of Theorem 4 (sketch). This proof is very similar to the previous proof: $P_p^D(M)$ can be split into parts P_1 , P_2 , P_3 , and a new part P'_1 consisting of rules (11). P_1 and P'_1 are independent from each another and from the other parts. An answer set I' of (11) influences (14) and (15), corresponding to deactivated ($d_1(r) \in I'$) or unconditionally added ($d_2(r) \in I'$) bridge rules. The rest of $P_p^D(M)$ eliminates all answer sets I' which do not yield at least one equilibrium in the modified M . Therefore an answer set I of $P_p^D(M)$ corresponds to an output-projected equilibrium of M after application of a diagnosis, and this diagnosis is indicated in I by $d_1(r)$ and $d_2(r)$ atoms. \square

References

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge, UK: Cambridge University Press.

Bikakis, A., and Antoniou, G. 2008. Distributed defeasible contextual reasoning in ambient computing. In *Ambient Intelligence*, 308–325.

Brain, M.; Gebser, M.; Pührer, J.; Schaub, T.; Tompits, H.; and Woltran, S. 2007. Debugging ASP programs by means of ASP. In *Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 31–43.

Brewka, G., and Eiter, T. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In *AAAI Conference on Artificial Intelligence (AAAI)*, 385–390.

Brewka, G.; Roelofsen, F.; and Serafini, L. 2007. Contextual default reasoning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 268–273.

Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2008. Inconsistency tolerance in P2P data integration: An epistemic logic approach. *Information Systems* 33(4-5):360–384.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3):374–425.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2005. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *International Joint Conference on Artificial Intelligence*, 90–96.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2006. Effective integration of declarative rules with external evaluations for semantic-web reasoning. In *European Semantic Web Conference (ESWC)*, 273–287.

Eiter, T.; Brewka, G.; Dao-Tran, M.; Fink, M.; Ianni, G.; and Krennwallner, T. 2009. Combining nonmonotonic knowledge bases with external sources. In *Frontiers of Combining Systems (FroCoS)*, 18–42.

Faber, W.; Leone, N.; and Pfeifer, G. 2004. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *European Conference on Logics in Artificial Intelligence (JELIA)*, 200–212.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9(3/4):365–386.

Giunchiglia, F., and Serafini, L. 1994. Multilanguage hierarchical logics, or: How we can do without modal logics. *Artificial Intelligence* 65(1):29–70.

Inoue, K., and Sakama, C. 1995. Abductive framework for nonmonotonic theory change. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 204–210.

Lifschitz, V., and Turner, H. 1994. Splitting a logic program. In *International Conference on Logic Programming (ICLP)*, 23–37.

Przymusiński, T. 1991. Stable semantics for disjunctive programs. *New generation computing* 9(3):401–424.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32:57–95.

Roelofsen, F., and Serafini, L. 2005. Minimal and absent information in contexts. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 558–563.

Syrjänen, T. 2006. Debugging inconsistent answer set programs. In *International Workshop on Nonmonotonic Reasoning (NMR)*, 77–83.