

Inference Graphs: A New Kind of Hybrid Reasoning System*

Daniel R. Schlegel and Stuart C. Shapiro

Department of Computer Science and Engineering
University at Buffalo, Buffalo NY, 14260
<drschleg,shapiro>@buffalo.edu

Abstract

Hybrid reasoners combine multiple types of reasoning, usually subsumption and Prolog-style resolution. We outline a system which combines natural deduction and subsumption reasoning using Inference Graphs implementing a Logic of Arbitrary and Indefinite Objects.

1 Introduction

Inference Graphs (IGs) (Schlegel and Shapiro 2013; 2014) are a graph-based forward/backward reasoning mechanism supporting concurrency built upon propositional graphs (Shapiro and Rapaport 1992). IGs have previously been described for grounding predicate logic. We present a new technique for combining natural deduction reasoning with subsumption reasoning by extending IGs to implement a Logic of Arbitrary and Indefinite Objects (\mathcal{L}_A) (Shapiro 2004), a first order logic using structured quantified terms.

Natural deduction is a proof-theoretic reasoning technique with introduction and elimination rules for each connective, some of which use subproofs. Subsumption allows new beliefs about arbitrary objects to be derived directly from beliefs about other more general arbitrary objects.

Modern hybrid reasoners focus mostly on combining ontologies containing description logic classes with logic programming. This results in knowledge representations with expressiveness at the intersection of the combined reasoning techniques (Grosz et al. 2003), or some other decidable fragment of first order logic (Motik, Sattler, and Studer 2005). We assert that a human-level AI must be at least as expressive as first order logic (FOL), so are not as concerned about decidability. In addition, we believe an AI should store intermediate beliefs when reasoning (as a human does), and so should use a proof-theoretic technique, unlike the above which use Prolog-style resolution.

*This work has been supported by a Multidisciplinary University Research Initiative (MURI) grant (Number W911NF-09-1-0392) for Unified Research on Network-based Hard/Soft Information Fusion, issued by the US Army Research Office (ARO) under the program management of Dr. John Lavery.
Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2 Background¹

\mathcal{L}_A is a FOL designed for use as the logic of a KR system for natural language understanding, and for commonsense reasoning. It is sound and complete, and makes use of structured arbitrary and indefinite terms (collectively, quantified terms) instead of the universally and existentially quantified variables familiar in first order predicate logic. Instead of reasoning about *all* members of a class, \mathcal{L}_A reasons about a *single* arbitrary member of a class. Indefinite members are reasoned about much like Skolem functions with dependencies on arbitraries. Quantified terms range over their set of restrictions, $R(q_i)$, providing structure, and allowing for their placement in a subsumption hierarchy. A generic term is a term with a quantified or generic sub-term.

Propositional graphs are graphs in which every well-formed expression in the knowledge base is represented by a node in the graph. A rule is represented in the graph as a node for the rule itself, nodes for the argument formulas, and arcs emanating from the rule node, terminating at the argument nodes. Arcs are labeled with an indication of the role (*e.g.*, antecedent or consequent) the argument plays in the rule, itself. No two nodes represent syntactically identical expressions; rather, the same node is used in all cases.

IGs for ground predicate logic are propositional graphs in which directed communication *channels* have been added between nodes along possible inference paths. *Messages* flow through channels to relay new information from one node to another, and are combined in rule nodes to apply rules of inference.

Channels come in two varieties: i-channels carry messages saying “I am asserted/negated;” and u-channels carry messages saying “You are asserted/negated.” Each channel contains a valve, which controls inference by (when open) allowing or (when closed) preventing message flow through a channel. Messages wait at a closed valve until it is opened.

When a deductive rule is added to the graph, channels are built within it. A rule consists of one or more antecedents, one or more consequents, and a rule node, r , for the rule itself. I-channels are created from each antecedent to r , and u-channels from r to each consequent.

Channels carry messages: *i-infer* and *u-infer* messages are sent along i- or u-channels to communicate infer-

¹This section is adapted from (Schlegel and Shapiro 2014).

ences and contain a *flagged node set* which provides a mapping from each antecedent to its truth value (when known).

Messages are combined in rule nodes, which implement the rules of inference. When a combined message has an appropriate number of true or negated entries in its flagged node set, the rule fires, sending *u-infer* messages to some or all of its consequents via its u-channels, causing them to become asserted.

3 Hybrid Reasoning with Inference Graphs

Since generic terms contain quantified terms, supporting them in IGs (both for natural deduction and subsumption inference) requires several additions. Channels must be created between nodes for terms which *match*, or which unify and meet certain subsumption and type restrictions, so that those nodes may communicate. Messages must carry substitutions between terms, and channels must ensure those substitutions are relevant to and in the proper variable context of the destination. When messages are combined in rule nodes, their substitutions must be taken into account so that only compatible substitutions are combined. Arbitrary objects must be able to produce instances of themselves by also combining compatible substitutions from their restrictions.

When a term is added to the graph it is matched with all other terms to determine if channels should be created between two terms. First, the added term is unified with all other terms in the graph, then the resulting substitutions are checked for appropriate type and subsumption relationships. That is, more specific terms may share their instances only with equally or less specific ones they unify with.

Messages are extended to carry a substitution, and channels are enhanced with operations to maintain those substitutions. i-channels can now be thought of as carrying messages reporting “I have a new (negated) substitution instance,” and u-channels as carrying messages reporting “you have a new (negated) substitution instance”. Additionally we add a new type of channel, g-channels, which are i-channels, but exist only within generics to distinguish between channels to other generic terms, and to non-generics.

In addition to the existing valve, we add a filter and switch to each channel. The filter ensures a message’s substitution is relevant to the destination node, discarding it if it is not. The switch alters the variable context of the passing message from that of the originator, to that of the destination.

In addition to within rules, and between matching terms, channels are added within generic terms. Each quantified term, q_i , and each generic term, g_k which is a direct subterm of a generic term g has an outgoing g-channel to g . g has outgoing g-channels to other generic terms which it is a direct subterm of. Finally, each indefinite term has incoming g-channels from each arbitrary term which it depends on.

Messages are combined in three types of nodes: in a rule node, as previously discussed; in a quantified term, to determine if substitutions from each of the restrictions are compatible with each other, and thus instantiate the term; and in a generic term to determine if substitutions it has received on its incoming g-channels are compatible, satisfying it.

Most interestingly for hybrid reasoning, a generic term g also collects substitutions, ϕ , for asserted terms which match

g . When g is satisfied by a substitution ψ which is compatible with ϕ , an *i-infer* message with substitution ψ is sent out all of g ’s i-channels, regardless of whether g itself is asserted. This allows g to be used as the antecedent for a deductive rule, where g ’s requirements are satisfied, but nothing new is derived by g .

These enhancements allow inferences like the following. Consider: if the arbitrary small dog yips, then that it is scared. If the arbitrary dog is scared, then it is carried. Toto is a dog, is small, and yips. We can derive that Toto is scared, since it satisfies the conditions to instantiate the arbitrary small dog, and also yips. The arbitrary small dog who is scared matches the arbitrary dog who is scared, because of the arbitrary dog is more general than the arbitrary small dog. Therefore, since the arbitrary dog is instantiated by Toto, and Toto is scared, we know Toto is carried. Using a similar strategy, we can derive that if the arbitrary *small* dog is scared, then it is carried.

We have previously evaluated the concurrent performance of IGs for ground predicate logic (Schlegel and Shapiro 2014), and found a near-linear speedup with the number of processors. We plan in the future to extend this evaluation to IGs which use hybrid reasoning.

4 Conclusion

IGs have previously been shown to allow for efficient forward and backward reasoning through the extension of propositional graphs, but only implementing reasoning over ground predicate logic. By implementing an algorithm to determine if two terms match each other, using unification, subsumption, and a type hierarchy; augmenting messages with substitutions, and channels with a way to ensure that messages are relevant and in the proper context when received; and adding additional channels between matching terms and within generic terms, we have shown that Inference Graphs may be extended to hybrid reasoning that combines subsumption reasoning with natural deduction over a logic as expressive as FOL.

References

- Grosz, B. N.; Horrocks, I.; Volz, R.; and Decker, S. 2003. Description logic programs: Combining logic programs with description logic. In *Proceedings of WWW 2003*, 48–57.
- Motik, B.; Sattler, U.; and Studer, R. 2005. Query answering for OWL-DL with rules. *Web Semantics: Science, Services and Agents on the World Wide Web* 3(1):41–60.
- Schlegel, D. R., and Shapiro, S. C. 2013. Inference graphs: A roadmap. In *Poster Collection of the Second Annual Conference on Advances in Cognitive Systems*, 217–234.
- Schlegel, D. R., and Shapiro, S. C. 2014. Concurrent reasoning with inference graphs. In Croitoru, M.; Rudolph, S.; Woltran, S.; and Gonzales, C., eds., *Graph Structures for Knowledge Representation and Reasoning*, volume 8323 of *LNAI*. Switzerland: Springer International Publishing. 138–164.
- Shapiro, S. C., and Rapaport, W. J. 1992. The SNePS family. *Computers & Mathematics with Applications* 23(2–5):243–275.
- Shapiro, S. C. 2004. A logic of arbitrary and indefinite objects. In Dubois, D.; Welty, C.; and Williams, M., eds., *Proceedings of KR2004*, 565–575. Menlo Park, CA: AAAI Press.