DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-9703

Cooperation Between Knowledge Based Systems

Flávio S. Corrêa da Silva
Wamberto W. Vasconcelos
David Robertson

Abril 97

Cooperation Between Knowledge Based Systems

Flávio S. Corrêa da Silva
Instituto de Matemática e Estatística da Universidade de São Paulo
Cid. Universitária "ASO" - São Paulo (SP) - Brazil
email: fcs@ime.usp.br

Wamberto W. Vasconcelos
Departamento de Ciência de Computação da Universidade Estadual do Ceará
Cid. Universitária - Fortaleza (CE) - Brazil
email: wamb@uece.br

David Robertson
Department of Artificial Intelligence, University of Edinburgh
Edinburgh - Scotland
email: dr@aisb.ed.ac.uk

Abstract

For as long as there has been interest in knowledge based systems there has been interest in sharing formally expressed knowledge. It is traditional for this to require a high degree of social interaction between the suppliers and recipients of such information but the Internet has brought with it an interest in more opportunistic, semi-automatic cooperation between systems. This raises a variety of technical problems relevant to logic programming. We discuss these - concentrating on the use of protocols and interlinguas.

1 Introduction

One of the benefits of formally represented knowledge is its potential to be shared. The opportunity presented by the Internet (and related technologies) is for locally produced knowledge bases and inference mechanisms to interact in solving problems which are larger than each local system could tackle on its own. The problem which this creates is how we can ensure (or at least assess) the reliability of this form of cooperation, given that the cooperating systems may have been developed by different groups of people. To avoid adding an extra layer of complexity to our discussion we shall assume that all the systems we

$\mathrm{KBS_{FOHC}}$
$\Sigma' = \{q_0, \ldots, q_n, x_0, \ldots, x_m, a_0, \ldots, a_r, \wedge, \leftarrow, \top, \bot\}$ Components $x_0, \ldots, a_0, \ldots, a_r$ are generically addressed as *terms* $t_0, \ldots,$.
Φ' is the set comprising of 1. $q(\vec{T}) \leftarrow \bigwedge_{i=0}^{n} q_i(\vec{T_i})$ where $\vec{T}$ and $\vec{T_i}$ are sequences of terms. 2. $\leftarrow \bigwedge_{i=0}^{n} q_i(\vec{T_i})$ where $\vec{T_i}$ is a sequence of terms. 3. $\square$ (the empty formula $\bot \leftarrow \top$)
$\Delta' = \emptyset$
$\Gamma' = \{SLD\}$ where SLD is SLD-resolution rule

Figure 1: Formalisation of $\mathrm{KBS_{FOHC}}$

are dealing with are already represented in a computational logic of some kind. We shall refer to the set of axioms and inference rules defining a knowledge base and inference mechanism as a "knowledge based system" or KBS for short.

A large number of formal languages exist for KBSs, each being chosen according to its adequacy for the representation of the knowledge related to specific domains, ease-of-use, popularity, and existing supporting tools. For instance, bayesian and probabilistic languages [Pea88] are convenient for uncertain knowledge; linear logics [Gir87] are convenient for representing knowledge whose inferences may be resource-bounded, and so on. Attempts to provide heterogeneous sources of knowledge with the ability to cooperate has arisen in different contexts. Some projects related to this idea can be found in [Sub, NG, Gra]. The approach described in Section 2 is akin to the proposal found in [Sub, Gra] in the sense that we achieve global cooperation by resolving local communication issues. A more formal counterpart to these proposals, oriented to formal specification of software, can be found in [OPE97, CM97, GB92].

In the discussions which follow we shall describe each KBS as the tuple $\langle \Sigma, \Phi, \Delta, \Upsilon \rangle$, standing, respectively, for the alphabet, the set of formulae, the axioms and the inference rules. The first of these, $\mathrm{KBS_{FOHC}}$, is shown in Figure 1. Its alphabet contains those predicate names $(q_0, \ldots, q_n)$, variable names $(x_0, \ldots, x_m)$ and constants $(a_0, \ldots, a_r)$ used in defining the formulae. The formulae are Horn clauses. There are no axioms and the proof rule is simply SLD resolution.

Perhaps the most obvious way of controlling cooperation is by viewing each KBS as a module, with restrictions on the predicates which are allowed to be imported or exported. Potential connections between pairs of KBS's can then be detected automatically by matching elements of the import description of the "client" KBS to elements of the export description of the "server" KBS.

This sort of simple interface between modules allows KBS designers to distinguish which sorts of information they wish to exclude from cooperation (by leaving them out of the import/export definitions) but does not guarantee that those predicates which are included can safely be used between systems. This is because the match between predicates is syntactic whereas we are concerned to ensure that the semantics of the predicates imported from the donor KBS conforms to that expected by the recipient KBS. One way to do this is to agree, in advance, on a protocol for communication between pairs of KBS systems. We give an example of this type of protocol in Section 2.

2 Protocols for Communication between Knowledge Bases

Given KBS descriptions of the form introduced in Figure 1, there are a number of ways of being more precise about permitted channels of communication between systems. In this section we describe one way of doing this, based on *external descriptions*, *query-answer sets* and *protocols* for cooperation between knowledge bases.

The external description of a knowledge base formalism comprises the minimal descriptive information necessary for sharing its knowledge. Both server and client knowledge bases are required to provide their corresponding external descriptions in order to establish a connection. An *external description* is composed of:

- a presentation of the alphabet of the corresponding formalism, given in some standard notation;
- the formal syntax of the input/output expressions to be offered by the knowledge-based system, i.e. the external queries accepted by the system and corresponding answers provided by this system – also given in some standard notation (e.g. in Backus-Naur Form).

What portions of a given knowledge-based system are going to be shared with other systems is a matter of choice: given the constraints imposed by each formalism on what alphabet and expressions can be recognised, it is left to the system designer to decide what will indeed be turned into public knowledge.

A *query-answer set* is a collection of pairs of expressions that the client system is going to process, and thus these expressions are presented using the syntax of that system. Query expressions represent queries that the client can send to the server, and answer expressions represent the corresponding answers provided by the server.

Finally, a *protocol* is a set of connections between elements of the alphabets of the client and the server systems, and of the queries and answers in the query-answer set and corresponding input/output expressions in the server external description. To each connection can be associated a procedure to adapt the corresponding information, by translating its content from one formalism to the other, eliminating possible incompatibilities (and thus possibily loosing information in the communication) or even processing and transforming the transferred information so that external knowledge becomes compatible with the internal representations of the client system.

The expressions accepted for a query-answer set are constrained by the external descriptions of both client and server systems. Nevertheless, the client system designer may prefer to constrain the system capabilities for connecting to the server even further, given the restrictions on information privacy and computational costs. Each system designer is free to do so according to their intentions and interests.

The process of building an external description consists of:

1. presenting the alphabet and input/output expressions for the corresponding formalism – that will typically consist of subsets of the alphabet and syntax of the formalism itself;

2. linking each element of the alphabet and each input/output expression to their actual representations within the formalism, i.e. linking the external description with its corresponding knowledge-based system, so that it can act as an interface for that system; and

3. providing for each element of the alphabet and for each input/output expression an explanation of their meaning. The meaning of expressions and symbols in an external description is not used as a piece of software code to activate any system. For this reason, and also given the plurality of features and concepts to be found in different knowledge base formalisms, its format is left free. Nevertheless, this meaning is essential to ensure the proper use of knowledge bases by external clients, and hence it must always be carefully and precisely stated within external descriptions.

The last step to allow two systems to act as knowledge client and knowledge server is the establishment of the communication protocol itself. Given two external descriptions – one for the server and one for the client – and a query-answer set for the client knowledge-based system, each element of the alphabet of the client system that occurs in the query-answer set must be related to a corresponding element of the alphabet of the external description of the server system. Additionally, each expression in the query-answer set must be related to a corresponding input/output expression (or set of expressions) in the external description of the server system.

As a working example let us consider establishing the communication between two knowledge-based systems that employ different formalisms: $\mathrm{KBS}_{\mathrm{FOHC}}$, which we gave in Figure 1, and $\mathrm{KBS}_{\mathrm{PML}}$, which is given in Figure 2 and uses a propositional modal logic. Suppose that two different design groups have each developed a KBS - one based on $\mathrm{KBS}_{\mathrm{FOHC}}$ to perform risk assessment on geographical areas; the other based on $\mathrm{KBS}_{\mathrm{PML}}$ to describe the occurrence of diseases in

$\mathrm{KBS}_{\mathrm{PML}}$
$\Sigma = \{p_0, \ldots, p_n, \Box, \Diamond, \vee, \wedge, \neg, \rightarrow, \top, \bot\}$
$\Phi = \{\varphi_0, \ldots\}$ such that 1. $p_i \in \Phi, i \geq 0$; 2. if $\varphi \in \Phi$ then $\Box\varphi \in \Phi, \Diamond\varphi \in \Phi$ and $\neg\varphi \in \Phi$; 3. if $\varphi_1, \varphi_2 \in \Phi$ then $\varphi_1 \vee \varphi_2 \in \Phi, \varphi_1 \wedge \varphi_2 \in \Phi$ and $\varphi_1 \rightarrow \varphi_2 \in \Phi$ 4. $\top, \bot \in \Phi$
$\Delta = PA \cup S4$, where PA = propositional axioms and $S4$ = axioms for modal system $S4$
$\Gamma = \{\frac{\varphi_i, \varphi_i \rightarrow \varphi_j}{\varphi_j}, \frac{\varphi}{\Diamond\varphi}\}$

Figure 2: Formalisation of $\mathrm{KBS}_{\mathrm{PML}}$

$\widehat{\Sigma} = \{p_1, p_2, p_3, \Box, \Diamond, \neg\}$
$\mathcal{INPUT} = \{\Box p_i, \Diamond p_i, \Box\neg p_i, \Diamond\neg p_i | 1 \leq i \leq 3\}$
$\mathcal{OUTPUT} = \{\bot, \top\}$

Figure 3: External Description of $\mathrm{KBS}_{\mathrm{PML}}$

particular countries. At the risk of overloading terminology, we shall refer to these two KBS instances as $\mathrm{KBS}_{\mathrm{FOHC}}$ and $\mathrm{KBS}_{\mathrm{PML}}$.

A selected subset of the alphabet and the expected input/output expressions comprise the external description to be made available to potential clients of the knowledge-based system. In our example, $\mathrm{KBS}_{\mathrm{PML}}$ is assumed to have the external description shown in Figure 3. Similarly, $\mathrm{KBS}_{\mathrm{FOHC}}$ has the external description shown in Figure 4.

These are strong specialisations of the full sets of queries accepted by each formalism and their corresponding possible answers. Such specialisations are deliberate: we are assuming that the designers of $\mathrm{KBS}_{\mathrm{FOHC}}$ and $\mathrm{KBS}_{\mathrm{PML}}$ *do not want to* offer their full knowledge bases, nor their full knowledge processing capabilities, to client systems – instead, they are offering restricted possibilities selected from all they can offer.

Assuming that $\mathrm{KBS}_{\mathrm{FOHC}}$ wants to become a client of $\mathrm{KBS}_{\mathrm{PML}}$ its corresponding query-answer set would have to be interactively specified by its designer, taking into account the associated meaning of each symbol of the alphabets and expressions built from them. A list of components and their meanings

$\widehat{\Sigma'} = \{q_1, q_2, x_0, \ldots, a_2, a_5, \wedge, \leftarrow\}$
$\mathcal{INPUT}' = \{\leftarrow q_1(\vec{T_1}), \leftarrow q_2(\vec{T_2})\}$ where $\vec{T_1}$ and $\vec{T_2}$ are variable-free;
$\mathcal{OUTPUT}' = \{\bot, \top\}$

Figure 4: External Description of $\mathrm{KBS}_{\mathrm{FOHC}}$

$\widehat{\Sigma}' = \{q_1, q_2, q_1^1, q_2^1, q_3^1, x_0, \ldots, a_2, a_5, \wedge, \leftarrow\}$
where q_1^1, q_2^1, q_3^1 are unary predicates
$\mathcal{INPUT}'$ and $\mathcal{OUTPUT}'$ remain as before

Figure 5: Extended External Description of KBS_{FOHC}

$\mathcal{QUERY} = \{\leftarrow q_1^1(x_i), \leftarrow q_2^1(x_j), \leftarrow q_3^1(x_k)\}$
$\mathcal{ANSWER} = \{\perp, \top\}$

Figure 6: Query-Answer Set for KBS_{FOHC}

is made available for inspection and subsequent association. The specification of the query-answer set is determined by three factors:

1. Input/output expressions made available by the server knowledge-based system;

2. The full syntax of the client knowledge-based system formalism; and

3. The specific interests of the designer of the client knowledge-based system

Let us assume that, in our example, the designer of KBS_{FOHC} wishes to consult each one of the $p_i, 1 \leq i \leq 3$, made available by KBS_{PML}. This requires the external description of KBS_{FOHC} to be extended, so that its alphabet contemplates one symbol for each of those p_i. Thus, the external description of KBS_{FOHC} is extended to what is presented in Figure 5. With this in hand we can build a query-answer set for KBS_{FOHC}. In our example, it is assumed to have the specification shown in Figure 6. Notice that the $\mathcal{ANSWER}$ set does not comprise substitutions, and therefore may lead to a theorem prover based on SLD-resolution with a peculiar (perhaps desirable) behaviour. This shall be taken into account and solved by the communication system when establishing a connection. But first, we must define a protocol.

In this example, the definition of a protocol is simple: it consists of a characterisation of how elements of $\widehat{\Sigma}$ and $\widehat{\Sigma}'$ correspond to each other, and of how elements of $\mathcal{QUERY}$ (resp. $\mathcal{ANSWER}$) and of $\mathcal{INPUT}$ (resp. $\mathcal{OUTPUT}$) are connected. Our protocol is shown in Figure 7. Now KBS_{FOHC} is prepared to act as a knowledge client with KBS_{PML} as the server. All necessary information to automatically generate the code that will control the interaction between these two systems is given in their external descriptions and in the corresponding query-answer set and protocol.

q_1^1	$\Leftrightarrow$	p_1
q_2^1	$\Leftrightarrow$	p_2
q_3^1	$\Leftrightarrow$	p_3
$\leftarrow q_1^1(x)$	$\Leftrightarrow$	$\Diamond p_1$
$\leftarrow q_2^1(x)$	$\Leftrightarrow$	$\Diamond p_2$
$\leftarrow q_3^1(x)$	$\Leftrightarrow$	$\Diamond p_3$
$\bot$	$\Leftrightarrow$	$\bot$
$\top$	$\Leftrightarrow$	$\top$

Figure 7: Protocol between KBS_{FOHC} and KBS_{PML}

2.1 Sample Session

Suppose that KBS_{PML} has part of its knowledge base as follows:

$$\begin{array}{l} p_4 \wedge p_7 \wedge p_9 \rightarrow \Diamond p_1 \\ \vdots \\ p_4 \\ p_7 \\ p_9 \\ \vdots \end{array}$$

The meaning of each propositional variable is:

- p_1 – outbreak of yellow fever;
- p_4 – swampy conditions;
- p_7 – warm climate;
- p_9 – pollution.

The modal rule presented above could be read as "*if there are swampy conditions and a warm climate and pollution then it is possible that an outbreak of yellow fever may occur*". Let us further suppose that KBS_{FOHC} has part of its knowledge base as follows:

$$\begin{array}{l} q_{10}(x) \leftarrow q_1^1(\_) \wedge q_4(x) \\ q_4(x) \leftarrow \ldots \\ \vdots \end{array}$$

The meaning of each predicate is:

- $q_{10}(x)$ – area x is threatened by yellow fever;

- $q_1^1(x)$ – area x has had an outbreak of yellow fever;
- $q_4(x)$ – area x fulfils conditions for an outbreak of yellow fever.

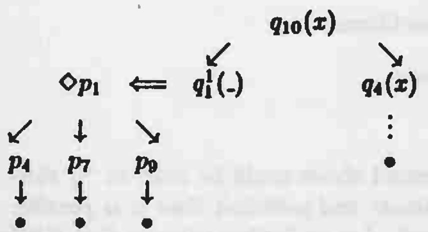
The rule presented above could be read as "*area x is under threat of a yellow fever outbreak if there have been cases of yellow fever in the world (i.e. the disease has not yet been eradicated) and it fulfils the necessary conditions for an yellow fever outbreak*". If query

$$\leftarrow q_{10}(x)$$

is posed to KBS_{FOHC}, it causes the following behaviour:

1. $q_{10}(x)$ matches the head of the rule; the inference mechanism of KBS_{FOHC} proceeds to prove $q_1^1(\_)$ and then $q_4(x)$;
2. KBS_{FOHC} fails to prove $q_1^1(\_)$ locally, and resorts to the established protocol to communicate with other KBSs; $q_1^1(\_)$ is automatically translated via the protocol as $\Diamond p_1$ and sent to KBS_{PML};
3. The inference mechanism of KBS_{PML} successfully proves $\Diamond p_1$ and its result $\top$ is then sent to KBS_{FOHC};
4. KBS_{FOHC} receives the translated result $\top$ (in our example, the original and translated results are the same) stating that $q_1^1(\_)$ holds and it proceeds to successfully prove $q_4(x)$, obtaining $\theta = \{x/a\}$, for some a.

This behaviour can be graphically illustrated as the proof tree below:



The proof-tree is presented in the usual notation, with "•" to indicate a closed branch. We have added the "$\Leftarrow$" to indicate the transfer of a query to a KBS different from the one in which it was originated.

3 Interlinguas as Alternatives to Protocols

A protocol, like that of Section 2 seems necessary as a means of recording agreements about transfer of information. However, accurate protocols require consensus, in advance of use, on the points at which different KBSs may interact.

This, in turn, requires work by each team of KBS designers for each different cooperation. The use of "interlinguas" has been suggested as a solution to this problem (see [NG] for an example or [UG96] for an overview). Instead of constructing protocols between each pair of KBSs we require the developers of each one to write general purpose translators into and out of a single interlingua language. If we have 100 dialects then we have to write 200 translators (less if some of the translators are bi-directional), which is much better than the 19800 translators we would need to convert directly between every possible pair of dialects. This sounds like an attractive alternative to protocols but the following counter-examples demonstrate that the interlingua ideal of automatic translation is not always achievable.

- Translation to an interlingua may suppress important information about predicates with a special role in inference. Suppose that we have written, in a domain-specific application language, the following rule which says that birds other than penguins can fly:

  ```
  if bird(X) then flies(X) unless penguin(X)
  ```

  To make this accessible, we might write a translation for such statements into the interlingua language, for which the rewrite rule might be:

  ```
  if P(X) then Q(X) unless R(X)
  ```

  $$\Downarrow$$

  $$(P(X) \wedge \neg ab(Q,X) \rightarrow Q(X)) \wedge (R(X) \rightarrow ab(Q,X))$$

  and applying this rule to obtain the appropriate statement in the interlingua gives:

  $$\begin{aligned} bird(X) \wedge \neg ab(flies,X) &\rightarrow flies(X) \\ penguin(X) &\rightarrow ab(flies,X)) \end{aligned}$$

  If the interlingua were truly independent of the application language then it should be possible to use it freely, without reference to the originating rules. This isn't the case because the *ab* predicate plays a special role - it differentiates default cases from special cases - and fiddling around with it without realising this is liable to cause problems. If we were to merge this knowledge with another knowledge base we would probably want to ensure that the treatment of defaults was uniform, so it is essential that we know that the original dealt with defaults and that we can trace how this fed through to the interlingua.

- The knowledge representation and inference methods of the original system are closely linked so the interlingua needs to apply to both and maintain the connection between them. An alternative (and in practice more common) way of building an application for the default example above would be to have two simpler rules:

```
if bird(X) then flies(X)
if penguin(X) then not(flies(X))
```

which would translate to the interlingua straightforwardly as:

$$\begin{aligned} bird(X) &\rightarrow flies(X) \\ penguin(X) &\rightarrow \neg flies(X) \end{aligned}$$

and define an inference mechanism which checked for special cases before allowing the default to be concluded. This is essential because the rules themselves could allow logically inconsistent conclusions (were we to add the obvious rule that penguins are birds). The appropriate control information could be expressed in the interlingua, for example as:

$$infer(A) \leftarrow rule(A \rightarrow B) \wedge infer(A) \wedge \neg infer(\neg B)$$

but we are now using the interlingua to prescribe how the knowledge should be used, with $rule(A \rightarrow B)$ being true if a rule exists in the knowledge base and $infer(X)$ holding if X can be inferred using the inference mechanism. Someone wanting to use the knowledge base would be well advised to use the inference mechanism designed for it or, at least, be aware of the intended forms of inference when merging the knowledge base with others.

- We need to know something of the potential capabilities and limitations of a system before we use it. Consider the following two definitions of a network, in which the predicate $path(X,Y)$ denotes that there is a path from node X to node Y in the network. Our network has only three nodes, a, b and c. The first definition is:

$$\begin{aligned} & path(a,b) \\ & path(b,c) \\ path(X,Y) &\leftarrow path(X,Z) \wedge path(Z,Y) \end{aligned}$$

This is capable of testing if a given path (*e.g.* $path(a,c)$) is allowed by the network. However, it is also capable of recursing infinitely if it is used to generate all potential paths (*e.g.* by asking for all instances of $path(a,F)$). Our second definition is:

$$\begin{aligned} & step(a,b) \\ & step(b,c) \\ path(X,Y) &\leftarrow step(X,Z) \wedge path(Z,Y) \end{aligned}$$

which will both test and generate paths, without fear of infinite recursion. This is a property of the second definition which we would wish to know about when deciding between the two definitions. We would not want to have to discover such properties from first principles because that might well be more effort than writing our own definition.

In short, it is impractical to rely on solely on interlinguas to support cooperating KBSs because we need to know more about the context of each knowledge base than we can tell by simply looking at its definition in some standard form.

4 Conclusions

The problem of managing cooperation between KBSs has been with us for much longer than the Internet. What has changed is that there is now a greater impulse to combine systems from diverse sources. Many of the methods currently being advocated for doing this appear to be unreliable because they require a global consensus on the interpretation of formal representations (either by ontological standardisation or through interlinguas). We have suggested, through a series of counter-examples, that this is impractical in general (although it may still be useful for certain closed-group cooperations, where design styles are relatively uniform). In response to this problem we have suggested two complementary remedies. The first is to define languages targeted at specific forms of cooperation, where protocols for transfer of information are constructed in advance. The second is to record along with the definitions of each KBS information about key properties which constrain its intended use - for example, that some predicates play a special role during inference; that particular proof strategies are appropriate; or that particular computational properties are (or are not) guaranteed. A challenge for logic programming is whether it can provide such support in styles suited to knowledge engineering practice.

Acknowledgements

The first author was partially supported by FAPESP grant 93/0603-01, and CNPq grant 300041/93-4. Part of these ideas were developed while the authors were working together at the University of Edinburgh during the (northern hemisphere) summer of 1996, and while the first and the second authors were working together at the State University of Ceará during (southern hemisphere) summer of 1997.

References

[CM97] M. Cerioli and J. Meseguer. May I Borrow your Logic? *Theoretical Computer Science*, 173:311–347, 1997.

[GB92] J. A. Goguen and R. M. Burstall. Institutions: Abstract Model Theory for Specification and Programming. *Journal of the ACM*, 39:95–146, 1992.

[Gir87] J. Y. Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987.

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

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

[OPE97] F. Orejas, E. Pino, and H. Ehrig. Institutions for Logic Programming. *Theoretical Computer Science*, 173:485–511, 1997.

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

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

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

RELATÓRIOS TÉCNICOS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 1994 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail(mac@ime.usp.br).

Flavio S. Corrêa da Silva
AN ALGEBRAIC VIEW OF COMBINATION RULES
RT-MAC-9401, Janeiro de 1994, 10 pp.

Flavio S. Corrêa da Silva e Junior Barrera
AUTOMATING THE GENERATION OF PROCEDURES TO ANALYSE BINARY IMAGES
RT-MAC-9402, Janeiro de 1994, 13 pp.

Junior Barrera, Gerald Jean Francis Banon e Roberto de Alencar Lotufo
A MATHEMATICAL MORPHOLOGY TOOLBOX FOR THE KHOROS SYSTEM
RT-MAC-9403, Janeiro de 1994, 28 pp.

Flavio S. Corrêa da Silva
ON THE RELATIONS BETWEEN INCIDENCE CALCULUS AND FAGIN-HALPERN STRUCTURES
RT-MAC-9404, abril de 1994, 11 pp.

Junior Barrera; Flávio Soares Corrêa da Silva e Gerald Jean Francis Banon
AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES
RT-MAC-9405, abril de 1994, 15 pp.

Valdemar W. Setzer; Cristina G. Fernandes; Wania Gomes Pedrosa e Flavio Hirata
UM GERADOR DE ANALISADORES SINTÁTICOS PARA GRAFOS SINTÁTICOS SIMPLES
RT-MAC-9406, abril de 1994, 16 pp.

Siang W. Song
TOWARDS A SIMPLE CONSTRUCTION METHOD FOR HAMILTONIAN DECOMPOSITION OF THE HYPERCUBE
RT-MAC-9407, maio de 1994, 13 pp.

Julio M. Stern
MODELOS MATEMATICOS PARA FORMAÇÃO DE PORTFÓLIOS
RT-MAC-9408, maio de 1994, 50 pp.

Imre Simon
STRING MATCHING ALGORITHMS AND AUTOMATA
RT-MAC-9409, maio de 1994, 14 pp.

Valdemar W. Setzer e Andrea Zisman
*CONCURRENCY CONTROL FOR ACCESSING AND COMPACTING B-TREES**
RT-MAC-9410, junho de 1994, 21 pp.

Renata Wassermann e Flávio S. Corrêa da Silva
TOWARDS EFFICIENT MODELLING OF DISTRIBUTED KNOWLEDGE USING EQUATIONAL AND ORDER-SORTED LOGIC
RT-MAC-9411, junho de 1994, 15 pp.

Jair M. Abe, Flávio S. Corrêa da Silva e Marcio Rillo
PARACONSISTENT LOGICS IN ARTIFICIAL INTELLIGENCE AND ROBOTICS.
RT-MAC-9412, junho de 1994, 14 pp.

Flávio S. Corrêa da Silva, Daniela V. Carbogim
A SYSTEM FOR REASONING WITH FUZZY PREDICATES
RT-MAC-9413, junho de 1994, 22 pp.

Flávio S. Corrêa da Silva, Jair M. Abe, Marcio Rillo
MODELING PARACONSISTENT KNOWLEDGE IN DISTRIBUTED SYSTEMS
RT-MAC-9414, julho de 1994, 12 pp.

Nami Kobayashi
THE CLOSURE UNDER DIVISION AND A CHARACTERIZATION OF THE RECOGNIZABLE Z-SUBSETS
RT-MAC-9415, julho de 1994, 29pp.

Flávio K. Miyazawa e Yoshiko Wakabayashi
AN ALGORITHM FOR THE THREE-DIMENSIONAL PACKING PROBLEM WITH ASYMPTOTIC PERFORMANCE ANALYSIS
RT-MAC-9416, novembro de 1994, 30 pp.

Thomaz I. Seidman e Carlos Humes Jr.
SOME KANBAN-CONTROLLED MANUFACTURING SYSTEMS: A FIRST STABILITY ANALYSIS
RT-MAC-9501, janeiro de 1995, 19 pp.

C.Humes Jr. and A.F.P.C. Humes
STABILIZATION IN FMS BY QUASI- PERIODIC POLICIES
RT-MAC-9502, março de 1995, 31 pp.

Fabio Kon e Arnaldo Mandel
SODA: A LEASE-BASED CONSISTENT DISTRIBUTED FILE SYSTEM
RT-MAC-9503, março de 1995, 18 pp.

Junior Barrera, Nina Sumiko Tomita, Flávio Soares C. Silva, Routo Terada
AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES BY PAC LEARNING
RT-MAC-9504, abril de 1995,16 pp.

Flávio S. Corrêa da Silva e Fabio Kon
CATEGORIAL GRAMMAR AND HARMONIC ANALYSIS
RT-MAC-9505, junho de 1995,17 pp.

Henrique Mongelli e Routo Terada
ALGORITMOS PARALELOS PARA SOLUÇÃO DE SISTEMAS LINEARES
RT-MAC-9506, junho de 1995, 158 pp.

Kunio Okuda
PARALELIZAÇÃO DE LAÇOS UNIFORMES POR REDUCÃO DE DEPENDÊNCIA
RT-MAC-9507, julho de 1995, 27 pp.

Valdemar W. Setzer e Lowell Monke
COMPUTERS IN EDUCATION: WHY, WHEN, HOW
RT-MAC-9508, julho de 1995, 21 pp.

Flávio S. Corrêa da Silva
REASONING WITH LOCAL AND GLOBAL INCONSISTENCIES
RT-MAC-9509, julho de 1995, 16 pp.

Julio M. Stern
MODELOS MATEMÁTICOS PARA FORMAÇÃO DE PORTFÓLIOS
RT-MAC-9510, julho de 1995, 43 pp.

Fernando Iazzetta e Fabio Kon
A DETAILED DESCRIPTION OF MAXANNEALING
RT-MAC-9511, agosto de 1995, 22 pp.

Flávio Keidi Miyazawa e Yoshiko Wakabayashi
POLYNOMIAL APPROXIMATION ALGORITHMS FOR THE ORTHOGONAL Z-ORIENTED 3-D PACKING PROBLEM
RT-MAC-9512, agosto de 1995, pp.

Junior Barrera e Guillermo Pablo Salas
SET OPERATIONS ON COLLECTIONS OF CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC PROGRAMMINIG OF MORPHOLOGICAL MACHINES
RT-MAC-9513, agosto de 1995, 84 pp.

Marco Dimas Gubitoso e Jörg Cordsen
PERFORMANCE CONSIDERATIONS IN VOTE FOR PEACE
RT-MAC-9514, novembro de 1995, 18pp.

Carlos Eduardo Ferreira e Yoshiko Wakabayashi
ANAIS DA I OFICINA NACIONAL EM PROBLEMAS COMBINATÓRIOS: TEORIA, ALGORITMOS E APLICAÇÕES
RT-MAC-9515, novembro de 1995, 45 pp.

Markus Endler and Anil D'Souza
SUPPORTING DISTRIBUTED APPLICATION MANAGEMENT IN SAMPA
RT-MAC-9516, novembro de 1995, 22 pp.

Junior Barrera, Routo Terada,
Flávio Corrêa da Silva and Nina Sumiko Tomita
*AUTOMATIC PROGRAMMING OF MMACH'S FOR OCR**
RT-MAC-9517, dezembro de 1995, 14 pp.

Junior Barrera, Guillermo Pablo Salas and Ronaldo Fumio Hashimoto
SET OPERATIONS ON CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC PROGRAMMING OF MMACH'S
RT-MAC-9518, dezembro de 1995, 14 pp.

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, 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.