ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-813

Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation

Kostyantyn Shchekotykhin and **Gerhard Friedrich** and **Patrick Rodler** and **Philipp Fleiss**[1]

Abstract. Sequential diagnosis methods compute a series of queries for discriminating between diagnoses. Queries are answered by probing such that eventually the set of faults is identified. The computation of queries is based on the generation of a set of *most probable* diagnoses. However, in diagnosis problem instances where the number of minimal diagnoses and their cardinality is high, even the generation of a set of minimum cardinality diagnoses is unfeasible with the standard conflict-based approach. In this paper we propose to base sequential diagnosis on the computation of *some* set of minimal diagnoses using the direct diagnosis method, which requires less consistency checks to find a minimal diagnosis than the standard approach. We study the application of this direct method to high cardinality faults in knowledge-bases. In particular, our evaluation shows that the direct method results in almost the same number of queries for cases when the standard approach is applicable. However, for the cases when the standard approach is not applicable, sequential diagnosis based on the direct method is able to locate the faults correctly.

1 Introduction

Model-based diagnosis (MBD) [12] is a general method which can be used to find errors in hardware, software, knowledge-bases, orchestrated web-services, configurations, etc. In particular, ontology (knowledge-base) debugging tools [11, 5, 8] can localize a (potential) fault by computing sets of axioms $\mathcal{D} \subseteq \mathcal{KB}$ called *diagnosis* for a knowledge-base $\mathcal{KB}$. At least all axioms of a minimal diagnosis must be modified or deleted in order to formulate a fault-free knowledge-base $\mathcal{KB}^*$. A knowledge-base (KB) is faulty if some requirements, such as consistency of $\mathcal{KB}$, presence or absence of specific entailments, are violated.

Sequential MBD methods [2] acquire additional information in order to discriminate between diagnoses. Queries are generated and answered either by automatic probing or by humans providing additional observations about the system to be diagnosed. As various applications show, the standard methods work very satisfactory for cases where the number of faults is low (single digit number), consistency checking is fast (single digit number of seconds), and sufficient possibilities for observations are available.

All the discrimination and diagnosis approaches listed above follow the standard model-based diagnosis technique [12] and compute diagnoses using minimal conflict sets, i.e. irreducible sets of axioms $CS \subseteq \mathcal{KB}$ that violate some requirements, by using a consistency checker (black-box approach). Furthermore, diagnoses are ordered and filtered by some preference criteria, e.g. probability or cardinality, in order to focus debugging on the most likely cases.

[1] Alpen-Adria Universität, Klagenfurt, 9020 Austria, email: firstname.lastname@aau.at

In the common KB development scenario where a user develops an ontology manually, the changes between validation steps are rather small. Therefore, the number of faulty axioms is in a range where standard sequential model-based methods are applicable [13].

However, there are situations when the changes of KBs are substantial. For example in ontology matching scenarios two KBs with several thousands of axioms are merged into a single one. High quality matchers (e.g. [9]) require the diagnosis of such substantially extended KBs, but could not apply standard diagnosis methods because of the large number of minimum cardinality diagnoses and their high cardinality. E.g. there are cases when the minimum cardinality of diagnoses is greater than 20.

In order to deal with hard diagnosis instances, we propose to relax the requirement for sequential diagnosis to compute a set of *preferred* minimal diagnoses, such as a set of most probable diagnoses. Instead, we compute *some* set of minimal diagnoses which can be used for query generation. This allows to use the direct computation of diagnoses [17] *without* computing conflict sets. The direct approach was applied for non-interactive diagnosis of ontologies [16, 15] and constraints [4]. The computation of a minimal diagnosis by a variant of QuickXPlain [10] requires $O(|\mathcal{D}| \log(\frac{|\mathcal{KB}|}{|\mathcal{D}|}))$ consistency checks, where $|\mathcal{D}|$ is the cardinality of the minimal diagnosis and $|\mathcal{KB}|$ the size of the knowledge base. If m minimal diagnoses are required for query generation, then only m calls to a direct diagnosis generator are needed. A recent approach [18] does not generate the standard HS-Tree, but still depends on the minimization of conflict sets, i.e. $|\mathcal{D}|$ minimized conflicts have to be discovered. Consequently, if $|\mathcal{D}| \gg m$, substantially more consistency checks are required.

Since we are replacing the set of most probable diagnoses by just a set of minimal diagnoses, some important practical questions have to be addressed. (1) Is a substantial number of additional queries needed, (2) is this approach able to locate the faults, and (3) how efficient is this approach?

In order to answer these questions we have exploited the most difficult diagnoses problems of the ontology alignment competition [3].

Our evaluation shows that sequential diagnosis by direct diagnosis generation needs approximately the same number of queries (± 1) in order to identify the faults. This evaluation was carried out for cases where the standard sequential diagnosis method was applicable. Furthermore, the evaluation shows that our proposed method is capable to locate faults in all cases correctly. Moreover, the computation costs which are introduced in addition to the computational costs of theorem proving are less than 7% of the total runtime.

The remainder of the paper is organized as follows: Section 2 gives a brief introduction to the main notions of sequential KB diagnosis. The details of the suggested algorithms and their applications are presented in Section 3. In Section 4 we provide evaluation results.

2 Basic concepts

In the following we present (1) the fundamental concepts regarding the diagnosis of KBs and (2) the interactive localization of axioms which must be changed.

Diagnosis of KBs. Given a knowledge-base $\mathcal{KB}$ which is a set of logical sentences (axioms), the user can specify particular requirements during the knowledge-engineering process. The most basic requirement is satisfiability, e.g. a logical model exists. A further frequently employed requirement is coherence. Coherence requires that there exists a model s.t. the interpretation of every unary predicate is non-empty. In other words, if we add $\exists Y\ a(Y)$ to $\mathcal{KB}$ for every unary predicate a, then the resulting KB must be satisfiable. In addition, as it is common practice in software engineering, the knowledge-engineer (user for short) may specify test cases. Test cases are axioms which must (not) be entailed by a valid KB.

Given a set of axioms P (called positive test cases) and a set of axioms N (called negative test cases), a knowledge-base $\mathcal{KB}^*$ is valid iff $\mathcal{KB}^*$ is satisfiable (and coherent if required) and

1. $\mathcal{KB}^* \models p \quad \forall p \in P$
2. $\mathcal{KB}^* \not\models n \quad \forall n \in N$

Let us assume that there is a non-valid knowledge-base $\mathcal{KB}$, then a set of axioms $\mathcal{D} \subseteq \mathcal{KB}$ must be removed and possibly some axioms EX must be added by user s.t. an updated $\mathcal{KB}^*$ becomes valid, i.e. $\mathcal{KB}^* := (\mathcal{KB} \setminus \mathcal{D}) \cup EX$. The goal of diagnosis is to provide information to the users which are the smallest sets of axioms that must be changed. Consequently, $\mathcal{D}$ (which is called a diagnosis) should be as small as possible. Furthermore, we allow the user to define a set of axioms $\mathcal{B}$ (called the background theory) which must not be changed (i.e. the correct axioms). More formally:

Definition 1. *Given a diagnosis problem instance (DPI) specified by $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$ where $\mathcal{KB}$ is a knowledge-base, $\mathcal{B}$ a background theory, P a set of axioms which must be implied by a valid knowledge-base $\mathcal{KB}^*$, and N a set of axioms which must* not *be implied by $\mathcal{KB}^*$.*

A diagnosis is a set of axioms $\mathcal{D} \subseteq \mathcal{KB}$ iff the set of axioms $\mathcal{KB} \setminus \mathcal{D}$ can be extended by a set of logical sentences EX such that:

1. *$(\mathcal{KB} \setminus \mathcal{D}) \cup \mathcal{B} \cup EX$ is satisfiable (and coherent if required)*
2. *$(\mathcal{KB} \setminus \mathcal{D}) \cup \mathcal{B} \cup EX \models p$ for all $p \in P$*
3. *$(\mathcal{KB} \setminus \mathcal{D}) \cup \mathcal{B} \cup EX \not\models n$ for all $n \in N$*

$\mathcal{D}$ is a minimal diagnosis *iff there is no $\mathcal{D}' \subset \mathcal{D}$ such that $\mathcal{D}'$ is a diagnosis. $\mathcal{D}$ is a* minimum cardinality diagnosis *iff there is no diagnosis $\mathcal{D}'$ such that $|\mathcal{D}'| < |\mathcal{D}|$.*

The following proposition of [13] characterizes diagnoses by replacing EX with the positive test cases.

Corollary 1. *Given a DPI $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$, a set of axioms $\mathcal{D} \subseteq \mathcal{KB}$ is a diagnosis iff $(\mathcal{KB} \setminus \mathcal{D}) \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\}$ is satisfiable (coherent) and $\forall n \in N : (\mathcal{KB} \setminus \mathcal{D}) \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \not\models n$*

In the following we assume that there is always a diagnosis.

Proposition 1. *A diagnosis $\mathcal{D}$ for a DPI $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$ exists iff $\mathcal{B} \cup \{\bigwedge_{p \in P} p\}$ is consistent (coherent) and $\forall n \in N : \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \not\models n$*

For the computation of diagnoses *conflict sets* are usually employed to constrain the search space. A conflict set is the part of the KB that preserves the inconsistency/incoherency.

Definition 2. *Given a DPI $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$, a set of axioms $CS \subseteq \mathcal{KB}$ is a conflict set iff $CS \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\}$ is inconsistent (incoherent) or there is a $n \in N$ s.t. $CS \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \models n$. CS is minimal iff there is no $CS' \subset CS$ such that CS' is a conflict set.*

Minimal conflict sets can be used to compute the set of minimal diagnoses as it is shown in [12]. The idea is that each diagnosis should include at least one element of each minimal conflict set.

Proposition 2. *$\mathcal{D}$ is a diagnosis for the DPI $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$ iff $\mathcal{D}$ is a minimal hitting set for the set of all minimal conflict sets of the instance.*

For the generation of a minimal conflict set, diagnosis systems use a divide-and-conquer method (e.g. QUICKXPLAIN [10], for short QX). In the worst case, QX requires $O(|CS| \log(\frac{|\mathcal{KB}|}{|CS|}))$ calls to the reasoner, where CS is the returned minimal conflict set.

The computation of minimal diagnoses in KB debugging systems is implemented using Reiter's Hitting Set HS-TREE algorithm [12]. The algorithm constructs a directed tree from the root to the leaves, where each non-leave node is labeled with a minimal conflict set and leave nodes are labeled by ✓ (*no conflicts*) or × (*pruned*).

Each (✓) node corresponds to a minimal diagnosis. The minimality of the diagnoses is guaranteed by the minimality of conflict sets used for labeling the nodes, the pruning rule and the breadth-first strategy of the tree generation [12]. Moreover, because of the breadth-first strategy the minimal diagnoses are generated in increasing order of their cardinality. Under the assumption that diagnoses with lower cardinality are more probable than those with higher cardinality HS-TREE generates most probable minimal diagnoses first.

Diagnoses discrimination. For many real-world DPIs, a debugger can return a large number of (minimal) diagnoses. Each minimal diagnosis corresponds to a different set of axioms that must be changed in order to formulate a valid KB. The user may extend the test cases P and N s.t. diagnoses are eliminated, thus identifying exactly those axioms that must be changed. For discriminating between diagnoses we assume that the user knows some of the sentences a valid KB $\mathcal{KB}^*$ must (not) entail, i.e. the user serves as an oracle.

Property 1. *Given a DPI $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$, a set of diagnoses $\mathbf{D}$, and a logical sentence Q representing the oracle query $\mathcal{KB}^* \models Q$. If the oracle gives the answer* yes *then every diagnosis $\mathcal{D}_i \in \mathbf{D}$ is a diagnosis for $\langle \mathcal{KB}, \mathcal{B}, P \cup \{Q\}, N \rangle$ iff both conditions hold:*

$$(\mathcal{KB} \setminus \mathcal{D}_i) \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \cup \{Q\} \text{ is consistent (coherent)}$$

$$\forall n \in N \ : \ (\mathcal{KB} \setminus \mathcal{D}_i) \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \cup \{Q\} \not\models n$$

If the oracle gives the answer no *then every diagnosis $\mathcal{D}_i \in \mathbf{D}$ is a diagnosis for $\langle \mathcal{KB}, \mathcal{B}, P, N \cup \{Q\} \rangle$ iff both conditions hold:*

$$(\mathcal{KB} \setminus \mathcal{D}_i) \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \text{ is consistent (coherent)}$$

$$\forall n \in (N \cup \{Q\}) \ : \ (\mathcal{KB} \setminus \mathcal{D}_i) \cup \mathcal{B} \cup \{\bigwedge_{p \in P} p\} \not\models n$$

However, many different queries might exist for some set of diagnoses $|\mathbf{D}| > 2$, in the extreme case exponentially many (in $|\mathbf{D}|$). To select the best query, the authors in [13] suggest two query selection strategies: SPLIT-IN-HALF (SPL) and ENTROPY (ENT). The first strategy is a greedy approach preferring queries which allow to remove half of the diagnoses in $\mathbf{D}$, for both answers to the query.

The second is an information-theoretic measure, which estimates the information gain for both outcomes of each query and returns the query that maximizes the expected information gain. The *prior fault probabilities* required for evaluating the ENTROPY measure can be obtained from statistics of previous diagnosis sessions. For instance, if the user has problems to apply "$\exists$", then the diagnosis logs are likely to contain more repairs of axioms including this quantifier. Consequently, the prior fault probabilities of axioms including "$\exists$" should be higher. Given the fault probabilities of axioms, one can calculate prior fault probabilities of diagnoses as well as evaluate ENTROPY (see [13] for more details). The queries for both strategies are constructed by exploiting so called classification and realization services provided by description logic reasoners. Given a knowledge-base $\mathcal{KB}$ and interpreting unary predicates as classes (rsp. concepts), the classification generates the inheritance (subsumption) tree, i.e. the entailments $\mathcal{KB} \models \forall X\, p(X) \rightarrow q(X)$, if p is a subclass of q. Realization computes, for each individual name t occurring in a knowledge-base $\mathcal{KB}$, a set of most specific classes p s.t. $\mathcal{KB} \models p(t)$ (see [1] for details).

Due to the number of diagnoses and the complexity of diagnosis computation, not all diagnoses are exploited for generating queries but a set of minimal diagnoses of size less or equal to some (small) predefined number m [13]. We call this set the *leading diagnoses* and denote it by $\mathbf{D}$ from now on. This set comprises the (most probable) minimal diagnoses which represent the set of all diagnoses.

The sequential KB debugging process can be sketched as follows. As input a DPI and some meta information, such as prior fault estimates $\mathcal{F}$, query selection strategy s_Q (SPL or ENT) and stop criterion σ, are given. As output a minimal diagnosis is returned that has a posterior probability of at least $1 - \sigma$. For sufficiently small σ this means that the returned diagnosis is highly probable whereas all other minimal diagnoses are highly improbable.

1. Using QX and HS-TREE calculate a set of leading diagnoses $\mathbf{D}$ of cardinality $\min(m, a)$, where a is the number of all minimal diagnoses for the DPI and m is the number of leading diagnoses predefined by a user.
2. Use the prior fault probabilities $\mathcal{F}$ and the already specified test cases to compute (posterior) probabilities of diagnoses in $\mathbf{D}$ by the Bayesian Rule (cf. [13]).
3. If some diagnosis $\mathcal{D} \in \mathbf{D}$ has probability greater or equal to $1-\sigma$ or the user accepts $\mathcal{D}$ as the axioms to be changed then stop and return $\mathcal{D}$.
4. Use $\mathbf{D}$ to generate a set of queries and select the best query Q according to s_Q.
5. Ask the user $\mathcal{KB}^* \models Q$ and, depending on the answer, add Q either to P or to N.
6. Remove elements from $\mathbf{D}$ violating the newly acquired test case.
7. Repeat at Step 1.

3 Interactive Direct Diagnosis of Ontologies

The novelty of our approach is the interactivity combined with the direct calculation of diagnoses. To this end we will utilize an "inverse" version of the QX algorithm [10] called INV-QX and an associated "inverse" version of HS-TREE termed INV-HS-TREE.

This combination of algorithms was first used in [4]. However, we introduced two modifications: (i) a depth-first search strategy instead of breadth-first and (ii) a new pruning rule which moves axioms from $\mathcal{KB}$ to $\mathcal{B}$ instead of just removing them from $\mathcal{KB}$, since not adding them to $\mathcal{B}$ might result in losing some of the minimal diagnoses.

INV-QX – Key Idea. INV-QX relies on the monotonic semantics of the used knowledge representation language. The algorithm takes a DPI $\langle \mathcal{KB}, \mathcal{B}, P, N \rangle$ and a ranking heuristic as input and outputs either one minimal diagnosis or *no-diagnosis-exists*. The ranking heuristic assigns a fault probability to each axiom in $\mathcal{KB}$, if this information is available; otherwise every axiom has the same rank. In the first step INV-QX verifies if a diagnosis exists, next whether $\mathcal{KB}$ is faulty and, if so, sorts all axioms in descending order. Ordering of axioms according to their fault probabilities allows the algorithm to compute an approximation of a most probable minimal diagnosis. Next, INV-QX enters the recursion in which $\mathcal{KB}$ is partitioned into two subsets S_1 and S_2 such that S_1 comprises axioms with higher fault probabilities and S_2 with lower. In our implementation $\mathcal{KB}$ is split in half. Then the algorithm verifies whether S_1 is a diagnosis of the input DPI according to Definition 1. The algorithm continues to operate in a divide-and-conquer strategy until a minimal diagnosis is found. INV-QX requires $O(|\mathcal{D}| \log(\frac{|\mathcal{KB}|}{|\mathcal{D}|}))$ calls to a reasoner to find a minimal diagnosis $\mathcal{D}$. Moreover, in opposite to SAT or CSP methods, e.g. [14], INV-QX can be used to compute diagnoses in cases when satisfiability checking is beyond NP. For instance, reasoning for most of KBs used in Section 4 is ExpTime-complete.

INV-QX is a deterministic algorithm. In order to obtain a different next diagnosis, the DPI used as input for INV-QX must be modified accordingly. To this end we employ INV-HS-TREE.

INV-HS-TREE – Construction. The algorithm is inverse to the HS-TREE algorithm in the sense that nodes are now labeled by minimal diagnoses (instead of minimal conflict sets) and a path from the root to an open node is a partial conflict set (instead of a partial diagnosis). The algorithm constructs a directed tree from the root to the leaves, where each node nd is labeled either with a minimal diagnosis $\mathcal{D}$ or $\times$ (*pruned*) which indicates that the node is closed. For each $s \in \mathcal{D}$ there is an outgoing edge labeled by s. Let $H(nd)$ be the set of edge labels on the path from the root to the node nd. Initially the algorithm generates an empty root node and adds it to a LIFO-queue, thereby implementing a *depth-first search* strategy. Until the required number m of minimal diagnoses is reached or the queue is empty, the algorithm removes the first node nd from the queue and labels the node by applying the following steps.

1. (*reuse*): $\mathcal{D} \in \mathbf{D}$ if $\mathcal{D} \cap H(nd) = \emptyset$, add for each $s \in \mathcal{D}$ a node to the LIFO-queue, or
2. (*pruned*): $\times$ if INV-QX$(\mathcal{KB} \setminus H(nd), \mathcal{B} \cup H(nd), P, N) =$ *no-diagnosis-exists*, (see Proposition 1), or
3. (*compute*): $\mathcal{D}$ if INV-QX$(\mathcal{KB} \setminus H(nd), \mathcal{B} \cup H(nd), P, N) = \mathcal{D}$; add $\mathcal{D}$ to $\mathbf{D}$ and add for each $s \in \mathcal{D}$ a node to the LIFO-queue.

Reuse of known diagnoses in Step 1 and the addition of $H(nd)$ to the background theory $\mathcal{B}$ in Step 2 allows the algorithm to: (i) force INV-QX to search for a minimal diagnosis that is different to all diagnoses in $\mathbf{D}$. Finally, if neither Step 1 nor Step 2 are applicable INV-HS-TREE calls INV-QX to compute a new minimal diagnosis $\mathcal{D}$ which is then added to the set $\mathbf{D}$. The depth-first search strategy maintains only a set of minimal diagnoses comprising at most m elements. No conflicts are stored. This allows a significant reduction of memory usage by INV-HS-TREE compared to HS-TREE. The worst case space complexity of INV-HS-TREE computing m minimal diagnoses is *linear* and amounts to $O(m)$, whereas the worst case space complexity of HS-TREE is $O(|CS_{\max}|^d)$ where $|CS_{\max}|$ is the maximal cardinality minimal conflict set (i.e. there is no minimal conflict set with larger cardinality) and d is the depth were m minimal diagnoses have been generated w.r.t. a DPI.

The disadvantage of INV-HS-TREE is that it cannot guarantee the

computation of diagnoses in a special order, e.g. minimum cardinality or maximum fault probability first.

INV-HS-TREE – Update Procedure for Interactivity. Since paths in INV-HS-TREE are (1) irrelevant and need not be maintained, and (2) only a small (linear) number of nodes/paths is in memory due to the application of a depth-first search, the update procedure after a query Q has been answered involves a reconstruction of the tree. In particular, by answering Q, $m-k$ of (maximally) m leading diagnoses are invalidated and deleted from memory. The k still valid minimal diagnoses are used to build a new tree. To this end, the root is labeled by any of these k minimal diagnoses. A tree is constructed as described above where the k diagnoses are incorporated for the *reuse* check. Note, the recalculation of a diagnosis that has been invalidated by a query is impossible as in subsequent iterations a new DPI is considered which includes the answered query as a test case.

Example. Consider a DPI with the following $\mathcal{KB}$:
$ax_1 : \forall X\, C(X) \rightarrow A(X).$ $ax_4 : \forall X\, B(X) \rightarrow C(X).$
$ax_2 : \forall X\, C(X) \rightarrow E(X).$ $ax_5 : \forall X\, B(X) \rightarrow \neg D(X).$
$ax_3 : \forall X\, A(X) \rightarrow \neg(C(X) \vee \neg B(X)).$
the background knowledge $\mathcal{B} = \{A(v)., B(w)., C(s).\}$, one positive $P = \{D(v).\}$ and one negative $N = \{E(w).\}$ test case. For the sample DPI the set of minimal conflict sets comprises four elements $\{CS_1 : \langle ax_1, ax_3\rangle, CS_2 : \langle ax_2, ax_4\rangle, CS_3 : \langle ax_3, ax_5\rangle, CS_4 : \langle ax_3, ax_4\rangle\}$, as well as the set of minimal diagnoses $\{\mathcal{D}_1 : [ax_2, ax_3], \mathcal{D}_2 : [ax_3, ax_4], \mathcal{D}_3 : [ax_1, ax_4, ax_5]\}$. Assume also that the number of leading diagnoses required for query generation is set to $m = 2$. Applied to the sample DPI, INV-HS-TREE computes a minimal diagnosis $\mathcal{D}_1 := [ax_2, ax_3] = \text{INV-QX}(\mathcal{KB}, \mathcal{B}, P, N)$ to label the root node, see Figure 1. Next, it generates one successor node that is linked with the root by an edge labeled with ax_2. For this node $\text{INV-QX}(\mathcal{KB} \setminus \{ax_2\}, \mathcal{B} \cup \{ax_2\}, P, N)$ yields a minimal diagnosis $\mathcal{D}_2 := [ax_3, ax_4]$ disjoint with $\{ax_2\}$. Now $|\mathbf{D}| = 2$ and a query is generated and answered as in Figure 1. Adding $C(w)$ to the negative test cases invalidates $\mathcal{D}_1$ since $(\mathcal{KB} \setminus \mathcal{D}_1) \cup \mathcal{B} \models C(w)$. In the course of the update, $\mathcal{D}_1$ is deleted and $\mathcal{D}_2$ used as the root of a new tree. An edge labeled with ax_3 is created and diagnosis $\mathcal{D}_3 := [ax_1, ax_4, ax_5]$ is generated. After the answer to the second query is added to the positive test cases, $\mathcal{D}_3$ is invalidated and all outgoing edge labels ax_3, ax_4 of the root $\mathcal{D}_2$ of the new tree are conflict sets for the current DPI $\langle \mathcal{KB}, \mathcal{B}, \{D(v)., \forall X\, A(X) \rightarrow C(X).\}, \{E(w)., C(w).\}\rangle$, i.e. all leaf nodes are labeled by $\times$ and the tree construction is complete. So, $\mathcal{D}_2$ is returned as its probability is 1.

4 Evaluation

We evaluated our approach DIR (based on INV-QX and INV-HS-TREE) versus the standard technique STD [13] (based on QX and HS-TREE) using a set of KBs created by automatic matching systems. Given two knowledge bases $\mathcal{KB}_i$ and $\mathcal{KB}_j$, a matching system outputs *alignment* M_{ij} which is a set of *mappings* (correspondences) between semantically related entities of $\mathcal{KB}_i$ and $\mathcal{KB}_j$. Let $Q(\mathcal{KB})$ denote the set of all elements of $\mathcal{KB}$ for which mappings can be produced, i.e. names of predicates. Each mapping is a tuple $\langle x_i, x_j, r, v\rangle$, where $x_i \in Q(\mathcal{KB}_i)$, $x_j \in Q(\mathcal{KB}_j)$ and x_i, x_j have the same arity, $r \in \{\leftarrow, \leftrightarrow, \rightarrow\}$ is a logical operator and $v \in [0, 1]$ is a confidence value. The latter expresses the probability of a mapping to be correct. Let $\overline{X}$ be a vector of distinct logical variables with a length equal to the arity of x_i, then each $\langle x_i, x_j, r, v\rangle \in M_{ij}$ is translated to the logical sentence $\forall \overline{X}\, x_i(\overline{X})\ r\ x_j(\overline{X})$. Let $\mathcal{KB}(M_{ij})$ be set of axioms for the alignment M_{ij}, then the result of the matching process is an aligned $\mathcal{KB}_{ij} = \mathcal{KB}_i \cup \mathcal{KB}(M_{ij}) \cup \mathcal{KB}_j$.

The KBs considered in this section were created by ontology matching systems participating in the Ontology Alignment Evaluation Initiative (OAEI) 2011 [3]. Each matching experiment in the framework of OAEI represents a scenario in which a user obtains an alignment M_{ij} by means of some (semi)automatic tool for two real-world *ontologies* $\mathcal{KB}_i$ and $\mathcal{KB}_j$. The latter are KBs expressed by the Web Ontology Language (OWL) [7] whose semantics is compatible with the $\mathcal{SROIQ}$ description logic (DL). This DL is a decidable fragment of first-order logic for which a number of effective reasoning methods exist [1]. Note that, $\mathcal{SROIQ}$ is a member of a broad family of DL knowledge representation languages. All DL KBs considered in this evaluation are expressible in $\mathcal{SROIQ}$.

The goal of the first experiment was to compare the performance of sequential STD and sequential DIR on a set of large, but diagnostically uncomplicated KBs, generated for the Anatomy experiment of OAEI[2]. In this experiment the matching systems had to find mappings between two KBs describing the human and the mouse anatomy. $\mathcal{KB}_1$ (`Human`) and $\mathcal{KB}_2$ (`Mouse`) include 11545 and 4838 axioms respectively, whereas the size of the alignment M_{12} produced by different matchers varies between 1147 and 1461 mappings. Seven matching systems produced a classifiable but incoherent output. One system generated a classifiable and coherent aligned KB. However, this system employes a built-in heuristic diagnosis engine which does not guarantee to produce minimal diagnoses. I.e. some axioms are removed without reason. Four systems produced KBs which could not be processed by current reasoning systems (e.g. HermiT) since these KBs could not be classified within 2 hours.

For testing the performance of our system we have to define the correct output of sequential diagnosis which we call the target diagnosis $\mathcal{D}_t$. We assume that the only available knowledge is M_{ij} together with $\mathcal{KB}_i$ and $\mathcal{KB}_j$. In order to measure the performance of the matching systems the organizers of OAEI provided a *golden standard* alignment M_t considered as correct. Nevertheless, we cannot assume that M_t is available since otherwise the matching system would have used this information. W.r.t. the knowledge available, any minimal diagnosis of $\mathcal{KB}(M_{ij})$ with $\mathcal{KB}_i \cup \mathcal{KB}_j$ as background theory can be selected as target diagnosis. However, for every alignment we selected a minimal diagnosis as target diagnosis which is outside the golden standard. By this procedure we mimic cases where additional information can be acquired such that no mapping of the golden standard is removed in order to establish coherence. We stress that this setting is unfavorable for diagnosis, since providing more information by exploiting the golden standard would reduce the number of queries to ask. Consequently, we limit the knowledge to $\mathcal{KB}_{ij}$ and use $\mathcal{KB}_{ij} \setminus \mathcal{D}_t$ to answer the queries.

In particular, the selection of a target diagnosis for each $\mathcal{KB}_{ij}$ output by a matching system was done in two steps: (i) compute the set of all minimal diagnoses $\mathbf{AD}$ w.r.t. the mappings which are not in the golden standard, i.e. $\mathcal{KB}(M_{ij} \setminus M_t)$, and use $\mathcal{KB}_i \cup \mathcal{KB}_j \cup \mathcal{KB}(M_{ij} \cap M_t)$ as background theory. The set of test cases are empty. I.e. the DPI is $\langle \mathcal{KB}(M_{ij} \setminus M_t), \mathcal{KB}_i \cup \mathcal{KB}_j \cup \mathcal{KB}(M_{ij} \cap M_t), \emptyset, \emptyset\rangle$. (ii) select $\mathcal{D}_t$ randomly from $\mathbf{AD}$. The prior fault probabilities of mapping axioms $ax \in \mathcal{KB}(M_{ij})$ were set to $1 - v_{ax}$ where v_{ax} is the confidence value provided by the matching system.

The tests were performed for the mentioned seven incoherent alignments where the input DPI is $\langle \mathcal{KB}(M_{ij}), \mathcal{KB}_i \cup \mathcal{KB}_j, \emptyset, \emptyset\rangle$ and

[2] All KBs and source code of programs used in the evaluation can be downloaded from http://code.google.com/p/rmbd/wiki/DirectDiagnosis. The tests were performed on Core i7, 64GB RAM running Ubuntu, Java 7 and HermiT as DL reasoner.

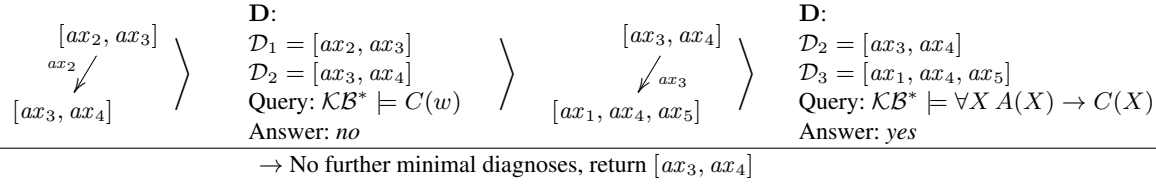

Figure 1. Identification of the target diagnosis $[ax_3, ax_4]$ using interactive direct diagnosis.

the output is a minimal diagnosis. We tested DIR and STD with both query selection strategies SPLIT-IN-HALF (SPL) and ENTROPY (ENT) in order to evaluate the quality of fault probabilities based on confidence values. Moreover, for generating a query the number of leading diagnoses was limited to $m = 9$.

The results of the first experiment are presented in Table 1. DIR computed $\mathcal{D}_t$ within 36 sec. on average and slightly outperformed STD which required 36.7 sec. The number of asked queries was equal for both methods in all but two cases resulting from KBs produced by the `MapSSS` system. For these KBs DIR required one query more using ENT and one query less using SPL. In general, the results obtained for the Anatomy case show that DIR and STD have similar performance in both runtime and number of queries. Both DIR and STD identified the target diagnosis. Moreover, the confidence values provided by the matching systems appeared to be a good estimate for fault probabilities. Thus, in many cases ENT was able to find $\mathcal{D}_t$ using one query only, whereas SPL used 4 queries on average.

In the first experiment the identification of the target diagnosis by sequential STD required the computation of 19 minimal conflicts on average. Moreover, the average size of a minimum cardinality diagnosis over all KBs in this experiment was 7. In the second experiment (see below), where STD is not applicable, the cardinality of the target diagnosis is significantly higher.

The second experiment was performed on KBs of the OAEI Conference benchmark which turned out to be problematic for STD. For these KBs we observed that the minimum cardinality diagnoses comprise 18 elements on average. In 11 of the 13 KBs of the second experiment (see Table 2) STD was unable to find any diagnosis within 2 hours. In the other two cases STD succeeded to find one minimal diagnosis for `csa-conference-ekaw` and nine for `ldoa-conference-confof`. However, DIR even succeeded to find 30 minimal diagnoses for each KB within time acceptable for interactive diagnosis settings. Moreover, on average DIR was able to find 1 minimal diagnosis in 8.9 sec., 9 minimal diagnoses in 40.83 sec. and 30 minimal diagnoses in 107.61 sec. (see Column 2 of Table 2). This result shows that DIR is a stable and practically applicable method even in cases where a knowledge base comprises high-cardinality faults.

In the Conference experiment we first selected the target diagnosis $\mathcal{D}_t$ for each $\mathcal{KB}_{ij}$ just as it was done in the described Anatomy case. Next, we evaluated the performance of sequential DIR using both query selection methods. The results of the experiment presented in Table 2 show that DIR found $\mathcal{D}_t$ for each KB. On average DIR solved the problems more efficiently using ENT than SPL because also in the Conference case the confidence values provided a reasonable estimation of axiom fault probabilities. Only in three cases ENT required more queries than SPL. Moreover, the experiments show that the efficiency of debugging methods depends highly on the runtime of the underlying reasoner. For instance, in the hardest case consistency checking took 93.4% of the total time whereas all other operations – including construction of the search tree, generation and selection of queries – took only 6.6% of time. Consequently, sequential DIR requires only a small fraction of computation effort. Runtime improvements can be achieved by advances in reasoning algorithms or the reduction of the number of consistency checks. Currently DIR requires $O(m * |\mathcal{D}| \log(\frac{|\mathcal{KB}|}{|\mathcal{D}|}))$ checks to find m leading diagnoses. A further source for improvements can be observed for the `ldoa-ekaw-iasted` ontology where both methods asked the same number of queries. In this case, ENT required only half of the consistency checks SPL did, but an average consistency check of ENT took almost twice as long as an average one for SPL. The analysis of this ontology showed that there is a small subset of axioms (hot spot) which made reasoning considerably harder. Identification of such hot spots [6] could result in a significant improvement of diagnosis runtime, since a hot spot can be resolved by suitable queries. This can be observed in the `ldoa-ekaw-iasted` case where SPL acquired appropriate test cases early and thereby found $\mathcal{D}_t$ faster.

5 Conclusions

In this paper we presented a sequential diagnosis method for faulty KBs which is based on the direct computation of minimal diagnoses. We reduce the number of consistency checks by avoiding the computation of minimized conflict sets and by computing *some* set of minimal diagnoses instead of a set of most probable diagnoses or a set of minimum cardinality diagnoses. The evaluation results presented in the paper indicate that the performance of the suggested sequential diagnosis system is either comparable with or outperforms the existing approach in terms of runtime and the number of queries in case a KB includes a large number of faults. The scalability of the algorithms was demonstrated on a set of large KBs including thousands of axioms.

REFERENCES

[1] *The Description Logic Handbook: Theory, Implementation, and Applications*, eds., Franz Baader et al., Cambridge Univ. Press, 2007.
[2] Johan de Kleer and Brian C. Williams, 'Diagnosing multiple faults', *Artif. Intel.*, **32**(1), 97–130, (1987).
[3] Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Cássia Trojahn dos Santos, 'Final results of the Ontology Alignment Evaluation Initiative 2011', in *Proceedings of OM-2011*, pp. 1–29. CEUR-WS.org, (2011).
[4] A. Felfernig, M. Schubert, and C. Zehentner, 'An efficient diagnosis algorithm for inconsistent constraint sets', *AI EDAM*, **26**(1), 53–62, (2012).
[5] Gerhard Friedrich and Kostyantyn Shchekotykhin, 'A General Diagnosis Method for Ontologies', in *ISWC*, pp. 232–246. (2005).
[6] Rafael S. Goncalves, Bijan Parsia, and Ulrike Sattler, 'Performance Heterogeneity and Approximate Reasoning in Description Logic Ontologies', in *ISWC*, pp. 82–98, (2012).
[7] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler, 'OWL 2: The next step for OWL', *J. Web Semant.*, **6**(4), 309–322, 2008.
[8] Matthew Horridge, Bijan Parsia, and Ulrike Sattler, 'Laconic and Precise Justifications in OWL', in *ISWC*, pp. 323–338, (2008).

System	Scoring	HS-TREE Time	HS-TREE #Queries	HS-TREE Reaction	INV-HS-TREE Time	INV-HS-TREE #Queries	INV-HS-TREE Reaction
AgrMaker	ENT	19.62	1	19.10	20.83	1	18.23
AgrMaker	SPL	36.04	4	8.76	36.03	4	8.28
GOMMA-bk	ENT	18.34	1	18.07	14.47	1	12.68
GOMMA-bk	SPL	18.95	3	6.15	19.51	3	5.91
GOMMA-nobk	ENT	18.26	1	17.98	14.26	1	12.49
GOMMA-nobk	SPL	18.74	3	6.08	19.47	3	5.89
Lily	ENT	78.54	1	77.71	82.52	1	72.83
Lily	SPL	82.94	4	20.23	115.24	4	26.93
LogMap	ENT	6.60	1	6.30	13.41	1	11.36
LogMap	SPL	6.61	2	3.17	15.13	2	6.82
LogMapLt	ENT	14.85	1	14.54	12.89	1	11.34
LogMapLt	SPL	15.59	3	5.05	17.45	3	5.29
MapSSS	ENT	81.06	4	19.86	56.17	3	17.32
MapSSS	SPL	88.32	5	17.26	77.59	6	12.43

Table 1. HS-TREE and INV-HS-TREE applied to Anatomy benchmark. Time is given in sec, **Scoring** stands for query selection strategy, **Reaction** is the average system reaction time between queries.

Ontology (Expressivity)	30 Diag	min $\|\mathbf{D}\|$	Scoring	Time	#Queries	Reaction	#CC	CC
ldoa-conference-confof	48.06	16	ENT	11.6	6	1.5	430	0.003
$\mathcal{SHIN}(\mathcal{D})$			SPL	11.3	7	1.6	365	0.004
ldoa-cmt-ekaw	42.28	12	ENT	48.6	21	2.2	603	0.016
$\mathcal{SHIN}(\mathcal{D})$			SPL	139.1	49	2.8	609	0.054
mappso-confof-ekaw	55.66	10	ENT	10	5	1.9	341	0.007
$\mathcal{SHIN}(\mathcal{D})$			SPL	31.6	13	2.3	392	0.021
optima-conference-ekaw	62.13	19	ENT	16.8	5	2.6	553	0.008
$\mathcal{SHIN}(\mathcal{D})$			SPL	16.1	8	1.9	343	0.012
optima-confof-ekaw	44.52	16	ENT	24	20	1.1	313	0.014
$\mathcal{SHIN}(\mathcal{D})$			SPL	17.6	10	1.7	501	0.006
ldoa-conference-ekaw	56.98	16	ENT	56.7	35	1.5	253	0.053
$\mathcal{SHIN}(\mathcal{D})$			SPL	25.5	9	2.7	411	0.016
csa-conference-ekaw	62.82	17	ENT	6.7	2	2.8	499	0.003
$\mathcal{SHIN}(\mathcal{D})$			SPL	22.7	8	2.7	345	0.02
mappso-conference-ekaw	70.46	19	ENT	27.5	13	1.9	274	0.028
$\mathcal{SHIN}(\mathcal{D})$			SPL	71	16	4.2	519	0.041
ldoa-cmt-edas	15.47	16	ENT	24.7	22	1	303	0.008
$\mathcal{ALCOIN}(\mathcal{D})$			SPL	11.2	7	1.4	455	0.002
csa-conference-edas	39.74	26	ENT	18.4	6	2.7	419	0.005
$\mathcal{ALCHOIN}(\mathcal{D})$			SPL	240.8	37	6.3	859	0.036
csa-edas-iasted	377.36	20	ENT	1744.6	3	349.2	1021	1.3
$\mathcal{ALCOIN}(\mathcal{D})$			SPL	7751.9	8	795.5	577	11.5
ldoa-ekaw-iasted	229.72	13	ENT	23871.5	9	1886	287	72.6
$\mathcal{SHIN}(\mathcal{D})$			SPL	20449	9	2100.1	517	37.2
mappso-edas-iasted	293.74	27	ENT	18400.3	5	2028.3	723	17.8
$\mathcal{ALCOIN}(\mathcal{D})$			SPL	159299	11	13116.6	698	213.2

Table 2. Sequential diagnosis using direct computation of diagnoses. **30 Diag** is the time required to find 30 minimal diagnoses, **min** $|\mathcal{D}|$ is the cardinality of a minimum cardinality diagnosis, **Scoring** indicates the query selection strategy, **Reaction** is the average system reaction time between queries, **#CC** number of consistency checks, **CC** gives average time needed for one consistency check. Time is given in sec.

[9] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau, 'LogMap: Logic-based and scalable ontology matching', in *ISWC*, pp. 273–288, (2011).

[10] Ulrich Junker, 'QUICKXPLAIN: Preferred Explanations and Relaxations for Over-Constrained Problems', in *AAAI*, pp. 167–172, (2004).

[11] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin, 'Finding all Justifications of OWL DL Entailments', in *ISWC*, pp. 267–280, (2007).

[12] Raymond Reiter, 'A Theory of Diagnosis from First Principles', *Artif. Intel.*, **32**(1), 57–95, (1987).

[13] Kostyantyn Shchekotykhin, Gerhard Friedrich, Philipp Fleiss, and Patrick Rodler, 'Interactive ontology debugging : two query strategies for efficient fault localization', *J. Web Semant.*, **12-13**, 88–103, (2012).

[14] Iulia Nica, Ingo Pill, Thomas Quaritsch and Franz Wotawa, 'The route to success: a performance comparison of diagnosis algorithms', in *IJCAI*, pp. 1039–1045, (2013).

[15] Franz Baader, Martin Knechtel, and Rafael Peñaloza, 'Context-dependent views to axioms and consequences of Semantic Web ontologies', *J. Web Semant.*, **12-13**, 22–40, (2012).

[16] Jianfeng Du, Guilin Qi, Jeff Z. Pan, and Yi-Dong Shen, 'A Decomposition-Based Approach to OWL DL Ontology Diagnosis', in *Proceedings of 23rd ICTAI*, pp. 659–664, (2011).

[17] Ken Satoh and Takeaki Uno, 'Enumerating Minimally Revised Specifications Using Dualization', in *JSAI'05*, pp. 182–189, (2005).

[18] Roni Stern, Meir Kalech, Alexander Feldman, and Gregory Provan, 'Exploring the Duality in Conflict-Directed Model-Based Diagnosis', in *AAAI'12*, pp. 828–834, (2012).