Logical Deduction using the Local Computation Framework

Nic Wilson and Jérôme Mengin
School of Computing and Mathematical Sciences
Oxford Brookes University
Gipsy Lane, Headington, Oxford OX3 0BP, U.K.
{nic.wilson,jmengin}@brookes.ac.uk

A Introduction

Computation in a number of uncertainty formalisms has recently been revolutionized by the notion of *local computation*. [9] and [6] showed how Bayesian probability could be efficiently propagated in a network of variables; this has already lead to sizeable successful applications, as well as a large body of literature on these Bayesian networks and related issues (e.g., the majority of papers in the *Uncertainty in Artificial Intelligence* conferences over the last ten years).

In the late 'Eighties, Glenn Shafer and Prakash Shenoy [14] abstracted these ideas, leading to their *Local Computation framework*. Remarkably, the propagation algorithms of this general framework give rise to efficient computation in a number of spheres of reasoning: as well as Bayesian probability [12], the Local Computation framework can be applied to the calculation of Dempster-Shafer Belief [14, 8], infinitesimal probability functions [17], and Zadeh's Possibility functions.

This paper describes how the framework can be used for the computation of logical deduction.

Local Computation is based on a structural decomposition of knowledge into a network of variables, in which there are two fundamental operations, combination and marginalization. The combination of two pieces of information is another piece of information which gives the combined effect; it is a little like conjunction in classical logic. Marginalization projects a piece of information relating a set of variables, onto a subset of the variables: it gives the impact of the piece of information on the smaller set of variables. Axioms are given which are sufficient for the propagation of these pieces of information in the network. General propagation algorithms can be defined using results in the Bayesian network literature and elsewhere. These algorithms are often efficient, depending, roughly speaking, on topological properties of the network. The reason that Local Computation can be very fast is that the propagation is expressed in terms of much smaller ('local') problems, involving only a small part of the network.

Finite sets of possibilities (or constraints) can be propagated with this framework, and so deduction in a finite propositional calculus can be performed by considering sets of possible worlds; this is implemented in, for example, PULCINELLA [11], and described formally in [13]. However, dealing with sets of possible worlds is often not computationally efficient; it is only very recently [5] that it has been shown how to use Local Computation to directly propagate sets of formulae in a finite propositional calculus.

In the next section we introduce the Local Computation framework. We describe in section C how a logic can be embedded in the framework, given that its semantics verifies certain properties. This is applied to first-order predicate calculus in section D. The last section discusses applications and advantages of this approach.

B Axioms for Local Computation

The primitive objects in the Local Computation framework are an index set χ (often called *the set of variables*) and for each $r \subseteq \chi$ a set $\mathcal{V}_r$, called *the set of r-valuations,* or, *the valuations on r*.The set of valuations $\mathcal{V}$ is defined to be $\bigcup_{r \subseteq \chi} \mathcal{V}_r$. We assume a function $\otimes : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$, called *combination*, such that if $A \in \mathcal{V}_r$ and $B \in \mathcal{V}_s$ then $A \otimes B \in \mathcal{V}_{r \cup s}$, for $r, s \subseteq \chi$. If A and B represent pieces of information then $A \otimes B$ is intended to represent an aggregation of the two pieces of information. We also assume that, for each $r \subseteq \chi$, there is a function $\downarrow r : \bigcup_{s \supseteq r} \mathcal{V}_s \to \mathcal{V}_r$, called *marginalization to r*.

The framework assumes that the following axioms are verified:

Axiom LC1 *(Combination and associativity of combination):* Suppose A, B and C are valuations. Then $A \otimes B = B \otimes A$ and $A \otimes (B \otimes C) = (A \otimes B) \otimes C$.

Axiom LC2 *(Consonance of marginalization):* Suppose A is a t-valuation and $r \subseteq s \subseteq t \subseteq \chi$. Then $(A^{\downarrow s})^{\downarrow r} = A^{\downarrow r}$.

Axiom LC3 *(Distributivity of marginalization over combination)*[1]*:* Suppose A is an r-valuation and B is an s-valuation and $r \subseteq t \subseteq r \cup s \subseteq \chi$. Then $(A \otimes B)^{\downarrow t} = A \otimes B^{\downarrow s \cap t}$.

Let $A_1, \ldots, A_n$ be valuations with, for $i = 1, \ldots, n$, $A_i \in \mathcal{V}_{r_i}$. Many problems can be expressed as calculating $(A_1 \otimes \cdots \otimes A_n)^{\downarrow r_0}$ for some $r_0 \subseteq \chi$; in Bayesian probability this computes the marginal of a joint probability distribution; we will see below how testing the consistency of a set of formulae in propositional or predicate calculus can be expressed in this way.

Elements of $\mathcal{V}_r$ will generally be simpler objects when r is small; for example they may be sets of formulae using only a small number of propositional symbols; also combination and marginalization will generally be much easier on the simpler objects (the computational complexity of these operations is typically exponential in $|r|$). Direct computation of $(A_1 \otimes \cdots \otimes A_n)^{\downarrow r_0}$ will very often be infeasible as it involves a valuation in $\mathcal{V}_r$ where $r = \bigcup_{i=0}^{n} r_i$. It can be seen that axioms **LC1**, **LC2** and **LC3** allow the computation of $(A_1 \otimes \cdots \otimes A_n)^{\downarrow r_0}$ to be broken down into a sequence of combinations and marginalizations, each within some $\mathcal{V}_{r_i}$ (i.e., *local* computations), *if* $\mathcal{H} = \{r_0, \ldots, r_n\}$ *is a hyperforest.* Briefly, $\mathcal{H}$ is said to be a hyperforest if its elements can be ordered as $s_0, \ldots, s_n$ where, for $i = 1, \ldots, n$, there exists $k_i < i$ with $s_i \cap \bigcup_{j<i} s_j \subseteq s_{k_i}$. The complexity of the computation will typically be roughly exponentially related to $\max_i |r_i|$.

If $\mathcal{H}$ is not a hyperforest then we can perform the computations in a hyperforest $\mathcal{G}$ which covers $\mathcal{H}$, i.e., such that for all $r \in \mathcal{H}$, there exists $s \in \mathcal{G}$ with $s \supseteq r$. Finding a good hyperforest cover has been studied in e.g., the graph theory, and statistics literature, see [6].

C Similarity Model Structures

A Similarity Model Structure is defined to be a triple $(\mathcal{M}, \chi, (\approx_r)_{r \subseteq \chi})$, where $\mathcal{M}$ is a set, the elements of which are called *models*, χ is an indexing set, and each $\approx_r$ is an equivalence relation on $\mathcal{M}$. For this paper we will also assume the following monotonicity property: *for* $r \subseteq s \subseteq \chi$, $\approx_r \supseteq \approx_s$ .For $M, N \in \mathcal{M}$, $r \subseteq \chi$, define $M^{\downarrow r}$ to be $\{N : N \approx_r M\}$, and for $A \subseteq \mathcal{M}$, define $A^{\downarrow r}$ to be $\bigcup_{M \in A} M^{\downarrow r}$. If $A^{\downarrow r} = A$ we say that A *is r-closed.*

Embedding Similarity Model Structures in the Local Computation Framework To embed Similarity Model Structures in the Local Computation Framework we need to define r-valuations and the operations Combination and Marginalization. We use the same

[1] **LC3** is slightly stronger than the corresponding axiom **A3** given in [14], (their axiom is **LC3** but with the restriction that $r = t$); it turns out to be occasionally useful to have this stronger axiom.

indexing set χ; for $r \subseteq \chi$, the set of r-valuations $\mathcal{V}_r$ is defined to be the set of r-closed subsets of $\mathcal{M}$. For $A \in \mathcal{V}_s$, we have already defined its result under r-marginalization, $A^{\downarrow r}$. For $A \in \mathcal{V}_r$ and $B \in \mathcal{V}_s$ define $A \otimes B$ to be $A \cap B$ which can be shown to be an element of $\mathcal{V}_{r \cup s}$.

It can easily be seen that axioms **LC1** and **LC2** are automatically satisfied for this embedding of Similarity Model Structures, but **LC3** does not always hold, and is more problematic. Similarity Model Structure $(\mathcal{M}, \chi, (\approx_r)_{r \subseteq \chi})$ is said to satisfy the *Independence Property* if

for any $r, s \subseteq \chi$ and $M, N \in \mathcal{M}$ such that $M \approx_{r \cap s} N$, there exists $L \in \mathcal{M}$ such that $L \approx_r M$ and $L \approx_s N$.

This property may be paraphrased as: *knowing the $\approx_r$-equivalence class A of an unknown model L doesn't tell us anything about its $\approx_s$-equivalence class B, except that B and A are both subsets of the same $\approx_{r \cap s}$-equivalence class (i.e., that containing L).*

The main result of this section is that a Similarity Model Structure satisfies the Independence Property if and only if its embedding in the Local Computation framework satisfies the distributivity axiom **LC3**.

Example: the propositional calculus

Consider the propositional calculus based on set of propositional symbols $\chi = \{P_1, P_2, \ldots\}$. Let $\mathcal{M}$ be the set of truth functions, i.e., functions from χ to $\{\mathrm{T}, \mathrm{F}\}$. For $r \subseteq \chi$, define $\approx_r$ by $M \approx_r N$ iff M and N agree on r, i.e., for all $P_i \in r$, $M(P_i) = N(P_i)$. Each $\approx_r$-equivalence class corresponds to an r-partial model, i.e, a function from r to $\{\mathrm{T}, \mathrm{F}\}$. Hence r-closed sets may be thought of as sets of r-partial models. Using the above embedding, marginalising a set A of s-partial models to $r \subseteq s$ amounts to restricting them to r. If B is a set of t-models then $A \otimes B$ is the set of all $M \otimes N$, with $M \in A$, $N \in B$ such that M and N agree on $r \cap s$, where $M \otimes N$ is the $r \cup s$-valuation which agrees with M on r and with N on s. The fact that such an $r \cup s$-valuation exists implies that the Independence Property is satisfied, so the Local Computation axioms hold.

Suppose, for $i = 1, \ldots, n$, Γ_i is a set of formulae involving only finite number of propositional symbols $r_i \subseteq \chi$. We can check if $\bigcup_i \Gamma_i$ is consistent or not by seeing if $(\bigotimes_i [\Gamma_i])^{\downarrow \emptyset}$ is non-empty (where $[\Gamma_i]$ is the set of truth functions satisfying Γ_i, which is an r_i-closed set). To do this we find a hyperforest cover of $\{r_i : i = 1, \ldots, n\}$ using a standard algorithm, and perform local computations with sets of partial models.

The same approach can be used for a wide range of monotonic logics for which partial models can be defined.

D Application to first-order theorem proving

We consider a set χ of function and predicate symbols. Let $\mathcal{L}$ be the usual set of first-order formulae built using these symbols together with individual variables from a set Var. For $r \subseteq \chi$, let $\mathcal{L}_r$ be the sublanguage of $\mathcal{L}$ comprising formulae, the function and predicate symbols of which are all in r. Let $\mathcal{M}$ be the set of models on χ: each model $M \in \mathcal{M}$ is defined by its universe $\mathcal{U}_M$ and, for each n-ary function symbol $f \in \chi$, an n-ary function on $\mathcal{U}_M$, and for each n-ary predicate symbol P an n-ary relation P on $\mathcal{U}_M$. The set of models of a subset Γ of $\mathcal{L}$ is noted $[\Gamma]$. For each $r \subseteq \chi$ an equivalence relation $\approx_r$ on $\mathcal{M}$ can naturally be defined by: $M \approx_r N$ if and only if M and N have the same universe and give the same interpretation to the symbols of r.

For any $r, s \subseteq \chi$ and $M, N \in \mathcal{M}$ such that $M \approx_{r \cap s} N$, let L be the model of universe $\mathcal{U}_M = \mathcal{U}_N$ which gives to each symbol in $\chi - (s - r)$ the same interpretation as M, and gives to each symbol in $s - r$ the same interpretation as N: clearly $L \approx_r M$ and $L \approx_s N$.

Thus the similarity model structure $(\mathcal{M}, \chi, (\approx_r)_{r \subseteq \chi})$ satisfies the Independence Property, and can be embedded in the Local Computation Framework. Notice that if we consider the set $\mathcal{M}_H$ of Herbrand models of $\mathcal{L}$, $(\mathcal{M}_H, \chi, (\approx_r)_{r \subseteq \chi})$ still satisfies the Independence Property, since the model L constructed above is a Herbrand model if M and N are.

Suppose now that we have a family $(\Gamma_i)_i$ of subsets of $\mathcal{L}$, each Γ_i being more precisely a subset of some $\mathcal{L}_{r_i}$ with $r_i \subseteq \chi$, and that we want to check the satisfiability of $\bigcup_i \Gamma_i$. It can easily be checked that the set of models of $\bigcup_i \Gamma_i$ is empty if and only if $(\bigotimes_i [\Gamma_i])^{\downarrow \emptyset}$ is empty. Performing marginalization and combination on sets of models would often not be practical. However it is possible to work with first-order representations of sets of models whenever it is possible to define a function MARG such that $\mathrm{MARG}(\Gamma, r) \subseteq \mathcal{L}_r$ and $[\mathrm{MARG}(\Gamma, r)] = [\Gamma]^{\downarrow r}$. In this case $(\bigotimes_i [\Gamma_i])^{\downarrow \emptyset} = [\mathrm{MARG}(\cup_i \Gamma_i, \emptyset)]$. The formulas in $\mathrm{MARG}(\cup_i \Gamma_i, \emptyset)$ do not contain any predicate or function symbols (except possibly the equality predicate). More importantly we can look for a hyperforest cover of $\{r_i : i = 1, \ldots, n\}$ using a standard algorithm, and perform local computations of MARG on sets of formulae. In the remainder of this section, we review some existing algorithms to compute the marginalization of sets of formulae.

Marginalization can be computed using algorithms of for example [7, 16, 2, 1]. These algorithms eliminate existentially quantified predicate symbols. More precisely, suppose that ϕ is a formula containing the predicate and function symbols contained in a finite set $r \cup \{P\}$, with $P \notin r$, then it can be shown that $[\phi]^{\downarrow r} = [\exists P.\phi]$. The algorithms mentioned above are designed to compute a first-order formula equivalent to $\exists P.\phi$. The elimination of function symbols is the reverse of Skolemization, and is also performed by these algorithms. The algorithms of [7, 1] always terminate but succeed only in cases where ϕ can be put under disjunctive normal form such that each conjunct contains no positive occurrence of P or no negative occurrence of P. The algorithms of [16, 2] apply to general formulas but do not always halt.

In the case of clauses without the equality predicate, marginalization can be defined using the notion of *production field* [15, 4]: a production field $\mathcal{P}$ is defined by a set $\mathbf{L}$ of literals closed under instantiation; we then write $\mathcal{P} = \langle \mathbf{L} \rangle$. A clause C belongs to $\mathcal{P}$ if every literal in C belongs to $\mathbf{L}$. Given a set of clauses Σ, [4] defines the set of characteristic clauses of Σ with respect to $\mathcal{P}$, noted $\mathrm{Carc}(\Sigma, \mathcal{P})$, to be the set of clauses belonging to $\mathcal{P}$ that are entailed by Σ and that are not subsumed by any other consequence of Σ belonging to $\mathcal{P}$. If we define $\mathbf{L}_r$ to be the set of literals whose predicate and function symbols are all in r, it can be shown that, if $[\Sigma]_H$ is the set of Herbrand models of Σ, $[\Sigma]_H^{\downarrow r} = [\mathrm{Carc}(\Sigma, \langle \mathbf{L}_r \rangle)]_H$. Algorithms to compute Carc can be found in e.g. [10, 4]. Notice that these algorithms will not always terminate, since $\mathrm{Carc}(\Sigma, \langle \mathbf{L}_r \rangle)$ may be infinite.

E Discussion

Although this paper focuses on first-order theorem proving, the same approach can be applied to modal, conditional, probabilistic [18] and possibilistic logics, all of which have important applications in Artificial Intelligence. The approach also applies to certain, restrictive, non-monotonic logics, which are based on simple conditional logics. Apart from theorem proving, a number of problems can be expressed in terms of marginalization to a non-empty set of variables, for example, in power structures, correspondence theory, semantics for Hilbert calculi [2], circumscriptive query-answering [3], abduction [4].

Local Computation methods allow us to break down problems into smaller ones to which classical theorem proving techniques can be efficiently applied; for example, the framework gives strategies for choosing in which order to perform resolutions. It is also possible to get information about the complexity of a particular calculation, by considering the size of the largest set in the hyperforest; in the same way, some formulae which make the computation much worse can be recognised as those that increase the size of this largest set.

Although it is not yet clear how theorem proving algorithms based on Local Computation compare with standard ones, the generality of the framework has a number of benefits. In particular, there are problems where different kinds of information are more suitably expressed using different logical representations, e.g. sets of models, constraints, clausal forms, terminological descriptions etc. A major difficulty of mixing representations is that moving between them tends to be computationally expensive; the Local Computation framework suggests good places for mixing representations (namely in sub-languages corresponding to small intersections between neighbouring sets in the hyperforest).

References

[1] P. Doherty, W. Lukasewicz, and A. Szalas. Computing circumscription revisited: A reduction algorithm. In C. S. Mellish, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1502–1508. I.J.C.A.I., Morgan Kaufmann, 1995.

[2] D.M. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. Report MPI-I-92-231, Max-Planck Institut für Informatik, Saarbrücken, Germany, 1992.

[3] M. Gelfond, H. Przymusińska, and T. Przymusiński. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38(1):75–94, 1989.

[4] K. Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56:301–353, 1992.

[5] J. Kohlas and S. Moral. Propositional information systems. Unpublished report, May 1996.

[6] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to Expert Systems (with discussion). *Journal of the Royal Statistical Society ser. B*, 50(2):157–224, 1988.

[7] V. Lifschitz. Computing circumscription. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 121–127. I.J.C.A.I., Morgan Kaufmann, 1985.

[8] S. Moral and N. Wilson. Markov Chain Monte-Carlo algorithms for the calculation of Dempster-Shafer belief. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 269–274, 1994.

[9] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., 1988.

[10] T. Przymusiński. An algorithm to compute circumscription. *Artificial Intelligence*, 38(1):49–73, 1989.

[11] A. Saffiotti and E. Umkehrer. PULCINELLA: A general tool for propagating uncertainty in valuation networks. In *Uncertainty in Artificial Intelligence: Proceedings of the 7th Conference*, pages 323–331. Morgan Kaufmann, 1991.

[12] G. Shafer and P. P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–352, 1990.

[13] P. P. Shenoy. Consistency in valuation-based systems. *ORSA Journal on Computing*, 6(3):281–291, 1994.

[14] P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In G. Shafer and J. Pearl, editors, *Readings in Uncertain Reasoning*, pages 575–610. Morgan Kaufmann, 1990.

[15] P. Siegel. *Représentation et Utilisation de la Connaissance En Calcul Propositionnel.* Thèse d'état, Université d'Aix-Marseille II, Luminy, 1987.

[16] A. Szalas. On correspondence between modal and classical logic: Automated approach. Technical Report MPI-I-92-209, Max-Planck-Institut für Informatik, Saarbrücken, 1992.

[17] N. Wilson. Extended Probability. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96)*, pages 667–771. John Wiley and Sons Ltd, 1996.

[18] N.Wilson and S. Moral. A logical view of probability. In *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI-94)*, pages 386–390, 1994.