Using a chart to record intermediate results in a model generation theorem prover

Allan Ramsay
Dept of Computation, UMIST,
PO Box 88, Manchester M60 1QD, UK
Allan.Ramsay@umist.ac.uk

Abstract

Model generation [Manthey and Bry, 1988] provides a very straightforward basis on which to construct more elaborate theorem provers where some rules are applied backwards and some forwards. Unfortunately the basic algorithm can lead to a considerable amount of repeated and redundant work. The current paper exploits ideas from chart parsing [Kay, 1973, Kaplan, 1973] to minimise this problem.

1 Background

Various tasks in natural language processing require you to be able to reason: if you want to do anything as a consequence of hearing/reading an utterance/text, you have to be able to work out its consequences[Appelt, 1985, Asher and Lascarides, 1995, Baumgartner and Kühn, 2000, Blackburn et al., 1997, Crouch et al., 2001, Gardent and Konrad, 2000, Hirst, 1987, Hobbs et al., 1993, McRoy and Hirst, 1995, Schubert and Pelletier, 1984, Sperber and Wilson, 1986, Wedekind, 1996]. The situation is complicated by the fact that this reasoning often involves intensional axioms and logical forms. This happens when reasoning about lexical items that explicitly express propositional attitudes (words like *'want'*, *'regret'*, *'expect'*); for dealing with discourse relations such as the theme:rheme distinction or the significance of phonological marks such as contrastive stress; and for providing an account of terms like *'only'* or *'even'* which express an attitude to parts of the utterance [Pulman, 1993]. We have exploited the distinction between backward- and forward-rules that is implicit in Manthey and Bry [1988]'s MODEL GENERATION theorem prover to develop an inference engine for a highly intensional logic, namely a constructive version of Turner [1987]'s PROPERTY THEORY. This language is more expressive than a logic like λProlog [Nadathur and Miller, 1998] in that it allows for NON-UNIFORM PROOFS [Miller et al., 1991, Loveland and Nadathur, 1998]: the price we pay is that inference is even more difficult than usual.

The version of this inference engine reported in [Ramsay, 2001] incorporates a number of optimisations, mostly techniques that have been developed for standard first-order inference and adapted to work with the intensional language we are using. This theorem prover has been used to solve a number of problems in NLP [Ramsay, 1999a,b,c, 2000, Ramsay and Seville, 2000b,a, 2001b,a, Seville and Ramsay, 2000, Gaylard and Ramsay, 2001], but the basic model generation algorithm can lead to a considerable amount of repeated and redundant work. We therefore propose to exploit ideas from chart parsing [Kay, 1973, Kaplan, 1973] to minimise this problem.

2 Chart parsing

Chart parsing was introduced to minimise the extent to which partial and failed analyses of fragments of an utterance are repeatedly investigated during the process of obtaining a global structural analysis of an input sentence. The key observation behind chart parsing is that most parsing algorithms contain a step in which an incomplete analysis is extended by incorporating a new fragment. The so-called FUNDAMENTAL RULE OF CHART PARSING says that if you have an incomplete constituent $\mathtt{X/[Y_1,\ Y_2\ \ldots,\ Y_n]}$ adjacent to an item of type $\mathtt{Y_1}$ then you can merge them to form a new, more complete item $\mathtt{X/[Y_2,\ \ldots,\ Y_n]}$[1]. The crucial point is that if you retain *every* such object that is created during the course of parsing, and use this to block the creation of new but repetitive items, then you can have a cache of positive and negative lemmas which will enable you to avoid carrying out repeated work.

3 Chart inference

The fundamental rule of chart parsing looks extraordinarily like an application of resolution. It should therefore be possible to exploit the same mechanism in a theorem prover. The question is whether the overheads associated with keeping such a collection of lemmas pays for itself in terms of cutting down the search space. In the long run it must do: the argument that chart parsing provides an N^3 algorithm for analysing text in terms of a context-free, i.e. propositional, grammar whereas simple-minded chronological backtracking is exponential has to be adapted when the framework is first-order or worse, but the argument still applies. Thus for the backward-chaining part of the model generation algorithm, which applies solely to the Horn subset of the problem, there will be some performance gains from using a chart.

But for model generation the improvement should be even more dramatic than that. One of the problems with model generation is that it repeatedly generates new literals to be added to the set of Horn sequents and then tries to (a) prove $\perp$, and (b) if that fails then find a new literal to add. This, in the standard presentation, involves attempting the same proofs over and over again. By using a chart the addition of a new literal directly restart suspended, but failed, proofs of $\perp$ and of the availability of forward rules. We no longer have to repeat old proofs: we resume partial proofs immediately at the point at which they were suspended. This will also work for proofs which were suspended because two items failed to unify when further analysis reveals that the items in question are equal. The treatment of equality within the chart-based approach is more awkward than it was in the standard implementation, where we carry the current equivalence sets around in the LABEL [Gabbay, 1996] associated with the proof, but at the same time it allows more effective use of discoveries about what is equal to what.

The current state of the implementation of this notion is that we have the chart in place for avoiding repetition in the backward chaining part of the proof procedure, but that the integration with the forward chaining phase is not yet complete.

[1] The notation $\mathtt{X/[Y_1,\ \ldots,\ Y_n]}$ is a variant on the way CATEGORIAL GRAMMAR describes an item of type $\mathtt{X}$ which requires to combine with items of type $\mathtt{Y_1,\ \ldots,\ Y_n}$ in order to become 'complete' [Steedman, 2000, Wood, 1993]. In parsing, of course, you need directional information about whether $\mathtt{Y_1}$ should be to the right or left of $\mathtt{X/[Y_1,\ Y_2\ \ldots,\ Y_n]}$, but this doesn't make things much more complex.

References

D Appelt. *Planning English Sentences.* Cambridge University Press, Cambridge, 1985.

N Asher and A Lascarides. Lexical disambiguation in a discourse context. *Journal of Semantics*, 1995.

P Baumgartner and M Kühn. Abducing coreference by model construction. *Journal of Language and Computation*, 1(2), 2000.

P Blackburn, J Bos, M Kohlhase, and H de Nivelle. Inference and computational semantics. In H Bunt and E Thijsse, editors, *3rd International Workshop on Computational Semantics*, pages 5–19, University of Tilburg, 1997.

R Crouch, A Frank, and J van Genabith. Linear logic based transfer and structural misalignment. In H C Bunt, I van der Sluis, and E Thijsse, editors, *4th International Workshop on Computational Semantics*, pages 35–49, University of Tilburg, 2001.

D M Gabbay. *Labelled Deductive Systems.* Oxford University Press, Oxford, 1996.

C Gardent and K Konrad. Interpreting definites using model generation. *Journal of Language and Computation*, 1(2):215–230, 2000.

H Gaylard and A M Ramsay. *Any*: The hearer's role in discourse update. In G Angelova, K Bontcheva, R Mitkov, N Nicolov, and N Nikolov, editors, *Recent Advances in Natural Language Processing*, pages 108–114, Tzigov Chark, Bulgaria, 2001.

G Hirst. *Semantic interpretation and the resolution of ambiguity.* Studies in natural language processing. Cambridge University Press, Cambridge, 1987.

J R Hobbs, M E Stickel, D E Appelt, and P Martin. Interpretation as abduction. *Artificial Intelligence*, 63:69–142, 1993.

R M Kaplan. A general syntactic processor. In R. Rustin, editor, *Natural language processing*, pages 193–241, New York, 1973. Algorithmics Press.

M Kay. The MIND system. In R. Rustin, editor, *Natural Language Processing*, pages 155–188, New York, 1973. Algorithmics Press.

D W Loveland and G Nadathur. Proof procedures for logic programming. In D M Gabbay, C J Hogger, and J A Robinson, editors, *Handbook of logic in artificial intelligence and logic programming: volume 5, logic programming*, pages 163–234. Clarendon Press, Oxford, 1998.

R Manthey and F Bry. Satchmo: a theorem prover in Prolog. In *Proceedings of the 9th International Conference on Automated Deduction (CADE-9)*, volume 310 of *Lecture Notes in Artificial Intelligence*, pages 415–434, Berlin, 1988. Springer-Verlag.

S W McRoy and G Hirst. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, 21(4):435–478, 1995.

D Miller, G Nadathur, F Pfennig, and A Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

G Nadathur and D Miller. Higher-order logic programming. In D M Gabbay, C J Hogger, and J A Robinson, editors, *Handbook of logic in artificial intelligence and logic programming: volume 5, logic programming*, pages 499–590. Clarendon Press, Oxford, 1998.

S G Pulman. Higher order unification and the interpretation of focus. *Linguistics & Philosophy*, 20(1):73–115, 1993.

A M Ramsay. Direct parsing with discontinuous phrases. *Natural Language Engineering*, 5(3): 271–300, 1999a.

A M Ramsay. Dynamic and underspecified semantics without dynamic and underspecified logic. In H Bunt, L Kievit, R Muskens, and M Verlinden, editors, *Computing Meaning*, volume 1, pages 208–220, Dordrecht, 1999b. Kluwer Academic Publishers (SLAP 73).

A M Ramsay. Weak lexical semantics and multiple views. In H C Bunt and E G C Thijsse, editors, *3rd International Workshop on Computational Semantics*, pages 205–218, University of Tilburg, 1999c.

A M Ramsay. Speech act theory and epistemic planning. In W J Black and H Bunt, editors, *Abduction, Beliefs and Context in Dialogue: Studies in Computational Pragmatics*, pages 293–310, Amsterdam, 2000. John Benjamins.

A M Ramsay. Theorem proving for untyped constructive λ-calculus: implementation and application. *Logic Journal of the Interest Group in Pure and Applied Logics*, 9(1):89–106, 2001. ISSN ISSN: 1367-0751.

A M Ramsay and H Seville. Models and discourse models. *Journal of Language and Computation*, 1(2):167–181, 2000a.

A M Ramsay and H Seville. Unscrambling English word order. In M Kay, editor, *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 656–662, Universität des Saarlandes, July 2000b.

A M Ramsay and H Seville. Relevant answers to wh-questions. In *3rd International Conference on Inference in Computational Semantics*, pages 73–86, Siena, 2001a.

A M Ramsay and H Seville. What situational and discourse relations entail. In H C Bunt, I van der Sluis, and E Thijsse, editors, *4th International Workshop on Computational Semantics*, pages 417–432, University of Tilburg, 2001b.

L Schubert and F J Pelletier. From English to logic: Context-free computation of 'conventional' logical translations. *American Journal of Computational Linguistics*, pages 165–176, 1984.

H Seville and A M Ramsay. Making sense of reference to the unfamiliar. In M Kay, editor, *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 775–781, Universität des Saarlandes, July 2000.

D Sperber and D Wilson. *Relevance: communication and cognition*. Blackwell, Oxford, 1986.

M Steedman. *The Syntactic Process*. MIT Press, Cambridge, Ma., 2000.

R Turner. A theory of properties. *Journal of Symbolic Logic*, 52(2):455–472, 1987.

J Wedekind. On inference-based procedures for lexical disambiguation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 980–985, Copenhagen, 1996.

M M Wood. *Categorial Grammars*. Routledge, London, 1993.