DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-9913

KNOWLEDGE SHARING BETWEEN A PROBABILISTIC LOGIC AND BAYESIAN BALIEF NETWORK

FLÁVIO S. CORRÊA DA SILVA
JAUME AGUSTÍ
ROBERTO CÁSSIO DE ARAÚJO
ANA CRISTINA V. DE MELO

Novembro de 1999

Knowledge Sharing Between a Probabilistic Logic and Bayesian Belief Networks *

Flávio S. Corrêa da Silva†
Instituto de Matemática e Estatística
Universidade de São Paulo, São Paulo, Brazil
fcs@ime.usp.br

Roberto Cássio de Araújo‡
Instituto de Matemática e Estatística
Universidade de São Paulo, São Paulo, Brazil
bob@ime.usp.br

Jaume Agustí
Institut d'Investigació en Intel.ligencia Artificial
Campus UAB 08193 Bellaterra, Barcelona, España
agusti@iiia.csic.es

Ana Cristina V. de Melo
Instituto de Matemática e Estatística
Universidade de São Paulo, São Paulo, Brazil
acvm@ime.usp.br

Abstract

Knowledge sharing and reuse has been considered an important issue for cost-effective use of knowledge-based systems, especially after the development and popularisation of object-based technologies and Internet-based decentralised computing. Up until now, the majority of research tackling this issue has been founded on the assumption that there can be a common domain description – a *shared ontology* – which suits everyone with an interest in the knowledge. Unfortunately, getting an agreed ontology for a collection of systems can be a difficult problem and, even when this problem can be solved, it may not be enough for effective knowledge sharing, since the way we represent knowledge is intimately linked to the inferences we expect to perform with it. A nice example of this situation can be found in systems for reasoning under uncertainty, where even if we do have a shared ontology for the problem being solved we must still establish semantic links between the inferences performed within each system to actually have knowledge being shared and reused.

In the present paper we study a significant instance of this problem. We introduce a simple yet effective logical system for interval-based probabilistic reasoning, then discuss the difficulties to have this system being able to consult bayesian belief networks to complete its own inferences, and how these difficulties can be remedied. Finally, some simple motivating examples are introduced to suggest "practical" applications for this knowledge sharing scenario.

1 Introduction

Mainstream research in knowledge sharing among knowledge-based systems concentrates on mapping different notations to describe problems and their solving procedures, while making the assumption that the inference mechanisms employed in these systems are compatible [Sub, NG, Gra, PHG+99, Sha97, UG96].

There is good reason for making this assumption because it is difficult without it to guarantee that the meaning of knowledge expressed in one system is preserved when used by another system. However, this is a rather strong assumption to be made when one considers the (re)use of previously existing systems for knowledge sharing.

Alternatively, we have considered within the research project **DECaFf-KB** – Distributed Environment for Cooperation Among Formalisms for Knowledge-based Systems – the situation in which the relationships between inferences performed in different systems are not evident (or even do not exist at all) [CdSVR98, CdSVA+99]. Consider for example two systems for uncertain reasoning: the first one prepared to compute probability intervals for first-order sentences via a resolution-based inference procedure, and the second one a bayesian belief network that can compute the probability of a statement given an arbitrary set of assumptions. Both systems share a lot of semantic information, however the differences between the internal procedures to construct answers to queries create difficulties to establish the connections between answers from each system.

We suggest that in order to establish proper connection between knowledge and information conveyed by such systems an analysis must be carried through, based on three levels of understanding of the connection:

1. **conceptual level:** in which the semantic links between information and knowledge taken into account within each system are established;
2. **architectural level:** in which the actual procedures to implement these links are designed – e.g. via a knowledge broker; and
3. **implementation level:** in which specific strategies to implement the procedures designed above are developed.

To make this article self-contained, in section 2 we recall some basic definitions of probability theory that will be needed in the following sections. In order to make this discussion more direct and effective, we specialise it to particular systems: in section 3 we introduce a simple logical system to calculate probability bounds for first-order statements based on a resolution-style inference procedure, and in section 4 we present a simplified model for bayesian belief networks, that are expected nevertheless to contain the main features and properties of those formalisms. In section 5 we detail the three-level-based

*Work sponsored by the Consortium British Council/CAPES (Brasil), Grant no. 070/98.
†Partially sponsored by FAPESP, Grant no. 93/0603-01
‡Partially sponsored by CAPES PhD scholarship

analysis proposed above, and illustrate how the first two levels characterise a solution for knowledge sharing between the logical system and the bayesian belief network. In sections 6 and 7 we present two illustrative examples, to give a flavour of how this specific knowledge sharing setting can be used for practical applications. Finally, in section 8 we present some final remarks and suggestions for future work.

2 Preliminary Definitions

Given a finite set D, an *algebra* χ_D on D is a set of subsets of D such that (i) $D \in \chi_D$; (ii) $A \in \chi_D \Rightarrow \neg A \in \chi_D$; (iii) $A, B \in \chi_D \Rightarrow A \cup B \in \chi_D$.

A subset of D is called an *event* on D. Events belonging to χ_D are called *measurable events*.

A *probability measure* on χ_D is a function $\mathcal{P} : \chi_D \to [0,1]$ such that (i) $\mathcal{P}(D) = 1$ (total probability); (ii) $A \cap B = \{\} \Rightarrow \mathcal{P}(A \cup B) = \mathcal{P}(A) + \mathcal{P}(B)$ (finite additivity).

Given two measurable events $A, B \in \chi_D$, the *conditional probability* $\mathcal{P}(A|B)$ is defined as:

$$\mathcal{P}(A|B) = \begin{cases} \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)}, & \mathcal{P}(B) \neq 0 \\ 0, & \mathcal{P}(B) = 0 \end{cases}$$

Two measurable events A, B are called *independent* iff $\mathcal{P}(A|B) = \mathcal{P}(A)$ which, as a corollary, gives that $\mathcal{P}(A \cap B) = \mathcal{P}(A) \times \mathcal{P}(B)$.

The set D can be partitioned into m subsets $D_1, ..., D_m$ such that (i) $D_i \cap D_j = \{\}, i, j = 1, ..., m, i \neq j$; and (ii) $D = \bigcup_1^m D_i$.

We can have independent probability measures $\mathcal{P}_i$ for algebras χ_{Di} (probability measures for two algebras χ_1 and χ_2 are *independent* iff each event $X_i \in \chi_1$ is independent of every event $Y_j \in \chi_2$ and vice-versa) for each set D_i. If we assume that all events in each χ_{Di} are *pairwise independent*, we can extend measures to cartesian products of the sets D_i of a partition of D: the cartesian product of a collection of bases of algebras of elements $D_1, ..., D_m$ of a partition of D is the basis $\vec{\chi}'$ of an algebra of the cartesian product of the sets $D_1, ..., D_m$, and the measure $\vec{\mathcal{P}}$ on the corresponding algebra $\vec{\chi}$ is defined as:

- $\vec{\mathcal{P}} : \vec{\chi} \to [0,1]$.
- $\vec{\mathcal{P}}(\vec{A}) = \prod_1^m \mathcal{P}_i(A_i)$.

  where $\vec{A} = [A_1, ..., A_m], A_i \in \chi_{Di}, \mathcal{P}_i$ is the probability measure defined on χ_{Di}.

Probability measures can be extended to *non-measurable events*, i.e. sets $A_j \in 2^D \setminus \chi_D$. Given D, χ_D and $\mathcal{P}$, we define the *inner* and *outer extensions to* $\mathcal{P}$ ($\mathcal{P}_*$ and $\mathcal{P}^*$, respectively) as [Dud89]:

- $\mathcal{P}_*, \mathcal{P}^* : 2^D \to [0,1]$
- $$\begin{aligned} \mathcal{P}_*(A) &= sup\{\mathcal{P}(X) : X \subseteq A, X \in \chi_D\} \\ &= \mathcal{P}(\bigcup X : X \subseteq A, X \in \chi_D') \end{aligned}$$
- $$\begin{aligned} \mathcal{P}^*(A) &= inf\{\mathcal{P}(X) : A \subseteq X, X \in \chi_D\} \\ &= \mathcal{P}(\bigcup X : X \cap A \neq \{\}, X \in \chi_D') \end{aligned}$$

Inner and outer measures can be extended to cartesian products of a partition of D. Given a collection $D_1, ..., D_m$ of elements of a partition of D, and given also the algebras χ_{Di} and probability measures $\mathcal{P}_i$ of each $D_i, i = 1, ..., m$, we have:

- $\mathcal{P}_{m*}, \mathcal{P}_m^* : 2^{D_1 \times ... \times D_m} \to [0,1]$
- $$\begin{aligned} \mathcal{P}_{m*}(A) &= sup\{\mathcal{P}_m(X) : X \subseteq A, X \in \vec{\chi}\} \\ &= \mathcal{P}_m(\bigcup X : X \subseteq A, X \in \vec{\chi}') \end{aligned}$$
- $$\begin{aligned} \mathcal{P}_m^*(A) &= inf\{\mathcal{P}_m(X) : A \subseteq X, X \in \vec{\chi}\} \\ &= \mathcal{P}_m(\bigcup X : X \cap A \neq \{\}, X \in \vec{\chi}') \end{aligned}$$

The measures $\mathcal{P}_{m*}$ and $\mathcal{P}_m^*$ can be regarded as approximations from below and from above to the probabilities of non-measurable events: if we could evaluate the probability $\mathcal{P}_m(A)$, then we would have that $\mathcal{P}_{m*}(A) \leq \mathcal{P}_m(A) \leq \mathcal{P}_m^*(A)$. Indeed, for measurable events we have that $\mathcal{P}_{m*}(A) = \mathcal{P}_m(A) = \mathcal{P}_m^*(A)$.

3 A Simple System for Probabilistic Reasoning

In this section we introduce a simple logical system for probabilistic reasoning, that employs a resolution-style SLDNF deductive system for clausal theories and can be implemented as a pure PROLOG meta-interpreter [Apt94, SS86]. This system is a simplified version of what was presented in [CdSRH94] and implemented in [CdS92].

The system we present here allows the deduction of degrees of belief apportioned by a rational agent to statements represented as normal clauses and queries. These degrees of belief are represented as probabilities on possible worlds, that can be loosely understood as hypothetical scenarios capable of accommodating the problem being solved, following the foundations for well-known formalisms like Incidence Calculus [Bun85, CdSB91], Probabilistic Logic [Nil86] and the Dempster-Shafer Theory of Evidence [Sha76, FH89].

Following [Kun89, Tur89, CdS92], let us consider the special class of clausal theories called *function-free normal programs* under restrictions of *call consistency*, *strictness with respect to queries*, and *allowedness*, having as inference procedure the *SLDNF Resolution Rule*, that is known to be *sound* and *complete* with respect to the model of the *Clark's completion* of programs in the language.

This defines a rich subset of first-order logic with a computationally efficient inference procedure and a formally specified declarative semantics. A program **P** is a theory consisting of a collection of normal clauses `H :- Q1, ..., Qn` and unit clauses $\mathtt{H}_i$, and proofs will be triggered by a query `Q1', ..., Qm'`, where `H`, $\mathtt{H}_i$ are atomic predicates and $\mathtt{Q}_j$, $\mathtt{Q}_k$' are atomic predicates or negations of atomic predicates.

A set of possible worlds is a collection of *worlds* (or *states*, or *interpretations*), each of them assigning different truth-values to the formulae in our language. Intuitively, a possible world should be viewed as a conceivable hypothetical scenario upon which we can construct our reasoning.

Given a program **P** and a set of possible worlds $\Omega = \{\omega_1, ...\}$, a *rigid formula* is a formula which is always assigned the same truth-value in all possible worlds.

We assume in our language that, given a program $\mathbf{P}$, each possible world ω_i can assign a different truth-value to the set of unit clauses in $\mathbf{P}$. We assume that the normal clauses occurring in $\mathbf{P}$ are rigid.

The degree of belief attached to a query is the probability of selecting a possible world at random and the query being true in that world. One possible way of evaluating the degree of belief in a query could then be to trigger repeatedly a program for each $\omega_i \in \Omega$, to identify the subset of Ω in which the query is true, and then, based on a probability measure defined for an algebra on Ω, to evaluate the probability of the query being true (or at least the inner and outer approximations of it).

This procedure becomes computationally intractable as the size of Ω grows. Alternatively, following [NS92], approximate solutions can be obtained for the inner and outer approximations for the probability of a query $\psi =$ `Q1', ..., Qm'` (henceforth queries will be denoted with greek letters), denoted as $\hat{\mathcal{P}}_*(\psi)$ and $\hat{\mathcal{P}}^*(\psi)$ respectively:

$$\texttt{H :- Q} \quad \Rightarrow \quad \begin{array}{l} \hat{\mathcal{P}}_*(\texttt{H}) = \hat{\mathcal{P}}_*(\texttt{Q}) \\ \hat{\mathcal{P}}^*(\texttt{H}) = \hat{\mathcal{P}}^*(\texttt{Q}) \end{array}$$

$$\texttt{H :- }\psi,\varphi \quad \Rightarrow \quad \begin{array}{l} \hat{\mathcal{P}}_*(\texttt{H}) = \max\{0, \hat{\mathcal{P}}_*(\psi) + \hat{\mathcal{P}}_*(\varphi) - 1\} \\ \hat{\mathcal{P}}^*(\texttt{H}) = \min\{\hat{\mathcal{P}}^*(\psi), \hat{\mathcal{P}}^*(\varphi)\} \end{array}$$

$$\begin{array}{l} \texttt{H :- }\psi \\ \texttt{H :- }\varphi \end{array} \quad \Rightarrow \quad \begin{array}{l} \hat{\mathcal{P}}_*(\texttt{H}) = \max\{\hat{\mathcal{P}}_*(\psi), \hat{\mathcal{P}}_*(\varphi)\} \\ \hat{\mathcal{P}}^*(\texttt{H}) = \min\{1, \hat{\mathcal{P}}^*(\psi) + \hat{\mathcal{P}}^*(\varphi)\} \end{array}$$

$$\texttt{H :- \textbackslash+ Q} \quad \Rightarrow \quad \begin{array}{l} \hat{\mathcal{P}}_*(\texttt{H}) = 1 - \hat{\mathcal{P}}^*(\texttt{Q}) \\ \hat{\mathcal{P}}^*(\texttt{H}) = 1 - \hat{\mathcal{P}}_*(\texttt{Q}) \end{array}$$

In a program $\mathbf{P}$, the proof of a query $\texttt{Q} = \texttt{Q}'_1, \ldots, \texttt{Q}'_m$ is expressed by the construction of an interval $[\hat{\mathcal{P}}_*(\texttt{Q}), \hat{\mathcal{P}}^*(\texttt{Q})]$ where $\hat{\mathcal{P}}_*(\texttt{Q})$ and $\hat{\mathcal{P}}^*(\texttt{Q})$ are calculated from the intervals associated to $\texttt{Q}'_1, \ldots, \texttt{Q}'_m$. If $\texttt{Q}'_i$ is a unit clause, the values $\hat{\mathcal{P}}_*(\texttt{Q}'_i)$ and $\hat{\mathcal{P}}^*(\texttt{Q}'_i)$ are explicitly given (if they're not given, they're assumed to be 0.0). If $\texttt{Q}'_i$ is a normal clause, its interval is recursively calculated using the clauses included in the program.

Example 3.1 *For example, considering the following program:*

```
p(X)  :-  q(X), \+ r(X)
q(X)  :-  s(X), u(X)
```

And the following intervals associated to unit clauses:

$$\begin{array}{lcl} \texttt{r(b)} & = & [0.0, 0.0] \\ \texttt{r(c)} & = & [0.8, 0.9] \\ \texttt{s(a)} & = & [0.1, 0.3] \\ \texttt{s(b)} & = & [0.2, 0.4] \\ \texttt{s(c)} & = & [0.6, 0.7] \\ \texttt{u(a)} & = & [0.3, 0.4] \\ \texttt{u(b)} & = & [0.2, 0.5] \\ \texttt{u(c)} & = & [0.4, 0.7] \end{array}$$

The proof of `p(X)` *is obtained by the combination of the proofs of* `p(a)`, `p(b)` *and* `p(c)` *(the values to which* `X` *unifies). The interval associated to* `p(c)` *is the interval associated to the query* `s(c),u(c),\+r(c)`. *Using the rules above, this interval is bound to* $[max\{0, (0.6 + 0.4 - 1) + 0.1 - 1\}, min\{0.7, 0.7, 0.2\}] = [0, 0.2]$. *The proofs of* `p(a)` *and* `p(b)` *are obtained in the same way, with corresponding intervals set to* $[0, 0.3]$ *and* $[0, 0.4]$. *Notice that the interval associated to* `r(a)` *is* $[0, 0]$, *since there's no unit clause defining it. To complete the proof of* `p(X)`, *the resulting intervals,* $[0, 0.3]$, $[0, 0.4]$ *and* $[0, 0.2]$, *are combined and the interval for the belief on* `p(X)` *is* $[0, 0.9]$. *The proof tree and corresponding interval for the query* `p(X)` *are as presented in figure 1.*

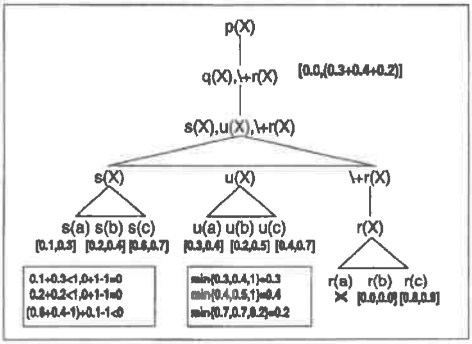


Figure 1: Proof Tree and Interval for `p(X)`

4 Bayesian Belief Networks

A *Bayesian belief network* is a directed acyclic graph in which nodes stand for propositional variables and an edge from node V_i to node V_j means that the beliefs about truth values assigned to the variable represented by the node V_i influences the ones of the variable represented by the node V_j.

For abbreviation, we will use v_i to denote the proposition $V_i = true$, and $\neg v_i$ to denote the proposition $V_i = false$. Given a set V of variables, a conjunction c_V of assignments to these variables is called a *configuration* of V. More formally, following [vdG96], a bayesian belief network is a tuple $B = (G, P)$ where:

- $G = (VG, AG)$ is an acyclic digraph with nodes $VG = \{V_1, ..., V_n\}$, $n \geq 1$, and edges AG;
- $P = \{p_{V_i} | V_i \in VG\}$ is a set of real-valued non-negative functions

  $$p_{V_i} : \{C_{V_i}\} \times \{C_{\rho_G(V_i)}\} \longrightarrow [0, 1]$$

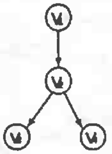
  called (conditional) probability assessment functions, such that for each configuration $c_{\rho_G(V_i)}$ of the set $\rho_G(V_i)$ of (immediate) predecessors of node V_i in G, we have that $p_{V_i}(\neg v_i | c_{\rho_G(V_i)}) = 1 - p_{V_i}(v_i | c_{\rho_G(V_i)})$.

The main application of belief networks is in representation and manipulation of knowledge via the calculation of how evidences support hypotheses, i.e., what is the probability of a hypothesis being true considering a specified collection of evidences. This is done via probability propagation along the network via the definition of conditional probability (cf. section 2) and specific values for probabilities given in the nodes and edges of the network.

Example 4.1 *The belief network shown in figure 2, that will be called PoorBN, represents a simple way to diagnose whether a person is poor based on their salary and household status.*

There are four variables represented in the network:

V_1 : person is poor;
V_2 : person rents a house;
V_3 : person owns a house;
V_4 : person has low salary.



$p(v_1) = 0.4$

$p(v_2|v_1) = 0.3$
$p(v_2|\neg v_1) = 0.8$

$p(v_3|v_2) = 0.6$
$p(v_3|\neg v_2) = 0.5$

$p(v_4|v_2) = 0.1$
$p(v_4|\neg v_2) = 0.6$

Considering the topology and the probabilities of the network, and observing that two disconnected variables are (conditionally) independent, it is possible to calculate every conditional probability involving these four variables. For example:

- $p(v_2) = p(v_2|v_1)p(v_1) + p(v_2|\neg v_1)p(\neg v_1) = 0.6$
- $p(\neg v_4) = p(\neg v_4|v_2)p(v_2) + p(\neg v_4|\neg v_2)p(\neg v_2) = 0.7$
- $p(v_1|v_4) = \frac{p(v_1)p(v_4|v_1)}{p(v_4)} = \frac{0.4\times(p(v_4|v_2)p(v_2|v_1)+p(v_4|\neg v_2)p(\neg v_2|v_1))}{0.3} = 0.64$
- $p(v_1, \neg v_2|v_4) = \frac{p(v_1,\neg v_2,v_4)}{p(v_4)} = \frac{p(v_4|\neg v_2)p(\neg v_2|v_1)p(v_1)}{p(v_4)} = 0.56$

Figure 2: The belief network PoorBN

Many interesting algorithms for efficient propagation of probabilities along belief networks have been developed. The interested reader is addressed e.g. to [Pea88] for a compilation of the major results in this field of research.

5 An Architecture for Knowledge Sharing

If we compare the formalisms for uncertain reasoning presented in the previous sections we observe they contain structural similarities, as presented in table 1.

	Logical System	Belief Network
atomic information	predicates and ground terms	boolean variables
relations	normal clauses	directed graph edges
degrees of belief	probability intervals	probability values
a priori information	intervals for unit clauses	probabilities on the network
output	intervals for queries	probabilities given evidential assumptions

Table 1: Structural Similarities Between Logical System and Belief Network for Uncertain Reasoning

Both systems were created to propagate their (probability-based) representations of degrees of belief along pieces of knowledge, so it is not surprising that they contain operators to perform conceptually equivalent tasks, as indicated by their structural similarities. However, the procedures employed within each system to effectively propagate degrees of belief is very dissimilar, as presented in table 2.

	Logical System	Belief Network
basis for propagation	logical implication	probabilistic conditioning
inference system	resolution	conditioning
representation of degrees of belief	interval based	single valued

Table 2: Operational Differences Between Logical System and Belief Network for Uncertain Reasoning

For both systems to share knowledge, we must establish the correspondences between their capabilities at the **conceptual level**, namely by connecting their semantics.

A broker is a module in this framework corresponding to the **architectural level**, containing the descriptions of the systems capabilities and their correspondences. The task of the broker is to accept queries, to format them accordingly and to provide the means to obtain answers based on capabilities correspondences.

In the present article we concentrate on these first two levels of analysis – especially the conceptual level – to construct correspondences between capabilities of knowledge based systems.

For instance, if we look again at example 3.1, we observe that `r(a)` and `r(b)` are indistinguishable, even though we have explicit information about the evaluated degree of belief for `r(b)` whereas the corresponding evaluation for `r(a)` is obtained by default given the *lack* of information about that unit clause.

Let us say we have a bayesian belief network, which can provide information about the probability interval related to `r(a)`. In this section we are going to present how and under what conditions the connection between queries from the logical system and answers from the belief network can be established.

The connection must be established as a functional transformation from atomic queries from the logical system to specified conditional probabilities in the belief network, and then back from the (single valued) probability estimations from the belief network to the (interval based) probability values in the logical system. The interpretation of queries and answers in each system must be compatible with each other, hence the connection must be established at the semantic level. We call this semantic view of the connection the **conceptual level** of the knowledge sharing problem (figure 3).

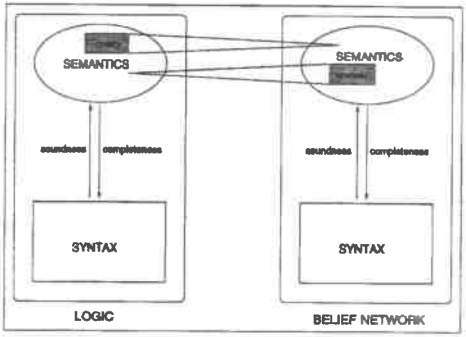


Figure 3: Connection Between Two KBSs: Conceptual Level

Once we have this connection established, we must provide the systems with an operational counterpart to their query and answer passing procedures. This must be designed at the syntactical level, in order to be liable to be implemented (figure 4), e.g. by means of a brokering system. A broker is a module in this architecture that contains the description of the systems that share knowledge. The task of the broker is to accept queries, format them accordingly and convey them to appropriate auxiliary systems based on the descriptions of the systems. For a detailed presentation of a brokering system, the reader is referred to [RACCdSV00]. This syntactical view of the connection gives rise to the **architectural level of the knowledge sharing problem.**

Finally, the actual implementation of the communication procedures must be designed and made, giving rise to the **implementation level** of the knowledge sharing problem (figure 5). A more detailed presentation of this level of the knowledge sharing problem will be left for future articles.

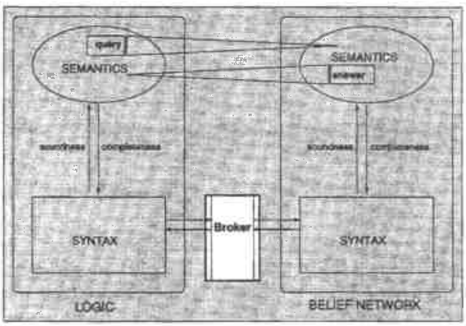


Figure 4: Connection Between Two KBSs: Architectural Level

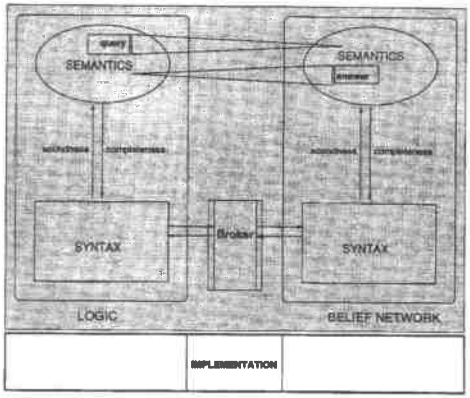


Figure 5: Connection Between Two KBSs: Implementation Level

The desired probability interval to be associated to `r(a)` – that is missing in the original program – characterises the inner and outer approximations for the probability of selecting at random a possible world and `r(a)` being true in that world, i.e. two nested events belonging to the algebra $\chi_{D_{r(a)}}$, where $D_{r(a)}$ is an appropriate subset of possible worlds "relevant" to `r(a)`, for which a probability measure is known. Since `r(a)` does not occur in the original program, it is assumed *not* to be true in any possible world initially employed to interpret the program. As a consequence, if auxiliary information is obtained from a bayesian belief network to evaluate the probability of `r(a)` it must be associated to a completely fresh subset of possible worlds, not used before to interpret the program.

We must (manually) link `r(a)` to a pair of events from the belief network $(v_1, ..., v_m | e_1, ..., e_r)$ and $(v_1, ..., v_m, v_{m+1}, ..., v_n | e_1, ..., e_r)$. The set of possible worlds selected by the evidences $e_1, ..., e_r$ must be disjoint with the set of possible worlds initially used to interpret the program, and the algebra and probability measure based on this set of possible worlds must be independent from the other algebras and measures used in the initial interpretation of the program, otherwise the semantics of the probability intervals generated by the belief network will be corrupted, because of duplicated information generated by the two systems.

Once we link `r(a)` and the pair of events from the belief network, we extend the set of possible worlds, algebras and probability measures considered initially for the logic program with the set of possible worlds relevant to $e_1, ..., e_r$, and evaluations provenient from the bayesian belief network can be employed within proofs in the logical system as if they belonged to the original program.

In order to clarify how we can use this semantic extension of a logical system with knowledge shared by a belief network, in the following sections we present two illustrative examples. In the first one, we use a belief network to complete a proof generated by a logical program. In the second one, we build a simple logical program that constructs its proofs based solely on information generated by three separate belief networks.

6 Consulting One Bayesian Belief Network

Suppose that a university wants to create a knowledge based system to decide whether a student shall receive free lodging in the university student accommodation facilities. The criteria upon which the decision is based considers the following information about the student:

financial situation : whether the student has means to pay for external accommodation;

academic performance : whether the student has good academic records;

medical condition : whether the student medical status can be considered potentially harmful to other students in the lodging. The diagnostic of the medical situation of a student is based on the student's vaccination records and on contagious diseases from which they are diagnosed to suffer.

A simple logical theory to decide whether a student is a good candidate to receive free lodging by the university can be given by:

```
deserveLodging(X)    :-  poor(X),
                         goodStudent(X),
                         healthy(X).
healthy(X)           :-  vaccinated(X),
                         \+disease(X).
```

`poor(John)`	=	$[0.7, 0.7]$
`poor(Paul)`	=	$[0.6, 0.9]$
`disease(John)`	=	$[0.5, 0.7]$
`vaccinated(John)`	=	$[0.5, 0.5]$
`vaccinated(Paul)`	=	$[0.7, 0.9]$

There are no explicit means to calculate the belief on `goodStudent(X)`, which will therefore assumed to be improbable by default. Alternatively, we could "build" the belief interval based on information available in the university students records, using knowledge shared by the belief network given below:

Example 6.1 *The belief network in figure 6, called GoodStudentBN, can be used to decide whether a student is a "good" student based on their academic information: exam results and lecture attendance information.*

Since we do not have specific information about the academic performance of John or Paul, we will replace both queries `goodStudent(John)` *and* `goodStudent(Paul)` *by a corresponding concept from the belief network, e.g. we can associate to both unit clauses the interval* $[p(a,b|c), p(a|c)]$.

The link from `goodStudent(John)` *and* `goodStudent(Paul)` *to the interval* $[p(a,b|c), p(a|c)]$ *is based on our interpretation of the systems (conceptual level) and it is included in the broker description of the connection (architectural level).*

The proof tree for `deserveLodging(John)`, *in this case, including a representation of the knowledge shared by the belief network, is given in figure 7.*

Description of the variables (nodes) of the network:

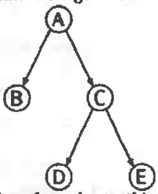
A : student is good

B : student goes to every lecture

C : student has good exam results

D : student has good university admission exam results

E : student has good exam results in university courses



$p(a) = 0.5$

$p(b|a) = 0.8$
$p(b|\neg a) = 0.4$

$p(c|a) = 0.9$
$p(c|\neg a) = 0.1$

$p(d|c) = 0.9$
$p(d|\neg c) = 0.5$

$p(e|c) = 0.9$
$p(e|\neg c) = 0.4$

Examples of queries to this belief network:

- $p(c) = p(c|a)p(a) + p(c|\neg a)p(\neg a) = 0.5$
- $p(a|c) = \frac{p(a)p(c|a)}{p(c)} = 0.9$
- $p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b|a)(p(c|a))}{p(c)} = 0.72$

Figure 6: Belief network to model the "quality" of a student

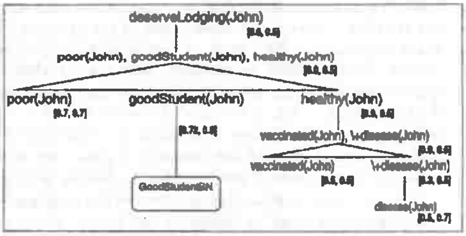


Figure 7: Proof tree for `deserveLodging(John)` with knowledge sharing

The role of the broker in this computation is to accept a query from the logical system and discover another system that can "prove" it. Since we have linked the formula `goodStudent(John)` *and a pair of queries to a belief network, the broker asks the belief network for the two values, builds an interval, and returns the interval to the logical system.*

7 Consulting Multiple Bayesian Belief Networks

Let us consider now a situation in which that several individual belief networks are available and we want to build a system that uses the capabilities of these networks to solve a complex problem. In example 7.1, three belief networks are used by a simple logical theory to decide whether a student shall receive free lodging by the university.

Example 7.1 *Let us assume we have access to the belief networks shown in examples 4.1 and 6.1 and, addition-*

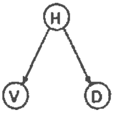
ally, to the network, called HealthBN, presented in figure 8:

Description of the variables (nodes) of the network:

H : person is healthy;

V : person is vaccinated against contagious diseases;

D : person is infected with contagious disease.



$p(h) = 0.8$

$p(v|h) = 0.9$
$p(v|\neg h) = 0.4$

$p(d|h) = 0.1$
$p(d|\neg h) = 0.6$

Examples of queries to this belief network:

- $p(d) = p(d|h)p(h) + p(d|\neg h)p(\neg h) = 0.2$
- $p(h|\neg d) = \frac{p(h)p(\neg d|h)}{p(\neg d)} = 0.9$
- $p(h, v|\neg d) = \frac{p(h,v,\neg d)}{p(\neg d)} = \frac{p(v|h)p(\neg d|h)p(h)}{p(\neg d)} = 0.81$

Figure 8: Belief network to model the health of a person

Based on these individual networks we can build a very simple logical theory to decide whether a student should receive free lodging by the university:

```
deserveLodging(X)  :-  poor(X),
                       goodStudent(X),
                       healthy(X).
```

Let us say we have engineered the relations between the needed knowledge and what can be conveyed by the belief networks. This is presented in table 3.

logic formula	network	associated interval
`poor(John)`	PoorBN	$[p(v_1, \neg v_2 \| v_4), p(v_1 \| v_4)]$
`poor(Paul)`	PoorBN	$[p(v_1, \neg v_2 \| v_4), p(v_1 \| v_4)]$
`goodStudent(John)`	GoodStudentBN	$[p(a, b \| c), p(a \| c)]$
`goodStudent(Paul)`	GoodStudentBN	$[p(a, b \| c), p(a \| c)]$
`healthy(John)`	HealthBN	$[p(h, v \| \neg d), p(h \| \neg d)]$
`healthy(Paul)`	HealthBN	$[p(h, v \| \neg d), p(h \| \neg d)]$

Table 3: Links between formulas and probability intervals

The broker in this example contains the information shown in table 3 and, for each query of the logical system, asks the appropriate belief network for the corresponding pair of values, builds an interval, and returns the interval to the logical system.

Using these links, the proof tree for the formula `deserveLodging(Paul)` *is presented in the figure 9. Notice that, although in this simple example the links associate the same intervals to Paul and John, this will not necessarily be the case in all problems.*

8 Conclusion

In the present article we have discussed a specific non-trivial knowledge sharing scenario, in which semantic

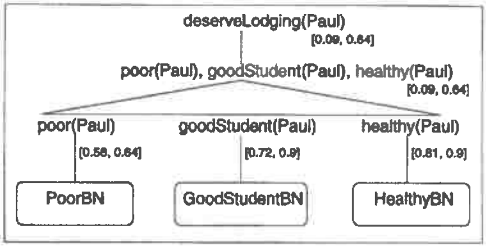


Figure 9: Proof tree for `deserveLodging(Paul)` with knowledge sharing

information about how inferences are performed within participating systems is essential for proper knowledge sharing to occur. Namely, we have considered how bayesian belief networks can offer the result of their inferences to a resolution-based logical system to assign probability intervals to clausal queries.

The results presented here take into account only a simple situation, in which information provided by the networks is completely missing in the logical system. Also, it is not considered how the connection between systems would have to be made so that logical theories could also provide useful information to belief networks. These two scenarios are being taken into account at the moment under the auspices of the project **DECaFf-KB** – Distributed Environment for Cooperation Among Formalisms for Knowledge-based Systems – and shall be presented in future papers.

Nevertheless, some interesting points should be highlighted:

- the approach described here can be regarded as a useful problem solving strategy when the problem does not belong to a single domain, and a good solution for it is achieved by the combination of the results obtained in its multiple domains;
- this approach can also be seen as an effective software (re)use strategy, where a brokering system can link information available in existing systems to specific queries of new ones;
- the possibility of use a logical system as an intelligent "interface" to combine information generated by a collection of belief networks;

References

[Apt94] K. R. Apt. Logic Programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 493–574. Elsevier – MIT Press, 1994.

[Bun85] A. Bundy. Incidence Calculus: a Mechanism for Probabilistic Reasoning. *Journal of Automated Reasoning*, 1(3):263–284, 1985.

[CdS92] F. S. Correa da Silva. *Automated Reasoning with Uncertainties*. PhD thesis, University of Edinburgh, Department of Artificial Intelligence, 1992.

[CdSB91] F. S. Correa da Silva and A. Bundy. A Rational Reconstruction of Incidence Calculus. Technical Report 517, University of Edinburgh, Department of Artificial Intelligence, 1991.

[CdSRH94] F. S. Correa da Silva, D. S. Robertson, and J. Hesketh. Automated Reasoning with Uncertainties. In M. Masuch and L. Pawlos, editors, *Knowledge*

Representation and Uncertainty - ch. 5 - a preliminary version was presented at the APLOC - Applied Logic Conference (Logic at Work), 1992. Springer Verlag LNAI 808, 1994.

[CdSVA+99] F. S. Correa da Silva, W. W. Vasconcelos, J. Agusti, D. S. Robertson, and A. C. V. Melo. Why Ontologies are not Enough for Knowledge Sharing. In Springer-Verlag, editor, 12^{th} *International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (Lecture Notes in Artificial Intelligence, v. 1611)*, pages 520–529, Cairo (Egypt), 1999.

[CdSVR98] F. S. Correa da Silva, W. W. Vasconcelos, and D. S. Robertson. Cooperation Between Knowledge Based Systems. In 4^{th} *World Congress on Expert Systems*, pages 819–825, Mexico, 1998.

[Dud89] R. M. Dudley. *Real Analysis and Probability.* Wadsworth & Brooks/Cole, 1989.

[FH89] R. Fagin and J. Y. Halpern. Uncertainty, Belief, and Probability. Technical Report RJ-6191, IBM Research Report, 1989.

[Gra] P. Gray. KRAFT – Knowledge Reuse and Fusion/Transformation. http:// www.csd.abdn.ac.uk/ apreece/ Research/ KRAFT/ KRAFTinfo.html.

[Kun89] K. Kunen. Signed Data Dependencies in Logic Programs. *Journal of Logic Programming*, 7:231–245, 1989.

[NG] R. Neches and D. Gunning. The Knowledge Sharing Effort. http:// www-ksl.stanford.edu/ knowledge-sharing/ papers/ kse-overview.html.

[Nil86] N. J. Nilsson. Probabilistic Logic. *Artificial Intelligence*, 28:71–87, 1986.

[NS92] R. Ng and V. S. Subrahmanian. Probabilistic Logic Programming. *Information and Computation*, 1992.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[PHG+99] A. Preece, K. Hui, A. Gray, Philippe Marti, T. Bench-Capon, D. Jones, and Z. Cui. The KRAFT Architecture for Knowledge Fusion and Transformation. In *Expert Systems 1999*, 1999.

[RACCdSV00] D. S. Robertson, J. Agusti-Cullel, F. S. Correa da Silva, and W. W. Vasconcelos. A Lightweight Capability Communication Mechanism. In *IEA-AIE'2000 (submitted)*, 2000.

[Sha76] G. Shafer. *A Mathematical Theory of Evidence.* Princeton Univ. Press, 1976.

[Sha97] M. J. R. Shave. Ontological Structures for Knowledge Sharing. *New Review of Information Networking (submitted)*, 1997.

[SS86] L. Sterling and E. Shapiro. *The Art of PROLOG.* MIT Press, 1986.

[Sub] V. S. Subrahmanian. Hermes – a Heterogeneous Reasoning and Mediator System. http:// www.cs.umd.edu/ projects/ hermes/index.html.

[Tur89] D. Turi. Logic Programs with Negation: Classes, Models, Interpreters. Technical Report CS-R8943, Centre for Mathematics and Computer Science, 1989.

[UG96] M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2):93–136, 1996. ISSN 0269-8889.

[vdG96] L. van der Gaag. Bayesian Belief Networks: Odds and Ends. *The Computer*, 39:97–113, 1996.

RELATÓRIOS TÉCNICOS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 1996 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail(mac@ime.usp.br).

Daniela V. Carbogim and Flávio S. Corrêa da Silva
FACTS, ANNOTATIONS, ARGUMENTS AND REASONING
RT-MAC-9601, janeiro de 1996, 22 pp.

Kunio Okuda
REDUÇÃO DE DEPENDÊNCIA PARCIAL E REDUÇÃO DE DEPENDÊNCIA GENERALIZADA
RT-MAC-9602, fevereiro de 1996, 20 pp.

Junior Barrera, Edward R. Dougherty and Nina Sumiko Tomita
AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES BY DESIGN OF STATISTICALLY OPTIMAL OPERATORS IN THE CONTEXT OF COMPUTATIONAL LEARNING THEORY.
RT-MAC-9603, abril de 1996, 48 pp.

Junior Barrera e Guillermo Pablo Salas
SET OPERATIONS ON CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC PROGRAMMINIG OF MMACH'S
RT-MAC-9604, abril de 1995, 66 pp.

Kunio Okuda
CYCLE SHRINKING BY DEPENDENCE REDUCTION
RT-MAC-9605, maio de 1996, 25 pp.

Julio Stern, Fabio Nakano e Marcelo Lauretto
REAL: REAL ATTRIBUTE LEARNING FOR STRATEGIC MARKET OPERATION
RT-MAC-9606, agosto de 1996, 16 pp.

Markus Endler
SISTEMAS OPERACIONAIS DISTRIBUÍDOS: CONCEITOS, EXEMPLOS E TENDÊNCIAS
RT-MAC-9607, agosto de 1996, 120 pp.

Hae Yong Kim
CONSTRUÇÃO RÁPIDA E AUTOMÁTICA DE OPERADORES MORFOLÓGICOS E EFICIENTES PELA APRENDIZAGEM COMPUTACIONAL
RT-MAC-9608, outubro de 1996, 19 pp.

Marcelo Finger
NOTES ON COMPLEX COMBINATORS AND STRUCTURALLY-FREE THEOREM PROVING
RT-MAC-9609, dezembro 1996, 28 pp.

Carlos Eduardo Ferreira, Flávio Keidi Miyazawa e Yoshiko Wakabayashi (eds)
ANAIS DA I OFICINA NACIONAL EM PROBLEMAS DE CORTE E EMPACOTAMENTO
RT-MAC-9610, dezembro de 1996, 65 pp.

Carlos Eduardo Ferreira, C. C. de Souza e Yoshiko Wakabayashi
REARRANGEMENT OF DNA FRAGMENTS: A BRANCH-AND-CUT ALGORITHM
RT-MAC-9701, janeiro de 1997, 24 pp.

Marcelo Finger
NOTES ON THE LOGICAL RECONSTRUCTION OF TEMPORAL DATABASES
RT-MAC-9702, março de 1997, 36 pp.

Flávio S. Corrêa da Silva, Wamberto W. Vasconcelos e David Robertson
COOPERATION BETWEEN KNOWLEDGE BASED SYSTEMS
RT-MAC-9703, abril de 1997, 18 pp.

Junior Barrera, Gerald Jean Francis Banon, Roberto de Alencar Lotufo, Roberto Hirata Junior
MMACH: A MATHEMATICAL MORPHOLOGY TOOLBOX FOR THE KHOROS SYSTEM
RT-MAC-9704, maio de 1997, 67 pp.

Julio Michael Stern e Cibele Dunder
PORTFÓLIOS EFICIENTES INCLUINDO OPÇÕES
RT-MAC-9705, maio de 1997, 29 pp.

Junior Barrera e Ronaldo Fumio Hashimoto
COMPACT REPRESENTATION OF W-OPERATORS
RT-MAC-9706, julho de 1997, 13 pp.

Dilma M. Silva e Markus Endler
CONFIGURAÇÃO DINÂMICA DE SISTEMAS
RT-MAC-9707, agosto de 1997, 35 pp

Kenji Koyama e Routo Terada
AN AUGMENTED FAMILY OF CRYPTOGRAPHIC PARITY CIRCUITS
RT-MAC-9708, setembro de 1997, 15 pp

Routo Terada e Jorge Nakahara Jr.
LINEAR AND DIFFERENTIAL CRYPTANALYSIS OF FEAL-N WITH SWAPPING
RT-MAC-9709, setembro de 1997, 16 pp

Flávio S. Corrêa da Silva e Yara M. Michelacci
MAKING OF AN INTELLIGENT TUTORING SYSTEM (OR METHODOLOGICAL ISSUES OF ARTIFICIAL INTELLIGENCE RESEARCH BY EXAMPLE)
RT-MAC-9710, outubro de 1997, 16 pp.

Marcelo Finger
COMPUTING LIST COMBINATOR SOLUTIONS FOR STRUCTURAL EQUATIONS
RT-MAC-9711, outubro de 1997, 22 pp.

Maria Angela Gurgel and E.M.Rodrigues
THE F-FACTOR PROBLEM
RT-MAC-9712, dezembro de 1997, 22 pp.

Perry R. James, Markus Endler, Marie-Claude Gaudel
DEVELOPMENT OF AN ATOMIC-BROADCAST PROTOCOL USING LOTOS
RT-MAC-9713, dezembro de 1997, 27 pp.

Carlos Eduardo Ferreira and Marko Loparic
A BRANCH-AND-CUT ALGORITHM FOR A VEHICLE ROUTING PROBLEM WITH CAPACITY AND TIME CONSTRAINTS
RT-MAC-9714, dezembro de 1997, 20 pp.

Nami Kobayashi
A HIERARCHY FOR THE RECOGNIZABLE M-SUBSETS
RT-MAC-9715, dezembro de 1997, 47 pp.

Flávio Soares Corrêa da Silva e Daniela Vasconcelos Carbogim
A TWO-SORTED INTERPRETATION FOR ANNOTATED LOGIC
RT-MAC-9801, fevereiro de 1998, 17 pp.

Flávio Soares Corrêa da Silva, Wamberto Weber Vasconcelos, Jaume Agustí, David Robertson e Ana Cristina V. de Melo.
WHY ONTOLOGIES ARE NOT ENOUGH FOR KNOWLEDGE SHARING
RT-MAC-9802, outubro de 1998, 15 pp.

J. C.de Pina e J. Soares
ON THE INTEGER CONE OF THE BASES OF A MATROID
RT-MAC-9803, novembro de 1998, 16 pp.

K. Okuda and S.W.Song
REVISITING HAMILTONIAM DECOMPOSITION OF THE HYPERCUBE
RT-MAC-9804, dezembro 1998, 17pp.

Markus Endler
AGENTES MÓVEIS: UM TUTORIAL
RT-MAC-9805, dezembro 1998, 19pp.

Carlos Alberto de Bragança Pereira, Fabio Nakano e Julio Michael Stern
A DYNAMIC SOFTWARE CERTIFICATION AN VERIFICATION PROCEDURE
RT-MAC-9901, março 1999, 21pp.

Carlos E. Ferreira e Dilma M. Silva
BCC DA USP: UM NOVO CURSO PARA OS DESAFIOS DO NOVO MILÊNIO
RT-MAC-9902, abril 1999, 12pp.

Ronaldo Fumio Hashimoto and Junior Barrera
A SIMPLE ALGORITHM FOR DECOMPOSING CONVEX STRUCTURING ELEMENTS
RT-MAC-9903, abril 1999, 24 pp.

Jorge Euler, Maria do Carmo Noronha e Dilma Menezes da Silva
ESTUDO DE CASO: DESEMPENHO DEFICIENTE DO SISTEMA OPERACIONAL LINUX PARA CARGA MISTA DE APLICAÇÕES.
RT-MAC-9904, maio 1999, 27 pp.

Carlos Humes Junior e Paulo José da Silva e Silva
AN INEXACT CLASSICAL PROXIMAL POINT ALGORITHM VIEWED AS DESCENT METHOD IN THE OPTIMIZATION CASE
RT-MAC-9905, maio 1999, pp.

Carlos Humes Junior and Paulo José da Silva e Silva
STRICT CONVEX REGULARIZATIONS, PROXIMAL POINTS AND AUGMENTED LAGRANGIANS
RT-MAC-9906, maio 1999, 21 pp.

Ronaldo Fumio Hashimoto, Junior Barrera, Carlos Eduardo Ferreira
A COMBINATORIAL OPTIMIZATION TECHNIQUE FOR THE SEQUENTIAL DECOMPOSITION OF EROSIONS AND DILATIONS
RT-MAC-9907, maio 1999, 30 pp.

Carlos Humes Junior and Marcelo Queiroz
ON THE PROJECTED PAIRWISE MULTICOMMODITY FLOW POLYHEDRON
RT-MAC-9908, maio 1999, 18 pp.

Carlos Humes Junior and Marcelo Queiroz
TWO HEURISTICS FOR THE CONTINUOUS CAPACITY AND FLOW ASSIGNMENT GLOBAL OPTIMIZATION
RT-MAC-9909, maio 1999, 32 pp.

Carlos Humes Junior and Paulo José da Silva e Silva
AN INEXACT CLASSICAL PROXIMAL POINT ALGORITHM VIEWED AS A DESCENT METHOD IN THE OPTIMIZATION CASE
RT-MAC-9910, julho 1999, 13 pp.

Markus Endler and Dilma M. Silva and Kunio Okuda
A RELIABLE CONNECTIONLESS PROTOCOL FOR MOBILE CLIENTS
RT-MAC-9911, setembro 1999, 17 pp.

David Robertson, Fávio S. Corrêa da Silva, Jaume Agustí and Wamberto W. Vasconcelos
A LIGHTWEIGHT CAPABILITY COMMUNICATION MECHANISM
RT-MAC-9912, novembro 1999, 14 pp.

Flávio S. Corrêa da Silva, Jaume Agustí, Roberto Cássio de Araújo and Ana Cristina V. de Melo
KNOWLEDGE SHARING BETWEEN A PROBABILISTIC LOGIC AND BAYESIAN BELIEF NETWORKS
RT-MAC-9913, novembro 1999, 13 pp.