SHIN ABox Reduction

Achille Fokoue, Aaron Kershenbaum, Li Ma,
Edith Schonberg, Kavitha Srinivas, Rose Williams
IBM Research, Hawthorne, NY 10532
achille, aaronk, malli, ediths, ksrinivs, rosemw@us.ibm.com

May 12, 2006

Abstract

We propose a technique to make consistency detection scalable for large Aboxes in secondary storage. We use static analysis of knowledge representation with summarization techniques to produce a dramatically reduced *proxy Abox*. We show that the *proxy Abox* is consistent only if the original Abox is also consistent. We also show that, in practice, our techniques dramatically reduce the time and space requirements for consistency detection.

1 Introduction

All common reasoning tasks in expressive DL ontologies reduce to consistency detection, which is well known to be intractable in the worst-case [2]. Given that the size of an Abox may be in the order of millions of assertions, this complexity poses a serious challenge for the practical use of DL ontologies.

We propose new techniques that make consistency detection scalable for SHIN Aboxes with millions of assertions. Ontology Aboxes often reside in transactional databases, so we do not assume that an Abox will fit into memory. With these new techniques, we are able to efficiently extract a small set of assertions from the database that represents the entire Abox, and reason over this small set in memory.

We first use static analysis to isolate a subportion of the Abox that captures all the *global* effects that can occur in reasoning, henceforth referred to as the global-effects Abox. By *global* effects, we mean effects that propagate through the Abox to affect an individual's membership in a given concept. In practice, isolating *global effects* results in substantial reductions in the size of the Abox, because most roles participate in local rather than global effects.

Once we've isolated *global* effects in the Abox, we use summarization techniques to dramatically reduce this subportion of the Abox further to produce a *proxy Abox*. For example, in the largest of the 4 ontologies that we studied, we reduced an Abox of 874K individuals and 3.5 million assertions to a *proxy Abox* with 18 individuals and 49 role assertions.

The power of creating such a *proxy* is that we can replace the consistency check on the global-effects Abox $\mathcal{A}'$ with a consistency check on a dramatically reduced *proxy*. Specifically, if the *proxy* is consistent, we are guaranteed that $\mathcal{A}'$ is consistent. If the *proxy* is inconsistent, it can still be used to partition $\mathcal{A}'$.

Our key contributions in this paper are as follows: (a) We present a technique to use static analysis of knowledge representation to isolate a portion of the Abox where *global* effects are possible. (b) We construct a dramatically reduced *proxy* of this portion using summarization techniques. (c) We present a method to further identify local effects in the global-effects Abox, based on the proxy Abox, to handle cases where the summarization may have been too conservative in building the *proxy*. (d) We show the efficiency of these techniques applied to 4 real ontologies.

2 Local/Global Effect Partitioning

We present several criteria for safely removing role assertions from an Abox $\mathcal{A}$. The reduced Abox is consistent iff $\mathcal{A}$ is consistent. In practice, applying these criteria results in both significantly shrinking the Abox and in partitioning it into many disconnected Aboxes, many of which consist of a single individual, which can be checked for consistency using concept satisfiability.

Our criteria for role assertion removal is based on the assumption that any concept in the $clos(\mathcal{A})$ can reach the concept set of any individual in $\mathcal{A}$ from the application of tableau expansion rules for SHIN. We define $clos(\mathcal{A})$ as $\bigcup_{a:C \in \mathcal{A}} clos(C)$ where $clos(C)$ is a set that contains C and all its sub-concepts (where C is in NNF). Note that our formal definition of $clos(\mathcal{A})$ differs from [6] in that we do not include $\neg C$ in $clos(C)$, due to lesser expressiveness of SHIN compared to SHIQ.

We assume that a and b are named individuals in the original $\mathcal{A}$, x is a new unnamed individual introduced as a result of tableau expansion rules, C is a concept in $clos(\mathcal{A})$, and R is a role. Let $\mathcal{L}$ be a mapping from each individual in $\mathcal{A}$ to a set of concepts in $clos(\mathcal{A})$, such that $a:C \in \mathcal{A}$ iff $C \in \mathcal{L}(a)$. $\mathcal{L}(a)$ is the *concept set* of a. An individual b is said to be an R-neighbor of a iff there is an assertion $Q(a,b)$ or $Q^{-}(b,a)$ in $\mathcal{A}$ where $Q \in \underline{R}$ (where $\underline{\mathrm{R}} = \{ Q | \ Q \sqsubseteq^{*} R\}$ and $\sqsubseteq^{*}$ is the reflexive transitive closure of the sub-role relation). The SHIN tableau expansion rules can merge individuals, add membership assertions of the form $a:C$ where $C \in clos(\mathcal{A})$, add unnamed individuals, and add new role assertions of the form $R(a, x)$ or $R(a, b)$ to $\mathcal{A}$.

We say that an expansion rule has a *global effect* if it uses an existing role assertion to add new assertions to $\mathcal{A}$ or to detect a clash. For example, a concept C will be propagated to the concept set $\mathcal{L}(b)$ of a named individual b if $a : \forall R.C$ and $R(a, b) \in \mathcal{A}$. The $\forall$-rule, $\leq$-rule, and $\forall_+$-rule can have global effects. In contrast, the $\exists$-rule and $\geq$-rule do not use any existing role assertions to alter the Abox, and are hence local effect rules; but these rules can generate new role assertions, so we call concepts of the form $(\exists R.C)$ and $(\geq nR)$ *R-generators*.

A role assertion $R(a, b)$ is said to be a global-effect role assertion iff there is at least one execution of the tableau algorithm in which it is used by a global-effect rule, or it is part of an explicit clash involving both a and b. Otherwise, it is a local-effect role assertion and cannot affect the outcome of a consistency check. Our criteria are designed to detect and remove local effect role assertions.

2.1 Role-based Local Effect Detection

We make the simple observation that if roles R and R^- are never used in any universal or maximum cardinality restrictions, then a role assertion $R(a, b)$ can never be used in a global-effect rule, so it can safely be ignored.

Definition 1: A role R is *part of a universal restriction* $\forall P.C$ iff $R \in \underline{P}$. (Similarly for maximum cardinality restriction). A role R is *part of the universal restrictions of an Abox* $\mathcal{A}$ iff there is a universal restriction $\forall P.C \in clos(\mathcal{A})$ such that $R \in \underline{P}$. (Similarly for maximum cardinality restrictions).

Theorem 2: A role assertion $R(a, b)$ can safely be removed from an Abox $\mathcal{A}$ if neither R nor its inverse R^- is part of the universal restrictions or maximum cardinality restrictions of $\mathcal{A}$.

Proof(*sketch*): Direct consequence of more the general Theorem 9. ∎

2.2 Assertion-based Local Effect Detection

We specify criteria for local-effect role assertion removal even in the presence of universal and maximum cardinality restrictions. The criteria for $\forall$-rule and $\forall_+$-rule is as follows: Let role R be part of a universal restriction $\forall P.C$. A role assertion $R(a, b)$ is *removable with respect to* $\forall P.C$ iff $b : C \in \mathcal{A}$ and R has no transitive superroles. $R(a, b)$ is *removable with respect to universal restrictions in an Abox* $\mathcal{A}$ iff it is removable with respect to all universal restrictions $\forall P.C$, where R or R^- is part of $\forall P.C$.

Next, we note that for the $\leq$-rule to have a global effect from merging there must be a maximum cardinality restriction $(\leq nP)$ in $clos(\mathcal{A})$, and an individual a with more than n P-neighbors. To ensure that an individual a has no more than n P-neighbors, we need to be able to compute an upper bound on P-neighbors of a safely. In particular, when counting $a's$ P-neighbors, it is

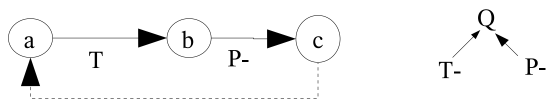

Figure 1: Effect of mergers on *a's* neighbors

important to include named individuals and unnamed individuals that can become P-neighbors through P-generators and mergers. One example of a merger is shown in Figure 1, where c can be merged with a because P^- is attracted to T^- through a shared super role Q which is part of a maximum cardinality restriction. We define the conditions under which the upper bound of $a's$ P-neighbors can be computed safely and efficiently below. First, we introduce the notion of an an *attractant* for P, to prevent new P-neighbors from mergers as shown in Figure 1:

Definition 3: For a given role P, we define $attractant(P)$ as follows: $T \in attractant(P)$ iff there is a role Q such that $P \sqsubseteq^* Q, T \sqsubseteq^* Q, \leq nQ \in clos(\mathcal{A})$.

We say that P is *safe* in $\mathcal{A}$ iff one of the following conditions is satisfied for the role P:

(a) the $attractant(P) \subseteq \{P\}$ and $attractant(P^-) \subseteq \{P^-\}$, or

(b) for all roles R such that either R or R^- is in $\underline{P}$, there are no R-generators in $clos(\mathcal{A})$.

Definition 4: An individual a in $\mathcal{A}$ is *mergeable* iff at some step of any execution of the tableau algorithm on $\mathcal{A}$, a is merged with a *named* individual b.

Lemma 5: Let P be a role that is safe in $\mathcal{A}$. During an execution of the tableau algorithm on $\mathcal{A}$ the following holds: if there is a unnamed individual x such that P or P^- is in the $\mathcal{L}(< parent(x), x >)$, then $|\mathcal{L}(< parent(x), x >)| = 1$, where $parent(x)$ denotes the parent node of x in the completion forest.

Proof: Proved by induction [3] on the iterations of the tableau algorithm.

Theorem 6: An individual a in $\mathcal{A}$ is *not mergeable* in $\mathcal{A}$ if, for any role P and any individual b in $\mathcal{A}$, the following conditions hold:

(1) if a is a P-neighbor of b there is no concept $(\leq nP)$ in $clos(\mathcal{A})$, and

(2) if b is a P-neighbor of a then P is safe in $\mathcal{A}$.

Proof *sketch*: By induction using Lemma 5 [3].■

Definition 7: Let role R be part of a maximum cardinality restriction $(\leq nP) \in clos(\mathcal{A})$. A role assertion $R(a, b)$ is *removable with respect to* $(\leq nP)$ iff

(1) if a is a Q-neighbor of a named individual c, there is no concept of the form $(\leq nQ)$ in $clos(\mathcal{A})$, and

(2) if a named individual c is a Q-neighbor of a then Q is safe in $\mathcal{A}$, and

(3) P is safe in $\mathcal{A}$ and its only mininum cardinality is of the form $\geq 1P$ $(\exists P.\top)$, and

(4) $|P(a)| + |Some(P, a)| \leq n$, where $Some(P, a) = \{\exists P.C \in clos(\mathcal{A}) \mid$ there is no P-neighbor c of a such that $c : C \in \mathcal{A}\}$.

$R(a,b)$ is *removable with respect to maximum cardinality restrictions in an Abox* $\mathcal{A}$ iff it is removable with respect to all maximum cardinality restrictions $\leq nR$, where R or R^- is part of $\leq nR$. Note that, by Theorem 6, (1) and (2) imply that a is not mergeable in $\mathcal{A}$.

Definition 8: A role assertion $R(a,b)$ is *removable with respect to maximum cardinality restrictions in an Abox* $\mathcal{A}$ iff the following holds: if R (resp. R^-) is part of a maximum cardinality restriction $(\leq nP) \in clos(\mathcal{A})$, then $R(a,b)$ (resp. $R^-(b,a)$) is removable with respect to $(\leq nP)$. $R(a,b)$ is *removable with respect to maximum cardinality restrictions in an Abox* $\mathcal{A}$ iff it is removable with respect to all maximum cardinality restrictions $\leq nP$, where R or R^- is part of $\leq nP$.

Theorem 9: A role assertion $R(a,b)$ can safely be removed from an Abox $\mathcal{A}$ if it is removable with respect to universal restrictions and removable with respect to maximum cardinality restrictions.

Proof Sketch: Let $R(a,b)$ be a role assertion removable w.r.t. maximum cardinality and universal restrictions in an Abox $\mathcal{A}$. Let $\mathcal{A}'$ be the Abox defined as $\mathcal{A}' = \mathcal{A} - \{R(a,b), R^-(b,a)\}$. If $\mathcal{A}$ is consistent, $\mathcal{A}'$ is obviously consistent. We show that if $\mathcal{A}'$ is consistent, a model of $\mathcal{A}$ can be constructed by applying the tableau algorithm rules in a particular way. [1]

First, for a root node c in the completion forest F, the root node $\alpha(c)$ is defined as follows: if $\mathcal{L}(c) \neq \emptyset$ then $\alpha(c) = c$; otherwise, $\alpha(c) = d$, where d is the unique root node in F with $\mathcal{L}(d) \neq \emptyset$ and $d \doteq c$. Since $\mathcal{A}'$ is consistent, we can apply the tableau expansion rules on $\mathcal{A}'$ without creating a clash in such a way that: (1) $\exists$-rule is never triggered to satisfy a constraint $\exists P.C \in \mathcal{L}(\alpha(a))$ (resp. $\mathcal{L}(\alpha(b))$) where $\leq nP \in clos(\mathcal{A})$, R (resp. R^-) is part of $\leq nP$, and $b : C \in \mathcal{A}$ (resp. $a : C \in \mathcal{A}$), and (2) $\geq$-rule is never triggered to satisfy a constraint $\geq nP \in \mathcal{L}(\alpha(a))$ (resp. $\mathcal{L}(\alpha(b))$) where $\leq nP \in clos(\mathcal{A})$, R (resp. R^-) is part of $\leq nP$, and, in the Abox $\mathcal{A}$, b (resp. a) is one of the n R-neighbors of a (resp. R^--neighbors of b) explicitly asserted to be distinct.

Such a rule application yields a clash-free completion forest F, and the only nodes on which expansion rules may be applicable are $\alpha(a)$ and $\alpha(b)$ (the only applicable rules are $\exists$-rule and $\geq$-rule). Next, we modify F to create a completion forest F' by adding to F the edge $< \alpha(a), \alpha(b) >$ if it was not already in F, and by adding R to $\mathcal{L}(< \alpha(a), \alpha(b) >)$, if it was not already there. We show that F' is complete (i.e. no rules are applicable) and clash-free.

The fact that, in F', $R \in \mathcal{L}(< \alpha(a), \alpha(b) >)$ ensures that the $\exists$ and $\geq$ rules, which may have been applicable on $\alpha(a)$ or $\alpha(b)$ in F, are not applicable on $\alpha(a)$ and $\alpha(b)$ in F'. However, the same fact may now make the $\forall$, $\forall_+$, $\leq$, and $\leq_r$ rules applicable on $\alpha(a)$ or $\alpha(b)$ in F'. We show that this cannot be the case.

The definition of removable w.r.t. universal restrictions obviously ensures that $\forall$ and $\forall_+$ rules are not applicable on $\alpha(a)$ or $\alpha(b)$ in F'. It can be shown

[1] A direct model-theoretic proof cannot easily provided here, see [3] for details.

[3] that $\leq$, and $\leq_r$ rules cannot be applicable on $\alpha(a)$ or $\alpha(b)$ in F' and that F' is still clash free. Thus, a tableau for $\mathcal{A}$ can be built from F' as in [6], which establishes that $\mathcal{A}$ has a model. ■

3 Proxy Abox

Intuitively, the Abox contains many redundant assertions from the point of view of consistency checking that can be collapsed to create a reduced *proxy Abox*. As an example, if the Abox contains assertions of the form $R(m,c)$ and $R(j,y)$, where m and j are both members of W and c and y are both members of U, we can replace m and j by a proxy individual $w:W$ that is connected by a R relation to a proxy individual $u:U$. Reasoning over the resulting proxy Abox corresponds to reasoning over the original Abox, as shown formally below.

Definition 10: A proxy Abox is an Abox $\mathcal{A}''$ that is generated from any SHIN Abox $\mathcal{A}$ using a mapping function $\mathbf{f}$ that satisfies the following constraints, where $\mathcal{R}$ is the set of all roles and their inverses in an Abox:

(1) if $a:C \in \mathcal{A}$ then $\mathbf{f}(a):C \in \mathcal{A}''$ (2) if $R(a,b) \in \mathcal{A}$ then $R(\mathbf{f}(a),\mathbf{f}(b)) \in \mathcal{A}''$

(3) if $a \neq b \in \mathcal{A}$ then $\mathbf{f}(a) \neq \mathbf{f}(b) \in \mathcal{A}''$

Theorem 11: If the proxy Abox $\mathcal{A}''$ obtained by applying the mapping function $\mathbf{f}$ to $\mathcal{A}$ is consistent then $\mathcal{A}$ is consistent. However, the converse of Theorem 11 does not hold.

Proof: Let us assume that $\mathcal{A}'$ is consistent w.r.t. $\mathcal{T}$ and $\mathcal{R}$. Therefore there is a model $\mathcal{I}' = (\Delta^{\mathcal{I}'}, .^{\mathcal{I}'})$ of $\mathcal{A}'$ w.r.t. $\mathcal{T}$ and $\mathcal{R}$. A model of $\mathcal{A}$ can easily be built from $\mathcal{I}'$ by interpreting an individual a in $\mathcal{A}$ in the same way as $\mathbf{f}(a)$ is interpreted by $\mathcal{I}'$. Formally, let $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ be the interpretation of the $\mathcal{A}$ w.r.t. $\mathcal{T}$ and $\mathcal{R}$ defined as follows: $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$; for a concept $C \in \mathcal{T}$, $C^{\mathcal{I}} = C^{\mathcal{I}'}$; for a role R in $\mathcal{R}$, $R^{\mathcal{I}} = R^{\mathcal{I}'}$; for an individual a in $\mathcal{A}$, $a^{\mathcal{I}} = \mathbf{f}(a)^{\mathcal{I}'}$. $\mathcal{I}$ is a model of $\mathcal{A}$ w.r.t. $\mathcal{T}$ and $\mathcal{R}$ as a direct consequence of the fact that $\mathcal{I}'$ is a model of $\mathcal{A}'$ and $\mathcal{A}'$ satisfies the 3 conditions stated in definition 1 (see [3] for more details). ■

The mapping function $\mathbf{f}$ that we use to create a proxy Abox from a global-effects Abox is defined such that if $\mathcal{L}(a) = \mathcal{L}(b)$ and $a \neq b \notin \mathcal{A}$, then $\mathbf{f}(a) = \mathbf{f}(b)$. That is, all individuals in the Abox $\mathcal{A}$ which have the same concept set and are not asserted to be distinct map to the same individual in the proxy Abox $\mathcal{A}''$. If a proxy Abox $\mathcal{A}''$ is not consistent, either there is a real inconsistency in $\mathcal{A}$ or the process of collapsing individuals to create $\mathcal{A}''$ caused an artificial inconsistency. To determine whether an inconsistency is real, we consider the global-effects Abox $\mathcal{A}'$. Like $\mathcal{A}'$, $\mathcal{A}''$ may consist of disconnected Aboxes and, since $\mathcal{A}''$ is typically small, it is not expensive to identify these partitions. Furthermore, we know from the consistency check which partitions in $\mathcal{A}''$ are inconsistent. For each inconsistent partition $\mathcal{A}_i''$ in $\mathcal{A}''$, we test for consistency the assertions in $\mathcal{A}'$ that map into it, which form a distinct partition $\mathcal{A}_i'$ in $\mathcal{A}'$. If any partition in $\mathcal{A}'$ is inconsistent, then $\mathcal{A}$ is inconsistent.

Table 1: Characteristics of $\mathcal{A}$, $\mathcal{A}$', and $\mathcal{A}$"

KB	Classes			Roles			Instances			Role Assertions		
	$\mathcal{A}$	$\mathcal{A}$'	$\mathcal{A}$"	$\mathcal{A}$	$\mathcal{A}$'	$\mathcal{A}$"	$\mathcal{A}$	$\mathcal{A}$'	$\mathcal{A}$"	$\mathcal{A}$	$\mathcal{A}$'	$\mathcal{A}$"
BioPax	31	14	14	40	2	2	261K	17K	39	582K	14K	106
LUBM	91	25	19	27	5	3	142K	44K	481	736K	45K	352
NIMD	19	2	2	28	1	1	1,278K	429K	2	2,000K	286K	1
ST	16	15	15	11	2	2	874K	547K	18	3,595K	580K	49

4 Computational Experience

We tested our approach on the four actual ontologies shown in Table 1. $\mathcal{A}$ corresponds to the original Abox, $\mathcal{A}'$ is the Abox obtained from $\mathcal{A}$ after removing local effects, and $\mathcal{A}''$ is the proxy Abox. Biopax includes the data for 11 organisms available at http://biocyc.org. We used a version of LUBM that was modified to SHIN [7] . The Abox of the NIMD ontology was generated from text analysis programs run over a large number of unstructured documents. The semantic traceability (ST) ontology Abox was generated from a program that extracted relationships between software artifacts of a large middleware application.

As can be seen in Table 1, in practice, $\mathcal{A}'$ is significantly smaller than $\mathcal{A}$, and $\mathcal{A}''$ is a substantial reduction over $\mathcal{A}'$. In most cases, $\mathcal{A}'$ was sufficient for the consistency check. For ST, a real inconsistency was detected in the local-effects consistency check. The Biopax and NIMD ontologies were consistent in $\mathcal{A}''$. For LUBM, we needed to check the consistency of $\mathcal{A}'$. The running time for our algorithm took from 12.6 seconds to 283 seconds, which included the time for building the proxy and checking it for consistency.

5 Related Work and Conclusion

There are many highly optimized reasoners such as Pellet [8], Racer [4], InstanceStore [1], and Kaon2 [10] designed for consistency checking, but only InstanceStore and Kaon2 can be extended to Aboxes in secondary storage. Kaon2 applies to deductive databases, whereas our techniques work with relational databases. InstanceStore is limited to role-free Aboxes. In theory, Instance Store can handle Aboxes with role assertions through a technique called precompletion [9], but this may not be practical for Aboxes stored in databases. Our techniques can be contrasted with optimization techniques such as model caching and Abox contraction [5], but again, it is unclear how such techniques can be applied to Aboxes in databases.

We have demonstrated a technique to scale consistency detection to large Aboxes in secondary storage by extracting a small representative Abox. Further, we have shown that, in practice, this technique works efficiently on four large ontologies. Our plan is to extend this approach to apply more accurate static analysis techniques, extend its applicability to more expressive languages, and to apply these techniques to query processing.

References

[1] Sean Bechhofer, Ian Horrocks, and Daniele Turi. The owl instance store: System description. *Proc. of 20th Int.Conf. on Automated Deduction*, pages 177–181, 2005.

[2] F. Donini. Complexity of reasoning. In F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors, *Description Logic Handbook*, pages 101–141. Cambridge University Press, 2002.

[3] Achille Fokoue, Aaron Kershenbaum, Li Ma, Edith Schonberg, and Kavitha Srinivas. Scalable reasoning:cutting ontologies down to size. In *http://www.research.ibm.com/iaaa/aaaiSubmission.ps*, 2006.

[4] V. Haarslev and R. Moller. Racer system description. *Conf. on Automated Reasoning (IJCAR 2001)*, pages 701–705, 2001.

[5] Volker Haarslev and Ralf Moller. An empirical evaluation of optimization strategies for abox reasoning in expressive description logics. *Proc. of the International Workshop on Description Logics*, pages 115–199, 1999.

[6] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with individuals for the description logic SHIQ*. *Proc. of 17th Int.Conf. on Automated Deduction*, pages 482–496, 2000.

[7] Li Ma, Yang Yang, Zhaomin Qiu, Guotong Xie, and Yue Pan. Towards a complete owl ontology benchmark. In *Proc. of the third European Semantic Web Conf.(ESWC 2006)*, 2006.

[8] Evren Sirin and Bijan Parsia. Pellet: An owl dl reasoner. In *Description Logics*, 2004.

[9] Sergio Tessaris and Ian Horrocks. Abox satisfiability reduced to terminological reasoning in expressive description logics. In *LPAR*, pages 435–449, 2002.

[10] U.Hustadt, B. Motik, and U. Sattler. Reducing shiq description logic to disjunctive datalog programs. *Proc. of 9th Intl. Conf. on Knowledge Representation and Reasoning (KR2004)*, pages 152–162.