Verifying time, memory and communication bounds in systems of reasoning agents

Natasha Alechina, Brian Logan, Nguyen Hoang Nga and Abdur Rakib*
School of Computer Science
University of Nottingham
Nottingham NG8 1BB, UK
({nza,bsl,hnn,rza}@cs.nott.ac.uk)

Abstract. We present a framework for verifying systems composed of heterogeneous reasoning agents, in which each agent may have differing knowledge and inferential capabilities, and where the resources each agent is prepared to commit to a goal (time, memory and communication bandwidth) are bounded. The framework allows us to investigate, for example, whether a goal can be achieved if a particular agent, perhaps possessing key information or inferential capabilities, is unable (or unwilling) to contribute more than a given portion of its available computational resources or bandwidth to the problem. We present a novel temporal epistemic logic, $BMCL\text{-}CTL$, which allows us to describe a set of reasoning agents with bounds on time, memory and the number of messages they can exchange. The bounds on memory and communication are expressed as axioms in the logic. As an example, we show how to axiomatise a system of agents which reason using resolution and prove that the resulting logic is sound and complete. We then show how to encode a simple system of reasoning agents specified in $BMCL\text{-}CTL$ in the description language of the Mocha model checker (Alur et al., 1998), and verify that the agents can achieve a goal only if they are prepared to commit certain time, memory and communication resources.

1. Introduction

A key application of multiagent systems research is distributed problem solving. Distributed approaches to problem solving allow groups of agents to collaborate to solve problems which no single agent could solve alone (e.g., because no single agent has all the information necessary to solve the problem), and/or to solve problems more effectively (e.g., in less time than a single agent). For a given problem and system of reasoning agents, many different solution strategies may be possible, each involving different commitments of computational resources (time and memory) and communication by each agent. For different multiagent systems, different solution strategies will be preferred depending on the relative costs of computational and communication resources for each agent. These tradeoffs may be different for different agents (e.g., reflecting their computational capabilities or network connection) and may reflect the agent's commitment to a particular problem. For a given system of agents with specified inferential abilities and resource bounds

* This work was supported by the UK Engineering and Physical Sciences Research Council [grant number EP/E031226].

it may not be clear whether a particular problem can be solved at all, or, if it can, what computational and communication resources must be devoted to its solution by each agent. For example, we may wish to know whether a goal can be achieved if a particular agent, perhaps possessing key information or inferential capabilities, is unable (or unwilling) to contribute more than a given portion of its available computational resources or bandwidth to the problem.

There has been considerable work in the agent literature on distributed problem solving in general (e.g., (Faltings and Yokoo, 2005; Jung and Tambe, 2005; Provan, 2002; Wooldridge and Dunne, 2006)) and on distributed reasoning in particular (e.g., (Adjiman et al., 2004; Amir and McIlraith, 2005)). Much of the work on distributed reasoning analyses the time and communication complexity of distributed reasoning algorithms. However, while we have upper bounds (and some lower bounds) on time and memory requirements for reasoning in distributed systems, we lack tools for reasoning about tradeoffs between computational and communication resources. In this paper we present a framework for reasoning about tradeoffs between time, memory and communication in systems of distributed reasoning agents. We assume that the agents reason using resolution. However this is not essential for the results in the paper, and we briefly sketch how reasoners using other inference methods can be formalised. We introduce a novel epistemic logic, $BMCL\text{-}CTL$, for specifying resource-bounded reasoners. Critically, the logic allows upper bounds on the resource commitments (time, memory and communication) of each agent in the system to be specified. We prove that the logic is sound and complete. Using simple resolution examples, we show how to encode systems of distributed reasoning agents specified in the logic in a model checker, and verify some example properties. In contrast to previous work, e.g., (Albore et al., 2006; Alechina et al., 2006a; Alechina et al., 2006b; Ågotnes and Alechina, 2006) which focused primarily on memory limitations of single reasoners, the approach proposed in this paper enables us to specify bounds on the number of messages the agents can exchange, allowing the investigation of tradeoffs between different resources. This allows us to determine whether, for example, giving a reasoner more memory will result in a shorter proof, or whether communication between agents can reduce either time or memory requirements or both. The logic presented in this paper is a revised and simplified version of that presented in (Alechina et al., 2008b). We changed the language of the logic by introducing communication counters and by replacing the underlying temporal logic $PCTL^*$ with CTL. We have also substantially simplified the axiomatisation of the logic.

The structure of the paper is as follows. In Section 2 we introduce the problem of resource bounds in distributed reasoning and in Section 3 we explain how we measure time, space and communication costs for a distributed reasoning problem. In Section 4 we introduce the epistemic logic

BMCL-CTL. Model checking experiments are described in Section 5. We survey related work in Section 6 and conclude in Section 7.

2. Distributed Reasoners

We define the 'shape' of a proof in terms of the maximum space requirement at any step in the proof and the number of inference steps it contains. The lower bound on space for a given problem is then the least maximum space requirement of any proof, and the lower bound on time is the least number of inference steps of any proof. In general, a minimum space proof and a minimum time proof will be different (have different shapes). Bounding the space available for a proof will typically increase the number of inference steps required and bounding the number of steps will increase the space required. For example, a proof which requires only the minimum amount of space may require rederivation of intermediate results.

We define the bounds on a reasoning agent in terms of its available resources expressed in terms of memory, time and communication. We assume that the memory required for a particular proof can be taken to be its space requirement (e.g., the number of formulas that must be simultaneously held in memory) times some constant, and the number of inference steps executed times some constant can be taken as a measure of the time necessary to solve the problem. The communication requirement of a proof is taken to be the number of messages exchanged with other agents. In what follows, we ignore the constants and assume that the units of problem size and resources are the same.

For a particular agent solving a particular problem, the space available for any given proof is ultimately bounded by the size of the agent's memory and the number of inference steps is bounded by the time available to the agent, e.g., by a response time guarantee offered by the agent, or simply the point in time at which the solution to the problem becomes irrelevant. The question then arises of whether a proof can be found which falls within the resource envelope defined by the agent's resource bounds.

For a single agent which processes a single goal at a time, the lower bounds on space for the goal determines the minimum amount of memory the agent must have if it is to solve the problem (given unlimited time); and the lower bound on time determines the time the agent must commit to solving the problem (given unlimited memory). In the general case in which the agent is attending to multiple goals simultaneously, the memory and time bounds may be given not by the environment, but by the need to share the available resources between multiple tasks. For example, the agent may need to share memory between multiple concurrent tasks and/or devote no more than a given proportion of CPU to a given task. In both cases, the agent designer

may be interested in tradeoffs between resource bounds; for example, whether more timely responses can be provided by pursuing fewer tasks in parallel (thereby making more memory available to each task) or whether more tasks can be pursued in parallel if each task is allowed to take longer.

In the distributed setting we distinguish between *symmetric problem distributions*, where all agents have the same premises, and *asymmetric problem distributions* where different premises may be assigned to different agents. We also distinguish between *homogeneous reasoners* (when all agents have the same rules of inference and resource bounds) and *heterogeneous reasoners*, (when different agents have different rules of inference and/or resource bounds).

Distribution does not necessarily change the shape (maximum space requirement and number of inference steps) of a proof. However, in a distributed setting the tradeoffs between memory and time bounds are complicated by communication. Unlike memory and time, communication has no direct counterpart in the proof. However like memory, communication can be substituted for time (e.g., if part of the proof is carried out by another agent), and, like time, it can be substituted for memory (e.g., if a lemma is communicated by another agent rather than having to be remembered). In the distributed setting, each agent has a minimum memory bound which is determined by its inference rules and which may be smaller than the minimum space requirement for the problem. If the memory bound for all agents taken individually is less than the minimum space requirement for the problem, then the communication bound must be greater than zero. (If the memory bound for all agents taken together is less than the minimum space requirement for the problem, then the problem is insoluble for any communication bound).

With a symmetric problem distribution, if the memory bound for at least one agent is greater than the minimum space requirement for the problem, the minimum communication bound is zero (with unbounded time). If the problem distribution is asymmetric, i.e., not all agents have all the premises, then the lower bound on communication may again be non-zero, if a necessary inference step requires premises from more than one agent.

In the next section, we present measures of space, time and communication for distributed reasoning agents which allow us to make these tradeoffs precise.

3. Measuring Resources

We assume a set of n agents. Each agent i has a set of propositional inference rules R_i (for example, R_i could contain conjunction introduction and modus ponens, or it could contain just a single rule of resolution) and a set of premises K_i.

For a single agent, the notion of a derivation, or a proof of a formula G from K_i is standard, and the time and space complexity of proofs are well studied (starting from the seminal paper by Haken on the length of resolution proofs (Haken, 1985)). Our model of space complexity is based on (Alekhnovich et al., 2002). We view the process of producing a proof of G from K_i as a sequence of *configurations* or states of a reasoner, starting from an empty configuration, and producing the next configuration by one of the following operations:

Read copies a formula from K_i into the current configuration (possibly overwriting a formula from the previous configuration)

Infer applies a rule from R_i to formulas in the current configuration (possibly overwriting a formula from the previous configuration)

The sequence of configurations constitutes a proof of G if G appears in the last configuration. Time complexity corresponds to the length of the sequence, and space complexity to the size of configurations.[1] The size of a configuration can be measured either in terms of the number of formulas appearing in the configuration or in terms of the number of symbols required to represent the configuration. Clearly, for some inference systems, for example, where the set of inference rules contains both conjunction introduction and conjunction elimination, the first size measure results in constant space usage. However, for other systems, such as resolution, counting formulas results in non-trivial space complexity (Esteban and Torán, 1999). In this paper, we take the size of a configuration to be the maximal number of formulas, since all the reasoning systems we consider have a non-trivial space complexity for this measure.

#	Configuration	Operation
1	$\{\,\}$	
2	$\{A_1\}$	**Read**
3	$\{A_1, A_2\}$	**Read**
4	$\{A_1, A_1 \wedge A_2\}$	**Infer**
5	$\{A_1 \wedge A_2, A_1 \wedge A_2 \rightarrow B_1\}$	**Read**
6	$\{A_1 \wedge A_2, B_1\}$	**Infer**

Figure 1. Example derivation using $\bigwedge_I$ and MP

[1] Note that we deviate from Alekhnovich et al. (2002) in that we do not have an explicit *erase* operation, preferring to incorporate erasing (overwriting) in the *read* and *infer* operations. This obviously results in shorter proofs; however including an explicit erase operation gives proofs which are no more than twice as long as our proofs if we do not require the last configuration to contain *only* the goal formula.

As an illustration, Figure 1 shows the space and time complexity of the derivation of the formula B_1 from $A_1,\ A_2,\ A_1 \wedge A_2 \rightarrow B_1$ in an inference system which contains only conjunction introduction and modus ponens. The length of the proof is 6 and the space usage is 2 (at most 2 formulas need to be present in the configuration at any given time). It is worth observing that the inference system consisting of just conjunction introduction and modus ponens does not have constant space complexity when space is measured as the number of formulas; a sequence of derivation examples requiring (logarithmically) growing space can easily be constructed starting from the example above, and continuing with a derivation of C_1 from $A_1, A_2, A_3, A_4, A_1 \wedge A_2 \rightarrow B_1, A_3 \wedge A_4 \rightarrow B_2, B_1 \wedge B_2 \rightarrow C_1$, etc.

#	Configuration	Operation
1	$\{\ \}$	
2	$\{A_1 \vee A_2\}$	**Read**
3	$\{A_1 \vee A_2, \neg A_1 \vee A_2\}$	**Read**
4	$\{A_1 \vee A_2, A_2\}$	**Infer**
5	$\{A_2, A_1 \vee \neg A_2\}$	**Read**
6	$\{A_2, A_1 \vee \neg A_2, \neg A_1 \vee \neg A_2\}$	**Read**
7	$\{A_2, \neg A_2, \neg A_1 \vee \neg A_2\}$	**Infer**
8	$\{\emptyset, \neg A_2, \neg A_1 \vee \neg A_2\}$	**Infer**

Figure 2. Example derivation using resolution

Most research in time and space complexity of proofs has focused on the lower bounds for the inference system as a whole. While we are interested in the lower bounds, we are also interested in the trade-offs between time and space usage for particular derivations. For example, consider a set of premises $A_1, A_2, A_3, A_4, A_1 \wedge A_2 \rightarrow B_1, A_3 \wedge A_4 \rightarrow B_2, B_1 \wedge B_2 \rightarrow C_1$ and a goal formula $A_1 \wedge A_2 \wedge C$. It is possible to derive the goal from the premises using conjunction introduction and modus ponens and configurations of size 3 in 17 steps (deriving $A_1 \wedge A_2$ twice). On the other hand, with configurations of size 4 the proof is 3 steps shorter.

Different inference systems have different complexity and different trade-offs. Figure 2 illustrates the (non-trivial) space complexity of resolution proofs in terms of the number of formulas in a configuration. The example, which is due to Esteban and Torán (1999), shows the derivation of an empty clause by resolution from the set of all possible clauses of the form

$$\sim A_1 \vee \sim A_2 \vee \ldots \vee \sim A_n$$

(where $\sim A_i$ is either A_i or $\neg A_i$), for $n = 2$. Its space usage is 3 and the length of the proof is 8.

In the multiagent case, when several reasoners can communicate to derive a common goal, an additional resource of interest is how many messages the reasoners must exchange in order to derive the goal. In the distributed setting, we assume that each agent has its own set of premises and inference rules and its own configuration, and that the reasoning of the agents proceeds in lock step. In addition to **Read** and **Infer**, each reasoner can perform two extra operations:

Skip which leaves its configuration unchanged

Copy if agent i has a formula A in its current configuration, then agent j can copy it to its next configuration

The goal formula is derived if it occurs in the configuration of one of the agents. Our model of communication complexity is based on (Yao, 1979), except that we count the number of formulas exchanged by the agents rather than the number of bits exchanged. The communication complexity of a joint derivation is then the (total) number of **Copy** operations in the derivation.

	Agent 1		Agent 2	
#	Configuration	Op.	Configuration	Op.
1	$\{\}$		$\{\}$	
2	$\{A_1 \vee A_2\}$	**Read**	$\{A_1 \vee \neg A_2\}$	**Read**
3	$\{A_1 \vee A_2, \neg A_1 \vee A_2\}$	**Read**	$\{\neg A_1 \vee \neg A_2, A_1 \vee \neg A_2\}$	**Read**
4	$\{A_1 \vee A_2, A_2\}$	**Infer**	$\{\neg A_2, A_1 \vee \neg A_2\}$	**Infer**
5	$\{A_1 \vee \neg A_2, A_2\}$	**Read**	$\{\neg A_2, A_2\}$	**Copy**
6	$\{A_1, A_2\}$	**Infer**	$\{\{\}, A_2\}$	**Infer**

Figure 3. Example derivation using resolution with two agents

In general, in a distributed setting, trade-offs are possible between the number of messages exchanged and the space (size of a single agent's configuration) and time required for a derivation. The total space use (the total number of formulas in all agent's configurations) clearly cannot be less than the minimal configuration size required by a single reasoner to derive the goal formula from the union of all knowledge bases using all of the available inference rules, however this can be distributed between the agents in different ways, resulting in different numbers of exchanged messages. Similarly, if parts of a derivation can be performed in parallel, the total derivation will be shorter, though communication of the partial results will increase the communication complexity. As an illustration, Figure 3 shows one possible distribution of the resolution example in Figure 2. As can be seen, two communicating agents can solve the problem faster than a single agent.

4. A bounded memory and communication logic BMCL-CTL

In this section we present $BMCL\text{-}CTL$, a temporal epistemic logic which allows us to describe a set of reasoning agents with bounds on memory and on the number of messages they can exchange. In this logic, we can express statements like 'the agents will be able to derive the goal formula in n inference steps' (see Section 5 for the translation). The bounds on memory and communication ability are expressed as axioms in the logic. In this paper, as an example, we have chosen to axiomatise a set of agents reasoning using resolution. Other reasoning systems can be axiomatised in a similar way, and we briefly sketch how to add model conditions and axioms for reasoners which reason using conjunction introduction and modus ponens to our logic at the end of this section.

The language of the logic contains belief operators B_i, for each agent i. The meaning of the belief operator reflects the purpose for which the logic is designed: namely, $B_i\alpha$ is true if the formula α is in agent i's memory. We interpret $B_i\alpha$ syntactically (as a property of a formula ϕ, rather than of a proposition denoted by ϕ). This is inevitable since we consider resource-limited reasoning agents, and we cannot assume that the agents can instantaneously identify logically equivalent formulas. For the same reason, we do not interpret beliefs using an accessibility relation, since this would cause beliefs to be immediately closed under logical consequence. We also do not consider nested belief operators because we do not model agents reasoning about each other's beliefs. However it is possible to model agents which reason using positive introspection in a similar way, see for example (Alechina et al., 2008a). Since this is a logic for reasoning about the way beliefs of agents change over time, we feel justified in calling it an epistemic (or doxastic) temporal logic.

Let the set of agents be $Ag = \{1, 2, .., n_{Ag}\}$. For simplicity, we assume that they agree on a finite set $PROP$ of propositional variables (this assumption can easily be relaxed, so that only some propositional variables are shared). Since each agent uses resolution for reasoning, we assume that all formulas of the internal language of the agents are in the form of *clauses*. For convenience, we define a clause as a set of *literals* in which a literal is a propositional variable or its negation. Then the set of literals is defined as $LPROP = \{p, \neg p \mid p \in PROP\}$. If L is a literal, then $\neg L$ is $\neg p$ if L is a propositional variable p, and p if L is of the form $\neg p$. Let Ω be the set of all possible clauses over $PROP$, i.e., $\Omega = \wp(LPROP)$. Note that Ω is finite.

The only rule of inference that each agent has is the *resolution rule* which is defined as follows:

$$\frac{\alpha \qquad \beta}{(\alpha \setminus \{L\}) \cup (\beta \setminus \{\neg L\})}\ \text{Res}$$

where α and β are clauses, $L \in \alpha$ and $\neg L \in \beta$.

Each agent i has a memory of size $n_M(i)$ where one unit of memory corresponds to the ability to store an arbitrary clause. Each agent i has a knowledge base or a set of premises $K_i \subseteq \Omega$ and can read clauses from K_i. The communication ability of the agents is expressed by the *copy* action which copies a clause from another agent's memory. The limit on each agent's communication ability is $n_C(i)$: in any valid run of the system, agent i can perform at most $n_C(i)$ copy actions.

4.1. SYNTAX OF BMCL-CTL

The syntax of $BMCL\text{-}CTL$ is defined inductively as follows.

- $\top$ is a well-formed formula (wff) of $BMCL\text{-}CTL$.
- If α is a clause, then $B_i\alpha$ is a wff of $BMCL\text{-}CTL$, for all $i \in Ag$.
- $c_i = n$ is a wff of $BMCL\text{-}CTL$, for all $i \in Ag$ and $n \in \mathbb{N}$.
- If ϕ and ψ are wffs, then so are $\neg\phi$, $\phi \wedge \psi$.
- If ϕ and ψ are wffs, then so are $EX\phi$, $E(\phi U \psi)$, and $A(\phi U \psi)$.

Classical abbreviations for $\vee$, $\rightarrow$, $\leftrightarrow$ and $\bot$ are defined as usual.

The language contains both temporal and epistemic modalities. For the temporal part of $BMCL\text{-}CTL$, we have CTL, a branching time temporal logic. Intuitively, CTL describes infinite trees, or all possible runs of the system, over discrete time. In the temporal logic part of the language, X stands for next step, U for until, A for 'on all paths' and E for 'on some path'. We will also use abbreviations for other usual temporal operators AX, EF, AF, EG and AG, in which F stands for 'some time in the future' and G for 'always from now'. The epistemic part of the language consists of belief modalities $B_i\alpha$, which means that agent i has α in its memory.

For convenience, we define the following sets:

$$B_i\Omega = \{B_i\alpha \mid \alpha \in \Omega\},$$

$$B\Omega = \bigcup_{i \in Ag} B_i\Omega,$$

$$CP_i = \{c_i = n \mid n = 0, \ldots, n_C(i)\},$$

$$CP = \bigcup_{i \in Ag} CP_i.$$

4.2. SEMANTICS OF BMCL-CTL

The semantics of $BMCL\text{-}CTL$ is defined by $BMCL\text{-}CTL$ transition systems. A $BMCL\text{-}CTL$ transition system $M = (S, R, V)$ is defined as follows.

- S is a non-empty set of states.
- $R \subseteq S \times S$ is a serial binary relation, that is for all $s \in S$, there exists $s' \in S$ such that $(s, s') \in R$.
- $V : S \times Ag \rightarrow \wp(\Omega \cup CP)$; we define the 'belief part' of the assignment $V^B(s,i) = V(s,i) \setminus CP$ and the communication counter part $V^C(s,i) = V(s,i) \cap CP$. V satisfies the following conditions:
  1. $|V^C(s,i)| = 1$ for all $s \in S$ and $i \in Ag$.
  2. If $(s,t) \in R$ and $c_i = n \in V(s,i)$ and $c_i = m \in V(t,i)$ then $n \leq m$.

For each model $M = (S, R, V)$, a path in M is a sequence of states $(s_0, s_1, \ldots)$ in which $(s_k, s_{k+1}) \in R$ for all $k \geq 0$.

The truth of a $BMCL$-CTL formula at a state $s \in S$ of a model $M = (S, R, V)$ is defined inductively as follows.

- $M, s \models B_i\alpha$ iff $\alpha \in V(s,i)$.
- $M, s \models c_i = n$ iff $c_i = n \in V(s,i)$.
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$.
- $M, s \models \phi \wedge \psi$ iff $M, s \models \phi$ and $M, s \models \psi$.
- $M, s \models EX\phi$ iff there exists $s' \in S$ such that $(s, s') \in R$ and $M, s' \models \phi$.
- $M, s \models E(\phi U \psi)$ iff there exists a path $(s_0, s_1, \ldots, s_n, \ldots)$ in M with $s = s_0$ and $n \geq 0$ such that $M, s_k \models \phi$ for all $k = 0, \ldots, n-1$ and $M, s_n \models \psi$.
- $M, s \models A(\phi U \psi)$ iff for all paths $(s_0, s_1, \ldots)$ in M with $s = s_0$, there exists $n \geq 0$ such that $M, s_k \models \phi$ for all $k = 0, \ldots, n-1$ and $M, s_n \models \psi$.

Now we describe conditions on the models. The first set of conditions refers to the accessibility relation R. The intuition behind the conditions is that R corresponds to the agents executing actions $\langle a_1, \ldots, a_{n_{Ag}} \rangle$ in parallel, where action a_i is a possible action (transition) for the agent i in a given state. The actions an agent i can perform are: $Read_{i,\alpha,\beta}$ (reading a clause α from the knowledge base and erasing β), $Res_{i,\alpha_1,\alpha_2,L,\beta}$ (resolving α_1 and α_2 on L and erasing β), $Copy_{i,\alpha,\beta}$ (copying α from another agent and erasing β), and $Idle_i$ (doing nothing), where $\alpha, \alpha_1, \alpha_2, \beta \in \Omega$ and $L \in LPROP$. Intuitively, β is an arbitrary clause which gets overwritten if it is in the agent's memory. If the agent's memory is full ($|V^B(s,i)| = n_M(i)$), then we require that β has

to be in $V^B(s,i)$. Not all actions are possible in any given state. For example, to perform a resolution step from state s, the agent has to have two resolvable clauses in s. The message counter of each agent i starts with the value 0 and is incremented every time i copies a clause. When the value of the counter becomes equal to $n_C(i)$, i cannot execute the $Copy$ action any more.

Let us denote the set of all possible actions by agent i in state s by $R_i(s)$. Below is the definition of $R_i(s)$:

DEFINITION 1 (Available actions). *For every state s and agent i,*

1. $Read_{i,\alpha,\beta} \in R_i(s)$ *iff* $\alpha \in K_i$ *and* $\beta \in \Omega$, *or if* $|V^B(s,i)| = n_M(i)$ *then* $\beta \in V^B(s,i)$,
2. $Res_{i,\alpha_1,\alpha_2,L,\beta} \in R_i(s)$ *iff* $\alpha_1, \alpha_2 \in \Omega$, $\alpha_1 \ni L$, $\alpha_2 \ni \neg L$, $\alpha_1, \alpha_2 \in V(s,i)$, $\alpha = (\alpha_1 \setminus \{L\}) \cup (\alpha_2 \setminus \{\neg L\})$; β *is as before,*
3. $Copy_{i,\alpha,\beta} \in R_i(s)$ *iff there exists* $j \neq i$ *such that* $\alpha \in V(s,j)$ *and* $c_i = n \in V(s,i)$ *for some* $n < n_C(i)$; β *is as before,*
4. $Idle_i$ *is always in* $R_i(s)$.

Now we define effects of actions on the agent's state, i.e., the assignment $V(s,i)$.

DEFINITION 2 (Effects of actions). *For each $i \in Ag$, the result of performing an action a in state s is defined if $a \in R_i(s)$ and has the following effect on the assignment of clauses to i in the successor state t:*

1. *if a is $Read_{i,\alpha,\beta}$:* $V(t,i) = V(s,i) \cup \{\alpha\} \setminus \{\beta\}$,
2. *if a is $Res_{i,\alpha_1,\alpha_2,L,\beta}$:* $V(t,i) = V(s,i) \cup \{\alpha\} \setminus \{\beta\}$ *where* $\alpha = (\alpha_1 \setminus \{L\}) \cup (\alpha_2 \setminus \{\neg L\})$,
3. *if a is $Copy_{i,\alpha,\beta}$, $c_i = n \in V(s,i)$ for some n:* $V(t,i) = V(s,i) \cup \{\alpha, c_i = (n+1)\} \setminus \{\beta, c_i = n\}$,
4. *if a is $Idle_i$:* $V(t,i) = V(s,i)$.

DEFINITION 3. $BMCM(K_1, .., K_{n_{Ag}}, n_M, n_C)$ *is the set of models* $M = (S, R, V)$ *such that:*

1. *For every s and t, $R(s,t)$ iff for some tuple of actions $\langle a_1, \ldots, a_{n_{Ag}} \rangle$, $a_i \in R_i(s)$ and the assignment in t satisfies the effects of a_i for every i in $\{1, \ldots, n_{Ag}\}$,*
2. *For every s and a tuple of actions $\langle a_1, \ldots, a_{n_{Ag}} \rangle$, if $a_i \in R_i(s)$ for every i in $\{1, \ldots, n_{Ag}\}$, then there exists $t \in S$ such that $R(s,t)$ and t satisfies the effects of a_i for every i in $\{1, \ldots, n_{Ag}\}$,*

3. *The bound on each agent's memory is set by the following constraint on the mapping V:*

$$|V^B(s,i)| \leq n_M(i) \textit{ for all } s \in S \textit{ and } i \in Ag.$$

Note that the bound $n_C(i)$ on each agent i's communication ability (no branch contains more than $n_C(i)$ $Copy$ actions by agent i) follows from the fact that $Copy_i$ is only enabled if i has performed fewer than $n_C(i)$ copy actions in the past.

4.3. AXIOMATISATION OF BMCL-CTL

The idea of the axiomatisation is as follows. Given that the internal language Ω of the agents is finite, we can describe the contents of the memory of each agent in each state by a formula, and describe all possible successor states of each state in the same way.

We can identify a state s with the set of all atomic formulas true in s. The set of all possible state descriptions x for a given Ω is given below:

$$\hat{S} = \{x \in \wp(B\Omega \cup CP) \mid |x \cap B\Omega| \leq n_M(i) \wedge |x \cap CP| = 1\}$$

(recall the definitions of $B\Omega$ and CP given at the end of Section 4.1). An example of an element x of $\hat{S}$ would be a set $\{B_1 p,\ c_1 = 0,\ B_2 q,\ c_2 = 0\}$ which describes a state where agent 1 believes p, agent 2 believes q, and none of the agents has copied any clauses. We implicitly assume that for every atomic formula $\phi \in \wp(B\Omega \cup CP) \setminus x$, ϕ is false in the corresponding state. So in particular, x above describes a state where agent 1 does not believe q.

Given $x \in \hat{S}$, we will denote by x_i the formulas in x which describe i's beliefs and the value of i's communication counter:

$$x_i = (x \cap B_i\Omega) \cup (x \cap CP_i)$$

For the example above, $x_1 = \{B_1 p, c_1 = 0\}$.

We define $R_i(x)$ similarly to Definition 1 for all $x \in \hat{S}$. Then, for each $a \in R_i(x)$, we define an effect function $a(x)$ as follows:

- $Read_{i,\alpha,\beta}(x) = x_i \cup \{B_i\alpha\} \setminus \{B_i\beta\}$,
- $Res_{i,\alpha_1,\alpha_2,L,\beta}(s) = x_i \cup \{B_i\alpha\} \setminus \{B_i\beta\}$ where $\alpha = (\alpha_1 \setminus \{L\}) \cup (\alpha_2 \setminus \{\neg L\})$,
- $Copy_{i,\alpha,\beta}(s) = x_i \cup \{B_i\alpha,\ c_i = (n+1)\} \setminus \{B_i\beta,\ c_i = n\}$,
- $Null_i(s) = x_i$.

The effect function for a joint action $a = \langle a_1, \ldots, a_{n_{Ag}} \rangle$ is defined as the union of the individual effect functions, i.e. $a(x) = \bigcup_{i \in Ag} a_i(x)$. We also define $R(x) = \{\langle a_1, \ldots, a_{n_{Ag}} \rangle \mid a_i \in R_i(x)\}$.

Consider the following set of axiom schemata:

A1 Axioms and rules of CTL as given in (Emerson and Halpern, 1985),

A2 $\bigwedge_{\alpha \in \Gamma} B_i\alpha \rightarrow \neg B_i\beta$ for all $\Gamma \subseteq \Omega$ such that $|\Gamma| = n_M(i)$ and $\beta \notin \Gamma$,

A3 $\bigvee_{n=0,\ldots,n_C(i)} c_i = n$,

A4 $c_i = n \rightarrow \neg c_i = m$ for any $m \neq n$,

A5 $\bigwedge_{\phi \in x} \phi \wedge \bigwedge_{\psi \in \overline{x}} \neg\psi \rightarrow EX(\bigwedge_{\phi \in a(x)} \phi \wedge \bigwedge_{\psi \in \overline{a(x)}} \neg\psi)$ where $x \in \hat{S}$, $\overline{x} = (B\Omega \cup CP) \setminus x$, $a \in R(x)$ and $\overline{a(x)} = (B\Omega \cup CP) \setminus a(x)$,

A6 $\bigwedge_{\phi \in x} \phi \wedge \bigwedge_{\psi \in \overline{x}} \neg\psi \rightarrow AX(\bigvee_{a \in R(x)} (\bigwedge_{\phi \in a(x)} \phi \wedge \bigwedge_{\psi \in \overline{a(x)}} \neg\psi))$, where $x \in \hat{S}$, $\overline{x} = (B\Omega \cup CP) \setminus x$, $a \in R(x)$ and $\overline{a(x)} = (B\Omega \cup CP) \setminus a(x)$.

Let $BMCL - CTL(K_1, .., K_{n_{Ag}}, n_M, n_C)$ be the logic defined by the our axiomatisation. Then we have the following result.

THEOREM 1. *$BMCL$-$CTL(K_1, .., K_{n_{Ag}}, n_M, n_C)$ is sound and weakly complete with respect to $BMCM(K_1, .., K_{n_{Ag}}, n_M, n_C)$.*

Proof. The proof of soundness is standard. Axiom **A2** assures that at a state, each agent can store maximally at most $n_M(i)$ formulas in its memory. Meanwhile, axioms **A3** and **A4** force the presence of a unique counter for each agent to record the number of copies it has performed so far. In particular, **A3** makes sure that at least a counter is available for any agent and **A4** guaranties that only one of them is present. Axiom **A5** states that if a joint action is available, then there exists a 'next' state which is the result of that action on the current state. Finally, **A6** guaranties that 'next' states can only be generated by their predecessor based on available joint actions.

The proof of completeness follows that for CTL in (Emerson and Halpern, 1985). That is, we construct a model for a consistent formula, then use the axioms to show that this model is in $BMCM(K_1, \ldots, K_{n_{Ag}}, n_M, n_C)$.

Given a consistent formula φ_0, we construct the generalised Fischer-Ladner closure of φ_0, $FL(\varphi_0)$, as the least set H of formulas containing φ_0 such that:

1. $\neg\varphi \in H$, then $\varphi \in H$,

2. $\varphi \wedge \psi \in H$, then $\varphi, \psi \in H$,
3. $E(\varphi U \psi) \in H$, then $\varphi, EXE(\varphi U \psi) \in H$,
4. $A(\varphi U \psi) \in H$, then $\varphi, AXA(\varphi U \psi) \in H$,
5. $EX\varphi \in H$, then $\varphi \in H$,
6. $AX\varphi \in H$, then $\varphi \in H$,
7. $\varphi \in H$ and φ is not of the form $\neg\psi$, then $\neg\varphi \in H$,
8. $B\Omega,\ CP \subseteq H$

Clearly, $FL(\varphi_0)$ is finite. A subset s of $FL(\varphi_0)$ is maximally consistent if s is consistent and for all φ, $\neg\varphi \in FL(\varphi_0)$, either φ or $\neg\varphi$ is in s. Repeat the construction of a model M for φ_0 as in (Emerson and Halpern, 1985) based on the set of maximally consistent sets of $FL(\varphi_0)$, with the condition that the assignment is defined as follows:

- $\alpha \in V(s,i)$ iff $B_i\alpha \in s$,
- $c_i = n \in V(s,i)$ iff $c_i = n \in s$.

Note that $|V^B(s,i)| \leq n_M(i)$ by axiom **A2**, and $|V^C(s,i)| = 1$ by axioms **A3** and **A4**.

The truth lemma for atomic formulas is trivial, for temporal formulas the proof is the same as in (Emerson and Halpern, 1985).

It remains to show that M satisfies the conditions on R and if $(s,t) \in R$ and $c_i = n \in V(s,i)$ and $c_i = m \in V(t,i)$ then $n \leq m$. This follows from the axioms **A5** and **A6** which exhaustively describe all, and only, the transitions which conform to the conditions on R. In particular, the value of communication counter never goes down. □

4.4. SYSTEMS OF HETEROGENEOUS REASONERS

Changing the logic to accommodate agents which reason using a different set of inference rules rather than resolution is relatively straightforward. As an illustration, we show how to add model conditions and axioms for agents which use modus ponens and conjunction introduction. We assume that the premises of these reasoners contain literals and implications of the form $L_1 \wedge \ldots \wedge L_n \rightarrow L$.

First of all, we need to change the conditions on models so that instead of using the Res action, an agent can change its state by performing MP and AND actions. Let i be an (MP, AND) reasoner. Define Ω as $\cup_i K_i$ closed under subformulas and the following conjunction introduction: if Q

is a set of distinct literals from $\cup_i K_i$, then $\wedge Q \in \Omega$. An agent i has actions $Read_{i,\phi,\psi}$ for any formula ϕ in K_i and $\psi \in \Omega$ (where ψ is the overwritten formula), $Copy_{i,\phi,\psi}$ for any formula $\phi \in \Omega$, $Idle_i$, and instead of Res it has $MP_{i,\phi_1,\phi_1\rightarrow\phi_2,\psi}$ and $AND_{i,\phi_1,\phi_2,\psi}$.

DEFINITION 4 (Availability of MP and AND). *For any $s \in S$:*

1. $MP_{i,\phi_1,\phi_1\rightarrow\phi_2,\psi} \in R_i(s)$ *iff* $\phi_1, \phi_1 \rightarrow \phi_2 \in V(s,i)$ *and* $\psi \in \Omega$,
2. $AND_{i,\phi_1,\phi_2,\psi} \in R_i(s)$ *iff* $\phi_1, \phi_2 \in V(s,i)$ *and* $\psi \in \Omega$.

DEFINITION 5 (Effects of MP and AND). *For every $s \in S$, the result of performing action a is defined if $a \in R_i(s)$ and has the following effect on the resulting state t:*

1. *If a is $MP_{i,\phi_1,\phi_1\rightarrow\phi_2,\psi}$ then* $V(t,i) = V(s,i) \cup \{\phi_2\} \setminus \{\psi\}$,
2. *If a is $AND_{i,\phi_1,\phi_2,\psi}$ then* $V(t,i) = V(s,i) \cup \{\phi_1 \wedge \phi_2\} \setminus \{\psi\}$.

The corresponding notations $R_i(x)$, $MP_{i,\phi_1,\phi_1\rightarrow\phi_2,\psi}(x)$ and $AND_{i,\phi_1,\phi_2,\psi}(x)$ for the (MP, AND) reasoner are defined similarly, where $x \in \hat{S}$. In particular, we have:

- $MP_{i,\phi_1,\phi_1\rightarrow\phi_2,\psi}(x) = x_i \cup \{\phi_2\} \setminus \{\psi\}$,
- $AND_{i,\phi_1,\phi_2,\psi}(x) = x_i \cup \{\phi_1 \wedge \phi_2\} \setminus \{\psi\}$.

Then, the corresponding axioms for the (MP, AND) reasoners can be easily adapted from the system for resolution reasoners and we obtain an axiomatisation for the heterogeneous system of reasoners.

5. Verifying Resource Bounds

The logic $BMCL$-CTL allows us to express precisely how the beliefs of a set of resource-bounded agents change over time, and, given a memory and communication bound for each agent, to verify formulas which state that a certain belief will or will not be acquired within a certain number of steps. For example, given a system of two agents which reason using resolution from premises $K_1 = \{\{p_1, p_2\}, \{\neg p_1, p_2\}\}$ and $K_2 = \{\{p_1, \neg p_2\}, \{p_1, \neg p_2\}\}$, with bounds $n_M(1) = 2$, $n_M(2) = 2$ (both agents have a memory of size 2) and $n_C(1) = 0$, $n_C(2) = 1$ (agent 1 cannot copy anything and agent 2 can copy one clause), we can prove $EX^5 B_2(\{\})$ (i.e., the agents can derive the empty clause in 5 steps—recall that we consider the goal achieved if one of the agents has the formula in its memory).

However, rather than deriving such properties by hand, it is more convenient to use an automatic method to verify them. In this section, we describe how the models in $BMCM(K_1, .., K_{n_{Ag}}, n_M, n_C)$ can be encoded as an input to a model checker to allow the automatic verification of the properties expressing resource bounds using standard model checking techniques.

The model checking complexity of $BMCL\text{-}CTL$ is the same as that of CTL, since all the additional constructs in the language are constant-time checkable state properties:

THEOREM 2. *The model checking problem for $BMCL\text{-}CTL$ (given M and ϕ, does $M \models \phi$) can be solved in time $O(|M| \times |\phi|)$.*

Proof. Direct from the CTL model checking algorithm from (Clarke et al., 1986). □

5.1. MODEL CHECKER ENCODING

It is straightforward to encode a $BMCM$ model of such a system for a standard model checker. For the examples reported here, we have used the Mocha model checker (Alur et al., 1998).

States of the $BMCM$ models correspond to an assignment of values to state variables in the model checker. The state variables representing an agent's memory are organised as a collection of 'cells', each holding at most one clause. For an agent i with memory bound $n_M(i)$, there are $n_M(i)$ cells. Each cell is represented by a pair of vectors of state variables, each of length $k = |PROP|$, representing the positive and negative literals in the clause in some standard order (e.g., lexicographic order). For example, if $PROP$ contains the propositional variables A_1, A_2 and A_3 with index positions 0, 1 and 2 respectively, the clause $A_1 \vee \neg A_3$ would be represented by two vectors of state variables: "100" for the positive literals and "001" for the negative literals. We assume that the premises do not include any tautologies, and our encoding of resolution (see below) does not produce tautologies as resolvents. The index corresponding to a propositional variable is therefore set to 1 in at most one of the vectors representing a memory cell. This gives reasonably compact states.

Actions by each agent such as reading a premise, resolution and communication with other agents are represented by Mocha *atoms* which describe the initial condition and transition relation for a group of related state variables. Reading a premise ($Read_{i,\alpha,\beta}$) simply sets the vectors representing an arbitrary cell in agent i's memory to the appropriate values for the clause α. Resolution ($Res_{i,\alpha_1,\alpha_2,L,\beta}$) is implemented using simple bit operations on cells containing values representing α_1 and α_2, with the results being assigned to an arbitrary cell in agent i's memory. Communication ($Copy_{i,\alpha,\beta}$)

is implemented by copying the values representing α from a cell of agent j to an arbitrary cell of agent i. To express the communication bound, we use a counter for each agent which is incremented each time a copy action is performed by the agent. After the counter for agent i reaches $n_C(i)$, the $Copy_{i,\alpha,\beta}$ action is disabled.

Mocha supports hierarchical modelling through composition of *modules*. A module is a collection of atoms and a specification of which of the state variables updated by those atoms are visible from outside the module. In our encoding, each agent is represented by a module. A particular distributed reasoning system is then simply a parallel composition of the appropriate agent modules.

The specification language of Mocha is ATL (Alur et al., 1997), which includes CTL. We can express properties such as 'agent i may derive belief α in n steps' as $EX^n\ tr(B_i\alpha)$ where EX^n is EX repeated n times and $tr(B_i\alpha)$ is a suitable encoding of the fact that the clause α is present in agent i's memory, either as a disjunction of possible values of cell vectors or as a special boolean variable which becomes true when one of the cells contains a particular value. For example, if α is the empty clause, then both of the vectors of state variables representing one of agent i's cells should contain all 0s. (In practice, the situation is slightly more complex, as we need to check that a memory cell which contains all 0s at the current step was actually used in the proof, i.e., it contained a literal at the previous step.) To obtain the actual derivation we can verify the negation of a formula, for example $AG\ \neg tr(B_i\alpha)$, and use the counterexample trace generated by the model checker to show how the system reaches the state where α is proved. Note that we are interested in the properties involving simple reachability (there is a path to a state where one of the agents has derived a property) rather than in more complicated properties of strategic ability (there is a sequence of actions by the agents, such that for all responses by other agents, a state satisfying the property can be reached). In future work, we plan to consider logics for resource-bounded reasoners where explicit coalitional ability can be expressed.

5.2. EXAMPLES

Consider a single agent (agent 1) whose knowledge base contains all clauses of the form $\sim A_1 \vee \sim A_2$ where $\sim A_i$ is either A_i or $\neg A_i$, and which has the goal of deriving the empty clause. We can express the property that agent 1 will derive the empty clause at some point in the future as $EF\ B_1\{\}$.

Using the model checker, we can show that deriving the empty clause requires a memory bound of 3 and 8 time steps (see Figure 2).[2] We can also

[2] The space required for problems of this form is known to be logarithmic in the number of premises (Esteban and Torán, 1999).

Table I. Tradeoffs between resource bounds

# agents	Distrib.	Memory	Comm.	Time
1	Symmetric	3	–	8
2	Symmetric	2, 2	1, 0	6
2	Symmetric	3, 3	1, 0	6
2	Symmetric	3, 3	0, 0	8
2	Symmetric	2, 1	1, 1	9
2	Asymmetric	2, 2	2, 1	7
2	Asymmetric	3, 3	2, 1	7
2	Asymmetric	3, 1	1, 0	8

show that these space and time bounds are minimal for a single agent; i.e., increasing the space bound does not result in a shorter proof.

With two agents and a symmetric problem distribution (i.e., each agent has all the premises $\sim A_1 \vee \sim A_2$), we can show that a memory bound of 2 (i.e., the minimum required for resolution) and a communication bound of 1 gives a proof of 6 steps (see Figure 3). Reducing the communication bound to 0 results in no proof, as, with a memory bound of 2 for each agent, at least one clause must be communicated from one agent to the other. Increasing the space bound to 3 (for each agent) does not shorten the proof, though it does allow the communication bound to be reduced to 0 at the cost of increasing the proof length to 8 (i.e., the single agent case). Reducing the total space bound to 3 (i.e., 2 for one agent and 1 for the other, equivalent to the single agent case) increases the number of steps required to find a proof to 9 and the communication bound to 1 for each agent. In effect, one agent functions as a cache for a clause required later in the proof, and this clause must be copied in both directions.

If the problem distribution is asymmetric, e.g., if one agent has premises $A_1 \vee A_2$ and $\neg A_1 \vee \neg A_2$ and the other has premises $\neg A_1 \vee A_2$ and $A_1 \vee \neg A_2$, then with a memory bound of 2 for each agent, we can show that the time bound is 7, and the communication bound is 2 for the first agent and 1 for the second. Increasing the memory bound for each agent to 3 does not reduce the time bound. However the memory bound can be reduced to 1 and the communication bound reduced to 1 for one agent and 0 for the other, if the time bound is increased to 8 (this is again equivalent to the single agent case, except that one agent copies the clause it lacks from the other rather than reading it). These tradeoffs are summarised in Table I.

Increasing the size of the problem increases the number of possible trade-offs, but similar patterns can be seen to the 2-variable case. For example, if the agent's premises contain all clauses of the form $\sim A_1 \vee \sim A_2 \vee \sim A_3$, then a single agent requires a memory bound of 4 and 16 steps to achieve the goal. In comparison, two agents, each with a memory bound of 2, require 13 steps and 4 messages to derive the goal.

While extremely simple, these examples serve to illustrate the interaction between memory, time and communication bounds, and between the resource distribution and the problem distribution.

6. Related work

There exist several approaches to epistemic logic which model reasoners as resource-bounded (not logically omniscient), including the deduction model of belief (Konolige, 1986), step logic and active logic (Elgot-Drapkin and Perlis, 1990; Grant et al., 2000), algorithmic knowledge (Halpern et al., 1994; Fagin et al., 1995; Pucella, 2004), and other syntactic epistemic logics (Duc, 1997; Ågotnes and Walicki, 2005; Alechina et al., 2004; Jago, 2006) where each inference step takes the agent into the next (or some future) moment in time. A logic where the depth of belief reasoning is limited is studied in (Fisher and Ghidini, 1999).

A considerable amount of work has also been done in the area of model checking multi-agent systems (see, e.g., (Bordini et al., 2004; Benerecetti et al., 1998)). However, this work lacks a clear connection between the way agent reasoning is modelled in the agent theory (which typically assumes that the agents are logically omniscient) and the formalisations used for model checking, and emphasises correctness rather than the interplay between time, memory and communication, and the ability of agents to derive a certain belief.

The current paper extends the work of Albore et al. (2006) which proposed a method of verifying memory and time bounds in a single reasoner which reasons in classical logic using natural deduction rather than resolution. We also extend the work in (Alechina et al., 2006c) which analyses a system of communicating rule-based reasoners and verifies time bounds for those systems, but assumes unlimited memory.

7. Conclusion

In this paper, we analyse the time, space and communication resources required by a system of reasoning agents to achieve a goal. We give a rigourous definition of the measures for each of those resources, and introduce the

epistemic logic $BMCL\text{-}CTL$ in which we can express properties of a system of resource-bounded reasoning agents. In particular, we can express bounds on memory and communication resources as axioms in the logic. We axiomatise a system of agents which reason using resolution (other reasoning systems can be axiomatised in a similar way), prove that the resulting logic is sound and complete, and show how to express properties of the system of reasoning agents in $BMCL\text{-}CTL$. Finally, we show how $BMCL\text{-}CTL$ transition systems can be encoded as input to the Mocha model-checker and how properties, such as the existence of derivations with given bounds on memory, communication, and the number of inference steps, can be verified automatically.

In future work, we plan to consider logical languages containing primitive operators which would allow us to state the agents' resource limitations as formulas in the language rather than axioms, and consider agents reasoning about each other's resource limitations. We also would like to consider agents reasoning in a simple epistemic or description logic. We also plan to investigate extending ATL to include reasoning about resources.

References

Adjiman, P., P. Chatalic, F. Goasdoué, M.-C. Rousset, and L. Simon: 2004, 'Distributed Reasoning in a Peer-to-Peer Setting'. In: R. L. de Mántaras and L. Saitta (eds.): *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*. pp. 945–946, IOS Press.

Ågotnes, T. and N. Alechina: 2006, 'Knowing minimum/maximum n formulae'. In: G. Brewka, S. Coradeschi, A. Perini, and P. Traverso (eds.): *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006)*. pp. 317–321, IOS Press.

Ågotnes, T. and M. Walicki: 2005, 'Strongly Complete Axiomatizations of "Knowing At Most" in Standard Syntactic Assignments'. In: F. Toni and P. Torroni (eds.): *Preproceedings of the 6^{th} International Workshop on Computational Logic in Multi-agent Systems (CLIMA VI)*. London, UK.

Albore, A., N. Alechina, P. Bertoli, C. Ghidini, B. Logan, and L. Serafini: 2006, 'Model-checking memory requirements of resource-bounded reasoners'. In: *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*. pp. 213–218, AAAI Press.

Alechina, N., P. Bertoli, C. Ghidini, M. Jago, B. Logan, and L. Serafini: 2006a, 'Verifying space and time requirements for resource-bounded agents'. In: P. Stone and G. Weiss (eds.): *Proceedings of the Fifth International Joint Conference on Aut onomous Agents and Multi-Agent Systems (AAMAS 2006)*. Hakodate, Japan, pp. 217–219, IEEE Press.

Alechina, N., P. Bertoli, C. Ghidini, M. Jago, B. Logan, and L. Serafini: 2006b, 'Verifying space and time requirements for resource-bounded agents'. In: S. Edelkamp and A. Lomuscio (eds.): *Proceedings of the Fourth Workshop on Model Checking and Artif icial Intelligence (MoChArt-2006)*. pp. 16–30.

Alechina, N., M. Jago, and B. Logan: 2006c, 'Modal logics for communicating rule-based agents'. In: G. Brewka, S. Coradeschi, A. Perini, and P. Traverso (eds.): *Proceedings of*

the 17th European Conference on Artificial Intelligence (ECAI 2006). pp. 322–326, IOS Press.

Alechina, N., B. Logan, N. H. Nga, and A. Rakib: 2008a, 'Reasoning about other agents' beliefs under bounded resources'. In: J.-J. C. Meyer and J. Broersen (eds.): *Pre-proceedings of the KR2008 Workshop on Knowledge Representation for Agents and Multi-Agent Systems (KRAMAS 2008)*. Sydney, Australia, pp. 4–18.

Alechina, N., B. Logan, N. H. Nga, and A. Rakib: 2008b, 'Verifying time, memory and communication bounds in systems of reasoning agents'. In: L. Padgham, D. Parkes, J. Müller, and S. Parsons (eds.): *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Vol. 2. Estoril, Portugal, pp. 736–743, IFAAMAS.

Alechina, N., B. Logan, and M. Whitsey: 2004, 'A Complete and Decidable Logic for Resource-Bounded Agents'. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*. New York, pp. 606–613, ACM Press.

Alekhnovich, M., E. Ben-Sasson, A. A. Razborov, and A. Wigderson: 2002, 'Space Complexity in Propositional Calculus.'. *SIAM Journal of Computing* **31**(4), 1184–1211.

Alur, R., T. A. Henzinger, and O. Kupferman: 1997, 'Alternating-time temporal logic'. In: *Journal of the ACM*. pp. 100–109, IEEE Computer Society Press.

Alur, R., T. A. Henzinger, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Tasiran: 1998, 'MOCHA: Modularity in Model Checking'. In: *Computer Aided Verification*. pp. 521–525.

Amir, E. and S. A. McIlraith: 2005, 'Partition-based logical reasoning for first-order and propositional theories'. *Artificial Intelligence* **162**(1-2), 49–88.

Benerecetti, M., F. Giunchiglia, and L. Serafini: 1998, 'Model Checking Multiagent Systems.'. *J. Log. Comput.* **8**(3), 401–423.

Bordini, R., M. Fisher, W. Visser, and M. Wooldridge: 2004, 'State-space reduction techniques in agent verification'. In: N. R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe (eds.): *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2004)*. New York, NY, pp. 896–903, ACM Press.

Clarke, E. M., E. A. Emerson, and A. P. Sistla: 1986, 'Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications'. *ACM Trans. Program. Lang. Syst.* **8**(2), 244–263.

Duc, H.: 1997, 'Reasoning About Rational, but not Logically Omniscient, Agents'. *Journal of Logic and Computation* **5**, 633–648.

Elgot-Drapkin, J. J. and D. Perlis: 1990, 'Reasoning Situated in Time I: Basic Concepts'. *Journal of Experimental and Theoretical Artificial Intelligence* **2**, 75–98.

Emerson, E. A. and J. Y. Halpern: 1985, 'Decision Procedures and Expressiveness in the Temporal Logic of Branching Time'. *J. Comput. Syst. Sci.* **30**(1), 1–24.

Esteban, J. L. and J. Torán: 1999, 'Space Bounds for Resolution.'. In: C. Meinel and S. Tison (eds.): *STACS 99, 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March 4-6, 1999, Proceedings*, Vol. 1563 of *Lecture Notes in Computer Science*. pp. 551–560, Springer.

Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi: 1995, *Reasoning about Knowledge*. Cambridge, Mass.: MIT Press.

Faltings, B. and M. Yokoo: 2005, 'Introduction: Special Issue on Distributed Constraint Satisfaction'. *Artificial Intelligence* **161**(1-2), 1–5.

Fisher, M. and C. Ghidini: 1999, 'Programming Resource-Bounded Deliberative Agents'. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*. pp. 200–206, Morgan Kaufmann.

Grant, J., S. Kraus, and D. Perlis: 2000, 'A Logic for Characterizing Multiple Bounded Agents'. *Autonomous Agents and Multi-Agent Systems* **3**(4), 351–387.

Haken, A.: 1985, 'The intractability of resolution'. *Journal of Theoretical Computer Science* **39**(2–3), 297–308.

Halpern, J. Y., Y. Moses, and M. Y. Vardi: 1994, 'Algorithmic Knowledge'. In: R. Fagin (ed.): *Proceedings of the 5th Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, USA, March 1994*. pp. 255–266, Morgan Kaufmann.

Jago, M.: 2006, 'Logics for Resource-Bounded Agents'. Ph.D. thesis, University of Nottingham.

Jung, H. and M. Tambe: 2005, 'On Communication in Solving Distributed Constraint Satisfaction Problems'. In: M. Pechoucek, P. Petta, and L. Z. Varga (eds.): *Multi-Agent Systems and Applications IV, 4th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2005, Budapest, Hungary, September 15-17, 2005, Proceedings*, Vol. 3690 of *Lecture Notes in Computer Science*. pp. 418–429, Springer.

Konolige, K.: 1986, *A Deduction Model of Belief*. San Francisco, Calif.: Morgan Kaufmann.

Provan, G. M.: 2002, 'A Model-Based Diagnosis Framework for Distributed Embedded Systems'. In: D. Fensel, F. Giunchiglia, D. L. McGuinness, and M.-A. Williams (eds.): *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*. pp. 341–352, Morgan Kaufmann.

Pucella, R.: 2004, 'Deductive Algorithmic Knowledge'. In: *AI&M 1-2004, Eighth International Symposium on Artificial Intelligence and Mathematics, January 4-6, 2004, Fort Lauderdale, Florida, USA*.

Wooldridge, M. and P. E. Dunne: 2006, 'On the computational complexity of coalitional resource games'. *Artif. Intell.* **170**(10), 835–871.

Yao, A. C.-C.: 1979, 'Some Complexity Questions Related to Distributive Computing (Preliminary Report)'. In: *Conference Record of the Eleventh Annual ACM Symposium on Theory of Computing, 30 April-2 May, 1979, Atlanta, Georgia, USA*. pp. 209–213, ACM.