A General Artificial Intelligence Approach for Skeptical Reasoning

Éric Grégoire, Jean-Marie Lagniez, and Bertrand Mazure

CRIL
Université d'Artois - CNRS UMR 8188
rue Jean Souvraz SP18 F-62307 Lens France
{gregoire,lagniez,mazure}@cril.fr

Abstract. We propose a general artificial intelligence approach for handling contradictory knowledge. Depending on the available computational resources, reasoning ranges from credulous to forms of skepticism with respect to the incompatible branches of alternatives that the contradictions entail. The approach is anytime and can be declined according to various knowledge representation settings. As an illustration of practical feasibility, it is experimented within a Boolean framework, using the currently most efficient computational tools and paradigms.

Keywords: General Artificial Intelligence, Inconsistency Handling, Credulous Reasoning, Skeptical Reasoning.

1 Introduction

The ability to reason and act in an appropriate coherent way despite contradictory knowledge is a fundamental paradigm of intelligence. A general artificial intelligence reasoner needs the ability to detect and overcome conflicting knowledge. Especially, its deductive capacities should not collapse due to conflicting premises (a standard-logic reasoner can infer any conclusion and its contrary from contradictory information). Similarly, when some contradictory knowledge forbids the existence of global solutions to a problem, the reasoner should be able to locate this problematic information and adopt a specific policy: at the extreme, it might for example decide to use its problem-solving capacities on the non-contradictory part of the knowledge, only.

In this context, the focus in this paper is on adopting a cautious or so-called *skeptical* stance towards contradictory knowledge. This requires the reasoner to locate the conflicting information and adopt what is shared by all possible branches of alternatives that are underlied by the contradictions. Such a paradigm has long been studied in A.I. (see the seminal work on default logic [1] and e.g. [2] [3]), but suffers from a lack of practical implemented tools, mainly due to discouraging theoretical results: even in the simple Boolean logic setting, skeptical reasoning belongs to the second-level of the polynomial hierarchy [4,5], making it intractable in the worst-case, unless P = NP.

B. Goertzel et al. (Eds.): AGI 2014, LNAI 8598, pp. 33–42, 2014.

We claim that the usual definition of skeptical reasoning should be revisited and refined in order to both better mimic human abilities and improve computationally viability. Especially, we argue for anytime reasoning abilities that are intended to translate a progressive range of attitudes in line with the elapsed computing time. Like human beings, general artificial intelligence systems also need to recognize that some reasoning tasks would require more time and resources to be successively conducted, when intractability threatens. In this context, we revisit skeptical reasoning to include various parameters that, on the one hand, translate reasonable assumptions about the targeted reasoning paradigms and, on the other hand, often entail more tractable treatments. For example, a skeptical reasoner might only be able to detect contradictions that are formed of less than a given preset maximal number of informational entities. As it is often assumed in model-based diagnosis [6,7], it might also recognize that its reasoning is satisfactory when a preset maximal number of different extracted contradictions is reached that would allow satisfiability to be recovered if these contradictions were removed. More generally, various parameters classifying contradictions can be included in the models of skepticism in such a way that various forms of bounded skepticism can be defined depending on the considered values for the parameters; when all possible values are considered, these forms converge to ideal skepticism, which excludes all the possibly controversial knowledge.

Interestingly, there has been much progress these last years on the practical computation of the basic building blocks required by this anytime model, making it even more realistic in many situations.

The paper is organized as follows. In the next section, the concepts of MSS (Maximal Satisfiable Subset), CoMSS (Complement of a Maximal Consistent Subset) and MUS (Minimal Unsatisfiable Subset), which are cornerstones in this study, are presented. The any-time reasoning approach is described in section 3, with a focus on various paradigms leading to useful weakened forms of skepticism. Section 4 details our experimental study showing the feasibility of the any-time approach, before promising paths for further research are presented in the last section.

2 Basic Concepts and Tools

As an illustration of practical feasibility, we have implemented and experimented this anytime reasoning architecture within a Boolean reasoning framework, using its currently most efficient computational tools and paradigms. As far as any additional representational mechanism does not yield intractability, other knowledge representation formalisms and problem-solving frameworks can be covered by the approach. For example, the knowledge representational setting can be at least as expressive as constraint networks for Constraint Satisfaction Problems in discrete domains [8,9], including other finite representation systems. All that is needed is an encoding of contradictory knowledge as unsatisfiable informational entities, together with a compactness result about this (finite) satisfiability concept: any proper subset of a satisfiable subset is itself satisfiable.

In the following we assume that knowledge takes the form of a CNF Σ, namely a finite set of clauses that is conjunctively interpreted, where a clause is a finite disjunction of possibly negated Boolean variables and constants. SAT is the NP-complete problem that consists in checking whether or not Σ is satisfiable, and in the positive case in delivering one assignment of values to all variables that makes all clauses *true*. The following MSS, CoMSS and MUS concepts are cornerstones in the study of credulous and skeptical reasonings.

Definition 1 (MSS). *$\Gamma \subseteq \Sigma$ is a Maximal Satisfiable Subset (*MSS*) of Σ iff Γ is satisfiable and $\forall \alpha \in \Sigma \setminus \Gamma$, $\Gamma \cup \{\alpha\}$ is unsatisfiable.*

Definition 2 (CoMSS). *$\Gamma \subseteq \Sigma$ is a Complement of a* MSS*, also called Minimal Correction Subset (*MCS *or* CoMSS*), of Σ iff $\Sigma \setminus \Gamma$ is satisfiable and $\forall \alpha \in \Gamma$, $\Sigma \setminus (\Gamma \setminus \{\alpha\})$ is unsatisfiable.*

Accordingly, Σ can always be partitioned into a pair made of one MSS and one CoMSS. Obviously, such a partition needs not be unique. A *core* of Σ is a subset of Σ that is unsatisfiable. Minimal cores, with respect to set-theoretical inclusion, are called MUSes.

Definition 3 (MUS). *$\Gamma \subseteq \Sigma$ is a Minimal Unsatisfiable Subset (*MUS*) of Σ iff Γ is unsatisfiable and $\forall \alpha \in \Gamma$, $\Gamma \setminus \{\alpha\}$ is satisfiable.*

Under its basic form where no specific information entity is compulsory present in any solution, MAX-SAT consists in delivering one maximal (with respect to cardinality) subset of Σ. Note that MSSes are maximal with respect to $\subseteq$: accordingly, any solution to MAX-SAT is one MSS, but not conversely. Note also that there is a hitting set duality between CoMSSes and MUSes. All the above concepts have been thoroughly studied in A.I., especially with respect to worst-case complexity properties: MAX-SAT is NP-complete; splitting Σ into one (MSS,CoMSS) pair is in P^{NP} [10], checking whether or not a CNF is a MUS is *DP*-complete [11] and checking whether or not a clause belongs to a MUS is in Σ_2^P [4]. Moreover, there can be $\mathcal{O}(C_m^{m/2})$ MUSes in Σ when Σ is made of m clauses. Despite these bad worst-time complexity results, much progress has been made this last decade, allowing the efficient extraction of MSSes, CoMSSes and MUSes from Σ, at least in many situations (see e.g. [12,13,14,15,16]).

3 An Anytime Progression of Reasoning

Let us turn back to the possible agent's stances towards handling some possibly conflicting knowledge. A first step from the agent is its recognition of the existence of a contradictory situation, namely unsatisfiability. This requires a successful call to a SAT-solver.

3.1 Fool Attitude

When the SAT-solver does not deliver any result within the available computing time, a first attitude towards inconsistency can however be obtained although it is of a limited interest. Usual CDCL SAT solvers [17] record the current assignment that satisfies the largest number of clauses. When the solver is stopped because of time-out limits, clauses from Σ can be classified according to this so-called *progressive interpretation*. Under tougher time-limits, the agent can also classify clauses of Σ according to any random assignment. Clauses that are satisfied by this assignment form a non-conflicting subset of information from Σ. The agent can then reason in a deductive way from this subset: every conclusion is only guaranteed to belong to one possible branch of reasoning amongst the incompatible ones that are underlied by the incompatible knowledge in Σ. All the next following forms of reasoning assume that Σ has been shown unsatisfiable.

3.2 Credulous Reasoning

In the A.I. literature, the concept of credulous (or brave) reasoner can be traced back at least to default logic [1]. In the standard Boolean framework, a credulous agent adopts one MSS of Σ and reasons on its basis. When the agent is running short of time to compute one MSS, it can be downgraded to the fool attitude and adopt the progressive interpretation or the largest satisfiable approximate MSS computed so-far. Note that a specific MSS is delivered by MAX-SAT: it is an MSSes that is one of the largest in terms of cardinality. From a practical point of view, computing one MSS is easier and proves more efficient in many cases.

3.3 Ideal Skepticism

Contrary to credulous reasoners, an ideally skeptical attitude rejects any piece of information that does not follow from every MSS. Equivalently, it thus rejects all MUSes of Σ. One MUS represents one smallest possible set of unsatisfiable clauses of Σ. In the general case, the number of MUSes can be exponential in the number of clauses of Σ. Since MUSes can intersect, simply removing a number of MUSes such that the remaining part of Σ becomes satisfiable is generally not sufficient to yield an ideally skeptical reasoner. However, in some situations, computing all MUSes remains tractable from a practical point of view (see for example [18]).

3.4 Practical Skepticism

Instead of computing ideal skepticism which is often out of practical reach, we propose to compute in an any-time manner forms of weakened skepticism, which converge to ideal skepticism when enough computing resources are provided. They can be defined based on assumptions about the nature and topology of unsatisfiability in Σ and the cognitive abilities of the reasoner. Let us examine some of the most direct and natural assumptions: as we shall see, they can often be mixed together.

Assumption 1. Each MUS contains less than k clauses.
The smaller the parameter k, the easier SAT-checking and the computation of MUSes (at least, expectedly). This assumption also translates a natural variable feature of human intelligence: the size of a MUS is the smallest number of clauses that are necessary in the proof of the contradiction embodied inside

the MUS. Clearly, human beings find it easier to detect e.g. two pieces of information that are directly conflicting rather than a proof of unsatisfiability that involves many clauses. Accordingly, an anytime skeptical reasoner might extract MUSes, both successively and according to their increasing sizes. When computational resources are running short, the skeptical reasoner might drop from Σ all MUSes computed so far when this leads to satisfiability: the reasoner is then aware that no MUS that is "easier" (in the sense that its size is lower than k), does exist in Σ. This is at least as smart as a human beings that are not capable of finding out a proof of unsatisfiability that involves at least k different pieces of information. This assumption is orthogonal to the next ones that we are going to present, and can be mixed with them.

Assumption 2. The number of MUSes in Σ is lower than l.
In real-life situations, it might not often be expected that Σ is polluted by many different reasons for unsatisfiability, hence the maximal value for l might be far more lower than the one that is exponential in the number of clauses of Σ in the worst-case complexity scenario. Again, an anytime skeptical reasoner can consider increasing values for the parameter (here, l), successively. It needs to recognize that it has dropped a sufficient number of MUSes to regain satisfiability, and, at the same time, that it has limited its investigations to a limited number l of MUSes, only. We believe that this policy is best mixed together with Assumption 1, so that MUSes of lower cardinality are considered first. This corresponds to a realistic skeptical human reasoner that finds out smallest proofs of unsatisfiability first: after having exhausted all its resources, it is aware that the number of MUSes in Σ was too large in order to compute all of them. At the same time, it might estimate that Σ without all the extracted MUSes forms an acceptable basis for further reasoning or problem-solving, despite this restriction.

Assumption 3. Two MUSes are pairwise independent iff they do not intersect. The maximal cardinality of a set made of MUSes such that any pair of them is pairwise independent, is lower than m.
As m translates the maximal number of totally different causes of unsatisfiability in Σ, it is expected that this number can be small in real-life, especially for many situations where a new piece of information is to be inserted within Σ, switching the status of Σ from satisfiable to unsatisfiable. Again, if m is large, human and artificial intelligence reasoners may lack sufficient resources to compute all of them. Contrary to Assumptions 1 & 2, we do not expect the reasoner to consider increasing values for m. Instead, it might attempt to find a lower bound for m by finding out a sufficient number of non-intersecting MUSes such that removing all these MUSes makes Σ become satisfiable. To this end, whenever a MUS is extracted, it is sufficient to remove it from Σ, and iterate until satisfiability is reached. A skeptical reasoner running out of resources might make do with such a *cover* of MUSes, being aware of the valuable property that any MUS not extracted so far shares at least one clause with this cover (otherwise, removing the computed MUSes would not have allowed satisfiability to be regained).

Assumption 4. The set-theoretic union of all MUSes of Σ forms at most n non-intersecting subparts of Σ.
Even when two MUSes are pairwise independent, each of them can intersect with a same third MUS. When $n > 1$, this assumption ensures that under a given (a priori) value for n and when the currently extracting MUSes form n non-intersecting subparts of Σ, the reasoner is sure that all the other MUSes can be found successively and are forming sequences of intersecting MUSes with the already discovered ones. Accordingly, this gives a hint about which candidate MUS are to be considered next. When this value for n is a mere belief of the reasoner about Σ, this latter one might start by considering candidate MUSes that obey this assumption before considering all other candidate MUSes.

Obviously, parameters in Assumptions 2, 3 and 4 are connected through the $n \leq m \leq l$ relation.

Assumption 5. Each MUS intersects with at most p other MUSes of Σ or **Each clause in a MUS belongs to at most q MUSes of Σ.**
This assumption relies on the idea that any clause does not belong to a large number of different minimal proofs of unsatisfiability. Intuitively, this assumption requires that every piece of information is not a cause of many, i.e. more than q, contradictions. When its actual value is unknown, the q parameter can be set to increasing values, and MUSes that obey that assumption can be searched first. However, this assumption might prove too strong since for example the insertion of a new piece of information might actually contradict with many existing clauses in Σ. Hence, we believe that this assumption should be refined as follows.

Assumption 6. When a MUS contains v clauses, at most v' (with $v' \leq v$) clauses in this MUS belong also to other MUSes of Σ. Actually, each of these clauses belongs to at most s MUSes.
Again, the anytime reasoner might consider increasing values for v' and s, successively, and search first for MUSes obeying the assumptions. The intuitive idea about the topology of unsatisfiability is that in any MUS, only v' clauses are "actual" reasons for unsatisfiability. These clauses might for example coincide with the information that has been added inside Σ when Σ was satisfiable, leading to at most s different minimal sets of conflicting information.

Assumption 7. There is at most t clauses that are "actual" reasons for unsatisfiability. Each clause in a MUS is a reason for unsatisfiability in the sense that removing the clause would lead to "break" the MUS. The concept of "actual" reason specifies which clauses are the reasons in the real world leading to unsatisfiability. For example, assume that Σ is satisfiable and correct. A less reliable new piece of information under the form of a clause i is inserted within Σ that leads to Σ to become unsatisfiable. Accordingly, i is the real reason for unsatisfiability, although several MUSes can exist and removing one clause per MUS would allow Σ to regain satisfiability, even when these latter selected clauses are not i. In many actual situations, it might make sense to reason under a given a priori value for t. Moreover, under this assumption, when t non-intersecting MUSes are extracted, dropping them already excludes from Σ all actual reasons for unsatisfiability (but not necessarily all MUSes, which must include at least one of these reasons).

4 Experimental Study

In order to study the feasibility of the proposed any-time reasoner, we have implemented the above multi-level reasoning architecture and experimented it with a large set of challenging benchmarks involving various problem-solving and reasoning issues. More precisely, we have considered the 393 unsatisfiable challenging CNF benchmarks (`http://logos.ucd.ie/Drops/ijcai13-bench.tgz`) used and referred to in [13] for MSSes and CoMSSes computation and enumeration. They encode various problems related to e.g. distributed fault diagnosis for multiprocessor systems, planning issues, etc. Although most basic building blocks of reasoning in the architecture involve solving instances of NP-hard problems and are thus intractable in the worst case, much practical progress has been made these last decades making the tasks become feasible for many instances, as it was already envisioned a long time ago for example in [19]. We thus used some of the currently best performing tools in the Boolean search and satisfiability domains: MINISAT [17] was selected as the CDCL SAT-solver. We used our own tool called CMP as partitioner [16]: it is the currently best performing tool for the extraction of one MSS and its related CoMSS in the clausal Boolean setting. We also used a MUS extractor called MUSER2 (`http://logos.ucd.ie/web/doku.php?id=muser`) [12] and used ELS [13] as a tool for extracting the set of clauses made of all MUSes (or equivalently, all CoMSSes). All experimentations have been conducted on Intel Xeon E5-2643 (3.30GHz) processors with 8GB RAM on Linux CentOS. Time limit was set to 30 minutes for each single test.

First, we have computed the fool attitude (according to the progressive interpretation delivered by the SAT-solver when unsatisfiability is proved) and the credulous one embodied by one MSS. Then we considered forms of skepticism. In this last respect, we have concentrated on the expectedly computationally-heaviest schemas. First, we have implemented the ideally skeptical attitude as an ideal model of competence; not surprisingly, it remained out of computational reach for many instances. Then we have implemented a simple algorithm, called INDEPENDENT-MUSES to take Assumption 3 into account: it computes a sufficient number m of independent MUSes so that dropping them leads to satisfiability. To this end, we have exploited a direct relationship between CoMSS and MUSes: any CoMSS contains at least one clause of each MUS. We took advantage of the efficiency of the novel CMP tool to partition Σ into one pair (MSS, CoMSS). INDEPENDENT-MUSES computes one such a partition and then iterates the following schema until satisfiability is reached: compute one MUS containing one clause from the CoMSS, retract this MUS from Σ, ends if the remaining part of Σ is satisfiable. Note that under Assumption 7, when the number of detected MUSes reaches t, then we are sure that all actual reasons for unsatisfiability are located inside these MUSes.

Since relying on Assumption 3 and INDEPENDENT-MUSES leads the reasoner to miss MUSes intersecting with the other detected ones, we have modified this algorithm so that when a clause in a CoMSS does not lead to the extraction of a MUS that is not intersecting with the already computed ones, one MUS is computed that includes this clause and possibly intersects with already computed MUSes. This algorithm is called More-MUSes: it can be the starting schema for implementing Assumptions 5 and 6 (actually, it implements a part of Assumption 6 where $v' = 1$ and $s = 1$). Taking additional additional assumptions would expectedly lead to even better results since they prune the search space by restricting the number of MUSes (Assumption 2) and/or their possible contents (Assumption 1), or can lead to additional divide and conquer elements in the search strategy (Assumption 4).

The full experimentations data and results are available from `www.cril.univ-artois.fr/documents/fulltab-agi14.pdf` . In Table 1, we provide a sample of results. The columns give the name of the instance Σ, the number of Boolean variables ($\#v$) and clauses ($\#c$), the time rounded up to the second to compute the fool attitude (i.e., compute the progressive interpretation) (t_{fa}); in the worst case, this time is the time-out (1800 seconds) set for the satisfiability test. The next column gives the time also rounded up to the second to compute the credulous

Table 1. Some experimental results

Instance					Independent-MUSes		More-MUSes		All-MUSes	
Name	$\#v$	$\#c$	t_{fa}	t_{ca}	$\#c^-$	t_g	$\#c^-$	t_g	$\#c^-$	t_g
bf0432-005	1040	3667	0	0	782	0.00	871	0.32	1668	1067.07
bf0432-011	1053	3740	0	0	288	0.00	1070	0.19	time-out	
t5pm3-7777.spn	125	750	0	0	4	0.00	420	1.28	time-out	
C210_FW_SZ_91	1789	6709	0	0	282	0.18	283	0.94	time-out	
C202_FW_SZ_123	1799	7437	0	0	37	0.00	37	0.06	40	0.00
C208_FA_UT_3254	1876	6153	0	0	40	0.00	68	0.00	98	0.08
C202_FW_SZ_100	1799	7484	0	0	36	0.00	25	0.04	time-out	
ssa2670-136	1343	3245	0	0	1246	0.08	1301	0.44	1396	20.40
ssa2670-134	1347	3273	0	0	1242	0.08	1312	0.51	1404	200.73
22s.smv.sat.chaff.4.1.bryant	14422	40971	0	0	725	2.14	736	2.19	time-out	
cache.inv14.ucl.sat.chaff.4.1.bryant	69068	204644	4	1	17314	394.28	17331	548.96	time-out	
ooo.tag10.ucl.sat.chaff.4.1.bryant	15291	43408	0	0	1882	4.00	1891	4.10	time-out	
ooo.tag14.ucl.sat.chaff.4.1.bryant	62505	179970	1	1	4971	68.39	4968	68.31	time-out	
dlx2_ca	1225	9579	0	0	2230	1.51	2230	1.21	2970	810.99
dlx2_cs	1313	10491	1	0	2481	1.69	2480	2.28	3358	934.11
dlx1_c	295	1592	0	0	560	0.00	560	0.00	656	0.59
divider-problem.dimacs_5.filtered	228874	750705	0	2	6854	1477.39	31630	1720.09	time-out	
divider-problem.dimacs_7.filtered	239640	787745	1800	1800	6134	1642.59	time-out		time-out	
rsdecoder_fsm1.dimacs.filtered	238290	936006	1800	1800	time-out		time-out		time-out	
dspam_dump_vc1093	106720	337439	7	2	391	388.07	524	217.92	time-out	
dspam_dump_vc950	112856	360419	66	2	393	485.62	369	1394.81	memory overflow	
dividers4.dimacs.filtered	45552	162947	0	0	18	38.09	152	38.46	time-out	
dividers10.dimacs.filtered	45552	162646	0	1	937	38.19	4595	61.68	time-out	
dividers5.dimacs.filtered	45552	162946	0	0	652	37.99	650	38.35	time-out	
mem_ctrl1.dimacs.filtered	1128648	4410313	1800	1800	time-out		time-out		time-out	
wb-debug.dimacs	399591	621192	1800	1800	28	929.02	time-out		time-out	
mem_ctrl-debug.dimacs	381721	505523	0	0	71	569.54	118	570.00	time-out	
3pipe_1_ooo	2223	26561	1	0	5221	14.75	5253	13.04	time-out	
5pipe_5_ooo	10113	240892	10	0	28614	651.80	28933	634.39	time-out	
4pipe_3_ooo	5233	89473	7	0	15963	162.75	16226	177.82	time-out	
4pipe	5237	80213	45	0	17217	284.30	17086	302.10	time-out	
4pipe_2_ooo	4941	82207	8	0	14926	166.99	14422	187.88	time-out	
4pipe_4_ooo	5525	96480	4	0	17559	460.53	17598	211.92	time-out	
4pipe_1_ooo	4647	74554	27	0	12748	139.48	12622	318.02	time-out	
C168_FW_UT_2463	1909	6756	0	0	44	0.00	45	0.00	time-out	
C202_FS_SZ_121	1750	5387	0	0	23	0.00	23	0.00	26	0.00
C170_FR_SZ_92	1659	4195	0	0	131	0.00	131	0.00	131	0.00
C208_FC_RZ_70	1654	4468	0	0	212	0.00	212	0.00	212	0.08
term1_gr_rcs_w3.shuffled	606	2518	0	0	22	0.00	719	2.56	time-out	
alu2_gr_rcs_w7.shuffled	3570	73478	1800	1800	1254	681.31	time-out		time-out	
too_large_gr_rcs_w5.shuffled	2595	36129	1800	1800	109	1.36	time-out		time-out	
too_large_gr_rcs_w6.shuffled	3114	43251	0	0	751	7.61	4529	257.81	time-out	
9symml_gr_rcs_w5.shuffled	1295	24309	0	0	151	0.57	4914	386.24	time-out	
ca256.shuffled	4584	13236	0	0	9882	10.20	9882	10.28	9882	81.55
ca032.shuffled	558	1606	0	0	1173	0.08	1173	0.06	1173	0.99

attitude (i.e., compute one (MSS,CoMSS) partition) (t_{ca}). The next columns give the main parameters resulting from running Independent-MUSes: the total number of clauses in the discovered MUSes ($\#c^{-}$) and the global time taken by the algorithm (t_g). The same parameters are provided for More-MUSes. The columns for All-MUSEs provide the number of clauses in the set-theoretical union of all MUSes ($\#c^{-}$) and the time to complete the algorithm (t_g).

Not surprisingly, All-MUSes and thus ideal skepticism proved unsuccessful for most (i.e., 300/393) instances. The required time to compute the fool attitude with respect to the progressive interpretation was often done in less than one second. The average time for computing one MSS and thus a credulous attitude was 0.16 seconds for the 319/393 successively addressed instances. The number of additional clauses in MUSes dropped by More-MUSes clearly shows that Independent-MUSes often misses MUSes. Although the latter Algorithm is often more resources consuming, it remained successful for most (i.e., 338/393) instances. For example, ideal skepticism could not be completed within the 30 minutes maximal computing time for the `too_large_gr_rcs_w6.shuffled` benchmark whereas Independent-MUSes and More-MUSes extracted 751 and 4529 clauses belonging to the set-theoretic union of all MUSes of this instance, respectively. As a general lesson, this basic experimental study thus shows the viability of the any-time architecture and of forms of practical, weakened, skepticism, at least for the tested benchmarks, which are reputed challenging.

5 Conclusions and perspectives

In this paper, we have proposed and experimented with an any-time architecture to handle contradictory knowledge. Like human beings who can handle conflicting knowledge in various rational ways depending on the actual available time to complete the reasoning tasks, the architecture implements an any-time range of reasoning capabilities that depend on the elapsed computing time and resources. Especially, we have revisited skeptical reasoning to include human-like progressive ways that attempt to reach ideal skepticism. At this point, it is important to stress that the architecture and its main paradigms are not sticked to a logical representation of knowledge and reasoning. It can apply to any set of informational entities, provided that a finite satisfiability relationship can be defined and that a compactness result on this relation does exist, allowing for maximal satisfiable subsets and minimal unsatisfiable subsets concepts to be defined and computed. We believe that the Assumptions described in this paper, although instantiated to the clausal Boolean framework, could be easily transposed into such other formalisms. The actual efficiency of the architecture on these other representational mechanism will depend on the practical computational cost to handle these concepts. For example, the architecture applies to the general languages of constraint solving problems in discrete domains, where techniques for computing MSSes and MUSes have been studied for a long time [8,9]. Accordingly, we claim that the ways of handling contradictory knowledge that we have proposed can take part into a general artificial intelligence system that would be able to overcome contradictory knowledge.

References

1. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13(1-2), 81–132 (1980)
2. Grégoire, É.: Skeptical inheritance can be more expressive. In: Proceeding of the 9th European Conference on Artificial Intelligence (ECAI 1990), pp. 326–332 (1990)
3. Makinson, D., Schlechta, K.: Floating conclusions and zombie paths: Two deep difficulties in the "directly skeptical" approach to defeasible inheritance nets. Artificial Intelligence 48(2), 199–209 (1991)
4. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence 57(2-3), 227–270 (1992)
5. Cayrol, C., Lagasquie-Schiex, M.C., Schiex, T.: Nonmonotonic reasoning: From complexity to algorithms. Annals of Mathematics and Artificial Intelligence 22(3-4), 207–236 (1998)
6. Hamscher, W., Console, L., de Kleer, J. (eds.): Readings in Model-Based Diagnosis. Morgan Kaufmann (1992)
7. Feldman, A., Kalech, M., Provan, G. (eds.): Proceedings of the 24th International Workshop on Principles of Diagnosis, DX 2013 (2013), `http://www.dx-2013.org/proceedings.php` (electronic proceedings)
8. Rossi, F., van Beek, P., Walsh, T.: Handbook of Constraint Programming. Elsevier, Amsterdam (2006)
9. Lecoutre, C.: Constraint Networks: Techniques and Algorithms. Wiley (2009)
10. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1993)
11. Papadimitriou, C.H., Wolfe, D.: The complexity of facets resolved. Journal of Computer and System Sciences 37(1), 2–13 (1988)
12. Belov, A., Marques-Silva, J.: Accelerating MUS extraction with recursive model rotation. In: Proceedings of the Eleventh International Conference on Formal Models in Computed-Aided Design (FMCAD 2011), pp. 37–40 (2011)
13. Marques-Silva, J., Heras, F., Janota, M., Previti, A., Belov, A.: On computing minimal correction subsets. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013 (2013)
14. Liffiton, M.H., Malik, A.: Enumerating infeasibility: Finding multiple MUSes quickly. In: Gomes, C., Sellmann, M. (eds.) CPAIOR 2013. LNCS, vol. 7874, pp. 160–175. Springer, Heidelberg (2013)
15. Lagniez, J.-M., Biere, A.: Factoring out assumptions to speed up mus extraction. In: Järvisalo, M., Van Gelder, A. (eds.) SAT 2013. LNCS, vol. 7962, pp. 276–292. Springer, Heidelberg (2013)
16. Grégoire, É., Lagniez, J.M., Mazure, B.: A computational method for (MSS,CoMSS) partitioning (2014) (submitted to AAAI 2014)
17. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
18. Grégoire, É., Mazure, B., Piette, C.: Boosting a complete technique to find MSS and MUS thanks to a local search oracle. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 2300–2305 (2007)
19. Kautz, H.A., Selman, B.: Pushing the envelope: Planning, propositional logic and stochastic search. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference (AAAI 1996), vol. 2, pp. 1194–1201 (1996)