On the implementation of a Fuzzy DL Solver over Infinite-Valued Product Logic with SMT Solvers

Teresa Alsinet[1], David Barroso[1], Ramón Béjar[1], Félix Bou[2], Marco Cerami[3], and Francesc Esteva[2]

[1] Department of Computer Science – University of Lleida
C/Jaume II, 69 – 25001 Lleida, SPAIN
{tracy, david, ramon}@diei.udl.cat
[2] Artificial Intelligence Research Institute (IIIA-CSIC)
Campus UAB - 08193 Bellaterra, Barcelona, SPAIN
{fbou,esteva}@iiia.csic.es
Department of Computer Science – Palacký University in Olomouc
17. listopadu 12 – CZ-77146 Olomouc, CZECH REPUBLIC
marco.cerami@upol.cz

Abstract. In this paper we explain the design and preliminary implementation of a solver for the positive satisfiability problem of concepts in a fuzzy description logic over the infinite-valued product logic. The same solver also works for 1-satisfiability in quasi-witnessed models. The solver works by first performing a direct reduction of the problem to a satisfiability problem of a quantifier free boolean formula with non-linear real arithmetic properties, and secondly solves the resulting formula with an SMT solver. We show that the satisfiability problem for such formulas is still a very challenging problem for even the most advanced SMT solvers, and so it represents an interesting problem for the community working on the theory and practice of SMT solvers. We briefly explain a possible way of improving the performance of the solver by an alternative implementation under development, based on a reduction to a boolean formula but with linear real arithmetic properties.

Keywords: description logics, fuzzy product logic, SMT solvers

1 Introduction

In the recent years, the development of solvers for reasoning problems over description logics (DLs) has experienced an important growth, with very succesful approaches. We have two main approaches, the most traditional one, able to handle very expressive DLs, is the one based on Tableaux-like algorithms [1]. For certain DLs, the approach based on translations of the problem to more basic logical reasoning problems, like the ones based on a translation to propositional clausal forms has shown to be very sucessful [8]. Very recently, the approach based on doing translations to less simple knowledge representation formalisms and then using Sat Modulo Theory (SMT) solvers, has started to receive high interest [6].

In the case of DLs over fuzzy logics (Fuzzy DLs), the state-of-the-art on solvers can be summarized mainly on the work of Straccia et al. with *fuzzyDL*, their solver for Fuzzy DL over Łuckasiewicz Logic [9] (available in Straccia's home page), that is based on a mixture of tableau rules and Mixed Integer Linear Programming. Note that the problem faced in [9], called *concept satisfiability w.r.t. knowledge bases with GCI*, is more general than the one faced in the present paper. Unfortunately, the general problem of concept satisfiability w.r.t. knowledge bases with GCI over infinite-valued Łukasewicz semantics has been proved to be undecidable in [3]. Nevertheless, as proved in [3], the solver proposed in [9] can solve the concept satisfiability problem without knowledge bases.

In this work, we present a solver for the concept satisfiability problem without knowledge bases, in the Fuzzy DL $\mathcal{ALE}$ over the infinite-valued product logic. This problem has been studied in [2] and the FDL under exam has been denoted Π-$\mathcal{ALE}$. Our approach is based on the last work, where the authors show that the positive and 1-satisfiability problems in Π-$\mathcal{ALE}$ limited to quasi-witnessed models are decidable.

To prove the above result the authors give a reduction (inspired by the one given by Hájek for witnessed models [7]) of the concept satisfiability problem in Π-$\mathcal{ALE}$ with respect to quasi-witnessed models to an entailment problem between two set of propositional formulas. The algorithm presented in [2] takes a description concept C_0 as input and recursively produces a pair of propositional theories as output. The propositional theories produced as output jointly represent a description of an FDL interpretation (a kind of Kripke model) that is supposed to satisfy concept C_0 (in the case, obviously that C_0 is satisfiable) in the sense that C_0 is satisfiable if and only if it can proved that one of the propositional theories is not entailed by the other. The novelty of the algorithm presented in [2] is that it can describe possibly infinite models by means of a finite set of propositional formulas.

For this reason, the algorithm is much more complex than the one of Hájek. The algorithm proposed by Hájek, indeed, just produces one propositional theory with the property of being satisfiable if and only if the concept C_0 is satisfiable with respect to witnessed models. This is the advantage of dealing with witnessed models, which provide the calculus considered with the finite model property. But in the case of quasi-witnessed models this property is missing and there can be the case of dealing with infinite models of a certain shape. In this sense the two propositional theories in the output of the algorithm presented in [2] represent positive and negative constraints that this kind of structures must respect in order to be models for the concepts considered. In this sense, the problem of finding a propositional evaluation that satisfies the set of propositions $QWT(C_0)$ but not the set Y_{C0}, is exactly the problem of deciding whether Y_{C0} is not entailed by $QWT(C_0)$.

Moreover they prove that positive satisfiability in first order product logic (and, as a consequence, in Π-$\mathcal{ALE}$) coincide with positive satisfiability with respect to the quasi-witnessed models of first order product logic. If the same completeness result holds for the notion of 1-satisfiability is still an open problem.

In this paper we present a solver that works by first performing a direct reduction of the problem to a satisfiability problem of a boolean formula with real valued variables and non-linear terms, more concretely boolean formulas valid in $(\mathbb{R}, +, -, \cdot, /, \{q : q \in \{\mathbb{Q}\})$, and secondly solves the resulting formula with a SMT solver able to solve such formulas. Solving such formulas is still a very challenging problem for even the most advanced SMT solvers, and in this work we show results that indicate that this satisfiability problem for Π-$\mathcal{ALE}$ is a real challenging problem for SMT solvers, and so it represents an interesting problem for the community working of the theory and practice of SMT solvers.

2 An SMT-based Solver for the Π-$\mathcal{ALE}$ Description Logic

2.1 Global System Architecture

For solving the satisfiability problem, with witnessed or quasi-witnessed models, of an input concept C_0 in Π-$\mathcal{ALE}$, our system follows the next steps:

1. The user introduces the expression of the concept C_0 to be solved, and selects a class of models to search: witnessed or quasi-witnessed.
2. From the parsing tree of C_0, we either generate the set WT_{C_0} (for witnessed models) or the set QWT_{C_0} (for quasi-witnessed models).
3. We obtain a corresponding formula F_{C_0}, from WT_{C_0} or QWT_{C_0}, such that it will have a solution in $(\mathbb{R}, +, -, \cdot, /, \{q : q \in \{\mathbb{Q}\})$ if C_0 is satisfiable with the class of models we have selected. This is explained in more detail in the next subsection.
4. The formula F_{C_0} is solved with a suitable SMT solver.

In our current implementation we use the SMT solver Z3 [5] although the formula F_{C_0} to be solved is generated in SMT 2.0 format, so we can use any SMT solver able to solve formulas in $(\mathbb{R}, +, -, \cdot, /, \{q : q \in \{\mathbb{Q}\})$. There is an on-line version of the solver available at the URL: http://arinf.udl.cat/fuzzydlsolver.

2.2 Translation of Fuzzy Propositional Axioms to Non-linear Real Arithmetic Formulas

In [2] the authors showed a translation of the r-satisfiability problem with respect to quasi-winessed models of a concept C_0 over the logic Π-$\mathcal{ALE}$ to an entailment problem of a propositional theory QWT_{C_0} in Product Logic. Instead on trying to solve directly QWT_{C_0}, our approach is based on a reduction to the problem of solving the satisfiabilitity of a corresponding formula F_{C_0} built over quantifier-free real non-linear arithmetic logic such that F_{C_0} is satisfiable if and only if the concept C_0 is r-satisfiable in a quasi-witnessed model over Π-$\mathcal{ALE}$. We explain first the reduction for the particular case of witnessed models, presented in the work of Hájek, that is based on a different fuzzy propositional theory WT_{C_0}.

For every proposition $p \in WT_{C_0}$, we generate a corresponding formula $f(p)$ over quantifier-free non-linear real arithmetic logic. See Definition 3 in [7] for a detailed explanation of all the axioms in WT_{C_0} obtained from an input concept C_0 or Definition 10 in [2] for the corresponding explanation of the axioms in QWT_{C_0} for the more general case of quasi-witnessed models. The formulas to generate depend on the form of the proposition p, and are indicated in Table 1. In the table, $ite(C, A, B)$ is a shorthand for: if condition C is true, then A must be true, else B must be true and $it(C, A)$ is a shorthand for: if condition C is true, then A must be true. For example, the formula of the first row indicates that real value assigned to the propositional variable of an universal concept, $pr(\forall R.C(d_\sigma))$, must be equal to 1 if $pr(R(d_\sigma, d_{\sigma,n})) \leq pr(C(d_{\sigma,n}))$ and $pr(C(d_{\sigma,n}))/pr(R(d_\sigma, d_{\sigma,n}))$ otherwise.

$p \in WT_{C_0}$	$f(p) \in F_{C_0}$
$(\forall R.C(d_\sigma) \equiv (R(d_\sigma, d_{\sigma,n}) \rightarrow C(d_{\sigma,n})))$	$ite(pr(R(d_\sigma, d_{\sigma,n})) \leq pr(C(d_{\sigma,n})), pr(\forall R.C(d_\sigma)) = 1,$ $pr(\forall R.C(d_\sigma)) \cdot pr(R(d_\sigma, d_{\sigma,n})) = pr(C(d_{\sigma,n})))$
$(\exists R.C(d_\sigma) \equiv (R(d_\sigma, d_{\sigma,n}) \boxdot C(d_{\sigma,n})))$	$pr(\exists R.C(d_\sigma)) = pr(R(d_\sigma, d_{\sigma,n})) * pr(C(d_{\sigma,n}))$
$\forall R.C(d_\sigma) \rightarrow (R(d_\sigma, d_{\sigma,m}) \rightarrow C(d_{\sigma,m}))$	$it(pr(R(d_\sigma, d_{\sigma,m}) > pr(C(d_{\sigma,m})),$ $pr(\forall R.C(d_\sigma)) \leq \frac{pr(C(d_{\sigma,m}))}{pr(R(d_\sigma, d_{\sigma,m}))})$
$(R(d_\sigma, d_{\sigma,m}) \boxdot C(d_{\sigma,m}))) \rightarrow \exists R.C(d_\sigma)$	$pr(R(d_\sigma, d_{\sigma,m})) \cdot pr(C(d_{\sigma,m})) \leq pr(\exists R.C(d_\sigma))$

Table 1. Reduction of formulas from the propositional theory WT_{C_0} to formulas in the corresponding set of non-linear arithmetic boolean formulas F_{C_0}.

Then, to solve the r-satisfiability problem of concept C_0 we must determine whether:

$$F_{C_0} \cup \{0 \leq pr(E) \leq 1 \mid pr(E) \in Vars(F_{C_0})\} \cup \{pr(C_0) = r\}$$

is satisfiable in $(\mathbb{R}, +, -, \cdot, /, \{q : q \in \{\mathbb{Q}\})$, where $Vars(F_{C_0})$ denotes the set of all the propositional variables used in formulas of F_{C_0} .

When we ask instead to solve the problem over quasi-witnessed models, we consider then the theory $QWT(C_0)$. In that case, we change the formula produced in the first row of Table 1 for:

$$(pr(\forall R.C(d_\sigma)) = 0) \vee\ (ite(pr(R(d_\sigma, d_{\sigma,n})) \leq pr(C(d_{\sigma,n})),\ pr(\forall R.C(d_\sigma)) = 1,$$
$$pr(\forall R.C(d_\sigma)) \cdot pr(R(d_\sigma, d_{\sigma,n})) = pr(C(d_{\sigma,n}))))$$

And we have also to consider the additional set of propositions in Y_{C_0} of Definition 10 in [2], that are of the form:

$$\neg \forall R.C(d_\sigma) \boxdot (R(d_\sigma, d_{\sigma,n}) \rightarrow C(d_{\sigma,n}))$$

that must not be equal to 1 in any solution of the satisfiability problem in order to encode valid quasi-witnessed models. What we want to enforce with the propositions in Y_{C_0} is that when $pr(\forall R.C(d_\sigma)) = 0$, in order to finitely encode a model with infinite individuals $d^1_{\sigma,n}, d^2_{\sigma,n}, \ldots$ such that:

$$lim_{i \rightarrow \infty} \frac{pr(C(d_{\sigma,n}))^i}{pr(R(d_\sigma, d_{\sigma,n}))^i} = 0$$

we need that $pr(R(d_\sigma, d_{\sigma,n})) > pr(C(d_{\sigma,n}))$ So, for each such proposition we introduce this additional formula in F_{C_0}:

$$it(pr(\forall R.C(d_\sigma)) = 0,\ pr(R(d_\sigma, d_{\sigma,n})) > pr(C(d_{\sigma,n}))$$

which translates the fact that propositions in Y_{C_0} should not be satisfied in terms of satisfiability of non-linear arithmetic boolean formulas.

3 Preliminary Evaluation

Consider the following family of $1-$satisfiable concepts, indeed satisfiable with witnessed models, in our logic Π-$\mathcal{ALE}$, that use the relation symbol $friend$ and the atomic concept symbol $popular$, determined by the following regular expression:

$$\left(\forall friend\cdot\right)^{n+1} popular \sqcap \left(\exists friend\cdot\right)^{n} \neg popular \tag{1}$$

where the expresion $(E)^n$ means n nested concatenations of the expression E, and n is an integer parameter with $n \geq 1$. So that with $n = 1$ we have the concept:

$$\forall friend\cdot\forall friend\cdot popular \sqcap \exists friend\cdot\neg popular$$

and with $n = 2$ the concept:

$$\forall friend\cdot\forall friend\cdot\forall friend\cdot popular \sqcap \exists friend\cdot\exists friend\cdot\neg popular$$

Consider also the following family of $1-$satisfiable concepts, but only with quasi-witnessed models, determined by the regular expression:

$$\left(\forall friend\cdot\right)^{n} popular \sqcap \neg\left(\forall friend\cdot\right)^{n} (popular \boxdot popular) \tag{2}$$

where n is as before an integer parameter with $n \geq 1$.

Table 2 shows the computation times [3], obtained when using the SMT solver Z3 (version 4.3.2) with a memory limit of 7GB per execution, when solving the instances from our benchmarks in the range $n \in [3, 10]$. We have solved the instances with both encodings, the one for only witnessed models and the one for quasi-witnessed models. The table also shows the size of the resulting formulas F_{C_0} obtained from each encoding. We observe that on the first benchmark, with both encodings we solve the instances within the time limit of 20 minutes up to $n = 7$, but with the quasi-witnessed encoding is always harder to solve it. For the second benchmark, the situation is even more different between both encodings. The witnessed encoding correctly solves the instances (find that they are not satisfiable with witnessed models) up to $n = 8$. By contrast, the quasi-witnessed encoding solves the instances only up to $n = 6$ and always with more time.

4 Conclusions and Future Work

Our results show that the performance of our SMT-based approach, that works by solving a non-linear real arithmetic boolean formula is really problematic. So, we are now developing a version of our tool that will consider a translation of the problem to a satisfiability problem over a linear real arithmetic problem. This new tool is based on the results shown in [4] and follows a similar approach to the one proposed in [10] to develop a satisfiability solver for different many-valued propositional logics, that uses SMT solvers as well.

[3] A solving time equal to > 1200 means that the execution was aborted after 20 minutes without being able to solve the instance.

	Benchmark (1)				Benchmark (2)				
	WT_{C_0}		QWT_{C_0}			WT_{C_0}		QWT_{C_0}	
n	size	solving time	size	solving time	n	size	solving time	size	solving time
3	20	0.033	24	0.029	3	16	0.023	20	0.036
4	40	0.041	44	0.063	4	32	0.055	36	0.105
5	80	0.118	92	0.216	5	64	0.143	76	0.450
6	164	0.379	184	0.806	6	132	0.465	152	2.101
7	332	1.327	372	3.094	7	264	1.639	304	> 1200
8	672	5.080	756	> 1200	8	528	11.301	616	> 1200
9	1400	> 1200	1500	> 1200	9	1100	> 1200	1300	> 1200
10	2700	> 1200	3100	> 1200	10	2200	> 1200	2500	> 1200

Table 2. Formula size (in Kbytes) and solving times (in seconds) for F_{C_0} obtained with our two benchmarks of concepts with Z3 SMT solver. The generation time of the formula F_{C_0} was less than 0.08 seconds up to $n = 8$ and less than 0.2 seconds for the other sizes.

References

1. Franz Baader. Tableau algorithms for description logics. In *Proceedings of TABLEAUX'2000*, pages 1–18, 2000.
2. Marco Cerami, Francesc Esteva, and Félix Bou. Decidability of a description logic over infinite-valued product logic. In *Proceedings of KR'2010*, 2010.
3. Marco Cerami and Umberto Straccia. On the (un)decidability of fuzzy description logics under łukasiewicz t-norm. *Information Sciences*, 227:1–21, 2013.
4. Roberto Cignoli and Antoni Torrens. An algebraic analysis of product logic. *Multiple-Valued Logic*, 5:45–65, 2000.
5. Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *Proceedings of TACAS'2008*, pages 337–340, 2008.
6. Volker Haarslev, Roberto Sebastiani, and Michele Vescovi. Automated reasoning in *alcq* via smt. In *Proceedings of CADE'2011*, pages 283–298, 2011.
7. Petr Hájek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154(1):1–15, 2005.
8. Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Reducing SHIQ-description logic to disjunctive datalog programs. In *Proceedings of KR'2004*, pages 152–162, 2004.
9. Umberto Straccia and Fernando Bobillo. Mixed integer programming, general concept inclusions and fuzzy description logics. In *Proceedings of EUSFLAT'2007*, pages 213–220, 2007.
10. Amanda Vidal, Félix Bou, and Lluis Godo. An SMT-based solver for continuous t-norm based logics. In *Proceedings of SUM'2012*, pages 633–640, 2012.