Using a Description Logic with Concept Inclusions

Ian Horrocks and **Alan Rector**
Medical Informatics Group
Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, UK
(horrocks—rector@cs.man.ac.uk)

Abstract

The GALEN project is constructing a conceptual schema for medical terminology using a Description Logic (DL) which supports transitive roles and general concept inclusions. Tableaux calculus subsumption testing algorithms for such a DL are well understood but serious worst case intractability would appear to limit their practical applicability. However by using heuristic speedup techniques adapted from constraint satisfaction problem solving procedures it is possible to dramatically improve performance with a realistic knowledge base.

1 Introduction

The European GALEN project aims to promote the sharing and re-use of medical data by using a concept model as a flexible and extensible classification schema [Rector *et al.*, 1993]. The concept model is being built using a Description Logic (DL) which, due to the identification of critical design requirements [Nowlan, 1993], supports transitive roles and general concept inclusions (GCIs).

The theory of extended tableaux calculus subsumption algorithms which are able to deal with transitive roles and GCIs is well understood [Baader, 1990; Buchheit *et al.*, 1993; Sattler, 1996; Schild, 1991] but severe worst case intractability appears to limit their practical applicability. However it seems worthwhile to investigate how well such an algorithm performs with a real knowledge base when a range of optimisations are employed. The adaption of heuristic speed up techniques from constraint satisfaction problems (CSPs), and in particular propositional satisfiability (SAT) testing procedures [Freeman, 1996], seems to be a promising method for dealing the most serious cause of intractability, the large numbers of disjunctive constraints introduced by GCIs.

2 Subsumption testing

Most DLs restrict terminological axioms to acyclic concept introductions[1] of the form $PN \sqsubseteq C$ or $CN \doteq C$, where PN (CN) is a new primitive (non-primitive) concept name and C is a concept term [Baader *et al.*, 1991]. Concept terms can be expanded until they contain only primitives and subsumption between fully expanded terms can then be evaluated independently of the remainder of the terminology. The subsumption problem can be transformed into a satisfiability problem—C_1 subsumes C_2 if $C_2 \sqcap \neg C_1$ is *not* satisfiable—and solved using an algorithm based on the tableaux calculus.

GCIs are a more general axiom of the form $A \sqsubseteq B$ where both A and B are arbitrary concept terms[2]. The satisfiability of a concept term is no longer independent of the terminology when it includes GCIs—any model must also satisfy all of the GCIs. GCIs act as constraints $x : (B \sqcup \neg A)$ which apply to every variable x in a constraint system. These constraints can be added either directly, using a new type of constraint and expansion rule [Buchheit *et al.*, 1993], or indirectly, by extending the logic to include role union and transitive reflexive closure [Baader, 1990; Schild, 1991]. The algorithm being studied uses the former mechanism as it has a simple control structure which is amenable to the proposed optimisation techniques.

3 Performance

In practice most of the GCI constraints can be eliminated using a filtering technique which makes use of negated primitives in disjunctions [Horrocks *et al.*, 1996], leaving satisfiability problems which are easily solved. However the performance of classification as a whole is still dominated by a relatively small number of 'hard cases' of which there are two distinct sorts: critically constrained problems and under-constrained prob-

[1] The concept term in an introduction may not refer, either directly or indirectly, to the concept name being introduced.

[2] A concept equation $A \doteq B$ is equivalent to a pair of GCIs, $A \sqsubseteq B$ and $B \sqsubseteq A$.

lems. Critically constrained problems are ones in which there are neither so few constraints that most choices lead easily to a solution (under-constrained) nor so many constraints that unsatisfiability is easily demonstrated (over-constrained) [Bresciani *et al.*, 1995; Hogg *et al.*, 1996]. Critically constrained problems are consistently hard to solve due to subtle interactions between constraints. Under-constrained problems on the other hand, while generally easy to solve, can also prove extremely difficult in a small number of cases due to a phenomenon called thrashing [Gent and Walsh, 1996; Baker, 1996].

Thrashing occurs when the order in which non-deterministic constraints are expanded results in large amounts of unproductive backtracking. This problem is exaggerated in tableaux algorithms as inherent unsatisfiability caused by a disjunctive expansion choice may be concealed in a sub-problem and not discovered until many more disjunctions have been expanded. A naive backtracking strategy would try all possible combinations of these intervening disjunctions before getting back to the real cause of the unsatisfiability.

4 Optimisation techniques

Standard optimisation techniques [Baader *et al.*, 1992] do little to tackle the major cause of intractability in DLs which support GCIs—searching the large number of possible expansions offered by sets of disjunctive constraints. This problem is similar to searching possible variable assignments in SAT problems and adapted SAT optimisation techniques can be used to improve the performance of tableaux concept satisfiability testing algorithms. In fact it can be seen that the GCI constraint filtering technique [Horrocks *et al.*, 1996] corresponds to a common DPL[3] SAT optimisation, the deletion of clauses containing pure literals whose complements do not occur in the formula.

4.1 Heuristic guided search

Heuristic guided search can be used to try to increase the effectiveness of pruning and minimise the number of constraint systems which are explored. There are two steps to this procedure: firstly the pruning effect of earlier expansion choices is maximised by eliminating 'obviously satisfiable' disjunctions and 'obvious clashes' from the remaining disjunctions[4]; secondly the pruning effect of future expansion choices is maximised by preferentially expanding highly constraining disjunctions [Freeman, 1996].

A variation on the MOMS[5] heuristic [Freeman, 1995] can be used to estimate the degree to which each concept term in a disjunction constrains possible solutions. The heuristic assigns the highest value to concept terms which occur most frequently in disjunctive clauses of minimum size; it also prefers terms with similar numbers of positive and negative occurrences[6]. The most constraining disjunctions are estimated to be those containing concept terms with the highest MOMS value.

The search strategy, applied to variable x in a constraint system S, would proceed by expanding deterministic constraints (using the enhanced blocking strategy described in [Baader *et al.*, 1996]) and performing constraint propagation—disjunctive constraints $x : (C_1 \sqcup \ldots \sqcup C_n)$ are eliminated when one of $x : C_1, \ldots, x : C_n$ is in S and concept terms C are eliminated from disjunctions when $\neg C$ is in S[7]. The most constraining disjunctive constraint is then selected for expansion.

4.2 SAT based search

Recent work has shown how SAT procedures can be used directly in a decision procedure for $\mathcal{ALC}$ [Giunchiglia and Sebastiani, 1996a; 1996b]. This technique can also be used to provide a more efficient mechanism for dealing with disjunctive constraints in tableaux algorithms.

A pre-processing step is performed which lexically sorts, simplifies and explicitly encodes concept subterms. This allows lexically equivalent concept terms to be identified and quickly eliminates many unsatisfiable over-constrained problems[8]. For example $(C \sqcap D) \sqcap (\neg D \sqcup \neg C) \Rightarrow C' \sqcap \neg C' \Rightarrow \bot$. The pre-processing has two additional advantages: it allows search guiding heuristics to operate much more efficiently; it maximises the effect of the optimisation described in [Baader *et al.*, 1992], which uses a combination of delayed expansion and the retention of expanded names to provide earlier unsatisfiability detection.

The tableaux expansion algorithm proceeds normally through the deterministic expansion and constraint propagation phases. Heuristics are then used to select a term from one of the disjunction constraints (the current algorithm uses a combination of MOMS and Jeroslow and Wang heuristics with binary constraint propagation look-ahead as described in [Freeman, 1995]) and branching is then performed by recursively testing satisfiability with the chosen term added as a positive or negated constraint—the heuristic also determines in which order the two possibilities are explored.

[3] The Davis-Putnam procedure in Lovlands form, a widely used SAT algorithm.

[4] This process is called constraint propagation.

[5] *M*aximum *O*ccurences of *M*inimum *S*ize.

[6] This is computationally expensive without the pre-processing step described in section 4.2.

[7] If this results in any zero length disjunctions the algorithm can backtrack immediately.

[8] This was one of the major causes of the performance improvement exhibited by the SAT based $\mathcal{ALC}$ algorithm described in [Giunchiglia and Sebastiani, 1996b].

4.3 Intelligent backtracking

The thrashing problem can be addressed by using a form of backjumping [Baker, 1996; Ginsberg, 1993] to return rapidly to a branching point where a different choice could remove the source of the conflict. In backjumping constraints are labeled to indicate the disjunctive expansion choices from which they were derived. The initial $x : C$ constraint is labeled $\{0\}$, a constraint chosen from the nth disjunction expanded is labeled $\{n\}$ and constraints derived from deterministic expansion rules are labeled with the (union of the) label(s) from the triggering constraint(s).

Instead of simply returning *failed* the satisfiability testing procedure can return a dependency list—the (union of the) label(s) from the constraint(s) which caused the clash. If an expansion of the nth disjunction causes a clash such that the maximum value in the dependency list is less than n the expansion can terminate immediately and return the dependency list. If all expansions of the nth disjunction cause clashes the returned dependency list is the union of the disjunction constraint's own label and the dependency lists generated by all the clashes with the value n removed.

4.4 Obvious satisfiability

While optimisations which enhance the detection of obvious subsumption relations (unsatisfiable concept terms) are well known, it is difficult for tableaux algorithms to detect obvious non-subsumption relations (satisfiable concept terms) [Baader *et al.*, 1992]. This is unfortunate as failed subsumption tests predominate in realistic KBs where concepts typically have many more children than parents (a ratio of over 3:1 in the Galen KB) and as failed tests are typically more expensive (a ratio of over 7:1 in the Galen KB).

This problem can be tackled by caching a partial model for each concept and its negation—to keep space requirements within reasonable bounds only root constraints (those which apply to the first variable in a model) need to be cached. When testing the satisfiability of $C \sqcap \neg D$ a quick positive result can be returned by demonstrating that the known models for C and $\neg D$ can be merged.

If D is primitive, all that is required is that the model of C does not have a root constraint of the form $x : D$. When D is not primitive it is necessary to test for possible interactions between the two models—for example root constraints of the from xRy and $x : \forall R.C'$ in the models of C and D respectively. If there are no interactions a model of $C \sqcap \neg D$ could be formed by joining the models of C and $\neg D$ at their roots.

If there are interactions between the model roots, tableaux expansion must proceed. However a similar technique can still be applied at each model node in order to avoid solving obviously satisfiable sub-problems. It may also be worth maintaining a temporary cache to try and avoid the repeated solution of sub-problems during a single tableaux expansion or across the set of tableaux expansions required to classify of a concept.

5 Preliminary results

In spite of the filtering technique [Horrocks *et al.*, 1996] some satisfiability tests still produce constraint systems in which variables are subject to a considerable number of disjunctive constraints introduced by GCIs. For example demonstrating the satisfiability of the concept `BodySpace` (i.e. testing if $\perp$ subsumes `BodySpace`) produced a constraint system containing 31 disjunctive constraints[9] with a theoretical 5×10^{12} possible expansions. Using an unoptimised algorithm with this constraint system leads to severe thrashing and effective non-termination.

The introduction of backjumping causes a dramatic improvement and allows the satisfiability of `BodySpace` to be quickly demonstrated—a complete expansion is discovered after only 4 backtracks. Guided search succeeds in further reducing the number of backtracks (to 3 in this case) but the processing time actually increases due to the cost of the heuristics.

The analysis of SAT based search is still at an early stage but improvements only seem to be in the region of 20% and are certainly not of the magnitude demonstrated for random K-SAT problems in [Giunchiglia and Sebastiani, 1996b]. This may be explained by the structure of the KB: easy over-constrained problems are rare and the number of concept terms[10] is large. This means that cases which flatter the SAT procedure occur only infrequently. However the lexical pre-processing does facilitate the operation of search heuristics and it should be possible to improve on the general SAT heuristics currently employed by using knowledge about the structure of concepts in the KB.

The current implementation only detects obvious satisfiability at the top level (before any tableaux expansion is performed). This does however allow over 75% of failed subsumption tests (almost 60% of all subsumption tests) to be avoided when classifying the Galen model and leads to a reduction of approximately 60% in total classification time.

6 Conclusion

Using a DL which supports GCIs may at first seem unpromising due to theoretical intractability but heuristic speed up techniques can be used to improve performance with a realistic knowledge base. A considerable amount

[9] The topology of hollow body structures is represented in detail in the Galen model.

[10] Equivalent to propositional variables in SAT problems.

of work remains to be done to evaluate and improve the optimisation procedures but early results show considerable promise.

The introduction of backjumping seems to be particularly effective and it may be worth using more sophisticated backtracking procedures, such as dynamic backtracking [Ginsberg, 1993], which try to preserve useful work while returning to the source of discovered contradictions. Caching of partial constraint systems also proved a useful technique and allowed large numbers of failed subsumption tests to be avoided—extending this technique to tableaux sub-problems should provide further performance improvements. The guided search techniques were less effective but this may be due in part to the use of inappropriate heuristics directly adapted from SAT procedures—work is continuing on the design of improved heuristics. It is also hoped to investigate the effectiveness of the optimisation techniques with other knowledge bases and with other DLs.

The worst case complexity of the problem is immutable but many instances of observed poor performance may actually be artifacts of the algorithm and its implementation. Although there will be some hard problems which are not amenable to optimisation techniques it seems likely that these will be rare in realistic KBs. Detecting the characteristic thrashing of the algorithm which these cases cause and terminating the satisfiability test would result in a degree of incompleteness (if *success* is returned) or unsoundness (if *failed* is returned) which may be acceptable in applications where the expressive power provided by GCIs is of critical importance.

References

[Baader *et al.*, 1991] F. Baader, H.-J. Heinsohn, B. Hollunder, J. Muller, B. Nebel, W. Nutt, and H.-J. Profitlich. Terminological knowledge representation: A proposal for a terminological logic. Technical Memo TM-90-04, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), 1991.

[Baader *et al.*, 1992] F. Baader, B. Hollunder, B. Nebel, and H.-J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems. In B. Nebel, C. Rich, and W. Swartout, editors, *Principals of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92)*, pages 270–281. Morgan-Kaufmann, 1992. Also available as DFKI RR-93-03.

[Baader *et al.*, 1996] F. Baader, M. Buchheit, and B. Hollunder. Cardinality restrictions on concepts. *Artificial Intelligence*, 1996. To appear.

[Baader, 1990] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. Research Report RR-90-13, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), 1990.

[Baker, 1996] A. B. Baker. Intelligent backtracking in the hardest constraint problems. *Journal of Artificial Intelligence Research*, 1996. To appear.

[Bresciani *et al.*, 1995] P. Bresciani, E. Franconi, and S. Tessaris. Implementing and testing expressive description logics: a preliminary report. In Gerard Ellis, Robert A. Levinson, Andrew Fall, and Veronica Dahl, editors, *Knowledge Retrieval, Use and Storage for Efficiency: Proceedings of the First International KRUSE Symposium*, pages 28–39, 1995.

[Buchheit *et al.*, 1993] M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.

[Freeman, 1995] J. W. Freeman. *Improvements to propositional satisfiability search algorithms*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA, 1995.

[Freeman, 1996] J. W. Freeman. Hard random 3-SAT problems and the Davis-Putnam procedure. *Artificial Intelligence*, 81:183–198, 1996.

[Gent and Walsh, 1996] I. P. Gent and T. Walsh. The satisfiability constraint gap. *Artificial Intelligence*, 81:59–80, 1996.

[Ginsberg, 1993] M. L. Ginsberg. Dynamic backtracking. *Journal of Artificial Intelligence Research*, 1:25–46, 1993.

[Giunchiglia and Sebastiani, 1996a] F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures—the case study of modal K. In Michael McRobbie and John Slaney, editors, *Proceedings of the Thirteenth International Conference on Automated Deduction (CADE-13)*, number 1104 in Lecture Notes in Artificial Intelligence. Springer, 1996.

[Giunchiglia and Sebastiani, 1996b] F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for $\mathcal{ALC}$. In *Principals of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR'96)*. Morgan Kaufmann, 1996.

[Hogg *et al.*, 1996] T. Hogg, B. A. Huberman, and C. P. Williams. Phase transitions and the search problem. *Artificial Intelligence*, 81:1–15, 1996. Editorial.

[Horrocks *et al.*, 1996] I. Horrocks, A. Rector, and C. Goble. A description logic based schema for the classification of medical data. In F. Baader, M. Buchheit, M.A. Jeusfeld, and W. Nutt, editors, *Reasoning about structured objects—knowledge representa-*

tion meets databases. Proceedings of the 3rd Workshop KRDB'96, pages 24–28, 1996.

[Nowlan, 1993] W. A. Nowlan. *Structured methods of information management for medical records*. PhD thesis, University of Manchester, 1993.

[Rector *et al.*, 1993] A. L. Rector, W A Nowlan, and A Glowinski. Goals for concept representation in the GALEN project. In *Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care (SCAMC'93)*, pages 414–418, Washington DC, USA, 1993.

[Sattler, 1996] U. Sattler. The complexity of concept languages with different kinds of transitive roles. In G. Görz and S. Hölldobler, editors, *20. Deutsche Jahrestagung für Künstliche Intelligenz*, number 1137 in Lecture Notes in Artificial Intelligence. Springer Verlag, 1996.

[Schild, 1991] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proceedings of IJCAI'91*, pages 466–471, 1991.