# A general method for diagnosing axioms

**Gerhard Friedrich, Stefan Rass,** and **Kostyantyn Shchekotykhin**
Universitaet Klagenfurt, Universitaetsstrasse 65, 9020 Klagenfurt, Austria, Europe
firstname.lastname@ifit.uni-klu.ac.at

## Abstract

Full support of debugging knowledge bases must not stop at the level of axioms. In this work, we present a general theory for diagnosing faulty knowledge bases, which not only allows the identification of faulty axioms, but also the pinpointing of those parts of axioms, which must be changed. Based on our theory, we present methods for computing these diagnoses and show the feasibility by extensive test evaluations. The proposed approach is broadly applicable, since it is independent of a particular logical language (with monotonic semantics) and independent of a particular reasoning system.

## 1 Introduction

A broad adoption of knowledge based systems requires effective methods to support the test and debug cycle of knowledge bases. In this cycle the knowledge engineer has to diagnose the knowledge base (KB) in order to identify those parts, which must be changed such that the intended behavior is achieved. This task becomes challenging even in moderately sized knowledge bases with hundreds of axioms. Therefore, considerable research effort [Schlobach and Cornet, 2003; Parsion *et al.*, 2005; Friedrich and Shchekotykhin, 2005; Haase *et al.*, 2005; Wang *et al.*, 2005] was put into the improvement of debugging. All these approaches have in common that they either are not based on a well-founded diagnosis theory or consider an axiom as the smallest entity, which could be faulty. Consequently, no general diagnosis approach exists that identifies those parts of axioms which must be changed such that all tests and requirements for a KB are fulfilled. Diagnosing axioms becomes especially important in knowledge bases, where axioms are of remarkable size, e.g. the Galen ontology[1] comprises axioms with more than 100 arguments in a logical operator. In this case, debugging is still difficult even if diagnosis provides a set of axioms that need further investigation by the knowledge engineer.

Consequently, an important research question is, whether it is possible to develop a general theory and practically applicable algorithms for the diagnosis of knowledge bases, which improve the resolution of diagnoses by identifying faulty parts of axioms. Such an algorithm could be very supportive as an interactive debugging tool for knowledge engineers who want to inspect diagnoses and axioms on a more fine grained resolution level.

We address this question by the development of a general theory for the diagnosis of knowledge bases, which are expressed in a declarative knowledge representation language. This theory not only allows the identification of faulty axioms, but also the faulty parts of axioms. Furthermore, this theory is applicable for all declarative knowledge representation languages, which are based on a variant of first-order logic (FOL), e.g. description logic (DL) or the OWL language family, which plays an important role for the implementation of the Semantic Web. We will show that this theory is a generalization of the diagnosis theory of [Friedrich and Shchekotykhin, 2005], which on one hand provides a general theory for diagnosing knowledge bases but on the other hand considers just axioms as the finest granularity of diagnoses. For the implementation, we employ a transformation of the axioms in a set of axioms such that the original diagnosis algorithms presented in [Friedrich and Shchekotykhin, 2005] can be applied. As a consequence, this approach provides a sound and complete method for the generation of diagnoses for axioms independently of a particular reasoning system. Therefore, we are not limited to special properties of the knowledge bases (e.g. acyclic) or restrictions of the representation language. We present enhancements of this algorithm, which lead to considerable improvements of the running time for the diagnosis of axioms. Finally, we show the feasibility of our methods by exploiting a standard test library.

In the following Section, we present our basic idea to diagnose axioms. In Section 3, we introduce a general theory for diagnosing axioms and relate this theory to the existing theory of diagnosing knowledge bases. The foundation for computing axiom diagnoses is given in Section 4, followed by an evaluation in Section 5. The paper closes with a discussion of related work and final conclusions.

[1]A test version is included in a benchmark suite for description logic, e.g. see RACER's version at `http://www.racer-systems.org`

## 2 Limitation of current approach

For the introduction of our concepts, we consider the DL knowledge base[2] `bike2.tkb`, available through the benchmark suite for the RACER system. This KB comprises 154 axioms. Let us assume that in the knowledge acquisition process one of the axioms was incorrectly stated. In our exemplification, the axiom defining concept `C13` is incorrectly specified, i.e. in the depicted Axiom 66 the correct expression `(SOME R11 C75)` has been replaced by the faulty expression `(ALL R11 C75)`:

```
[66:]  (DEFINE-CONCEPT C13
       (AND (SOME R22 C63)
         (SOME R11 C74) (ALL R11 C75)
         (AT-MOST 3 R11) (AT-LEAST 2 R11)
         (SOME R14 *TOP*) (SOME R30 *TOP*)
         (AT-LEAST 2 R19) (SOME R4 *TOP*)
         (SOME R23 *TOP*) (SOME R2 *TOP*)))
```

As a consequence the KB `bike2.tkb` becomes incoherent because of Axiom 66 and the axioms

```
[145:](IMPLIES C74 (NOT C75))
[146:](IMPLIES C75 (NOT C74))
```

A knowledge base is incoherent, iff there exists a concept or role which is incoherent. A concept or role is incoherent, iff it has an empty extension in all models. In our example `C13` is incoherent in `bike2.tkb`. The sets of axioms $\langle$`66, 145`$\rangle$ and $\langle$`66, 146`$\rangle$ are the minimal incoherent subsets of the example KB. Such sets are called *minimal conflicts* in the terminology of model-based diagnosis [Friedrich and Shchekotykhin, 2005]. As expected, the diagnosis-engine of [Friedrich and Shchekotykhin, 2005] correctly returns Axiom `66` as the only single fault diagnosis. More formally, a KB-Diagnosis problem is defined as follows.

**Definition 1 (KB-Diagnosis Problem)** *A* KB-Diagnosis Problem *(Diagnosis Problem for a Knowledge Base, [Friedrich and Shchekotykhin, 2005]) is a tuple $(KB, B, TC^+, TC^-)$, where $KB$ is a knowledge base, B is a background theory, $TC^+$ is a set of positive and $TC^-$ a set of negative test cases, which the $KB$ has to be consistent or inconsistent with, respectively. The test cases are given as sets of logical sentences. We assume that each test case and the background theory on its own are consistent.*

The principal idea of the following definition of a diagnosis for a KB is to find a set of axioms that must be changed (respectively deleted) and, possibly, some axioms that must be added s.t. all test cases are satisfied. The symbol $\perp$ expresses a contradiction.

**Definition 2 (KB-Diagnosis)** *[Friedrich and Shchekotykhin, 2005] A* KB-Diagnosis *for a KB-Diagnosis Problem $(KB, B, TC^+, TC^-)$ is a set $D \subseteq KB$ of sentences such that there exists an extension $EX$, where $EX$ is a set of logical sentences added to the knowledge base, such that*

1. $\forall e^+ \in TC^+ : (KB - D) \cup B \cup EX \cup e^+ \not\models \perp$
2. $\forall e^- \in TC^- : (KB - D) \cup B \cup EX \cup e^- \models \perp$

A minimal diagnosis is a diagnosis such that no proper subset is a diagnosis. A minimum cardinality diagnosis is a diagnosis such that there exists no diagnosis with smaller cardinality.

According to [Friedrich and Shchekotykhin, 2005, Corollary 1], we may characterize $EX$ by the conjunction of all negated negative test cases. In particular, $D$ is a diagnosis for $(KB, B, TC^+, TC^-)$ *iff* $\forall e^+ \in TC^+ : (KB - D) \cup B \cup e^+ \cup \bigwedge_{e^- \in TC^-} (\neg e^-)$ is consistent. In order to keep the example simple, we do not specify a background theory or test cases; but we require coherence and consistency.

Presenting Axiom `66` as the only minimal single fault diagnosis does not provide information about the parts of Axiom `66`, which caused the incoherence. In particular, we would like to identify those parts of a faulty axiom, which must be changed such that the requirements (e.g. coherence and compliance with test cases) are fulfilled.

In order to achieve this task, we base our principal idea on the observation that axioms are composed by structures according to a predefined grammar. E.g. Axiom `66` consists of an `AND`-structure with 11 arguments. Each argument is a structure itself. In general, such structures are defined by a grammar, where the terminal symbols are literals. Exploiting this observation, we recognize various possibilities for restoring coherence. Either the `AND`-structure must be changed (e.g. deleting the second argument of the `AND`-structure) or one of its arguments. Analyzing these arguments, we recognize that only the arguments `(SOME R11 C74)` and `(ALL R11 C75)` are relevant for producing an incoherent KB. Changes to arguments or operators of these structures will resolve the incoherence, e.g. by replacing the `ALL` operator by a `SOME` operator or by changing the names of concepts or roles. As a consequence, we can pinpoint parts of the axiom, which must be changed, thus exonerating the greater part of Axiom `66`. In the following, we will generalize the ideas presented in the example.

## 3 Diagnosis of axioms

The refinement of a KB-Diagnosis is based on the structure of the axioms according to the underlying syntax of the knowledge representation language. For our methods, we assume that the syntax of the language is defined by a context-free grammar $G$. In particular, we assume the usual structuring of logic-based languages expressed by production rules according to the following prototypical rule:

$$V_0 \rightarrow op(V_1, \ldots, V_n\{, N_j\}^*),$$

where $V_0$ is a non-terminal, and the right-hand side is a *logical structure* defined by a logical operator $op$ and arguments $V_1, \ldots, V_n$, which are logical structures. $\{, N_j\}^*$ denotes a possibly empty list of *non-logical arguments*, i.e.

[2] We assume the reader to be familiar with the basics of description logic. For otherwise, [Baader *et al.*, 2003] provides an excellent introduction. In addition, we would like to stress that using DL knowledge bases as examples does not imply any limitation of our approach to DL. We used DL because of its importance for the Semantic Web.

numbers. Logical structures correspond to literals or otherwise can be recursively composed by applying operators to simpler structures. Consequently, a logical structure is either an operator application or a literal. Context free grammars may have also single non-terminal symbol on the right-hand side ($V_0 \rightarrow V_m$). In this case the syntax-tree [Linz, 1996] comprises intermediate nodes, which can be replaced by the successor of this node. Therefore, we assume the following structuring of axioms, where $L$ is a logical structure, and $LI$ is a literal:

$$L \rightarrow op(L, \ldots, L\{, N_j\}^*)|LI$$

Note that $op$ depends on the language. In case of FOL, $op$ corresponds to the usual logical connectives and to quantifications of variables (e.g. $\exists\ x$). However, in DL, $op$ corresponds to one of the logical operators defined in specific DL variants (e.g. `ALL`, `SOME`, `AND`, `OR`). Furthermore, a LISP-like notation may be chosen. A simple grammar used for DL within the RACER system can be found in [Patel-Schneider and Swartout, 1993], e.g.

$$C_0 \rightarrow CN \mid (\texttt{AND}\ C_1 \ldots C_n) \mid (\texttt{ALL}\ R\ C_0) \mid \ldots$$

where $CN$ is an atomic concept (i.e. a literal), $C_i$ are concepts and $R$ is a role.

In addition to logical arguments $V_i$ of an operator, a language may exploit *non-logical* arguments $N_j$, which are modifying the meaning of this operator e.g. a DL could comprise the `(AT-LEAST N R)` operator, where `N` is a natural number and `R` is a role. Viewing it as a first order logic statement, a logical structure is introduced, which depends on `N`. Regarding the meaning, we make the usual assumption that the semantics of *logical structures* is given denotationally using an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a domain (non-empty universe) of values and $(\cdot)^{\mathcal{I}}$ is an interpretation function. Literals are subsets of $\Delta^{\mathcal{I}}$ or relations over $\Delta^{\mathcal{I}}$, and the semantics is inductively defined. Since we are interested in a general theory for diagnosing logical knowledge bases, we constraint a semantics as little as possible. In order to deal with variable symbols, we have to define a partial function $\mu$, which provides a substitution of some variables by elements of $\Delta^{\mathcal{I}}$. We assume that the semantics of $op$ is defined by $op(V_1, \ldots, V_n\{, N_j\}^*)^{\mathcal{I},\mu} := op(V_1^{\mathcal{I},\mu}, \ldots, V_n^{\mathcal{I},\mu}\{, N_j\}^*)$, where $op$ is defined as a partial function that maps $V_1^{\mathcal{I},\mu}, \ldots, V_n^{\mathcal{I},\mu}\{, N_j\}^*$ to a value, depending on the logic defined, e.g. to truth values in case $op$ is a logical connective in FOL or subsets of $\Delta^{\mathcal{I}}$ in case $op$ is an operator of a DL. An axiom is satisfied by $\mathcal{I}, \mu$ if it is true. Note that non-logical structures are not interpreted by $\mathcal{I}$. E.g. the semantics of $(\texttt{AT-LEAST}\ N\ R)^{\mathcal{I}}$ is defined as $\{a \in \Delta^{\mathcal{I}} \big|\ |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \geq N\}$, where $N$ is a non-logical argument. Finally, we assume a monotonic semantics of the logic.

As a consequence, every axiom can be represented by its syntax-tree (cf. fig. 1. Every operator, literal, and non-logical argument of an axiom is represented by a node. There is a directed arc $n_1 \longrightarrow n_2$ from node $n_1$ to node $n_2$, iff $n_2$ is an argument of $n_1$. We then say that $n_1$ is the predecessor of $n_2$. More generally, we can view the whole KB as one tree (called the $KB$-tree), where the first level (root node) is

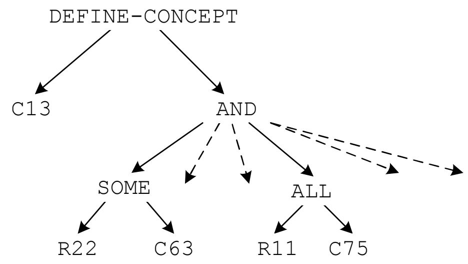


Figure 1: Syntax tree for our example axiom. Arcs represent relationships between operators and their corresponding arguments.

the start symbol of the grammar, and the second level corresponds to the operators (logical structures) defining axioms (e.g. `DEFINE-CONCEPT`). A complete subtree rooted at a node in the $KB$-tree defines a *logical expression*. An axiom is a logical expression itself.

Based on this representation, we can generalize the concept of *KB-Diagnosis* in order to identify faulty logical structures. For this generalization, let every logical structure $L$ (which is either represented by an operator or by a literal) of a $KB$ and its associated logical expression $E$ be uniquely identified by a marker $M(L)$ rsp. $M(E)$. The following example shows a possible marking of Axiom 66:

$(\texttt{DEFINE-CONCEPT}_{66.0}\ \texttt{C13}_{66.1}$
$(\texttt{AND}_{66.2}\ (\texttt{SOME}_{66.3}\ \texttt{R22}_{66.4}\ \texttt{C63}_{66.5}\ )$
$\quad (\texttt{SOME}_{66.6}\ \texttt{R11}_{66.7}\ \texttt{C74}_{66.8}\ ) \ldots ))$

Furthermore, we will exploit a replacement operator $KB[L/R]$, where the logical structure $L$ is replaced by a logical expression $R$ in the knowledge base $KB$. Replacements of logical structures are regarded as repairs for faulty descriptions. Therefore, we will only consider syntactically valid replacements of a structure. In our example some of these *syntactically valid* replacements for $\texttt{AND}_{66.2}$ could be the change to an `OR` operator, the addition of a `NOT` before the `AND`, or the introduction of a new operator (e.g. an `OR`) combining some arguments. $KB[L_1/R_1, \ldots, L_n/R_n]$ denotes the simultaneous replacement of $L_1, \ldots, L_n$ by $R_1, \ldots, R_n$. The principal idea of the following definition is to identify those logical structures, which are the cause for not satisfying the requirements (e.g. failed test cases or incoherent KB). A set of logical structures is a cause for not satisfying the requirements iff there exists a replacement of these structures (a possible repair) such that all requirements are satisfied. More formally:

**Definition 3 (AX-Diagnosis)** *Let $LS$ be the set of logical structures of the knowledge base $KB$. An AX-Diagnosis for a KB-Diagnosis Problem $(KB, B, TC^+, TC^-)$ is a set $D = \{L_1, \ldots, L_n\} \subseteq LS$ such that there exist syntactically valid replacements $R_i$ for each logical structure $L_i \in D$ and an extension $EX$, where $EX$ is a set of logical sentences added to the knowledge base, such that*

1. $\forall e^+ \in TC^+$: $KB[L_1/R_1, \ldots, L_n/R_n] \cup B \cup EX \cup e^+ \not\models \perp$

2. $\forall e^- \in TC^-$: $KB[L_1/R_1, \ldots, L_n/R_n] \cup B \cup EX \cup e^- \models \perp$

We say that such a replacement $KB[L_1/R_1, \ldots, L_n/R_n]$ *clears* (or repairs) the fault. A minimal AX-Diagnosis $D$ is defined as usual by requiring that no subset $D' \subset D$ is an AX-Diagnosis. Likewise, $D$ is a minimum cardinality AX-Diagnosis, if there exists no AX-Diagnosis with smaller cardinality than $D$.

In our running example, there exist eight minimum cardinality (single fault) AX-diagnoses, marked by boxes:

$$
\begin{array}{l}
(\boxed{\texttt{DEFINE-CONCEPT}}_{66.0}\ \texttt{C13}\\
(\ \boxed{\texttt{AND}}_{66.2}\ \texttt{(SOME R22 C63)}\\
\qquad (\boxed{\texttt{SOME}}_{66.6}\ \boxed{\texttt{R11}}_{66.7}\boxed{\texttt{C74}}_{66.8})\\
\qquad (\ \boxed{\texttt{ALL}}_{66.9}\ \boxed{\texttt{R11}}_{66.10}\ \boxed{\texttt{C75}}_{66.11})\\
\qquad \texttt{(AT-MOST 3 R11) (AT-LEAST 2 R11)}\\
\qquad \texttt{(SOME R14 *TOP*) (SOME R30 *TOP*)}\\
\qquad \texttt{(AT-LEAST 2 R19) (SOME R4 *TOP*)}\\
\qquad \texttt{(SOME R23 *TOP*) (SOME R2 *TOP*)))}
\end{array}
$$

Note that in fact each of these structures can be replaced in order to restore the coherence of `bike2`, e.g. in $\texttt{AND}_{66.2}$ the second and third argument could be replaced by a new logical structure combining them by an `OR`, the concepts and roles $\texttt{C74}_{66.8}$, $\texttt{C75}_{66.11}$, $\texttt{R11}_{66.7}$, and $\texttt{R11}_{66.10}$ could be replaced by other roles and concepts not mentioned in `bike2`. $\texttt{ALL}_{66.9}$ could be replaced by `SOME` and $\texttt{SOME}_{66.6}$ may be preceded by a `NOT`. Finally, with respect to a replacement of $\texttt{DEFINE-CONCEPT}_{66.0}$, a complete deletion of Axiom 66 (the axiom may be out-dated) is a possible repair.

Note that replacements of the logical structures in the example axiom, which are not one of the eight single fault diagnoses do not restore the coherence of `bike2`. More generally, logical structures that are not contained in minimal diagnoses need not be considered for replacement. Consequently, a large fraction of the example axiom does not require further investigations by the knowledge engineer.

Note that if we focus the diagnosis process on minimum cardinality AX-Diagnoses then the exoneration of logical structures can be extended. Let $N$ be the cardinality of the minimum cardinality diagnoses. Then it is clear that any replacement of a logical structure not contained in a minimum cardinality diagnosis can only resolve the fault if at least $N + 1$ repairs are performed in total.

As it is generally the case in diagnosis, additional information as test cases and extensions to the background theory could be exploited to reduce the number of most likely diagnoses (e.g. the number of minimum cardinality diagnoses). In addition, one could imagine a more sophisticated estimation regarding the likelihood of AX-diagnoses. However, since we focus on the foundations of diagnosing axioms both tasks are out of the scope of this work.

In case the knowledge engineer wants to change a logical structure in addition to those contained in a minimal AX-Diagnosis, then no additional changes are necessary because of the following property:

**Remark 1** *Every superset of a minimal AX-diagnosis for a KB-Diagnosis Problem is an AX-diagnosis.*

Since a replacement of a logical structure by a logical expression may also include changes in the arguments, the concept of AX-Diagnosis shares some similarities with hierarchical diagnosis:

**Remark 2** *Let* $D = \{L_1, \ldots, L_i, \ldots, L_n\}$ *be an AX-Diagnosis for a KB-Diagnosis Problem* $(KB, B, TC^+, TC^-)$*, and let the logical structure* $L'_i$ *be a predecessor of* $L_i$ *in the* $KB$*-tree. Then* $\{L_1, \ldots, L'_i, \ldots, L_n\}$ *(i.e. replacing* $L_i$ *by* $L'_i$ *in D) is an AX-Diagnosis.*

Roughly speaking, if we regard logical arguments as sub components, then the previous remark says that a faulty sub component also implies a faulty super component. However, the converse is not necessarily true. This converse is usually assumed in hierarchical diagnosis, i.e. if a super component is faulty then at least one of its sub components is faulty. An AX-Diagnosis $\{L_1, \ldots, L'_i, \ldots, L_n\}$ might contain a logical structure $L'_i$ for, which there does not exist an AX-Diagnosis $\{L_1, \ldots, L_i, \ldots, L_n\}$, where $L'_i$ is replaced by a successor $L_i$ w.r.t. the $KB$-tree, e.g. operators could be wrongly defined leading to an inconsistency independently of the logical arguments. In the following section, we will show the basic methods for the computation of $AX$-Diagnoses.

## 4 Computation of Axiom Diagnoses

The principal idea of diagnosing axioms, is to translate an axiom into a set of axioms, which allows the application of the diagnosis methods introduced in [Friedrich and Shchekotykhin, 2005]. For the translation, we assume that the logic contains the equivalence operator (which may be simulated by exploiting implication and conjunction). More formally we require $(V_1 \equiv V_2)^{\mathcal{I},\mu} := (V_1^{\mathcal{I},\mu} = V_2^{\mathcal{I},\mu})$. We apply this operator in order to decompose logical expressions. Let $E_i^{\overline{y}_i}$ be logical expressions with free variables $\overline{y}_i$ and $X_i(\overline{y}_i)$ be a unique literal with variables $\overline{y}_i$ as arguments. A logical expression $op(E_1^{\overline{y}_1}, \ldots, E_n^{\overline{y}_n}\{, N_j\}^*)$ is replaced by $op(X_1(\overline{y}_1), \ldots, X_n(\overline{y}_n)\{, N_j\}^*)$ and a set of additional axioms $\{X_1(\overline{y}_1) \equiv E_1^{\overline{y}_1}, \ldots, X_n(\overline{y}_n) \equiv E_n^{\overline{y}_n}\}$. A complete decomposition for our sample Axiom `66` would be the following:

$$
\begin{array}{ll}
\texttt{[66.0:]} & \texttt{(DEFINE-CONCEPT X}_{66.1}\ \texttt{X}_{66.2}\texttt{)}\\
\texttt{[66.1:]} & \texttt{(EQUIVALENT X}_{66.1}\ \texttt{C13)}\\
\texttt{[66.2:]} & \texttt{(EQUIVALENT X}_{66.2}\ \texttt{(AND X}_{66.3}\ \texttt{X}_{66.6}\ \ldots\ \texttt{))}\\
\texttt{[66.3:]} & \texttt{(EQUIVALENT X}_{66.3}\ \texttt{(SOME X}_{66.4}\ \texttt{X}_{66.5}\texttt{))}\\
\texttt{[66.4:]} & \texttt{(ROLES-EQUIVALENT X}_{66.4}\ \texttt{R22)}\\
\texttt{[66.5:]} & \texttt{(EQUIVALENT X}_{66.5}\ \texttt{C63)}\\
\texttt{[66.6:]} & \texttt{(EQUIVALENT X}_{66.6}\ \texttt{(SOME X}_{66.7}\ \texttt{X}_{66.8}\texttt{))}\\
 & \ldots\\
\texttt{[66.32:]} & \texttt{(EQUIVALENT X}_{66.32}\ \texttt{*TOP* )}
\end{array}
$$

This transformation preserves the interpretation of the original logical expression.

**Remark 3** *Let the logical expression* $op(E_1^{\overline{y}_1}, \ldots, E_n^{\overline{y}_n}\{, N_j\}^*)$ *be decomposed as described above,* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}} \rangle$ *be an interpretation, and* $\mu$ *a variable substitution for the free variables. Then* $op(X_1(\overline{y}_1), \ldots, X_n(\overline{y}_n)\{, N_j\}^*)^{\mathcal{I},\mu} = op(E_1^{\overline{y}_1}, \ldots, E_n^{\overline{y}_n}\{, N_j\}^*)^{\mathcal{I},\mu}$.

FUNCTION DECOMP($E$): **Return** a set of axioms
**Input:** a logical expression $E$
**if** $E$ is a literal with free variables $\overline{y}$ **then**
  **if** $E$ is an axiom **then return** $\{[M(E):]E\}$
  **else return** $\{[M(E):]X_{M(E)}(\overline{y}) \equiv E\}$
**else let** $E = op(E_1, \ldots, E_n\{, N_j\}^*)$, **where**
  $\overline{y_i}$ are the free variables of $E_i$, and
  $y = \bigcup_i \overline{y_i}$ are the free variables of $E$
  **if** $E$ is an axiom **then return**
    $\{[M(E):]op(X_{M(E_1)}(\overline{y_1}), \ldots, X_{M(E_n)}(\overline{y_n})\{, N_j\}^*)\} \cup \bigcup_{i=1,\ldots,n} \text{DECOMP}(E_i)$
  **else return** $\{[M(E):]X_{M(E)}(\overline{y}) \equiv op(X_{M(E_1)}(\overline{y_1}), \ldots, X_{M(E_n)}(\overline{y_n})\{, N_j\}^*)\} \cup \bigcup_{i=1,\ldots,n} \text{DECOMP}(E_i)$

Figure 2: Decomposition of an axiom

The complete decomposition of an axiom and all its subexpressions can be performed by the recursive function depicted in Figure 2. We exploit the markers of the logical expressions $E$ of a $KB$ to mark corresponding axioms by $[M(E):]$.

Based on a decomposition of the axioms of a KB, we can show that there is a one-to-one correspondence between the *KB-Diagnoses* of the *decomposed* knowledge base and the *AX-Diagnoses* of the *original* one. For this correspondence, we make the reasonable assumption that for every logical structure there is a syntactically valid replacement by a unique literal. This assumption holds for FOL and DL. Following definition 3, we use the newly introduced literals as replacements $R_i$ for the logical sub-structures $L_i$ of the original axiom in $KB$. Let every axiom $Ax$ of the decomposition be identified by $M(Ax)$ and $\text{DECOMP}(KB) = \bigcup_{Ax \in KB} \text{DECOMP}(Ax)$ is the decomposition of the knowledge base $KB$.

**Proposition 1** *Provided the knowledge representation language allows for every logical structure a syntactically valid replacement by a literal,* $D_{AX} = \{L_1, \ldots, L_n\}$ *is an* ***AX****-Diagnosis for the KB-Diagnosis Problem* $(KB, B, TC^+, TC^-)$*, iff* $D_{KB} = \{M(L_1), \ldots, M(L_n)\}$ *is a* ***KB****-Diagnosis for the KB-Diagnosis Problem* $(\text{DECOMP}(KB), B, TC^+, TC^-)$.

Note that the inverse transformation of the composition can be obtained by backsubstituting the equivalent expressions for each newly introduced literal. For our example, the original axiom arises by substituting the expressions for $\text{X}_{66.1}$ (which is `C13`) and $\text{X}_{66.2}$ (which is (`AND` $\text{X}_{66.3}$ $\text{X}_{66.6}$ $\ldots$)), and recursively substituting the equivalent expressions for all X-literals therein. By this backsubstitution, we are able to pinpoint the logical structures in the axioms, which correspond to the elements of *AX-Diagnoses*. Furthermore, it is not needed to decompose the whole knowledge-base, but to apply the decomposition on demand, e.g. if the knowledge engineer has an interest to investigate an axiom more deeply, because on the level of axioms, a single leading diagnosis was identified.

As a consequence, we can exploit all the methods for computing KB-Diagnoses as described in [Friedrich and Shchekotykhin, 2005] for the computation of axiom diagnoses. Based on Proposition 1, it follows easily that the complete and correct algorithm of [Friedrich and Shchekotykhin, 2005] for the generation of minimal KB-Diagnoses can be applied to generate soundly the set of all minimal AX-Diagnoses. It remains the question regarding the practical applicability, which will be addressed in the next section.

# 5 Evaluation

We implemented the diagnostic engine in Java as described in [Friedrich and Shchekotykhin, 2005], with extensions for calculating axiom diagnoses. Benchmarks were run on a PC (Pentium IV with 2GHz and 512MB RAM) with Windows XP SP2 as operating system. We conducted numerous tests using the files from the RACER test suite.

In order to be comparable, we applied the same test setting as described in [Friedrich and Shchekotykhin, 2005], i.e. we conducted 30 tests for each knowledge base, where each test randomly changed the knowledge-base such that each change on its own leads to an incoherent KB. The diagnosis task is to find minimum cardinality diagnoses in order to restore coherence. We employed QUICKXPLAIN [Junker, 2004] for finding minimal conflict sets, and RACER for coherence checks. Minimal conflict sets are computed on demand in order to label the hitting set tree[3] (HS-Tree) exploited for the computation of minimal diagnoses.

The evaluation was carried out in two steps, the first of which determined the minimum cardinality KB-Diagnoses for a test. In the second step, we emulated the behavior of a knowledge engineer who wants to investigate an axiom $Ax$ of a KB-Diagnosis $D$ more deeply, e.g. to question, which parts of $Ax$ must be changed provided that $D$ is the preferred diagnosis. Therefore, we selected an axiom $Ax$ from a KB-Diagnosis found in the first step. Since we are only interested in AX-Diagnoses w.r.t. $Ax$, this question corresponds to computing AX-Diagnoses, where the background theory is extended by $KB - D$, i.e. $B_e := B \cup (KB - D)$, and the knowledge base, for which we are calculating AX-diagnoses is $\text{DECOMP}(Ax)$.

In case we want to compute the minimal AX-Diagnoses of an axiom contained in a KB-Diagnosis, the following heuristic allows a faster computation by reusing minimal conflict sets found in the KB-Diagnosis step. Informally, a conflict set $CS$ of a knowledge base $KB$ is a minimal subset of $KB$, which is inconsistent with at least one positive test case unified with the background theory, or just the background theory in case there are no positive test cases. Such a set is called *minimal*, if no proper subset of $CS$ is a conflict set. Let $D$ be a KB-diagnosis and let $CS_{KB} = CS_1, \ldots, CS_k$ be the minimal conflict sets found in the KB-Diagnosis step, which contain $Ax$. It follows that $(CS_i - D) \cup \text{DECOMP}(Ax)$ must contain a minimal conflict set. Therefore when computing a label for the HS-Tree, in a first step, we use $B'_e := B \cup [\bigcup_{i=1}^{k} CS_i - D]$ as a reduced background theory. If we cannot find a minimal conflict w.r.t. this reduced background theory, in a second step, we use the full background theory $B_e$ to generate a conflict set. Note that if for a node $n$ in the HS-Tree, we cannot find a conflict w.r.t. the reduced background

[3]We assume the reader to know the basics of model-based diagnosis [Reiter, 1987].

| KB | | $\#KBC$ | $\|KBC\|$ | $\#AD$ | $\#AC$ | $\#IAD$ | DTRB | H-ADT | ADT | KB-DT |
|---|---|---|---|---|---|---|---|---|---|---|
| bike1 | min | 4 | 4 | 5 | 1 | 0 | 0,66 | 2,15 | 8,56 | 6,17 |
| 81 ax | avg | 4,07 | 4,27 | 5,43 | 1,57 | 0 | 1,92 | 3,71 | 34,02 | 27,58 |
| | max | 5 | 4 | 15 | 2 | 0 | 3,48 | 8,18 | 87,25 | 59,77 |
| bike2 | min | 6 | 4 | 4 | 1 | 0 | 1,02 | 3,63 | 8,14 | 21,69 |
| 154 ax | avg | 5,67 | 4,1 | 3,13 | 1,43 | 0,1 | 1,86 | 5,4 | 17,43 | 52,84 |
| | max | 3 | 4 | 3 | 1 | 1 | 1,41 | 18,67 | 31,89 | 100,85 |
| bike3 | min | 4 | 3 | 5 | 1 | 0 | 0,77 | 4,01 | 7,31 | 24,34 |
| 109 ax | avg | 3,97 | 3 | 5 | 1 | 0 | 0,86 | 4,66 | 53,16 | 33,02 |
| | max | 5 | 3 | 5 | 1 | 0 | 0,89 | 6,66 | 62,13 | 53,65 |
| bike4 | min | 7 | 4 | 1 | 3 | 0 | 2,87 | 3,79 | 10,86 | 33,35 |
| 166 ax | avg | 4,97 | 5,4 | 4,7 | 1,7 | 0,07 | 1,95 | 7,33 | 107,2 | 98,07 |
| | max | 5 | 4 | 3 | 1 | 2 | 1,18 | 23,04 | 426,19 | 157,99 |
| bike5 | min | 3 | 4 | 3 | 2 | 0 | 3,25 | 6,61 | 36,03 | 30,98 |
| 184 ax | avg | 3,6 | 3,97 | 6,2 | 1,33 | 0 | 1,83 | 12,85 | 101,84 | 114,05 |
| | max | 3 | 4 | 9 | 1 | 0 | 1,58 | 20,5 | 420,2 | 160,15 |
| bike6 | min | 4 | 4 | 3 | 2 | 0 | 3,33 | 7,12 | 36,06 | 55,63 |
| 207 ax | avg | 3,33 | 3,8 | 5,33 | 1,33 | 0,2 | 2,23 | 15,14 | 148,49 | 140,11 |
| | max | 2 | 3 | 6 | 1 | 3 | 1,72 | 64,7 | 493,52 | 217,1 |
| bike7 | min | 1 | 3 | 3 | 1 | 0 | 0,49 | 3,64 | 30,25 | 24,43 |
| 162 ax | avg | 2,63 | 3 | 5,27 | 1,23 | 0 | 1,76 | 11,89 | 59,56 | 63,05 |
| | max | 3 | 3 | 9 | 1 | 0 | 5,47 | 27,18 | 97,3 | 113,6 |
| bike8 | min | 1 | 3 | 3 | 1 | 0 | 0,48 | 3,92 | 33,03 | 30,99 |
| 185 ax | avg | 2,57 | 3 | 5,33 | 1,2 | 0 | 1,28 | 11,83 | 52,15 | 65,65 |
| | max | 2 | 3 | 9 | 1 | 0 | 1,84 | 26,35 | 92,17 | 115,47 |
| bike9 | min | 5 | 4 | 3 | 2 | 0 | 4,37 | 8,94 | 25,2 | 36,84 |
| 215 ax | avg | 3,43 | 3,97 | 4,4 | 1,47 | 0,33 | 3,64 | 14,92 | 181,02 | 154,7 |
| | max | 4 | 4 | 6 | 1 | 3 | 2,22 | 53,94 | 536,98 | 234,47 |
| galen | min | 4 | 2 | 3 | 2 | 0 | 12,98 | 52,27 | 124,78 | 142,03 |
| 3963 ax | avg | 3,47 | 2 | 5,4 | 1,2 | 2,07 | 11,48 | 146,98 | 252,71 | 244,52 |
| | max | 2 | 2 | 9 | 2 | 15 | 26,17 | 507,33 | 523,55 | 525,66 |
| galen2 | min | 3 | 2 | 5 | 1 | 0 | 5,95 | 39,2 | 98,58 | 90,3 |
| 3927 ax | avg | 3,03 | 2 | 5,2 | 1,3 | 2,5 | 12,7 | 94,61 | 146,27 | 141,83 |
| | max | 2 | 2 | 9 | 2 | 15 | 23,94 | 237,21 | 236,79 | 287,45 |
| bcs3 | min | 2 | 3 | 1 | 3 | 0 | 1,61 | 3,16 | 5,86 | 5,36 |
| 432 ax | avg | 3,33 | 25,63 | 2,03 | 1,67 | 0,87 | 1,68 | 24,15 | 154,45 | 126,73 |
| | max | 3 | 2 | 2 | 1 | 2 | 0,72 | 99,64 | 550,39 | 528,54 |

Table 1: Test results for diagnosing faulty axioms.

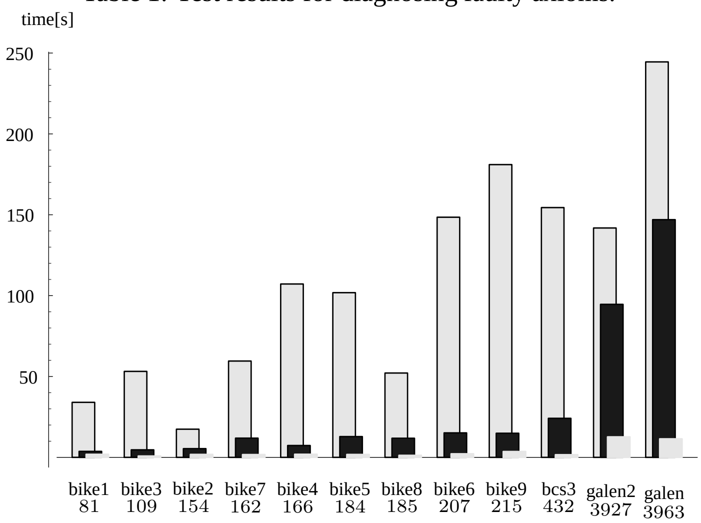


Figure 3: Comparing the performance with and without heuristic diagnosis according to the size of the KB. The bars in the back show the average times (in seconds) for finding minimum cardinality AX-Diagnoses without heuristic improvements. The black bars in the middle display the average time required when the heuristic is exploited. The small bars in the front show the average time required for finding minimum cardinality AX-Diagnoses with reduced background (i.e. before checking consistency w.r.t. $B \cup (KB - D)$).

theory $B'_e$, then this property holds also for all successors of $n$. Hence, we can omit the first step for these successor nodes.

Table 1 depicts the results using this heuristic for calculating diagnoses and compares the results to the plain method without heuristic speedup. Columns are to be interpreted as follows: H-ADT and ADT are the times for generating the minimum cardinality AX-Diagnoses *with* heuristic speed-up (H-ADT), and *without* heuristic speed-up (ADT). KB-DT is the time for running the initial knowledge base diagnosis. All times are given in seconds and include the time for decompositions. For each $KB$, we characterized the fastest (min) and slowest (max) heuristic axiom diagnosis by the number of minimum cardinality AX-Diagnoses (#AD) and the number of minimal conflicts (#AC) generated during diagnosing the axiom. As the heuristic requires the conflicts sets within the full $KB$, we additionally provide the number of $KB$-conflicts ($\#KBC$) and their maximum size ($|KBC|$) for the fastest and slowest case, as well as the number of invalid minimum cardinality AX-Diagnoses (i.e. those AX-Diagnoses of the reduced background theory $B'_e$ being inconsistent with the full one $B_e$, shown in column $\#IAD$) and the time for finding the minimum cardinality AX-Diagnoses w.r.t. the reduced background $B'_e$ (DTRB). The "avg" row shows these values averaged over all tests for a single $KB$. The unit "ax" denotes the number of axioms in a $KB$.

In addition, we improved the application of QUICKXPLAIN. QUICKXPLAIN takes two arguments (a $KB$ and a background theory $B$) and computes a minimal subset of $KB$ (called a minimal conflict set), which is inconsistent with $B$ provided that $B$ is consistent. If such a set does not exist (i.e. $KB \cup B$ is consistent), it outputs consistent. Experiments showed that QUICKXPLAIN performs better if $B$ is small due to the divide-and-conquer technique, which reduces $KB$ rapidly (but not $B$), and therefore reduces the costs for consistency checking. A reduction of the size of $KB$ results in *disproportionate* speed-up. Note that this is also the reason why QUICKXPLAIN works well for large knowledge bases. Consequently, if we diagnose DECOMP$(Ax)$ w.r.t. (a possibly large) $B_e$, a divide-and-conquer strategy will not provide significant acceleration. Hence, we adopt another strategy for generating minimal conflicts by leaving $B$ untouched; just replacing the axiom in $D$ we want to investigate by DECOMP$(Ax)$.

The effort for finding an AX-diagnosis by running a KB-diagnosis on $(KB - D) \cup$ DECOMP$(Ax)$ with background $B$ and no heuristic speedup, is approximately the same as for calculating a diagnosis for the plain knowledge-base (prior to the axiom-diagnosis). The additional cost if the heuristic fails (no diagnosis found by using the reduced background theory) is negligible, compared to the time spent for using the extended background theory $B_e$. The heuristic for generating minimum cardinality AX-diagnoses can be summarized as follows:

1. Let a KB-diagnosis $D$ be given for $KB$ and background $B$ (and possibly existing test-cases, which we omit for brevity). Let $Ax \in D$ be the axiom to be investigated.
2. Collect all conflicts $CS_1, \ldots, CS_k$ from the HS-tree, where $D$ comes from, and create $B'_e$ by adding all $CS_i - D$ to $B$.
3. Run a diagnosis on DECOMP$(Ax)$ with background $B'_e$.
4. Check the resulting diagnoses for consistency w.r.t. $B \cup (\mathcal{K} - D)$[4]. If none of them turns out to be consistent, run a diagnosis on $(KB - D) \cup$ DECOMP$(Ax)$ with background $B$.

For 310 experiments, we compared the times for generating minimum cardinality AX-Diagnoses with and without heuristic (see Figure 3). As we recognize, the heuristic always achieved a considerable improvement. In particular, the time for diagnosing w.r.t. the reduced background theory is almost negligible compare to the diagnosis time for the full background theory. The expensive step is (as expected) to check if diagnoses of the reduced background theory are consistent with the full background theory. Therefore, the speed up depends on the number of diagnoses which must be checked and the cost of consistency checking. Note that in our experiments the minimum cardinality diagnoses of the reduced background theory always contained those of the full background theory. The number of diagnoses deleted in the maximum average cases is rather small (around two, see Table 1), which means that the diagnoses of the reduced problem are already a very good approximation for the minimum cardinality diagnoses of the full problem.

The running time for finding an axiom diagnosis is strongly depending on the complexity of an axiom, as well as on the size and structure of the knowledge base. Note that in our evaluation, RACER does not provide any information on which axioms are used for discovering an incoherence. If theorem provers are available, which return the set of axioms applied during theorem proving (i.e. they return a conflict set, which is not necessarily minimal, but smaller than the whole $KB$) then QUICKXPLAIN could start with such a reduced set. This will result in significant speed-ups.

## 6 Related work

Most closely related to our approach is the work of [Schlobach and Cornet, 2003]. In this paper, a method for debugging faulty knowledge bases as well as faulty axioms is proposed. The approach is called *concept pinpointing*. Concepts are diagnosed by successive generalization of an axiom until the most general form that is still incoherent is achieved. Generalizations are based on a syntactic relation, which is assumed to exist and left to the knowledge engineer. Moreover, the approach will find concepts only.

In [Schlobach, 2005], three approaches are compared for generating diagnoses of terminologies, which differ in the size of the conflict sets returned by the theorem prover. The first approach always considers the complete knowledge base (which served as an input for the theorem prover) as a conflict set. In the second approach, the theorem prover returns a minimized (not necessarily minimal) conflict set, and in the third

[4]Prior to invoking RACER for checking coherence, one can search for known conflicts to appear in the set of axioms, and calculate a new conflict if nothing is found. This conflict can be re-used in subsequent checks.

approach, a special procedure for computing minimal conflicts in unfoldable (i.e. acyclic) $\mathcal{ALC}$-Tboxes is employed. Compared to these evaluations, we employ QUICKXPLAIN to generate minimal conflict sets in order to avoid the known problems of non-minimal conflicts for the HS-tree generation. Since we are not restricted to a special theorem prover, our approach is not restricted to unfoldable $\mathcal{ALC}$-Tboxes. However, a runtime comparison of generating minimal conflict sets by the method described in [Schlobach and Cornet, 2003] and the combination of QUICKXPLAIN with a highly optimized consistency checker is open. Note that this comparison also strongly depends on the ability of the consistency checker to return axioms which were involved in the generation of an inconsistency as described above.

In [Mateis *et al.*, 2000] model-based diagnosis of Java programs is discussed. Similar to our approach, the grammar of the Java language is the starting point of the transformation, but unlike our method, the approach is designed to be used with imperative semantics, while we focus on declarative semantics.

In the heuristic approaches of [Parsion *et al.*, 2005] and [Wang *et al.*, 2005], debugging cues respectively error patterns are exploited. In contrast to these approaches, our goal was to provide a general, complete, and correct method for diagnosis. Nevertheless, these heuristic approaches may help us to discriminate between minimal diagnoses. Based on a connection relation, [Haase *et al.*, 2005] present a method for computing a minimal subset of an ontology, in which a concept is unsatisfiable. However, they do not identify the parts of axioms, which cause this unsatisfiability.

## 7 Conclusions

We presented a general theory of diagnosis for faulty knowledge bases, which allows the identification of faulty parts of axioms. This approach subsumes current methods and is independent of FOL knowledge representation language variants or particular reasoning systems. Based on the roots of model-based diagnosis, we were able to develop correct and complete algorithms for the computation of axiom diagnoses. We have shown the feasibility of our approach by extensive test evaluation, and provided an extension of current diagnosis methods s.t. a considerable speed up for the diagnosis of axioms is achieved.


## Acknowledgments

The research project is funded partly by grants from the Austrian Research Promotion Agency, Programm Line FIT-IT Semantic Systems (www.fit-it.at), Project AllRight, Contract 809261 and the European Union, Project WS-Diamond, Contract 516933.


## References


[Baader *et al.*, 2003] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[Friedrich and Shchekotykhin, 2005] G. Friedrich and K. Shchekotykhin. A general diagnosis method for ontologies. In $4^{th}$ *ISWC 2005*, volume 3729. Springer LNCS, 2005.

[Haase *et al.*, 2005] P. Haase, F. Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In $4^{th}$ *ISWC 2005*, volume 3729. Springer LNCS, 2005.

[Junker, 2004] U. Junker. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *Proc. AAAI 04*, pages 167–172, San Jose, CA, USA, 2004.

[Linz, 1996] P. Linz. *An introduction to formal languages and automata*. Lexington Books, Mass., 2 edition, 1996.

[Mateis *et al.*, 2000] C. Mateis, M Stumptner, and F. Wotawa. Modeling java programs for diagnosis. In *ECAI 2000*, pages 171–175, 2000.

[Parsion *et al.*, 2005] B. Parsion, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *WWW 2005*, Chiba, Japan, May 2005. ACM.

[Patel-Schneider and Swartout, 1993] P.F. Patel-Schneider and B. Swartout. Description-logic knowledge representation system specification. Technical report, KRSS Group of the DARPA Knowledge Sharing Effort, November 1993.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 23(1):57–95, 1987.

[Schlobach and Cornet, 2003] S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. IJCAI 03*, pages 355–362, Acapulco, Mexico, 2003.

[Schlobach, 2005] S. Schlobach. Diagnosing terminologies. In *Proc. AAAI 05*, pages 670–675, Pittsburgh, PA, USA, 2005.

[Wang *et al.*, 2005] H. Wang, M. Horridge, A. Rector, N. Drummond, and J. Seidenberg. Debugging owl-dl ontologies: A heuristic approach. In $4^{th}$ *ISWC 2005*, volume 3729. Springer LNCS, 2005.