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; 2014a) are a graph-based forward/backward reasoning mechanism supporting concurrency built upon propositional graphs (Shapiro and Rapaport 1992). IGs have previously been described for ground 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. This paper is an expanded version of (Schlegel and Shapiro 2014b).

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 (Grosof 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.

2 Background[1]

2.1 The Logic of Arbitrary and Indefinite Objects

$\mathcal{L}_A$ 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 formulas 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 are structured – they consist of a quantifier indicating whether they are arbitrary or indefinite, a syntactic variable, and a set of *restrictions*. The range of a quantified term is dictated by its set of restrictions, taken conjunctively. A quantified term q_i has a set of restrictions $R(q_i) = \{r_{i_1}, \ldots, r_{i_k}\}$, where each r_i makes use of q_i's variable, v_i. Indefinite terms may be dependent on one or more arbitrary terms $D(q_i) = \{d_{i_1}, \ldots, d_{i_k}\}$. The syntax used for $\mathcal{L}_A$ is a version of CLIF (ISO/IEC 2007). An arbitrary term is written as `(every` v_{q_i} $R(q_i)$`)` and an indefinite term as `(some` v_{q_i} $D(q_i)$ $R(q_i)$`)`.

Since an arbitrary term represents an arbitrary entity, there are no two arbitrary terms with the same set of restrictions. That said, it is sometimes useful to discuss two different arbitrary members with the same restrictions. This can be done using the special restriction `(notSame` $q_1 \ldots q_n$`)`.

To give an idea as to the flavor of inference in $\mathcal{L}_A$, consider the following examples. First, we can represent the rule that "either Fido is owned, or Fido is feral" as:

```
(xor (Owned Fido) (Feral Fido))
```

From this, given either `(Owned Fido)`, or `(Feral Fido)` the negation of the other can be derived, or given either `(not (Owned Fido))`, or `(not (Feral Fido))`, the positive instance of the other can be derived.

Terms which are not deductive rules, but contain quantified terms are called *generic terms*. Structural subsumption can be used on these terms. The following is meant to mean that "every owned dog is a pet."

[1]Portions of the material in this section are adapted from (Schlegel and Shapiro 2013).

```
(Isa (every x (Owned x) (Isa x Dog))
     Pet)
```

Now, given that Fido is a Dog – `(Isa Fido Dog)` – and Fido is Owned – `(Owned Fido)` – we can derive that Fido is a Pet – `(Isa Fido Pet)` since Fido is subsumed by the arbitrary term `(every x (Isa x Dog) (Owned x))`.

Any rule which uses subsumption inference can be rewritten to use implication as the main connective (though the reverse is not true). For example, we can rephrase the above to mean "if a dog is owned, then it is a pet" as follows:

```
(if (Owned (every x (Isa x Dog)))
    (Isa x Pet))
```

As above, when given that Fido is a Dog, and Fido is Owned, we can derive that Fido is a Pet. This time the inference is hybrid – both subsumption and natural deduction are used in the derivation. Arbitrary terms take wide scope, allowing `x` to be used in the consequent of the rule without re-definition.

For trivial examples such as this, it may not be particularly appealing that there are two ways to write derivationally equivalent expressions, but some expressions in English are difficult to express without one or more propositional connectives, at least without first re-wording the English expression. For example, "Two people are colleagues if there is some committee they are both members of."

2.2 Knowledge Representation

In the tradition of the SNePS family (Shapiro and Rapaport 1992), 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 (henceforth, a *rule node*), 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. Every node is labeled with an identifier. Nodes representing individual constants, proposition symbols, function symbols, or relation symbols are labeled with the symbol itself. Nodes representing functional terms or non-atomic formulas are labeled `wft`*i*, for some integer, *i*. Every SNePS expression is a term, hence `wft` instead of `wff`. An exclamation mark, "`!`", is appended to the label if it represents a proposition that is asserted in the KB. Arbitrary and indefinite terms are labeled `arb`*i* and `ind`*i*, respectively. Restrictions for arbitrary and indefinite terms are represented in the graph with special arcs labeled "restrict." Dependencies of indefinites are represented in the graph with special arcs labeled "depend." No two nodes represent syntactically identical expressions; rather, the same node is used in all cases.

2.3 Inference Graphs for Ground Predicate Logic

Inference Graphs 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. Messages are processed concurrently to the greatest extent possible. We will now discuss this type of IG in only the detail necessary to further our discussion of hybrid reasoning.

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 inferences 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

Inference Graphs are modified in several ways to support quantified and generic terms, both for natural deduction and subsumption inference. Channels are created between nodes for terms which *match*, or which unify and meet certain subsumption and type restrictions. Messages carry substitutions between matching terms, and channels ensure those substitutions are relevant to, and in the proper variable context of, the destination. When messages are combined in rule nodes, their substitutions are taken into account so that only compatible substitutions are combined. Arbitrary objects produce instances of themselves by also combining compatible substitutions from their restrictions.

To motivate the discussion, we introduce an example inspired by the counter-insurgence domain, first only in the logical syntax of $\mathcal{L}_A$, and in a later subsection, using an IG.

```
;; A person is arrested if and only if
;;  they are held by a another person
;;   who is a corrections officer.
(iff
  (Arrested (every x (Isa x Person)))
  (heldBy x (some y (x)
              (Isa y Person)
              (Isa y CrctnsOfcr)
              (notSame x y))))

;; A person is detained if and only
;;  if they are held by another person.
(iff
  (Detained (every x (Isa x Person)))
  (heldBy x
         (some y (x) (Isa y Person)
```

```
                            (notSame x y))))

;; A person is either detained,
;;  on supervised release, or free.
(xor
  (Detained (every x (Isa x Person)))
  (onSupervisedRelease x)
  (Free x))

;; A person who is not free
;;  has travel constraints.
(hasTravelConstraints
  (every x (Isa x Person)
           (not (Free x))))

;; Azam is an arrested person.
(Arrested Azam)
(Isa Azam Person)
```

From this example KB, we'd like to reason that Azam has travel constraints. Here is an informal proof: since Azam has been arrested, he is held by some person who is a corrections officer, and therefore held by a person. Since he is held by a person who isn't himself, he is detained, and therefore is not free. Since Azam is a person who is not free, he has travel constraints. In the following several subsections we'll see how IGs are able to come to this same conclusion.

3.1 The Match Process

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 (treating quantified terms as just simple variables), then the resulting substitutions are checked for appropriate type and subsumption relationships.

When two terms, t_i and t_j, are unified, instead of producing an mgu, a factorization (McKay and Shapiro 1981) is produced which contains bindings for each of the terms being unified. To produce the bindings, instead of forming a single substitution during unification, form two – σ_i and σ_j – such that all and only quantified terms in t_i are given bindings in σ_i, and all and only quantified terms in t_j are given bindings in σ_j.

Once t_i and t_j have unified, it is determined in which direction(s) (if either) their substitutions are compatible in their subsumption relationship and in type. More specific terms may share their instances with less specific ones. From our example, consider that some person who is a corrections officer is still a person.

3.2 Channels

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, g-channels are added, which are i-channels, but exist only within generics to distinguish between channels to other generic terms, and to non-generics.

Channels now contain a valve (as before), a filter, and a switch. A filter ensures a message's substitution, ϕ, is relevant to the destination node, D, by ensuring that for every substitution pair $t_i/v_i \in \tau$ (where τ are the destination bindings) there is a substitution pair $t_j/v_i \in \phi$ such that either $t_i = t_j$ or t_j is a specialization of t_i, determinable through one-way pattern matching. If a message does not pass the filter, it is discarded. For each $t_i/v_i \in \phi$, the switch applies the σ, the originator bindings, to t_i, and stores the new substitution in the message being passed. This adjusts the substitution to use quantified terms required by D.

In addition to the discussed channels within deductive rules, and between matching terms, channels are added within generic terms. A generic term, g, is defined recursively as a term which has as a direct subterm one or more quantified terms $q_1, \ldots, q_n$, or one or more other generic terms, $g_1, \ldots, g_m$. Each q_i and g_k has an outgoing g-channel to g. Each indefinite term, ind_k, has incoming g-channels from each arbitrary term which it depends on.

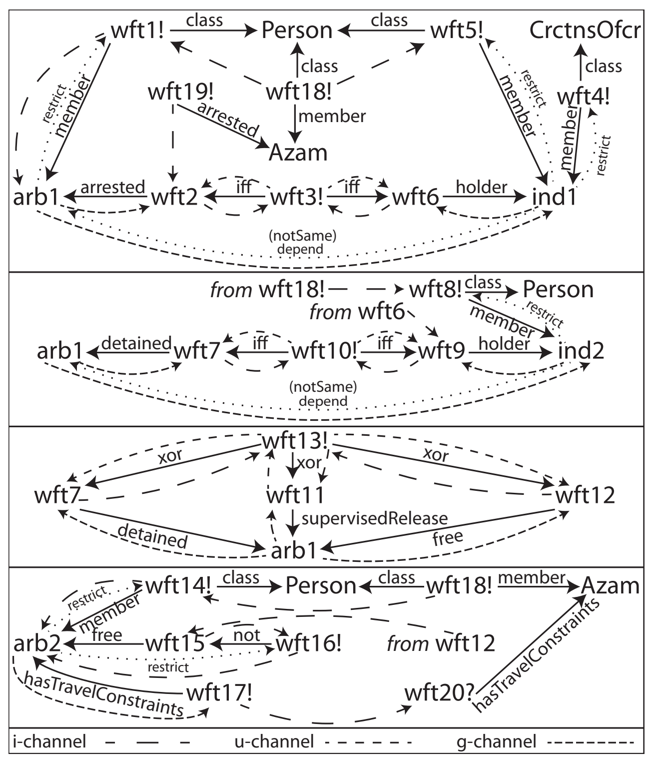

Figure 1: The IG for the example introduced in Section 3, split into four segments for easier understanding.

In Figure 1 the IG for our example is shown, with channels drawn as specified. The channels indicate what paths inference might take. For example, `wft2` has an i-channel to `wft3!` and `wft3!` has a u-channel to `wft2`, indicating that `wft2` may want to share a substitution it has with `wft3!`, and `wft3!` might derive `wft2`. Additionally, `wft19!` has an i-channel to `wft2`, indicating that `wft2` may be interested in `wft19!`'s assertional status and substitution.

3.3 Inference

Inference in an IG is essentially the combination of messages, and the determination of whether a resulting combination satisfies the requirements of a deductive rule, quantified term, or generic term.

Two messages, m_1 and m_2 may be combined if they are compatible – that is, if their substitutions and flagged node sets are compatible. We say that two substitutions, $\sigma = \{t_{\sigma_1}/v_{\sigma_1} \dots t_{\sigma_n}/v_{\sigma_n}\}$ and $\tau = \{t_{\tau_1}/v_{\tau_1} \dots t_{\tau_m}/v_{\tau_m}\}$, are compatible if whenever $v_{\sigma_i} = v_{\tau_j}$ then $t_{\sigma_i} = t_{\tau_j}$, and that two flagged node sets are compatible if they have no contradictory entries (that is, no antecedent of the rule is both true and false).

Messages are combined in three types of nodes: in a rule node, as previously discussed; in quantified terms, to determine if substitutions from each of a terms restrictions are compatible with each other, and thus instantiate the term; and in a generic term to determine if its received substitutions are compatible, satisfying it. Messages may be combined efficiently using several different data structures (Choi and Shapiro 1992).

Both indefinite and arbitrary terms combine messages. First, for arbitraries, the restrictions of an arbitrary term arb, represented by terms with i-channels to arb, are to be taken conjunctively, and as such messages must be combined from these nodes as they are available. In the example this is evident in `arb2`, which has two restrictions: `wft14!` and `wft16!`. Only when both of these terms report to `arb2` with compatible substitutions can the combined substitution be sent to `wft17!` for instantiation of the generic.

Indefinite terms operate differently, collecting instances of arbitraries from each of their dependencies. In ind_i these messages are combined, and when $m_{pos} = |D(ind_i)|$, a new indefinite term ind_j is produced which folds the dependencies of ind_i into the restrictions,

(some v_{ind_i} {}

$$\sigma R(ind_i) \cup \{\bigcup \sigma R(arb_k) | arb_k \in D(ind_i)\}),$$

where σ is the substitution in m, and $\sigma R(q)$ indicates the application of σ to each restriction of q. Finally m is sent along outgoing g-channels, with substitution $\sigma \cup \{ind_j/ind_i\}$. In our example, `ind2` reports its substitution to `wft9` as {Azam/arb1, ind3/ind2}, where ind3 is `(some x () (Isa x Person) (notSame x Azam))`.

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. `wft2` exemplifies this – neither just the fact that Azam was arrested (`wft19!`), nor the fact that Azam is a person (`wft1!`) would form a proper instance of `wft2`, only the two facts in combination do.

In the example, we can now derive that Azam has travel constraints. Messages flow forward from `wft18!` (through `wft1!` and `arb1`), and from `wft19!` to `wft2`, which then satisfies `wft3!`, deriving an instance of `wft6` – Azam is held by a corrections officer. We derive Azam is detained by messages flowing from `wft6`, and `arb1` (through `ind2`) to `wft9`, which satisfies `wft10!` and derives an instance of `wft7`. The message from `wft7` satisfies the `xor` rule `wft13!`, allowing a negated instance of `wft12` to be derived – Azam is not free. Finally we learn that Azam has travel restrictions since messages from `wft18!` (through `wft14!`), and from `wft12` (through `wft15` and `wft16!`) satisfy `arb2`, allowing a message to be sent asserting `wft20` (through `wft17!`).

4 Conclusion

Inference Graphs 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 IGs may be extended to hybrid reasoning that combines subsumption reasoning with natural deduction over a logic as expressive as FOL.

References

Choi, J., and Shapiro, S. C. 1992. Efficient implementation of non-standard connectives and quantifiers in deductive reasoning systems. In *Proc. HICSS-92*. Los Alamitos, CA: IEEE Computer Society Press. 381–390.

Grosof, 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.

ISO/IEC. 2007. *ISO/IEC 24707 – Common Logic (CL)*. ISO/IEC, Switzerland, First edition.

McKay, D. P., and Shapiro, S. C. 1981. Using active connection graphs for reasoning with recursive rules. In *Proc. IJCAI-81*, 368–374. Los Altos, CA: Morgan Kaufmann.

Motik, B.; Sattler, U.; and Studer, R. 2005. Query answering for OWL-DL with rules. *Web Semantics* 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. 2014a. 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. 138–164.

Schlegel, D. R., and Shapiro, S. C. 2014b. Inference graphs: A new kind of hybrid reasoning system (student abstract). In *Proc. AAAI-14*. Menlo Park, CA: AAAI Press/The MIT Press. In Press.

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., *KR2004*, 565–575. Menlo Park, CA: AAAI Press.