# Normative Multi-Agent Programs and Their Logics

**Article** · January 2009

DOI: 10.1007/978-3-642-05301-6_2 · Source: OAI

Mehdi Dastani
Utrecht University

John-jules Meyer
Utrecht University

# Pre-proceedings of the KR2008-workshop on Knowledge Representation for Agents and Multi-Agent Systems (KRAMAS), Sydney, September 2008

Edited by John-Jules Ch. Meyer and Jan Broersen

August 11, 2008

# Contents

# Preface

This collection presents the papers accepted for presentation at KRAMAS 2008, a workshop on Knowledge Representation for Agents and Multi-Agent Systems, held at KR 2008, Sydney, Australia, September 17, 2008. The collection will be used as the informal pre-proceedings to the workshop. The formal post-proceedings will appear as a volume in Springer's Lecture Notes in Computer Science Series.

The initiative was taken by this year's KR chairs to enhance cross-fertilization between the KR (Knowledge Representation and Reasoning) and agent communities. To enhance participation, in the KR schedule the workshop is conveniently 'sandwiched' between days with regular KR sessions. The topics solicited include:

- Knowledge Representation & Reasoning aspects of agent systems (languages, ontologies, techniques)
- Reasoning about (actions of) agents
- Reasoning methods (such as non-monotonic reasoning, abduction, argumentation, diagnosis, planning, decision-making under uncertainty, reasoning about preference, ...) applied to agents and multi-agent systems (MAS)
- Theory of negotiation, communication, cooperation, group decision-making, game theory for MAS
- Cognitive robotics
- Representations of other agents / opponent models
- Logics for intelligent agents and MAS
- Specification and verification techniques for agents
- Automated reasoning techniques for agent-based systems
- Logical foundations of agent-based systems, normative MAS and e-institutions
- Formal semantics of agent programming languages
- Formal techniques for agent-oriented programming and agent-oriented software engineering

We received 14 papers, of which 12 were deemed acceptable for presentation at the workshop by the program committee.

We are grateful to the members of the Program Committee for their service in reviewing papers and to the KR organization for taking the initiative for having KRAMAS and their support in its organization. Thanks, too, to Richard van de Stadt whose CyberChairPRO system was a very great help to us. More information can be found on http://www.cs.uu.nl/events/kramas2008/kramas.html

August 2008

John-Jules Ch. Meyer
Jan Broersen

## Workshop Organization

PROGRAM CHAIR

John-Jules Meyer

WORKSHOP CHAIRS

John-Jules Meyer (Utrecht University, The Netherlands)
Jan Broersen (Utrecht University, The Netherlands)

PROGRAM COMMITTEE

Thomas Agotnes (Berghen, Norway)
Natasha Alechina (Nottingham, UK)
Jamal Bentahar (Montreal, Canada)
Rafael Bordini (Durham, UK)
Jan Broersen (Utrecht, Netherlands)
Mehdi Dastani (Utrecht, Netherlands)
Giuseppe De Giacomo (Rome, Italy)
Hans van Ditmarsch (Otago, New Zealand)
Jurgen Dix (Clausthal, Germany)
Andreas Herzig (Toulouse, France)
Wiebe van der Hoek (Liverpool, UK)
Wojciech Jamroga (Clausthal, Germany)
Catholijn Jonker (Delft, Netherlands)
Yves Lesperance (York Univ., Toronto, Canada)
Alessio Lomuscio (London, UK)
Timothy Norman (Aberdeen, UK)
Henry Prakken (Utrecht, Netherlands)
Alessandro Ricci (Cesena, Italy)
Renate Schmidt (Manchester, UK)
Carles Sierra (Barcelona, Spain)
Francesca Toni (London, UK)
Rineke Verbrugge (Groningen, Netherlands)

# Reasoning about other agents' beliefs under bounded resources

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.** There exists a considerable body of work on epistemic logics for bounded reasoners where the bound can be time, memory, or the amount of information the reasoners can exchange. In much of this work the epistemic logic is used as a meta-logic to reason about beliefs of the bounded reasoners from an external perspective. In this paper, we present a formal model of a system of bounded reasoners which reason about each other's beliefs, and propose a sound and complete logic in which such reasoning can be expressed. Our formalisation highlights a problem of incorrect belief ascription in resource-bounded reasoning about beliefs, and we propose a possible solution to this problem, namely adding reasoning strategies to the logic.

## 1 Introduction

The purpose of this paper is to investigate a multi-agent epistemic logic which results from taking seriously the idea that agents have bounded time, memory and communication resources, and are reasoning about each other's beliefs. The main contribution of the paper is to generalise several existing epistemic logics for resource-bounded reasoners by adding an ability for reasoners to reason about each other's beliefs. We show that a problem of incorrect belief ascription arises as a result, and propose a possible solution to this problem.

To give the reader an idea where the current proposal fits into the existing body of research on epistemic logics for bounded reasoners, we include a brief survey of existing approaches, concentrating mostly on the approaches which have influenced the work presented here.

In standard epistemic logic (see e.g. [1, 2] for a survey) an agent's (implicit) knowledge is modelled as closed under logical consequence. This can clearly pose a problem when using an epistemic logic to model resource-bounded reasoners, whose set of beliefs is not generally closed with respect to their reasoning rules. Various proposals to modify possible worlds semantics in order to solve this problem of logical omniscience (e.g., introducing impossible worlds as in [3, 4], or non-classical assignment as in [5]) result in agent's beliefs still being logically closed, but with respect to a weaker logic.

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

Our work builds on another approach to solving this problem, namely treating beliefs as syntactic objects rather than propositions (sets of possible worlds). In [6], Fagin and Halpern proposed a model of limited reasoning using the notion of awareness: an agent explicitly believes only the formulas which are in a syntactically defined awareness set (as well as in the set of its implicit beliefs). Implicit beliefs are still closed under consequence, but explicit beliefs are not, since a consequence of explicit beliefs is not guaranteed to belong to the awareness set. However, the awareness model does not give any insight into the connection between the agent's awareness set and the agent's resource limitations, which is what we try to do in this paper.[1] Konolige [7] proposed a different model of non-omniscient reasoners, the deduction model of belief. Reasoners were parameterised with sets of rules which could, for example, be incomplete. However, the deduction model of belief still models beliefs of a reasoner as closed with respect to reasoner's deduction rules; it does not take into account the time it takes to produce this closure, or any limitations on the agent's memory. Step logic, introduced in [8], gives a syntactic account of beliefs as theories indexed by time points; each application of inference rules takes a unit of time. No fixed bound on memory was considered, but the issue of bounded memory was taken into account. An account of epistemic logic called algorithmic knowledge, which treats explicit knowledge as something which has to be computed by an agent, was introduced in [9], and further developed in e.g. [1, 10]. In the algorithmic knowledge approach, agents are assumed to possess a procedure which they use to produce knowledge. In later work [10] this procedure is assumed to be given as a set of rewrite rules which are applied to the agent's knowledge to produce a closed set, so, like Konolige's approach, algorithmic knowledge is concerned with the result rather than the process of producing knowledge. In [11, 12] Duc proposed logics for non-omniscient epistemic reasoners which will believe all consequences of their beliefs *eventually*, after some interval of time. It was shown in [13] that Duc's system is complete with respect to semantics in which the set of agent's beliefs is always finite. Duc's system did not model the agents' reasoning about each others' beliefs. Other relevant approaches where epistemic logics were given a temporal dimension and each reasoning step took a unit of time are, for example, [14], where each inference step is modelled as an action in the style of dynamic logic, and [15] which proposes a logic for verification of response-time properties of a system of communicating rule-based agents (each rule firing or communication takes a unit of time). In a somewhat different direction, [16] proposed a logic where agents reason about each others beliefs, but have no explicit time or memory limit; however there is a restriction on the depth of belief nestings (context switching by the agents). Epistemic logics for bounded-memory agents were investigated in, for example, [17–20], and the interplay between bounded recall and bounded memory (ability to store strategies of only bounded size) was studied in [21].

An epistemic logic BMCL for communicating agents with communication limits on the number of exchanged messages (and connections to space complexity of proofs and communication complexity) was investigated in [20]. In this paper we expand BMCL by adding rules for reasoning about other agents' beliefs, demonstrate that epistemic

[1] We also completely dispense with the notion of implicit beliefs.

reasoning done in resource-bounded fashion has an inherent problem of incorrect belief ascription, and propose the use of reasoning strategies as a solution to this problem.

## 2 Model of reasoning agents

The logic BMCL presented in [20] formalises reasoning about the beliefs of a system of reasoners who reason using propositional resolution and can exchange information to solve a problem together. The set up is similar to, for example, [22]. BMCL models each inference rule application as taking a single time step, introduces an explicit bound on the set of beliefs of each reasoner, and a bound on the number of messages the reasoners can exchange. In this paper, we generalise this approach by assuming that agents can also reason about each other's beliefs. Namely, they assume that other agents use a certain set of inference rules, and they reason about what another agent may believe at the next step. For example, if agent $A$ believes that agent $B$ believes two clauses $c_1$ and $c_2$ and these two clauses are resolvable to a clause $c$, and agent $A$ assumes that agent $B$ reasons using resolution, then it is reasonable for agent $A$ to believe that agent $B$ may believe $c$ at the next step.

We assume a set of $n$ agents. Each agent $i$ has a set of inference rules, a set of premises $KB_i$, and a *working memory*. To infer from the premises in $KB_i$, the relevant formulas must first be read into working memory. We assume that each agent's working memory is bounded by $n_M$, which is the maximal number of formulas an agent can believe at the same time. We also set a limit on the possible size of a formula, or rather on the depth of nesting of belief operators, $n_B$, and a limit, $n_C$, on the maximal number of communications an agent can make. For simplicity, we assume that these bounds are the same for all agents, but this can be easily relaxed by introducing functions $n_M(i)$, $n_B(i)$ and $n_C(i)$ which assign a different limit to each agent $i$.

The set of reasoning actions is as follows:

**Read KB:** an agent can retrieve information from its KB and put it into its working memory using the *Read* action. Since an agent has a fixed size memory, adding a formula to its memory may require erasing some belief already in memory (if the limit $n_M$ would otherwise be exceed). The same applies to other reasoning actions which add a new formula, in that adding a new formula may involve overwriting a formula currently in working memory.

**Resolution:** an agent can derive a new clause if it has two resolvable clauses in its memory.

**Copy:** an agent can communicate with another agent to request a clause from the memory of the other agent. We assume that communication is always successful if the other agent has the requested clause. If agent $A$ has clause $c$ in memory, then a copy by $B$ will result in agent $B$ believing that $A$ believes $c$. *Copy* is only enabled if the agent has performed fewer than $n_C$ copy actions in the past and the prefix of the resulting belief has nesting of at most $n_B$.

**Idle:** an agent may idle (do nothing) at any time step. This means that at the next time point of the system, the agent does not change its state of memory.

**Erase:** an agent may remove a formula from its working memory. This action is introduced for technical reasons to simplify the proofs.

In addition to the actions listed above, we introduce actions that enable agents to reason about other agents' beliefs, essentially epistemic axioms K (ascribing propositional reasoning to the other agent) and 4 (positive introspection about the agent's own beliefs, and ascribing positive introspection to other agents). The reasons we do not adopt for example KD45 are as follows. If the agent's knowledge base is inconsistent, we want it to be able to derive $B\bot$ (or $B[]$ where $[]$ is the empty clause). Negative introspection is also problematic in a resource-bounded setting, in that the agent may derive $\neg B\alpha$ if $\alpha$ is not in its current set of beliefs, and then derive $\alpha$ from its other beliefs, ending up with an inconsistent set of beliefs ($\neg B\alpha$ and $B\alpha$ by positive introspection from $\alpha$), even if its knowledge base is consistent. We could have adopted a restricted version of negative introspection (see, e.g., [12]) but in this paper we omit it for simplicity.

In addition to the reasoning actions listed above, we therefore add the following actions:

**Other's Resolution:** an agent $A$ can perform this action if it believes that another agent $B$ believes two resolvable clauses $c_1$ and $c_2$. Then $A$ can conclude that $B$ will believe in the resolvent clause $c$ of $c_1$ and $c_2$ in the next time point. As a general case, we can extend the chain *agent-believes ... agent-believes*. For example, if agent $A$ believes that agent $B$ believes that agent $C$ believes two resolvable clauses $c_1$ and $c_2$, then it is possible in the next time point that agent $A$ believes that agent $B$ believes that agent $C$ believes $c$ which is the resolvent of $c_1$ and $c_2$.

**Positive Introspection:** if an agent $A$ believes a clause $c$, it can perform this action to reach a state where it believes that it believes $c$.

**Other's Positive Introspection:** if an agent $A$ believes that another agent $B$ believes a clause $c$, it can perform this action to reach a state where it believes that $B$ believes that $B$ believes $c$.

The reasoning actions *Positive Introspection* and *Other's Positive Introspection* are only enabled if the derived formula has a depth of nesting of at most $n_B$.

Note that the assumption that the agents reason using resolution and positive introspection is not essential for the main argument of this paper. This particular set of inference rules has been chosen to make the logic concrete; we could have, for example, assumed that the agents reason using modus ponens and conjunction introduction instead of resolution. In what follows, we give a formal definition of an epistemic logic for communicating agents which reason in a step-wise, memory-bounded fashion using some well-defined set of inference rules.

## 3 Syntax and semantics of $ERBL$

In this section, we give the syntax and semantics of the logic $ERBL$ which formalises the ideas sketched in the previous section. ERBL (Epistemic Resource Bounded Logic) is an epistemic and temporal meta-language in which we can talk about beliefs expressed in the agents' internal language.

Let the set of agents be $A = \{1, 2, \ldots, n_A\}$. We assume that all agents agree on a finite set $PROP$ of propositional variables, and that all *belief formulas* of the internal language of the agents are in the form of *clauses* or clauses preceded by a prefix of belief operators of fixed length.

From the set of propositional variables, we have the definition of all literals as follows:

$$LPROP = \{p, \neg p \mid p \in PROP\}$$

Then, the set of all clauses is $\Omega = \wp(LPROP)$. Finally, the set of all belief formulas is defined as follows:

$$B\Omega ::= \{B_{i_1} \dots B_{i_k} c \mid c \in \Omega, 0 \le k \le n_B\},$$

where $i_j \in A$. Note that we only include in the set of belief formulas those whose belief operator nesting is limited by $n_B$. Therefore, $B\Omega$ is finite.

Each agent $i \in A$ is assumed to have a knowledge base $KB_i \subseteq B\Omega$.

For convenience, the negation of a literal $L$ is defined as $\neg L$, where:

$$\neg L = \begin{cases} \neg p & \text{if } L = p \text{ for some } p \in PROP \\ p & \text{if } L = \neg p \text{ for some } p \in PROP \end{cases}$$

The form of resolution rule which will be used in formal definitions below is as follows: given two clauses $c_1$ and $c_2 \in \Omega$ such that one contains a literal $L$ and the other has its negation $\neg L$, we can derive a new clause which is the union $c_1 \setminus \{L\} \cup c_2 \setminus \{\neg L\}$.

The syntax of $ERBL$ is then defined inductively as follows.

- $\top$ is a well-formed formula (wff) of $ERBL$.
- $start$ is a wff of $ERBL$; it is a marker for the start state.
- $cp_i^{=n}$ (the number of communication actions performed by agent $i$) is a wff of $ERBL$ for all $n = 0, \dots, n_C$, and $i \in A$; it is used as a communication counter in the language.
- If $\alpha \in B\Omega$, then $B_i\alpha$ (agent $i$ believes $\alpha$) is a wff of $ERBL$, $i \in A$.
- If $\varphi$ and $\psi$ are wffs, then so are $\neg\varphi$, $\varphi \wedge \psi$.
- If $\varphi$ and $\psi$ are wffs, then so are $X\varphi$ ($\varphi$ holds in the next moment of time), $\varphi U \psi$ ( $\varphi$ holds until $\psi$), and $A\varphi$ ($\varphi$ holds on all paths).

Classical abbreviations for $\vee$, $\rightarrow$, $\leftrightarrow$ are defined as usual. We also have $\bot \equiv \neg\top$, $F\varphi \equiv \top U \varphi$ ($\varphi$ holds some time in the future), $E\varphi \equiv \neg A \neg\varphi$ ($\varphi$ holds on some path). For convenience, let $CP_i = \{cp_i^{=n} | n = \{0, \dots, n_C\}\}$ and $CP = \bigcup_{i \in A} CP_i$.

The semantics of $ERBL$ is defined by $ERBL$ transition systems which are based on $\omega$-tree structures (standard CTL$^*$ models as defined in [23]).

Let $(T, R)$ be a pair where $T$ is a set and $R$ is a binary relation on $T$. Let the relation $<$ be the irreflexive and transitive closure of $R$, namely the set of pairs of states $\{(s,t) \in T \times T \mid \exists n \ge 0, t_0 = s, .., t_n = t \in T$ such that $t_i R t_{i+1}$ for all $i = 0, \dots, n-1\}$. $(T, R)$ is a $\omega$-tree frame iff the following conditions are satisfied.

1. $T$ is a non-empty set.
2. $R$ is total, i.e., for all $t \in T$, there exists $s \in T$ such that $tRs$.
3. For all $t \in T$, the past $\{s \in T \mid s < t\}$ is linearly ordered by $<$.
4. There is a smallest element called the root, denoted by $t_0$.

5. Each maximal linearly $<$- ordered subset of $T$ is order-isomorphic to the natural numbers.

A branch of $(T, R)$ is an $\omega$-sequence $(t_0, t_1, \ldots)$ such that $t_0$ is the root and $t_i R t_{i+1}$ for all $i \geq 0$. We denote by $B(T, R)$ the set of all branches of $(T, R)$.

A $ERBL$ transition system $M$ is defined as a triple $(T, R, V)$ where:

- $(T, R)$ is a $\omega$-tree frame,
- $V : T \times A \rightarrow \wp(B\Omega \cup CP)$ such that for all $s \in T$ and $i \in A$: $V(s, i) = Q \cup \{cp_i^{=n}\}$ for some $Q \subseteq B\Omega$ and $0 \leq n \leq n_C$. We denote by $V^*(s, i)$ the set $V(s, i) \setminus \{cp_i^{=n} | 0 \leq n\}$.

For a branch $\sigma \in B(T, R)$, $\sigma_i$ denotes the element $t_i$ of $\sigma$ and $\sigma_{\leq i}$ is the prefix $(t_0, t_1, \ldots, t_i)$ of $\sigma$.

The truth of a $ERBL$ formula at a point $n$ of a path $\sigma \in B(T, R)$ is defined inductively as follows:

- $M, \sigma, n \models \top$,
- $M, \sigma, n \models B_i\alpha$ iff $\alpha \in V(s, i)$,
- $M, \sigma, n \models start$ iff $n = 0$,
- $M, \sigma, n \models cp_i^{=m}$ iff $cp_i^{=m} \in V(s, i)$,
- $M, \sigma, n \models \neg\varphi$ iff $M, \sigma, n \not\models \varphi$,
- $M, \sigma, n \models \varphi \wedge \psi$ iff $M, \sigma, n \models \varphi$ and $M, \sigma, n \models \psi$,
- $M, \sigma, n \models X\varphi$ iff $M, \sigma, n + 1 \models \varphi$,
- $M, \sigma, n \models \varphi U \psi$ iff $\exists m \geq n$ such that $\forall k \in [n, m)$ $M, \sigma, k \models \varphi$ and $M, \sigma, m \models \psi$,
- $M, \sigma, n \models A\varphi$ iff $\forall \sigma' \in BR$ such that $\sigma'_{\leq n} = \sigma_{\leq n}$, $M, \sigma', n \models \varphi$.

The set of possible transitions in a model is defined as follows. Definition 1 below describes possible outcomes of various actions. For example, performing a resolution results in adding the resolvent to the set of beliefs. Definition 2 describes when an action is possible or enabled. For example, resolution is enabled if the agent has two resolvable clauses in memory.

**Definition 1.** *Let $(T, R, V)$ be a tree model. The set of effective transitions $R_a$ for an action $a$ is defined as a subset of $R$ and satisfies the following conditions, for all $(s, t) \in R$:*

*1.* $(s, t) \in R_{Read_{i,\alpha,\beta}}$ *iff* $\alpha \in KB_i$*,* $\alpha \notin V(s, i)$ *and* $V(t, i) = V(s, i) \setminus \{\beta\} \cup \{\alpha\}$. This condition says that $s$ and $t$ are connected by agent $i$'s $Read$ transition if the following is true: $\alpha$ is in $i$'s knowledge base but not in $V(s, i)$, $\alpha$ is added to the set of $i$'s beliefs at $t$, and $\beta \in B\Omega$ is removed from the agent's set of beliefs. The argument $\beta$ stands for a formula which is overwritten in the transition. If $\beta \in V(s, i)$ then the agent actually loses a belief in the transition, if $\beta \notin V(s, i)$ then the transition only involves adding a formula $\alpha$ without removing any beliefs.

*2.* $(s, t) \in R_{Res_{i,\alpha_1,\alpha_2,L,\beta}}$ *where* $\alpha_1 = B_{i_1} \ldots B_{i_{k-1}} B_{i_k} c_1$ *and* $\alpha_2 = B_{i_1} \ldots B_{i_{k-1}} B_{i_k} c_2$ *iff* $\alpha_1 \in V(s, i), \alpha_2 \in V(s, i), L \in c_1, \neg L \in c_2$*,* $\alpha = B_{i_1} \ldots B_{i_{k-1}} B_{i_k} c \notin V(s, i)$ *and* $V(t, i) = V(s, i) \setminus \{\beta\} \cup \{\alpha\}$ *where* $c = c_1 \setminus \{L\} \cup c_2 \setminus \{\neg L\}$. This condition says that $s$ and $t$ are connected by agent $i$'s $Res$ transition if in $s$ agent $i$ believes

two resolvable clauses $\alpha_1$ and $\alpha_2$ but not $\alpha$, possibly preceded by the same sequence of belief operators, and in $t$ agent $i$ believes their resolvent, preceded by the same prefix. Again, $\beta \in B\Omega$ is overwritten if it is in the set of agent's beliefs in $s$.
3. $(s,t) \in R_{Copy_{i,\alpha,\beta}}$ *iff* $\alpha \in V(s,j)$ *for some* $j \in A$ *and* $j \neq i$, *for any* $cp_i^{=n} \in V(s,i)$ *such that* $n < n_C$, $B_j\alpha \notin V(s,i)$ *and* $V(t,i) = V(s,i) \setminus \{cp_i^{=n} | cp_i^{=n} \in V(s,i)\} \cup \{cp_i^{=n+1} | cp_i^{=n} \in V(s,i)\} \setminus \{\beta\} \cup \{B_j\alpha\}$. $s$ and $t$ are connected by a $Copy$ transition of agent $i$ if in $t$, $i$ adds to its beliefs a formula $B_j\alpha$ where $\alpha$ is an agent $j$'s belief in $s$, and $i$ has previously copied fewer than $n_C$ formulas. Again some $\beta \in B\Omega$ is possibly overwritten.
4. $(s,t) \in R_{Idle_i}$ *iff* $V(t,i) = V(s,i)$. The $Idle$ transition does not change the state.
5. $(s,t) \in R_{Erase_{i,\beta}}$ *iff* $V(t,i) = V(s,i) \setminus \{\beta\}$. $Erase$ removes one of the agent's beliefs.
6. $(s,t) \in R_{PI_{i,\alpha,\beta}}$ *iff* $\alpha \in V(s,i)$, $B_i\alpha \notin V(s,i)$ *and* $V(t,i) = V(s,i) \setminus \{\beta\} \cup \{B_i\alpha\}$. $PI$ is $i$'s positive introspection: $s$ and $t$ are connected by $i$'s $PI$ transition if in $s$ it believes $\alpha$ but not $B_i\alpha$ and in $t$ it believes $B_i\alpha$.
7. $(s,t) \in R_{OPI_{i,B_{i_1}\dots B_{i_{k-1}},B_{i_k}\alpha,\beta}}$ *iff* $B_{i_1}\dots B_{i_{k-1}}B_{i_k}\alpha \in V(s,i)$ *but not* $B_{i_1}\dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha$, $V(t,i) = V(s,i) \setminus \{\beta\} \cup \{B_{i_1}\dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha\}$. This corresponds to ascribing positive introspection to agent $i_k$.

This specifies the effects of actions. Below, we specify when an action is possible. Note that we only enable deriving a formula if this formula is not already in the set of the agent's beliefs.

**Definition 2.** *Let* $(T,R,V)$ *be a tree model. The set* $Act_{s,i}$ *of possible actions that an agent* $i$ *can perform at a state* $s \in T$ *is defined as follows:*

1. $Read_{i,\alpha,\beta} \in Act_{s,i}$ *iff* $\alpha \notin V(s,i)$, $\alpha \in KB_i$ *and* $\beta \in V(s,i)$ *if* $|V^*(s,i)| \geq n_M$.
2. $Res_{i,\alpha_1,\alpha_2,L,\beta} \in Act_{s,i}$ *iff* $c = (c_1 \backslash L) \cup (c_2 \backslash \neg L) \notin V(s,i)$, $\alpha_1 = B_{i_1}\dots B_{i_{k-1}}B_{i_k}c_1$, $\alpha_2 = B_{i_1}\dots B_{i_{k-1}}B_{i_k}c_2$, $L \in c_1$, $\neg L \in c_2$, $\alpha_1, \alpha_2 \in V(s,i)$, *and* $\beta \in V(s,i)$ *if* $|V^*(s,i)| \geq n_M$.
3. $Copy_{i,\alpha,\beta} \in Act_{s,i}$ *iff* $B_j\alpha \notin V(s,i)$, $\alpha \in V(s,j)$ *for some* $j \in A$ *and* $j \neq i$, $n < n_C$ *for any* $cp_i^{=n} \in V(s,i)$ *and* $\beta \in V(s,i)$ *if* $|V^*(s,i)| \geq n_M$.
4. *It is always the case that* $Idle_i \in Act_{s,i}$.
5. $PI_{i,\alpha,\beta} \in Act_{s,i}$ *iff* ${}_i\alpha \notin V(s,i)$, $\alpha \in V(s,i)$ *and* $\beta \in V(s,i)$ *if* $|V^*(s,i)| \geq n_M$.
6. $OPI_{i,B_{i_1}\dots B_{i_{k-1}}B_{i_k}\alpha,\beta} \in Act_{s,i}$ *iff* $B_{i_1}\dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha \notin V(s,i)$, $B_{i_1}\dots B_{i_{k-1}}$ $B_{i_k}\alpha \in V(s,i)$ *and* $\beta \in V(s,i)$ *if* $|V^*(s,i)| \geq n_M$.

There are no specified conditions for enabling $Erase_{i,\beta}$. This action is introduced for technical reasons, to simplify the proofs.

Finally, the definition of the set of models corresponding to a system of reasoners is given below:

**Definition 3.** $M(KB_1,\dots,KB_{n_A},n_B,n_M,n_C)$ *is the set of models* $(T,R,V)$ *which satisfies the following conditions:*

1. $|V^*(s,i)| \leq n_M$ *for all* $s \in T$ *and* $i \in A$.
2. $cp_i^{=0} \in V(t_0,i)$ *where* $t_0$ *is the root of* $(T,R)$ *for all* $i \in A$.
3. $R = \bigcup_{\forall a} R_a$.
4. *For all* $s \in T$, $a_i \in Act_{s,i}$, *there exists* $t \in T$ *such that* $(s,t) \in R_{a_i}$ *for all* $i \in A$.

## 4 Axiomatisation

In this section, we introduce an axiom system which is sound and complete with respect to the set of models defined in the previous section.

Below are some abbreviations which will be used in the axiomatisation:

- $ByRead_i(\alpha, n) = \neg B_i\alpha \wedge cp_i^{=n}$. This formula describes the state before the agent comes to believe formula $\alpha$ by the $Read$ transition. $n$ is the value of $i$'s communication counter.
- $ByRes_i(\alpha, n) = \bot$ if $\alpha = B_{i_1} \dots B_{i_k - 1}\neg B_{i_k}c$ for some $c \in \Omega$ and $1 \leq k \leq n_B$; otherwise $ByRes_i(\alpha, n) = \neg B_i\alpha \wedge \bigvee_{(\alpha_1,\alpha_2)\in Res^{-1}(\alpha)} (B_i\alpha_1 \wedge B_i\alpha_2)$ where $Res^{-1}(B_{i_1} \dots B_{i_{k-1}}B_{i_k}c) = \{(B_{i_1} \dots B_{i_{k-1}}B_{i_k}c_1, B_{i_1} \dots B_{i_{k-1}}B_{i_k}c_2) \mid \exists L \in LPROP$ such that $c = c_1 \setminus \{L\} \cup c_2 \setminus \{\neg L\}\}$. This formula describes the state of the system before $i$ derives $\alpha$ by resolution. Note that it may not be possible to derive an arbitrary formula $\alpha$ by resolution; in that case, the state is described by falsum $\bot$.
- $ByCopy_i(\alpha, n) = \bot$ if $\alpha \neq B_j\alpha'$ for some $j \neq i$ or $n \leq 0$;
  otherwise $ByCopy_i(B_j\alpha', n) = \neg B_iB_j\alpha' \wedge B_j\alpha' \wedge cp^{=n-1}$.
- $ByPI_i(\alpha, n) = \bot$ if $\alpha \neq B_i\alpha'$; otherwise $ByPI_i(\alpha, n) = \neg B_iB_i\alpha' \wedge B_i\alpha' \wedge cp^{=n}$.
- $ByOPI_i(\alpha, n) = \bot$ if $\alpha \neq B_{i_1} \dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha'$;
  otherwise $ByOPI_i(\alpha, n) = \neg B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha' \wedge B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}\alpha' \wedge cp^{=n}$.

The axiomatisation is as follows.

**A1.** All axioms and inference rules of $CTL^*$ [24].

**A2.** $\bigwedge_{\gamma\in Q} B_i\gamma \wedge cp_i^{=n} \wedge \neg B_i\alpha \rightarrow EX(\bigwedge_{\gamma\in Q} B_i\gamma \wedge cp_i^{=n} \wedge B_i\alpha)$ for all $\alpha \in KB_i$, and $Q \subseteq B\Omega$ such that $|Q| < n_M$.

Intuitively, this axiom says that it is always possible to make a transition to a state where agent $i$ believes a formula from its knowledge base $KB_i$. In addition, the communication counter of the agent does not increase, and a set of beliefs $Q$ of cardinality less than $n_M$ can also be carried over to the same state.

Axioms **A3** - **A6** similarly describe transitions made by resolution (given that resolvable clauses are available), copy (with communication counter increased), and positive introspection (applied by agent $i$ or ascribed by $i$ to another agent).

**A3.** $\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}c_1 \wedge B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}c_2 \wedge cp_i^{=n} \wedge \neg B_iB_{i_1} \dots B_{i_{k-1}}\ B_{i_k}c \rightarrow EX(\bigwedge_{\gamma\in Q} B_i\gamma \wedge cp_i^{=n} \wedge B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}c)$ for all $c_1, c_2 \in \Omega$ such that $L \in c_1$, $\neg L \in c_2$ and $c = c_1 \setminus \{L\} \cup c_2 \setminus \{\neg L\}$, $k \geq 0$, and $Q \subseteq B\Omega$ such that $|Q| < n_M$.

**A4.** $\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_j\alpha \wedge cp_i^{=n} \wedge \neg B_iB_j\alpha \rightarrow EX(\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_iB_j\alpha \wedge cp_i^{=n+1})$ for any $\alpha \in B\Omega$, $j \in A$, $j \neq i$, $n < n_C$, and $Q \subseteq B\Omega$ such that $|Q| < n_M$.

**A5.** $\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_i\alpha \wedge cp_i^{=n} \wedge \neg B_iB_i\alpha \rightarrow EX(\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_iB_i\alpha \wedge cp_i^{=n})$ for any $\alpha \in B\Omega$ and $Q \subseteq B\Omega$ such that $|Q| < n_M$.

**A6.** $\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}\alpha \wedge cp_i^{=n} \wedge \neg B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha \rightarrow EX(\bigwedge_{\gamma\in Q} B_i\gamma \wedge B_iB_{i_1} \dots B_{i_{k-1}}B_{i_k}B_{i_k}\alpha \wedge cp_i^{=n})$ for any $\alpha \in B\Omega$, $k \geq 0$ and $Q \subseteq B\Omega$ such that $|Q| < n_M$.

**A7.** $EX(B_i\alpha \wedge B_i\beta) \rightarrow B_i\alpha \vee B_i\beta$.
This axiom says that at most one new belief is added in the next state.

**A8.** $EX(\neg B_i\alpha \wedge \neg B_i\beta) \rightarrow \neg B_i\alpha \vee \neg B_i\beta$.
This axiom says that at most one belief is deleted in the next state.

**A9.** $EX(B_i\alpha \wedge cp_i^{=n}) \rightarrow B_i\alpha \vee ByRead_i(\alpha, n) \vee ByRes_i(\alpha, n) \vee ByCopy_i(\alpha, n) \vee ByPI_i(\alpha, n) \vee ByOPI_i(\alpha, n)$ for $\alpha \in KB_i$.
This axiom says that a new belief which is an element of the agent's knowledge base can only be added by one of the valid reasoning actions.

**A10.** $EX(B_i\alpha \wedge cp_i^{=n}) \rightarrow B_i\alpha \vee ByRes_i(\alpha, n) \vee ByCopy_i(\alpha, n) \vee ByPI_i(\alpha, n) \vee ByOPI_i(\alpha, n)$ for $\alpha \notin KB_i$.
This axiom describes possible ways in which a new belief which is not in the agent's knowledge base can be added.

**A11.** $B_i\alpha_1 \wedge \ldots \wedge B_i\alpha_{n_M} \rightarrow \neg B_i\alpha_{n_M+1}$ for all $i \in A$, $\alpha_j \in B\Omega$ where $j = 1, \ldots, n_M + 1$ and all $\alpha_j$ are pairwise different.
This axiom states that an agent cannot have more than $n_M$ different beliefs.

**A12a** $start \rightarrow cp_i^{=0}$ for all $i \in A$.
In the start state, the agent has not performed any $Copy$ actions.

**A12b** $\neg EX start$ ($start$ only holds at the root of the tree).

**A13.** $\bigvee_{n=0\ldots n_C} cp_i^{=n}$ for all $i \in A$.
There is always a number $n$ between 0 and $n_C$ corresponding to the number of $Copy$ actions agent $i$ has performed.

**A14.** $cp_i^{=n} \rightarrow \neg cp_i^{=n'}$ for all $i \in A$ and $n' \neq n$.
The number of previous $Copy$ actions by $i$ in each state is unique.

**A15.** $\varphi \rightarrow EX\varphi$ where $\varphi$ does not contain $start$.
It is always possible to make a transition to a state where all agents have the same beliefs and communication counter values as in the current state (essentially an $Idle$ transition by all agents)

**A16.** $\bigwedge_{i \in A} EX(\bigwedge_{\gamma \in Q_i} B_i\gamma \wedge cp_i^{=n_i}) \rightarrow EX \bigwedge_{i \in A}(\bigwedge_{\gamma \in Q_i} B_i\gamma \wedge cp_i^{=n_i})$ for any $Q_i \subseteq B\Omega$ such that $|Q_i| \leq n_M$.
If each agent $i$ can separately reach a state where it believes formulas in $Q_i$, then all agents together can reach a state where for each $i$, agent $i$ believes formulas in $Q_i$.

Notice that since the depth of the nesting of belief operators is restricted by $n_B$, for any subformula $B_i\alpha$ appearing in any above axiom, $\alpha \in B\Omega$.

**Definition 4.** *$L(KB_1, \ldots, KB_{n_A}, n_B, n_M, n_C)$ is the logic defined by the axiomatisation **A1–A16**.*

We have the following result.

**Theorem 1.** *$L(KB_1, \ldots, KB_{n_A}, n_B, n_M, n_C)$ is sound and complete with respect to $M(KB_1, \ldots, KB_{n_A}, n_B, n_M, n_C)$.*

The proof is omitted due to lack of space; it is based on the proof technique used in [24].

## 5 Discussion

Systems of step-wise reasoners with bounded memory and a communication limit are faithful models of systems of distributed resource-limited reasoners, and various resource requirements of such systems can be effectively verified, e.g. by model-checking, as in for example [20]. However, adding reasoning about beliefs poses a significant challenge, both in the complexity of the system and in the way this reasoning is modelled. The branching factor of the models is much larger when reasoning about beliefs is considered, making model-checking less feasible. The main problem however has to do with the correctness of an agent's belief ascription. We describe this problem below and propose a tentative solution.

In the system proposed in this paper, agents correctly ascribe reasoning mechanisms to each other, and in the limit, their predictions concerning other agents' beliefs are correct: if agent $j$ believes that eventually agent $i$ will believe $\alpha$, then eventually agent $i$ will believe $\alpha$, and vice versa. More precisely, for every model $M$ and state $s$,

$$\{\alpha : M, s \models EFB_jB_i\alpha\} = \{\alpha : M, s \models EFB_i\alpha\}$$

However, in spite of this, the agents are almost bound to make wrong predictions when trying to second-guess what other reasoners will believe in the next state. More precisely,

$$\{\alpha : M, s \models B_jB_i\alpha\} \not\subseteq \{\alpha : M, s \models B_i\alpha\}$$

i.e. agent $j$ may believe that $i$ believes some $\alpha$ when $i$ does not believe $\alpha$.

Consider the following example. Suppose there are two agents, 1 and 2, each with a memory limit of two formulas, communication limit of one formula, belief nesting limit of two, and knowledge bases $KB_1 = \{p\}$ and $KB_2 = \{q\}$. A possible run of the system is shown in Figure 1.

| State | Agent 1 | Agent 2 |
|---|---|---|
| $t_0$ | { } | { } |
| transition: | Read | Read |
| $t_1$ | $\{p\}$ | $\{q\}$ |
| transition: | Copy | Copy |
| $t_2$ | $\{p, B_2q\}$ | $\{q, B_1p\}$ |

**Fig. 1.** A possible run of the system

Note that this is only one possible run, and other transitions are possible. For example, in $t_0$, one or both agents can idle. In $t_1$, one or both agents can idle, or make a positive introspection transition. In state $t_2$, the agents' beliefs about each other's beliefs are correct. However, in most successor states of $t_2$, agent 1 will have incorrect beliefs about agent 2's beliefs, and vice versa. Indeed, the options of agent 1 in $t_2$ are: read $p$, idle, erase $p$, erase $B_2q$, apply positive introspection to derive $B_1p$ or $B_1B_2q$, ascribe introspection to agent 2 to derive $B_2B_2q$. Agent 2 has similar choices. In only two of these cases do the agents make non-trivial (that is, new compared to the ones already existing in $t_2$) correct belief ascriptions, namely if agent 1 derives $B_1p$ and agent 2 derives $B_1B_1p$, and vice versa when agent 2 derives $B_2q$ and agent 1 derives $B_2B_2q$ (see Figure 2).

| State | Agent 1 | Agent 2 |
|---|---|---|
| $t_2$ | $\{p, B_2q\}$ | $\{q, B_1p\}$ |
| transition: | PI, overwrite $B_2q$ | OPI, overwrite $q$ |
| $t_3$ | $\{p, B_1p\}$ | $\{B_1p, B_1B_1p\}$ |

**Fig. 2.** Continuing from $t_2$: a correct ascription

Figure 3 shows one of many possible incorrect ascriptions. Note that agent 1's ascription is now incorrect because agent 2 has forgotten $q$, and agent 2's ascription is incorrect because it assumed agent 1 will use positive introspection to derive $B_1p$, which it did not.

| State | Agent 1 | Agent 2 |
|---|---|---|
| $t_2$ | $\{p, B_2q\}$ | $\{q, B_1p\}$ |
| transition: | Idle | OPI, overwrite $q$ |
| $t_4$ | $\{p, B_2q\}$ | $\{B_1p, B_1B_1p\}$ |

**Fig. 3.** Continuing from $t_2$: an incorrect ascription

This suggests an inherent problem with modelling agents reasoning about each other's beliefs in a step-wise, memory-bounded fashion. Note that this problem is essentially one of belief ascription, i.e., of correctly predicting what another agent will believe given limited information about what it currently believes (of deriving correct conclusions from correct premises), rather than a problem of belief revision [25], i.e., what an agent should do if it discovers the beliefs it has ascribed to another agent are incorrect. It is also distinct from the problem of determining the consequences of information updates as studied in dynamic epistemic logic (e.g. [26]). Adding new true beliefs in a syntactic approach such as ours is straightforward compared to belief update in dynamic epistemic logic, which interprets beliefs as sets of possible worlds. Essentially, in dynamic epistemic logic an agent acquires a new logically closed set of beliefs at the next 'step' after an announcement is made, while we model the gradual process of deriving consequences from a new piece of information (and the agent's previous beliefs).

The disparity between agent $i$'s beliefs and the beliefs agent $j$ ascribes to $i$ at each step is due both to the fact that at most one formula is derived by each agent at any given step (and agent $j$ may guess incorrectly which inference rule agent $i$ is going to use) and to memory limitations which cause agents to forget formulas. An obvious alternative is to do tentative ascription of beliefs to other agents, namely conclude that the other agent will be in *one of several* possible belief sets in the next state, e.g.

$$B_2 B_1 p \to EX(B_2((B_1 p \wedge B_1 B_1 p) \vee (B_1 p \wedge \neg B_1 B_1 p) \vee \ldots))$$

However, this implies that one of the agents (agent 2 in this case) has a much larger (exponentially larger!) memory and a more expressive internal language to reason about the other agent's beliefs.

It is clearly not sufficient for correct belief prediction for the reasoners to ascribe to other agents just a set of inferences rules or a logic such as KD45. They need to be able to ascribe to other agents a *reasoning strategy*, or a preference order on the set reasoning actions used by the other agents which constrains the possible transitions of each reasoner, and directs each agent's reasoning about the beliefs of other agents. As a simple example, suppose agent 2 believes that agent 1's strategy is to apply positive introspection to formula $p$ in preference to all other actions. Then in state $t_2$ agent 2 will derive $B_1 B_1 p$ from $B_1 p$. If agent 2's ascription of strategy to agent 1 is correct, agent 1 will indeed derive $B_1 p$ from $p$ in the next state, making agent 2's belief prediction correct.

## 6 *ERBL* with strategies

In this section, we modify the semantics of $ERBL$ to introduce reasoning strategies.

First we need to define strategies formally. A *reasoning strategy for agent* $i$, $\prec_i$, is a total order on the set $Act_i$ of all reasoning actions of $i$ and their arguments:

$$\begin{aligned} Act_i = \{ & Read_{i,\alpha,\beta},\ Res_{i,\alpha_1,\alpha_2,L,\beta},\ Copy_{i,\alpha,\beta}, \\ & Erase_{i,\beta},\ Idle_i,\ PI_{i,\alpha,\beta},\ OPI_{i,B_{i_1}\ldots B_{i_{k-1}},B_{i_k}\alpha,\beta} \mid \alpha,\beta,\alpha_1,\alpha_2 \in B\Omega\} \end{aligned}$$

A simple example of a reasoning strategy for $i$ would be a lexicographic order on $Act_i$ which uses two total orders: an order on the set of transitions, e.g. $Res < PI < OPI < Copy < Read < Idle$, and an order on $B\Omega$.

Recall that in Definition 2 we specified which actions are enabled in state $s$, $Act_{s,i} \subseteq Act_i$. We required in Definition 3 that for each enabled action, there is indeed a transition by that action out of $s$. The simple change that we make to Definition 3 is that for every agent $i$ we only enable *one* action, namely the element of $Act_{s,i}$ which is minimal in $\prec_i$.

**Definition 5.** *The set of reasoning strategy models* $M^{strat}(KB_1, \ldots, KB_{n_A}, n_B, n_M, n_C)$ *is the set of models* $(T, R, V)$ *which satisfies conditions 1-3 from Definition 3 and the following condition:*

**4'.** *For all* $s \in T$*, there exists a unique state* $t$ *such that* $(s,t) \in R_{a_i}$ *for all* $i \in A$*, where* $a_i$ *is the minimal element with respect to* $\prec_i$ *in* $Act_{s,i}$*.*

Observe that in the reasoning strategy models, the transition relation is a linear order.

Finally, we give one possible definition of a correct ascription of a reasoning strategy which allows an agent $j$ to have a correct and complete representation of the beliefs of another agent $i$, namely ensuring that $B_i\alpha \leftrightarrow B_jB_i\alpha$ at each step. Such perfect matching of $i$'s beliefs by $j$ is possible if

$$KB_j = \{B_i\alpha : \alpha \in KB_i\}$$

and agent $i$ does not use the $Copy$ action (intuitively, because in order to match $Copy$ by $i$, agent $j$ has to add two modalities in one step: when agent $i$ derives $B_l\alpha$ from $\alpha$ being in agent $l$'s belief set, agent $j$ has to derive $B_iB_l\alpha$). Below, we also assume that $j$ is allowed one extra nesting of belief modalities ($n_B(j) = n_B(i) + 1$).

**Definition 6.** *Agent $j$ has a strategy which* matches *the strategy of agent $i$ if for every natural number $k$, the following correspondence holds between the $k$th element of $\prec_j$ and the $k$th element of $\prec_i$:*

- *if the $k$th element of $\prec_i$ is $Read_{i,\alpha,\beta}$, then the $k$th element of $\prec_j$ is $Read_{j,B_i\alpha,B_i\beta}$*
- *if the $k$th element of $\prec_i$ is $Res_{i,\alpha_1,\alpha_2,L,\beta}$, then the $k$th element of $\prec_j$ is $Res_{j,B_i\alpha_1,B_i\alpha_2,L,B_i\beta}$*
- *if the $k$th element of $\prec_i$ is $PI_{i,\alpha,\beta}$, then the $k$th element of $\prec_j$ is $OPI_{j,B_i\alpha,B_i\beta}$*
- *if the $k$th element of $\prec_i$ is $OPI_{i,B_l\alpha,\beta}$, then then the $k$th element of $\prec_j$ is $OPI_{j,B_iB_l\alpha,B_i\beta}$.*
- *if the $k$th element of $\prec_i$ is $Erase_{i,\beta}$, then the $k$th element of $\prec_j$ is $Erase_{j,B_i\beta}$*
- *if the $k$th element of $\prec_i$ is $Idle_i$, then the $k$th element of $\prec_j$ is $Idle_j$.*

**Theorem 2.** *If agent $j$'s strategy matches agent $i$'s strategy and agent $j$ has complete and correct beliefs about agent $i$'s beliefs in state $s$: $M, s \models B_i\alpha \leftrightarrow B_jB_i\alpha$, then agent $j$ will always have correct beliefs about agent $i$'s beliefs: $M, s \models AG(B_i\alpha \leftrightarrow B_jB_i\alpha)$.*

Other more realistic matching strategies, for example, those which allow the agent to have a less than complete representation of other agent's beliefs, are possible, and their formal investigation is a subject of future work.

## 7 Conclusion

We presented a formal model of resource-bounded reasoners reasoning about each other's beliefs, and a sound and complete logic, $ERBL$, for reasoning about such systems. Our formalisation highlighted a problem of incorrect belief ascription, and we showed that this problem can be overcome by extending the framework with reasoning strategies. In future work we plan to extend the framework in a number of ways, including producing correct belief ascription under less strict matching between agents' strategies, and introducing reasoning about other agent's resource limitations. At the moment the agents have no way of forming beliefs about another agent's memory limit $n_M$ or belief nesting bound $n_B$ (note that we can also easily make those limits different for different agents). If they could represent those limitations, then one agent could infer that another agent does not believe some formula on the grounds that the latter agent's memory is bounded.

## References

1. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press, Cambridge, Mass. (1995)
2. Meyer, J.J., van der Hoek, W.: Epistemic Logic for Computer Science and Artificial Intelligence. Cambridge University Press (1995)
3. Hintikka, J.: Knowledge and belief. Cornell University Press, Ithaca, NY (1962)
4. Rantala, V.: Impossible worlds semantics and logical omniscience. Acta Philosophica Fennica **35** (1982) 106–115
5. Fagin, R., Halpern, J.Y., Vardi, M.Y.: A non-standard approach to the logical omniscience problem. In Parikh, R., ed.: Theoretical Aspects of Reasoning about Knowledge: Proceedings of the Third Conference, Morgan Kaufmann (1990) 41–55
6. Fagin, R., Halpern, J.Y.: Belief, awareness and limited reasoning: Preliminary report. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence. (1985) 491–501
7. Konolige, K.: A Deduction Model of Belief. Morgan Kaufmann, San Francisco, Calif. (1986)
8. Elgot-Drapkin, J.J., Perlis, D.: Reasoning situated in time I: Basic concepts. Journal of Experimental and Theoretical Artificial Intelligence **2** (1990) 75–98
9. Halpern, J.Y., Moses, Y., Vardi, M.Y.: Algorithmic knowledge. In Fagin, R., ed.: Theoretical Aspects of Reasoning about Knowledge: Proceedings of the Fifth Conference (TARK 1994). Morgan Kaufmann, San Francisco (1994) 255–266
10. Pucella, R.: Deductive algorithmic knowledge. J. Log. Comput. **16**(2) (2006) 287–309
11. Duc, H.N.: Logical omniscience vs. logical ignorance on a dilemma of epistemic logic. In Pinto-Ferreira, C.A., Mamede, N.J., eds.: Progress in Artificial Intelligence, 7th Portuguese Conference on Artificial Intelligence, EPIA '95, Funchal, Madeira Island, Portugal, October 3-6, 1995, Proceedings. Volume 990 of Lecture Notes in Computer Science., Springer (1995) 237–248
12. Duc, H.N.: Reasoning about rational, but not logically omniscient, agents. Journal of Logic and Computation **7**(5) (1997) 633–648
13. Ågotnes, T., Alechina, N.: The dynamics of syntactic knowledge. Journal of Logic and Computation **17**(1) (2007) 83–116
14. Sierra, C., Godo, L., de Mántaras, R.L., Manzano, M.: Descriptive dynamic logic and its application to reflective architectu res. Future Gener. Comput. Syst. **12**(2-3) (1996) 157–171
15. Alechina, N., Jago, M., Logan, B.: Modal logics for communicating rule-based agents. In Brewka, G., Coradeschi, S., Perini, A., Traverso, P., eds.: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), IOS Press (2006) 322–326
16. Fisher, M., Ghidini, C.: Programming resource-bounded deliberative agents. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI 1999), Morgan-Kaufmann (1999) 200–205
17. Ågotnes, T.: A Logic of Finite Syntactic Epistemic States. Ph.D. thesis, Department of Informatics, University of Bergen, Norway (2004)
18. Ågotnes, T., Alechina, N.: Knowing minimum/maximum $n$ formulae. In Brewka, G., Coradeschi, S., Perini, A., Traverso, P., eds.: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), IOS Press (2006) 317–321
19. Albore, A., Alechina, N., Bertoli, P., Ghidini, C., Logan, B., Serafini, L.: Model-checking memory requirements of resource-bounded reasoners. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006), AAAI Press (2006) 213–218
20. Alechina, N., Logan, B., Nga, N.H., Rakib, A.: Verifying time, memory and communication bounds in systems of reasoning agents. In Padgham, L., Parkes, D., Müller, J., Parsons,

S., eds.: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008). Volume 2., Estoril, Portugal, IFAAMAS, IFAAMAS (May 2008) 736–743

21. Ågotnes, T., Walther, D.: Towards a logic of strategic ability under bounded memory. In: Proceedings of the Workshop on Logics for Resource-Bounded Agents. (2007)
22. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.C., Simon, L.: Scalability study of peer-to-peer consequence finding. In Kaelbling, L.P., Saffiotti, A., eds.: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05), Edinburgh, Scotland, Professional Book Center (2005) 351–356
23. Emerson, E.A.: Temporal and modal logic. In: Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B). Elsevier and MIT Press (1990) 995–1072
24. Reynolds, M.: An axiomatization of full computation tree logic. Journal of Symbolic Logic **66**(3) (2001) 1011–1057
25. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet functions for contraction and revision. Journal of Symbolic Logic **50** (1985) 510–530
26. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge (TARK'98). (1998)

# An Argumentation-based Protocol for Conflict Resolution

Jamal Bentahar[1], Rafiul Alam[1], and Zakaria Maamar[2]

[1] Concordia University, Concordia Institute for Information Systems Engineering, Canada
[2] Zayed University, College of Information Systems, UAE

**Abstract.** This paper proposes an argumentation-based protocol for resolving conflicts between agents. These agents use assumption-based argumentation in which arguments are built from a set of rules and assumptions using backward deduction. Beyond arguments agents can handle, we propose the notion of *partial arguments* along with *partial attack* and *partial acceptability*. These notions allow agents to reason about partial information. Unlike existing protocols, this protocol merges inquiry and persuasion stages. In addition, by building and reasoning about partial arguments, agents can jointly find arguments supporting a new solution for their conflict, which is not known by any of them individually. Furthermore, during persuasion, agents can acquire new beliefs and drop attacked ones. The protocol is formally specified as a set of simple dialogue rules about which agents can reason using argumentation. We also provide termination, soundness and completeness results of the proposed protocol.

## 1 Introduction

In recent years, argumentation has been used in many applications such as legal reasoning, automatic negotiation, persuasion, online debates, medical applications, and Web services [19]. In multi-agent systems, some interesting argumentation-based protocols for persuasion and inquiry have been proposed. [2] proposes the Persuasive Argument for Multiple Agents (PARMA) Protocol, which enables participants to propose, attack, and defend an action or course of actions. This protocol is specified using logical consequence and denotational semantics, which maps statements in the syntax to mathematical entities. The focus of this work is more about the semantics of the protocol rather than the dynamics of interactions. [4] proposes a dialogue-game inquiry protocol that allows two agents to share knowledge in order to construct an argument for a specific claim. The protocol is declaratively specified as a function that returns the set of legal moves. A strategy function is also specified to allow an agent to select exactly one of the legal moves to make. This protocol considers only pure inquiry where no conflicting goals are identified. In the context of agent communication, [17] proposes an alternative view on argumentation and persuasion using a coherence theory. Argument generation, evaluation and integration within this theory are discussed in a pragmatic way. However, no protocol about using the proposed framework has been specified. [6] develops a dynamic, situation calculus-based argumentation model in which protocols describe, in a declarative way, which speech acts are legal in a particular state. The model is used to analyze a formal disputation. Except the [4]'s protocol, the others do not consider agents' strategies and the correctness and completeness properties are not discussed.

In a series of papers, researchers from Toulouse and Liverpool have developed an approach to specify persuasion and inquiry protocols. Particularly [16] uses propositional logic to define the underlying argumentation framework and only three locutions have been used in these protocols: *Assert*, *Accept*, and *Challenge*. The purpose is mainly to discuss the *pre-determinism* issue (i.e. to what extent the outcomes of dialogues are *predetermined* when using these protocols). The idea is to check if these outcomes are determined by agents' knowledge and the order agents utter locutions. The authors show that these protocols are not complete in the sense that the generated dialogues are not pre-determined. Also, agents' strategies on how to use the protocols are not considered, and persuasion and inquiry are dealt with as two different protocols without connection between them. [18] proposes formal dialogue games for persuasion protocols where arguments are assumed to be trees of deductive and/or defeasible inferences. Each dialogue move either attacks or surrenders to some earlier move of the addressee. The motivation of the framework is to ensure coherence and flexibility of dialogues. The protocol notion is specified as a function defining the legal moves, and the framework does not consider the agents' strategies.

All the protocols defined in the aforementioned proposals are either pure persuasion or pure inquiry. Also, the proposed persuasion protocols are not complete in the sense of pre-determinism. Except a few proposals such as [4], the notion of agents' strategies on how to use these protocols is disregarded. The purpose of this paper is to address these limitations. The contribution of this work is the proposition of a new sound and complete protocol combining persuasion and inquiry for conflict resolution. Agents use assumption-based argumentation in which arguments are built from a set of rules and assumptions using backward deduction. This protocol is different from all the aforementioned proposals in many points. It is operationally specified as simple *if-then rules* with conditions whose values determine the reply an agent can perform. The agent strategy is used when evaluating these conditions based on private agents' beliefs and publicly exchanged information between communicating agents. The whole protocol can be built by simply combining these rules. In addition, there are two original ideas behind it: (1) the notion of *partial arguments and their "acceptability" statuses* allowing agents to reason about incomplete information; and (2) the combination of persuasion and inquiry allowing agents to jointly find out new solutions for their conflicts that cannot be found by any of them individually. Consequently, new solutions can emerge by exchanging arguments and partial arguments, which allows resolving the problem of pre-determinism.

Section 2 presents the argumentation model and the notions of partial arguments and conflicts. Section 3 presents the protocol specification along with the different dialogue rules and strategies, and discusses its properties. Section 4 gives an illustrative example of the protocol. Section 5 concludes the paper by discussing some related work.

## 2 Argumentation Model

### 2.1 Language and Background

This section discusses the key elements of the formal argumentation system. Many argumentation systems have been proposed in the literature such as the abstract argumentation [10], logic-based argumentation [1, 3, 16], preference value-based argumentation [13], logic-programming-based argumentation [14, 12], and facts and rules-based argumentation [4] (see [7] for a survey). In this paper, we use assumption-based argumentation adapted from [5] and [8]. Assumption-based argumentation has been proven to be a powerful mechanism to understand commonalities and differences amongst many existing frameworks for non-monotonic reasoning, including logic programming [5]. This framework is built upon Dung's abstract argumentation by instantiating the primitive notions of *argument* and *attack* using notions of *deductive system* and corresponding *deductions*, *assumptions and contrary of assumptions*. Mechanisms for computing Dung's argumentation semantics [10] have been developed for this framework with some computational advantages, particularly in terms of avoiding re-computation by filtering out assumptions that have already been defended or defeated [9].

We use a formal language $\mathcal{L}$ to express agents' beliefs. This language consists of countably many sentences (or wffs). Also, the language is associated with an abstract *contrary mapping* like the one used in [11] (the negation is an example of this mapping, so the contrary of $p$ is $\neg p$). We do not assume this mapping to be necessarily symmetric. $\bar{x}$ denotes an arbitrary contrary of a wff $x$. By arbitrary we mean we do not need to specify this contrary. Agents build arguments using their beliefs. The set $Arg(\mathcal{L})$ contains all those arguments.

**Definition 1 (Argumentation Framework).** *An assumption-based argumentation framework is a tuple $\left\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \right\rangle$ where:*

- $(\mathcal{L}, \mathcal{R})$ *is a deductive system, with a countable set $\mathcal{R}$ of inference rules,*
- $\mathcal{A} \subseteq \mathcal{L}$ *is a $($non-empty$)$ set, whose elements are referred to as assumptions,*
- $^-$ *is a total mapping from $\mathcal{A}$ into $2^{\mathcal{L}} - \emptyset$, where $^-\alpha$ is the non-empty set of contraries of $\alpha$ and $\bar{\alpha}$ is an arbitrary contrary of $\alpha$ $(\bar{\alpha} \in^- \alpha)$.*

We will assume that the inference rules in $\mathcal{R}$ have the form: $c_0 \leftarrow c_1, \ldots, c_n$ with $n > 0$ or the form $c_0$ where each $c_i \in \mathcal{L}$ $(i = 0, \ldots, n)$.
$c_0$ is referred to as the head and $c_1, \ldots, c_n$ as the body of a rule $c_0 \leftarrow c_1, \ldots, c_n$. The body of a rule $c_0$ is considered to be empty. We will restrict attention to *flat assumption-based frameworks*, such that if $c \in \mathcal{A}$, then there exists no inference rule of the form $c \leftarrow c_1, \ldots, c_n \in \mathcal{R}$. Before defining the notions of argument and attack relation, we give here a formal definition of the backward deduction that is used in this framework.

The backward deduction can be represented by a top-down proof tree linking the conclusion to the assumptions. The root of the tree is labelled by the conclusion and the terminal nodes are labelled by the assumptions. For every non-terminal node in the tree, there is an inference rule whose head matches the sentence labelling the node. The children of the node are labelled by the body of the inference rule. Consequently, the backward deduction can be represented as a set of steps $S_1, \ldots, S_m$ and in each step we have a set of sentences to which we can apply inference rules because each sentence matches the head of a rule. From each step to the next one, the procedure consists of selecting one sentence and replacing it, if the sentence is not an assumption, by the body of the corresponding inference rule. The selection strategy is represented by the following function:

$$SS : Step \rightarrow \mathcal{L}$$

where $Step = \{S_1, \ldots, S_m\}$

**Definition 2 (Backward Deduction).** *Given a deduction system $(\mathcal{L}, \mathcal{R})$ and a selection strategy function $SS$, a backward deduction of a conclusion $c$ from a set of assumptions $X$ is a finite sequence of sets $S_1, \ldots, S_m$, where $S_1 = \{c\}$, $S_m = X$, and for every $1 \leq i < m$:*
***1.** If $SS(S_i) \notin X$ then $S_{i+1} = S_i - \{SS(S_i)\} \cup B$ for some inference rules of the form $SS(S_i) \leftarrow B$.*
***2.** Otherwise, $S_{i+1} = S_i$.*

**Definition 3 (Argument).** *Let $X \subseteq \mathcal{A}$ be a consistent subset of assumptions (i.e. $X$ does not include a formula and one of its contraries), and let $c$ be a sentence in $\mathcal{L}$. An argument in favor of $c$ is a pair $(X, c)$ such that $c$ is obtained by the backward deduction from $X$. $c$ is called the conclusion of the argument.*

In the rest of the paper, arguments will be denoted as pairs $(X, c)$ when the set of premises is needed for our analysis. Otherwise, arguments will be simply denoted by $a, b, d, \ldots$. The notion of conflicts between arguments is captured via the attack relation. The attack relation we define here is more general than the one used in the original assumption-based argumentation in the sense that we allow attacking, not only the assumptions, but also the conclusion of the argument.

**Definition 4 (Attack Relation).** *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments over the argumentation framework. The attack relation between arguments $\mathcal{AT} \subseteq Ar \times Ar$ is a binary relation over $Ar$ that is not necessarily symmetric. An argument $(X, c)$ attacks another argument $(X', c')$ denoted by $\mathcal{AT}((X, c), (X', c'))$ iff $c$ is a contrary of $c'$ or $c$ is a contrary of a sentence $c'' \in X'$.*

In our protocol (Section 3), by using this attack relation, agents can try to win the dispute by trying different arguments for the same conclusion. For example, if an assumption is attacked, and cannot be defended, the agent can try to defend the conclusion using another assumption.

As conflicts between arguments might occur, we need to define what an acceptable argument is. Different semantics for argument acceptability have been proposed in [10]. These are stated in definitions 5, 6, 7, and 8.

**Definition 5 (Defense).** *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments over the argumentation framework, and let $S \subseteq Ar$. An argument $a$ is defended by $S$ iff $\forall\, b \in Ar$ if $\mathcal{AT}(b, a)$, then $\exists\, c \in S : \mathcal{AT}(c, b)$.*

**Definition 6 (Admissible Set).** *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments over the argumentation framework. A set $S \subseteq Ar$ of arguments is admissible iff:*
*1) $\nexists\, a, b \in S$ such that $\mathcal{AT}(a, b)$ and*
*2) $\forall a \in S$ $a$ is defended by $S$.*

In other words, a *set of arguments* is admissible iff it is conflict-free and can counterattack every attack.

**Definition 7 (Characteristic Function).** *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments and let $S$ be an admissible set of arguments. The characteristic function of the argumentation framework is:*

$$F : 2^{Ar} \to 2^{Ar}$$
$$F(S) = \{a \mid a\, is\, defended\, by\, S\}$$

**Definition 8 (Acceptability Semantics).** *Let $S$ be an admissible set of arguments, and let $F$ be the characteristic function of the argumentation framework.*

- *$S$ is a complete extension iff $S = F(S)$.*
- *$S$ is the grounded extension iff $S$ is the minimal $($w.r.t. set-inclusion$)$ complete extension $($ the grounded extension corresponds to the least fixed point of $F)$.*
- *$S$ is a preferred extension iff $S$ is a maximal $($w.r.t. set-inclusion$)$ complete extension.*

Now we can define what are the acceptable arguments in our system.

**Definition 9 (Acceptable Arguments).** *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments, and let $G$ be the grounded extension in the argumentation framework. An argument $a$ over $Ar$ is acceptable iff $a \in G$.*

According to this acceptability semantics, which is based on the grounded extension, if we have two arguments $a$ and $b$ such that $\mathcal{AT}(a, b)$ and $\mathcal{AT}(b, a)$, then $a$ and $b$ are both non-acceptable. This notion is important in persuasion dialogues since agents should agree on an acceptable opinion, which is supported by an acceptable argument when a conflict arises. However, during the argumentative conversation, agents could use non-acceptable arguments as an attempt to change the status of some arguments previously uttered by the addressee, from acceptable to non-acceptable. This idea of using non-acceptable arguments in the dispute does not exist in the persuasion and inquiry protocols in the literature. For this reason, we introduce two new types of arguments based on the preferred extensions to capture this notion. We call these arguments *semi-acceptable and preferred semi-acceptable arguments*.

**Definition 10 ((Preferred) Semi-Acceptable Arguments).** *Let $G$ be the grounded extension in the argumentation framework, and let $E_1, \ldots, E_n$ be the preferred extensions in the same framework. An argument $a$ is:*

- *Semi-acceptable iff $a \notin G$ and $\exists E_i, E_j$ with $(1 \leq i, j \leq n)$ such that $a \in E_i \wedge a \notin E_j$.*
- *Preferred semi-acceptable iff $a \notin G$ and $\forall E_i$ $(1 \leq i \leq n)$ $a \in E_i$.*

In other words, an argument is *semi-acceptable* iff it is not acceptable and belongs to some preferred extensions, but not to all of them. An argument is *preferred semi-acceptable* iff it is not acceptable and belongs to all the preferred extensions. Preferred semi-acceptable arguments are stronger than semi-acceptable and grounded arguments are the strongest arguments in this classification.

**Proposition 1.** *the arguments defending themselves by only themselves against all the attackers (the set of attackers is supposed to be non-empty) are semi-acceptable.*

**Definition 11 (Eliminated Arguments).** *An argument is eliminated iff it does not belong to any preferred extension in the argumentation framework.*

We can easily prove that an argument is *eliminated* iff it is not acceptable, not preferred semi-acceptable, and also not semi-acceptable. Consequently, arguments take four exclusive statuses namely acceptable, preferred semi-acceptable, semi-acceptable, and eliminated. The dynamic nature of agents interactions is reflected by the changes in the statuses of *uttered arguments*. Let us now define the notions of conflict and conflict resolution.

**Definition 12 (Conflicts).** *Let $\alpha$ and $\beta$ be two argumentative agents and $AF_\alpha$ and $AF_\beta$ their respective argumentation frameworks. These two frameworks share the same contrary relation and the same rules, but not necessarily the same assumptions. There is a conflict between $\alpha$ and $\beta$ iff one of them $($e.g., $\alpha)$ has an acceptable argument $(X, c)$ relative to $AF_\alpha$ and the other $($i.e., $\beta)$ has an acceptable argument $(X', \bar{c})$ relative to $AF_\beta$. We denote this conflict by $\alpha_c \ncong \beta_{\bar{c}}$ $(\bar{c} \in^- c)$.*

For example, in an e-business setting if $c$ and $\bar{c}$ represent each a security policy $s_1$ and $s_2$ such that $s_1$ and $s_2$ cannot be used together, then there is a conflict if one agent has an acceptable argument for using $s_1$ while the other agent has an acceptable argument for using $s_2$. This conflict arises when both agents need to agree on which security policy to use. For simplification reasons and when the set of assumptions is not needed for our analysis, an argument $a$ supporting a conclusion $c$ will be denoted $a \uparrow c$.

**Definition 13 (Conflicts Resolution).** *A conflict between two agents is resolved after interaction iff they agree on a solution which is supported by an acceptable argument for both agents.*

In the aforementioned security example, the conflict is resolved iff (i) after interaction, one of the agents can build an acceptable argument from its knowledge base and the arguments exchanged during this interaction, supporting the use of the other policy, or (ii) when both agents agree on the use of a new policy such that each agent can build an acceptable argument, from its knowledge base and the exchanged arguments, supporting the use of this policy. The idea here is that by exchanging arguments, new solutions (and arguments supporting these solutions) can emerge. In this case, agents should update their beliefs by withdrawing attacked (i.e. eliminated) assumptions. However, there is still a possibility that each agent keeps its viewpoint at the end of the conversation.

### 2.2 Partial Arguments

The outcome of an interaction aiming to resolve a conflict depends on the status of the formula representing the conflict topic. As for arguments, a wff has four statuses depending on the statuses of the arguments supporting it (an argument supports a formula if this formula is the conclusion of that argument). A wff is *acceptable* if there exists an acceptable argument supporting it. If not, and if there exists a preferred semi-acceptable argument supporting it, then the formula is preferred semi-acceptable. Otherwise, the formula is semi-acceptable if a semi-acceptable argument supporting it exists, or eliminated if such an argument does not exist. Let $St$ be the set of these statuses. We define the following function that returns the status of a wff with respect to a set of arguments:

$$\varDelta : \mathcal{L} \times 2^{Ar} \rightarrow St$$

To resolve conflicts, it happens that agents do not have complete information on some facts. In similar situations, they can build *partial arguments* for some conclusions out of their beliefs. We define a partial argument as follows:

**Definition 14 (Partial Arguments).** *Let $X \subseteq \mathcal{A}$ be a consistent subset of assumptions, and $c$ a sentence in $\mathcal{L}$. A partial argument in favor of $c$ is a pair denoted by $(X, c)_\partial$ such that $\exists Y \subseteq \mathcal{A}$ $(Y \neq \emptyset)$ and $(X \cup Y, c)$ is an argument.*

*Example 1.* $\mathcal{L} = \{p, q, r, t, m\}$, $\mathcal{R} = \{p \leftarrow q, r; p \leftarrow q, t, m\}$, $\mathcal{A} = \{q, r, t, m\}$. $(\{q\}, p)_\partial$ is a partial argument.

Because agents can use different sets of rules to build the same argument, it is clear that there are may be several sets of assumptions leading to this argument. Consequently, the set $Y$ in Definition 14 is not unique. Agents can identify a set $Y$ by identifying possible rules leading to the conclusion. When we need to refer to a given set $Y$ (which is missing to complete the argument), we use the notation $(X, c)_\partial^Y$.

*Example 2.* The partial argument presented in example 1 can be completed by the assumption $r$ or by the assumptions $t$ and $m$. They can be denoted respectively by: $(\{q\}, p)_\partial^{\{r\}}$ and $(\{q\}, p)_\partial^{\{t,m\}}$

The idea behind building partial arguments is that in some situations, agents have partial information to build arguments for some conclusion, and the missing information can be obtained through interactions. If an agent misses a set of assumptions, it can check if the other agent can provide the missing part or a part of this missing part so that the complete argument could be jointly built progressively. This issue, already identified in [4], is a part of the inquiry dialogue and will be made clear in the protocol we define in the next section.

In the security scenario presented above, an example where partial arguments are needed is when the agent defending the security policy $s_1$ knows that the security policy $s_2$ that the other agent uses can be substituted by policy $s_1$ if some conditions are met when deploying $s_2$. Thus, this agent can build a partial argument supporting the fact that the second agent can use $s_1$. To be an argument, this partial argument needs the assumptions that implementing $s_2$ by the second agent meets these conditions.

As for arguments, we need to define the *status of partial arguments*. Our idea here, which is different from what is proposed in [4], is that if, considering the information an agent has at the current moment, there is no chance for the partial argument to be acceptable or at least to change the status of already uttered arguments, then there is no need to try to build such a partial argument. When the internal structure of these partial arguments is not needed, we use the notations $a_\partial$ and $a_\partial^Y$ to denote a partial argument and a partial argument that can be completed by the set $Y$ of assumptions respectively. The argument obtained by adding the assumptions $Y$ to the partial argument $a_\partial^Y$ is denoted by $a_\partial^Y.Y$. The following definitions establish the status of partial arguments.

**Definition 15 (Partial Attack).** -

- $\mathcal{AT}(a_\partial^Y, b)$ *iff* $\mathcal{AT}(a_\partial^Y.Y, b)$
- $\mathcal{AT}(b, a_\partial^Y)$ *iff* $\mathcal{AT}(b, a_\partial^Y.Y)$
- $\mathcal{AT}(a_\partial^Y, b_\partial^{Y'})$ *iff* $\mathcal{AT}(a_\partial^Y.Y, b_\partial^{Y'}.Y')$

In words, a partial argument $a_\partial^Y$ attacks (is attacked by) an argument $b$ iff $a_\partial^Y.Y$ attacks (is attacked by) $b$. Also, a partial argument $a_\partial^Y$ attacks another partial argument $b_\partial^{Y'}$ iff $a_\partial^Y.Y$ attacks $b_\partial^{Y'}.Y'$.

**Definition 16 (Status of Partial Arguments).** *A partial argument $a_\partial^Y$ is acceptable ((preferred) semi-acceptable, eliminated) iff $a_\partial^Y.Y$ is acceptable ((preferred) semi-acceptable, eliminated).*

*Example 3.* $\mathcal{L} = \{p, q, r, \bar{r}, s, t, u, \}$, $\mathcal{R} = \{p \leftarrow q; s \leftarrow r; t \leftarrow s; \bar{r} \leftarrow u\}$, $\mathcal{A} = \{q, r, u\}$.
The partial argument $(\emptyset, p)_\partial^{\{q\}}$ is acceptable. However the partial argument $(\emptyset, t)_\partial^{\{r\}}$ is not acceptable since the argument $(\{u\}, \bar{r})$ attacks the argument $(\{r\}, t)$.

Agents should consider the status of their partial arguments before using them. For example, if an agent has a partial argument $a_\partial^Y$ and by supposing that the assumptions in the set $Y$ are true, the resulting argument $a_\partial^Y.Y$ is already eliminated by considering the arguments already uttered, there is no need to try to establish these assumptions. However, if this is not the case, the agent will ask the addressee about these assumptions. Also, if an agent has an acceptable partial argument for (resp. against) a conclusion, and an acceptable argument against (resp. for) this conclusion, this agent will utter its acceptable argument only if the acceptable partial argument cannot emerge from the interaction. The motivation behind this *general rule* is that if the partial argument is acceptable, then if it becomes a complete argument (by establishing the missing assumptions), the status of the existing acceptable argument will change to non-acceptable. So, the agent should consider first the partial acceptable argument.

# 3 Protocol for Resolving Conflicts

## 3.1 Agent Configuration

For simplification reason, we suppose that only two agents take part in the argumentative conversation to resolve their conflict. We denote participating agents by $\alpha$ and $\beta$. Agents share the same set of rules and each agent has a possibly inconsistent belief base $\mathcal{A}_\alpha$ and $\mathcal{A}_\beta$ respectively containing assumptions, where $\mathcal{A}_\alpha \cup \mathcal{A}_\beta = \mathcal{A}$ the set of assumptions in the argumentation framework.

Agents use their argumentation systems to decide about the next move to play (e.g., accept or attack the arguments uttered during their interactions). Agents strategies are based on these systems. When an agent accepts an argument that an addressee suggests, this agent updates its knowledge base by adding the elements of this argument and removing all the elements that attack this argument. Each agent $\alpha$ has also a commitment store ($CS_\alpha$) publicly accessible for reading but only updated by the owner agent. Our protocol is to be used when a conflict is identified, for example as an outcome of a previous interaction. Consequently, when the protocol starts, the commitment stores of the two agents contain conflicting information. For example, $CS_\alpha = \{p\}$ and $CS_\beta = \{\bar{p}\}$ where $p$ is a wff representing the conflict topic. Commitment stores are updated by adding arguments and partial arguments that the agents exchange. $CS_\alpha$ refers to the commitment store of agent $\alpha$ *at the current moment*.

The possibility for an agent $\alpha$ to build an acceptable argument $a$ (respectively an acceptable partial argument $a_\partial^Y$) from its knowledge base and the commitment store of the addressee $\beta$ is denoted by $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a$ (respectively $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a_\partial^Y$). $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \not\rhd a$ (respectively $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \not\rhd a_\partial^Y$) means that agent $\alpha$ cannot build an acceptable argument $a$ (respectively an acceptable partial argument $a_\partial^Y$) from $\mathcal{A}_\alpha \cup CS_\beta$. For simplification reason, we associate the same symbols ($\unrhd$ and $\not\unrhd$) with (partial) preferred semi-acceptable and (partial) semi-acceptable arguments. However, agents consider first (partial) preferred semi-acceptable arguments.

### 3.2 Protocol Rules

In our framework, agents engage in persuasion and inquiry dialogues to resolve conflicts. We propose a persuasion-inquiry protocol, in which pre-determinism is considered. The protocol is modeled using a set of simple *dialogue rules* governing interactions between agents, in which each agent moves by performing utterances. These rules that correspond to dialogue games [15] are expressed as simple if-then rules that can be easily implemented. In this section, we define the notion of protocol and specify the protocol rules.

**Definition 17 (Protocol).** *A protocol is a pair $\langle \mathcal{C}, \mathcal{D} \rangle$ with $\mathcal{C}$ a finite set of allowed moves and $\mathcal{D}$ a set of dialogue rules.*

The moves in $\mathcal{C}$ are of $n$ different types ($n > 0$). We denote by $M^i(\alpha, \beta, a, t)$ a move of type $i$ played by agent $\alpha$ and addressed to agent $\beta$ at time $t$ regarding a content $a$. We consider four types of moves in our protocol: $Assert$, $Accept$, $Attack$, and $Question$. Generally, in the persuasion protocol agents exchange arguments. Except the $Question$ move whose content is not an argument, the content of other moves is an argument $a$ ($a \in Arg(\mathcal{L})$). When replying to a $Question$ move, the content of $Assert$ move can also be a partial argument or "?" when the agent does not know the answer. We use another special move $Stop$ with no content. It could be played by an agent to stop the interaction.

Intuitively, a dialogue rule in $\mathcal{D}$ is a rule indicating the possible moves that an agent could play following a move done by an addressee. To make agents deterministic, we specify these rules using conditions that reflect the agents' strategies. Each condition $C_j$ is associated with a single reply. This is specified formally as follows:

**Definition 18 (Dialogue Rule).** *A dialogue rule is either of the form:*

$$\bigwedge_{\substack{0<k\leq n_i \\ j\in J}} \big( M^i(\alpha, \beta, a, t) \wedge C_k \Rightarrow M_k^j(\beta, \alpha, a_k, t') \big)$$

*where $J$ is the set of move types, $M^i$ and $M^j$ are in $\mathcal{C}$ ($M_k^j$ is the $k^{th}$ move of type $j$), $t < t'$ and $n_i$ is the number of allowed communicative acts that $\beta$ could perform after receiving a move of type $i$ from $\alpha$; or of the form:*

$$\bigwedge_{\substack{0<k\leq n \\ j\in J}} \big(C_k \Rightarrow M_k^j(\alpha, \beta, a_k, t_0)\big)$$

*where $t_0$ is the initial time and $n$ is the number of allowed moves that $\alpha$ could play initially.*

In order to guarantee determinism and deadlock freedom, conditions $C_j$ need to be mutually exclusive and exhaustive. Agents use their argumentation systems to evaluate, in a private manner, these conditions. These argumentation systems are based on the private agents' beliefs and public commitments recorded in the commitment stores.

To simplify the notations, we omit the time parameter from the moves and use the notation $\cup CS$ as an abbreviation of $CS_\alpha \cup CS_\beta$. Also, $Ar$ denotes the set of all arguments that can be built by the two agents considering the union of there argumentation frameworks (the union of assumption sets) and the fact that they share the same rules. In our protocol, agents are not allowed to play the same move (with the same content) more than one time. We specify the different dialogue rules of our protocol as follows:

### 1- Initial Rule

$C_{in1} \Rightarrow Assert(\alpha, \beta, a)$

where:

$$\begin{aligned} C_{in1} = {} & \exists\, p, q \in \mathcal{L} : \\ & \alpha_p \ncong \beta_q \wedge \mathcal{AR}(\mathcal{A}_\alpha) \rhd a \wedge a \uparrow p \end{aligned}$$

The persuasion starts when a conflict is detected, for example as an outcome of a previous interaction. There is a conflict in the sense that agent $\alpha$ supports $p$ and agent $\beta$ supports $q$, a contrary of $p$. At first, one of the two agents asserts an acceptable argument supporting its position. In the remainder of this section, we suppose that the persuasion topic is represented by the wff $p$.

### 2- Assertion Rule

$$\begin{array}{ll} Assert(\alpha, \beta, \mu) \wedge C_{as1} \Rightarrow Attack(\beta, \alpha, b) & \wedge \\ Assert(\alpha, \beta, \nu) \wedge C_{as2} \Rightarrow Question(\beta, \alpha, Y) & \wedge \\ Assert(\alpha, \beta, \nu) \wedge C_{as3} \Rightarrow Accept(\beta, \alpha, a) & \wedge \\ Assert(\alpha, \beta, \nu) \wedge C_{as4} \Rightarrow Stop(\beta, \alpha) & \end{array}$$

where $\mu$ is an argument or partial argument, $\nu$ is an argument, partial argument, or "?" and:

$C_{as1} \quad = Op_{as_1}^{at_1} \vee (\neg Op_{as_1}^{at_1} \wedge Op_{as_1}^{at_2})$

$$\begin{aligned}
Op_{as_1}^{at_1} &= \exists\, b \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd b \\
&\quad \wedge\ \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b\}) \\
Op_{as_1}^{at_2} &= \exists\, b \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b \\
&\quad \wedge\ \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b\})
\end{aligned}$$

$$\begin{aligned}
C_{as2} &= \neg C_{as1} \wedge (Op_{as_2}^{qu_1} \vee (\neg Op_{as_2}^{qu_1} \wedge Op_{as_2}^{qu_2})) \\
Op_{as_2}^{qu_1} &= \exists\, b_\partial^Y,\ b_\partial^Y.Y \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd b_\partial^Y \\
&\quad \wedge\ \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b_\partial^Y.Y\}) \\
Op_{as_2}^{qu_2} &= \exists\, b_\partial^Y,\ b_\partial^Y.Y \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b_\partial^Y \\
&\quad \wedge\ \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b_\partial^Y.Y\})
\end{aligned}$$

$$\begin{aligned}
C_{as3} &= \exists\, a \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \wedge a \uparrow p \\
&\quad \wedge \neg Op_{as_2}^{qu_1} \wedge \neg Op_{as_2}^{qu_2}
\end{aligned}$$

$$\begin{aligned}
C_{as4} &= \neg Op_{as_1}^{at_1} \wedge \neg Op_{as_2}^{qu_1} \wedge \neg Op_{as_2}^{qu_2} \wedge \neg C_{as3} \\
&\quad \wedge \forall\, b \in Ar,\ \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b \Rightarrow \\
&\qquad\qquad \Delta(p, \cup CS) = \Delta(p, \cup CS \cup \{b\})
\end{aligned}$$

In this rule, the content of $Assert$ could be an argument, partial argument, or "?". Indeed agents can use this move to assert new arguments in the initial rule or to reply to a question in the question rule, which is a part of *inquiry* in our protocol. The move that agent $\beta$ can play as a reply to the $Assert$ move depends on the content of this assertion. When $\alpha$ asserts an argument or a partial argument, $CS_\alpha$ gets changed by adding the advanced (partial) argument. Agent $\beta$ can attack agent $\alpha$ if $\beta$ can generate an acceptable argument from its knowledge base and the $\alpha$'s commitment store so that this argument will change the status of the persuasion topic. Consequently, in this protocol agents do not attack only the last uttered argument, but any uttered argument during the interaction, which is still acceptable or (preferred) semi-acceptable ($Op_{as_1}^{at_1}$). This makes the protocol more flexible and efficient (for example agents can try different arguments to attack a given argument). If such an acceptable argument cannot be generated, $\beta$ will try to generate a (preferred) semi-acceptable argument changing the status of $p$ ($Op_{as_1}^{at_2}$). The idea here is that if $\beta$ cannot make $\alpha$'s arguments eliminated, it will try to make them semi-acceptable or at least preferred semi-acceptable. This is due to the following proposition whose proof is straightforward from the definition of (preferred) semi-acceptable arguments.

**Proposition 2.** *If $\beta$ plays the $Attack$ move with a semi-acceptable argument, then the status of the persuasion topic changes from acceptable to preferred semi-acceptable or semi-acceptable.*

We notice that in the Assertion rule changing the status of $p$ is a result of an attack relation:

**Proposition 3.** *In Assertion rule we have:* $\forall\, b \in Ar$,
$\Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b\}) \Rightarrow \exists\, a \in \cup CS : \mathcal{AT}(b, a)$.

If $\beta$ cannot play the $Attack$ move, then before checking the acceptance of an $\alpha$'s argument, it checks if no acceptable and then no (preferred) semi-acceptable argument in the union of the knowledge bases can attack this argument (inquiry part). For that, if $\beta$ can generate a partial argument changing the status of $p$, then it will question $\alpha$ about the missing assumptions ($Op_{as_2}^{qu_1}$ and $Op_{as_2}^{qu_2}$). This new feature provides a solution to the "pre-determinism" problem identified in [16]. If such a partial argument does not exist, and if $\beta$ can generate an acceptable argument supporting $p$, then it plays the $Accept$ move ($C_{as3}$).

**Proposition 4.** *An agent plays the Accept move only if it cannot play the Attack move and cannot play the Question move.*

***Proof.*** See Appendix

Agent $\beta$ plays the $Stop$ move when it cannot accept an $\alpha$'s argument and cannot attack it. This happens when an agent has a (preferred) semi-acceptable argument for $p$ and the other a (preferred) semi-acceptable argument against $p$, so the status of $p$ in the union of the commitment stores will not change by advancing the $\beta$'s argument ($C_{as4}$). Finally, we notice that if the content of $Assert$ move is "?", $\beta$ cannot play the $Attack$ move. The reason is that such an $Assert$ is played after a question in the Question rule, and agents play $Question$ moves only if an attack is not possible. By simple logical calculus, we can prove the following proposition:

**Proposition 5.** *In the protocol, an agent plays the* $Stop$ *move iff it cannot play another move.*

**3- Attack Rule**

$$\begin{array}{lll}
Attack(\alpha,\beta,a) \wedge C_{at1} \Rightarrow Attack(\beta,\alpha,b) & \wedge \\
Attack(\alpha,\beta,a) \wedge C_{at2} \Rightarrow Question(\beta,\alpha,Y) & \wedge \\
Attack(\alpha,\beta,a) \wedge C_{at3} \Rightarrow Accept(\beta,\alpha,a) & \wedge \\
Attack(\alpha,\beta,a) \wedge C_{at4} \Rightarrow Stop(\beta,\alpha) &
\end{array}$$

where:

$$\begin{array}{ll}
C_{at1} & = Op_{at_1}^{at_1} \vee (\neg Op_{at_1}^{at_1} \wedge Op_{at_1}^{at_2}) \\
Op_{at_1}^{at_1} & = Op_{as_1}^{at_1} \\
Op_{at_1}^{at_2} & = Op_{as_1}^{at_2}
\end{array}$$

$$\begin{array}{ll}
C_{at2} & = \neg C_{at1} \wedge (Op_{at_2}^{qu_1} \vee (\neg Op_{at_2}^{qu_1} \wedge Op_{at_2}^{qu_2})) \\
Op_{at_2}^{qu_1} & = Op_{as_2}^{qu_1} \\
Op_{at_2}^{qu_2} & = Op_{as_2}^{qu_2}
\end{array}$$

$$C_{at3} \quad = \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \triangleright a \wedge \neg Op_{at_2}^{qu_1} \wedge \neg Op_{at_2}^{qu_2}$$

$$\begin{array}{ll}
C_{at4} & = \neg Op_{at_1}^{at_1} \wedge \neg Op_{at_2}^{qu_1} \wedge \neg Op_{at_2}^{qu_2} \wedge \neg C_{at3} \\
 & \quad \wedge \forall\, b \in Ar,\ \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \trianglerighteq b \Rightarrow \\
 & \qquad\qquad \mathit{\Delta}(p, \cup CS) = \mathit{\Delta}(p, \cup CS \cup \{b\})
\end{array}$$

The conditions associated with the Attack rule are similar to the ones defining the Assert rule. The $Attack$ move also includes the case where the agent that initiates the persuasion puts forward a new argument, which is not attacking any existing argument but changing the status of the persuasion topic. This is useful when the advanced arguments cannot be attacked/defended, so that the agent tries another way to convince the addressee.

**4- Question Rule**

$$\begin{array}{l} Question(\alpha, \beta, Y) \wedge C_{qu1} \Rightarrow Assert(\beta, \alpha, a) \quad \wedge \\ Question(\alpha, \beta, Y) \wedge C_{qu2} \Rightarrow Assert(\beta, \alpha, d_{\partial}^{Y'}) \wedge \\ Question(\alpha, \beta, Y) \wedge C_{qu3} \Rightarrow Assert(\beta, \alpha, ?) \end{array}$$

where:

$$\begin{array}{ll} C_{qu1} & = \exists\, a \in Ar : \mathcal{AR}(\mathcal{A}_{\beta} \cup CS_{\alpha}) \rhd a \\ & \wedge (a \uparrow Y \vee a \uparrow \bar{Y}) \end{array}$$

$$\begin{array}{ll} C_{qu2} & = \exists\, d_{\partial}^{Y'}, d_{\partial}^{Y'}.Y' \in Ar : \mathcal{AR}(\mathcal{A}_{\beta} \cup CS_{\alpha}) \rhd d_{\partial}^{Y'} \\ & \wedge (d_{\partial}^{Y'} \uparrow Y \vee d_{\partial}^{Y'} \uparrow \bar{Y}) \end{array}$$

$$C_{qu23} = \neg C_{qu1} \wedge \neg C_{qu2}$$

Agent $\beta$ can answer $\alpha$'s question about the content $Y$ by asserting an argument for or against $Y$. If not, it answers by a partial argument if it can generate it. Otherwise, it answers by "?" which means that it does not know if $Y$ holds or not. We recall that this rule is played when an agent has a partial argument and asks the addressee about the missing assumptions, so that the answer could be the complete missing assumptions, a part of it, or nothing.

**5- Stop Rule**

$$\begin{array}{l} Stop(\alpha, \beta) \wedge C_{st1} \Rightarrow Question(\beta, \alpha, Y) \wedge \\ Stop(\alpha, \beta) \wedge C_{st2} \Rightarrow Stop(\beta, \alpha) \end{array}$$

where:

$$\begin{array}{ll} C_{st1} & = Op_{st_1}^{qu_1} \vee (\neg Op_{st_1}^{qu_1} \wedge Op_{st_1}^{qu_2})) \\ Op_{st_1}^{qu_1} & = Op_{as_2}^{qu_1} \\ Op_{st_1}^{qu_2} & = Op_{as_2}^{qu_2} \end{array}$$

$$C_{st2} = \neg C_{st1}$$

Before answering the $\alpha$'s $Stop$ move by another $Stop$ to terminate the protocol, $\beta$ checks if no other partial arguments changing the status of $p$ could be generated. The

$Stop$ move is played only if no such argument could be generated, which means that the conflict cannot be resolved.

**Theorem 1 (Termination).** *If $\langle \mathcal{C}, \mathcal{D} \rangle$ is a well-formed persuasion protocol about a wff $p$, then $\langle \mathcal{C}, \mathcal{D} \rangle$ always terminates either successfully by $Accept$ or unsuccessfully by $Stop$.*

***Proof.*** See Appendix

**Definition 19 (Soundness - Completeness).** *A persuasion protocol about a wff $p$ is sound and complete iff for some arguments $a$ for or against $p$ we have at the end of the protocol: $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta) \rhd a \Leftrightarrow \mathcal{AR}(\cup CS) \rhd a$.*

**Theorem 2 (Soundness and Completeness).** *If $\langle \mathcal{C}, \mathcal{D} \rangle$ is a well-formed persuasion protocol about a wff $p$, then $\langle \mathcal{C}, \mathcal{D} \rangle$ is sound and complete.*

***Proof.*** See Appendix

## 4 Illustrative Example

This example illustrates a B2B purchase-order scenario involving two businesses ($B_1$ and $B_2$). First, a customer places an order for products via `Customer-WS` (WS for Web service). Based on this order, `Customer-WS` obtains details on the customer's purchase history from `CRM-WS` (Customer Relationship Management) of $B_1$. Afterward, `Customer-WS` forwards these details to $B_1$'s `Billing-WS`, which calculates the customer's bill and sends the bill to `CRM-WS`. This latter prepares the detailed purchase order based on the bill and sends `Inv-Mgmt-WS` (Inventory Management) of $B_1$ this order for fulfillment. Then, `Inv-Mgmt-WS` sends `Shipper-WS` of $B_2$ a shipment request. `Shipper-WS` is now in charge of delivering the products to the customer.

The above scenario could be affected by the following conflict: $B_2$'s `Shipper-WS` may not deliver the products as agreed with $B_1$'s `Inv-Mgmt-WS`, perhaps due to lack of trucks. This is a conflict that could be resolved using our protocol by which, `Shipper-WS` tries to persuade `Inv-Mgmt-WS` about the new shipment time and then inform `Customer-WS` of the new delivery time.

Let $\alpha_{B_1}$ be the agent representing `Inv-Mgmt-WS` of $B_1$ and $\beta_{B_2}$ be the agent representing `Shipper-WS` of $B_2$. The resolution of the conflict along with the use of dialogue games are hereafter provided. For the lack of space reason, we will not give the agents' knowledge bases and we will also omit the arguments representation.

**1-** $\beta_{B_2}$ identifies the conflict (condition $C_{in1}$ is satisfied) and plays the **Initial game** by asserting an acceptable argument $a$ about lack of trucks from its knowledge base $\mathcal{A}_{\beta_{B_2}}$ supporting its position: $Assert(\beta_{B_2}, \alpha_{B_1}, a)$.

**2-** $\alpha_{B_1}$ has an argument $b$ attacking $\beta_{B_2}$'s argument which is about available trucks committed to others that could be used to ship the products (condition $C_{as1}$ is satisfied). $\alpha_{B_1}$ plays then the **Assertion game** by advancing the $Attack$ move: $Attack(\alpha_{B_1}, \beta_{B_2}, b)$.

**3-** $\beta_{B_2}$ replies by playing the **Attack game**. Because it does not have an argument to change the status of the persuasion topic (condition $C_{at1}$ is not satisfied), but has a partial argument for that, which is about the high price of these particular trucks that could be not accepted by $\alpha_{B_1}$ (condition $C_{at2}$ is satisfied), it advances the move: $Question(\beta_{B_2}, \alpha_{B_1}, x)$ where $x$ represents accepting or not the new prices. The idea here is that $\beta_{B_2}$ can attack $\alpha_{B_1}$, if it refuses the new prices that others have accepted.

**4-** $\alpha_{B_1}$ plays the **Question game** and answers the question by asserting an argument $c$ in favor of the increased shipment charges (condition $C_{qu1}$ is satisfied): $Assert(\alpha_{B_1}, \beta_{B_2}, c)$.

**5-** $\beta_{B_2}$ plays the **Assertion game**, and from $\mathcal{A}_{\beta_{B_2}} \cup CS_{\alpha_{B_1}}$, it accepts the argument and agrees to deliver the products as per the agreed schedule with the new price, which is represented by $d$ (condition $C_{as3}$ is satisfied): $Accept(\beta_{B_2}, \alpha_{B_1}, d)$. Consequently, the persuasion terminates successfully by resolving the conflict.

## 5 Related Work

The closest work to the protocol proposed in this paper is the one proposed by [4] for inquiry dialogues. However, there are many fundamental differences between the two protocols. Inquiry and persuasion settings are completely different since the objectives and dynamics of the two dialogues are different. In Black & Hunter's protocol, argumentation is captured only by the notion of argument with no attack relation between arguments. This is because agents collaborate to establish joint proofs. However, in our system, agents can reason about conflicting assumptions, and they should compute different acceptability semantics, not only to win the dispute, but also to reason internally in order to remove inconsistencies from their assumptions. From the specification perspective, there are no similarities between the two protocols. Our protocol is specified as a set of rules about which agents can reason using argumentation, which captures the agents' choices and strategies. However, Black & Hunter's protocol is specified in a declarative manner and the strategy is only defined as a function without specifying how the agents can use it. The adopted moves in the two proposals are also different. Although there is an equivalent notion of partial arguments in Black & Hunter's proposal, the statuses and dynamics we define for these arguments and the fact of considering these arguments as an attempt to change the statuses of uttered arguments are original in our work. Another technical, but fundamental difference in the two protocols is the possibility in our protocol of considering not only the last uttered argument, but any previous argument which allows agents to consider and try different ways of attacking each other.

## References

1. Amgoud, L., and Cayrol, C. 1998. On the acceptability of arguments in preferenced-based argumentation framework. In *Proc. of the 14th Conf. on Uncertainty in Art. Int.*, 1–7.
2. Atkinson, K., Bench-Capon, T., and McBurney, P. 2005. A Dialogue Game Protocol for Multi-Agent Argument over Proposals for Action. *J. of AAMAS*, 11(2):153–171.
3. Besnard, P., and Hunter, A. 2001. A logic-based theory of deductive arguments. *J. of Artificial Intelligence*, 128(1-2):203–235.
4. Black, E., and Hunter, A. 2007. A Generative Inquiry Dialogue System. In *The Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 1010–1017.

5. Bondarenko, A., Dung, P., Kowalski, R.; and Toni, F. 1997. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. *J. of Artificial Intelligence*, 93(1–2):63–101.
6. Brewka, G. 2001. Dynamic Argument Systems: A Formal Model of Argumentation Processes Based on Situation Calculus. *J. of Logic and Computation*, 11(2):257–282.
7. Chesnevar, C., Maguitman, A., and Loui, R. 2000. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383.
8. Dimopoulos, Y., Nebel, B., and Toni, F. 2002. On the computational complexity of assumption-based argumentation for default reasoning. *J. of Artificial Intelligence*, 141(1/2):57-78.
9. Dung, P., Kowalski, R., and Toni, F. 2006. Dialectic proof procedures for assumptionbased, admissible argumentation. *J. of Artificial Intelligence*, 170(2):114-159.
10. Dung, P. 1995. The Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning and Logic Programming and n-Person Game. *J. of Art. Int.*, 77:321–357.
11. Gaertner, D., and Toni, F. 2007. Computing Arguments and Attacks in Assumption-Based Argumentation. *IEEE Intelligent Systems*, 22(6):24–33.
12. Garcia, A., and Simari, G. 2004. Defeasible Logic Programming: an Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138.
13. Kaci, S., and van der Torre, L. 2007. Preference-based Argumentation: Arguments Supporting Multiple Values. *International Journal of Approximate Reasoning*, To appear.
14. Kakas, A., and Toni, F. 1999. Computing argumentation in logic programming. *Journal of Logic Programming*, 9(4):515–562.
15. McBurney, P., and Parsons, S. 2002. Games that Agents Play: A Formal Framework for Dialogues between Autonomous Agents. *Journal of Logic, Language, and Information*, 11(3):315–334.
16. Parsons, S., Wooldridge, M., and Amgoud, L. 2003. On the Outcomes of Formal Inter-Agent Dialogues. In *Proc. of The Int. Conf. on Autonomous Agents and Multiagent Systems*, 616–623.
17. Pasquier, P., Rahwan, I., Dignum, F., and Sonenberg, L. 2006. Argumentation and Persuasion in the Cognitive Coherence Theory. In *The 1st Int. Conf. COMMA*, 223–234. IOS Press.
18. Prakken, H. 2005. Coherence and Flexibility in Dialogue Games for Argumentation. *J. of Logic and Computation*, 15:1009–1040.
19. Rahwan, I., and McBurney, P. 2007. Argumentation Technology. *IEEE Intelligent Systems*, 22(6):21–23.

## Appendix

***Proof of Proposition 4.*** *To prove this we should prove that* $C_{as3} \Rightarrow \neg C_{as1} \wedge \neg C_{as2}$*. Using the logical calculation, we can easily prove that* $\neg C_{as1} \wedge \neg C_{as2} = \neg C_{as1} \wedge \neg Op^{qu_1}_{as_2} \wedge \neg Op^{qu_2}_{as_2}$*. Also, if an agent* $\beta$ *can build an acceptable argument* $a$ *from* $\mathcal{A}_\beta \cup CS_\alpha$*, then it cannot build an acceptable or (preferred) semi-acceptable argument attacking* $a$ *from the same set. Therefore,* $\mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \Rightarrow \neg C_{as1}$*. Thus the result follows.* $\Box$

***Proof of Theorem 1.*** *Agents' knowledge bases are finite and repeating moves with the same content is prohibited. Consequently, the number of* $Attack$ *and* $Question$ *moves that agents can play is finite. At a given moment, agents will have two possibilities only:* $Accept$ *if an acceptable argument can be built from* $CS_\alpha \cup CS_\beta$*, or* $Stop$*, otherwise. Therefore, the protocol terminates successfully by* $Accept$*, or unsuccessfully by* $Stop$

*when $Accept$ move cannot be played, which means that only semi-acceptable arguments are included in $CS_\alpha \cup CS_\beta$.* □

***Proof of Theorem 2.*** *For simplicity and without loss of generality, we suppose that agent $\alpha$ starts the persuasion.*
*Let us first prove the $\Rightarrow$ direction: $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta) \rhd a \Rightarrow \mathcal{AR}(\cup CS) \rhd a$. In the protocol, the persuasion starts when a conflict over $p$ occurs. Consequently, the case where $\mathcal{A}_\alpha \rhd a$ and $\mathcal{A}_\beta \rhd a$ does not hold. Indeed, the possible cases are limited to three:*

1. *$\mathcal{A}_\alpha \rhd a$ and $\mathcal{A}_\beta \ntriangleright a$. In this case, agent $\alpha$ starts the persuasion over $p$ by asserting $a$. Agent $\beta$ can either play the $Attack$ move or the $Question$ move. Because $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta \rhd a)$ all the $\beta$'s arguments will be counter-attacked. For the same reason, $\beta$ cannot play the $Stop$ move. Consequently, at the end, $\beta$ will play an $Accept$ move. It follows that $\mathcal{AR}(\cup CS \rhd a)$.*
2. *$\mathcal{A}_\alpha \ntriangleright a$ and $\mathcal{A}_\beta \rhd a$. In this case, agent $\alpha$ starts the persuasion by asserting an acceptable argument $b$ in its knowledge base against $p$ ($\mathcal{A}_\alpha \rhd b$). This argument will be attacked by agent $\beta$, and the rest is identical to case 1 by substituting agent roles.*
3. *$\mathcal{A}_\alpha \ntriangleright a$ and $\mathcal{A}_\beta \ntriangleright a$. To construct argument $a$ out of $\mathcal{A}_\alpha \cup \mathcal{A}_\beta$, two cases are possible. Either, (1) agent $\alpha$ has an acceptable partial argument $a^Y_\partial$ for $p$ and agent $\beta$ has the missing assumptions (or some parts of the missing assumptions, and agent $\alpha$ has the other parts), or (2) the opposite (i.e., agent $\beta$ has an acceptable partial argument $a^Y_\partial$ for $p$ and agent $\alpha$ has the missing assumptions (or some parts of the missing assumptions, and agent $\beta$ has the other parts)). Only the second case is possible since the first one is excluded by hypothesis. For simplicity, we suppose that agent $\alpha$ has all the missing assumptions, otherwise the missing assumptions will be built by exchanging the different partial arguments. Agent $\alpha$ starts the persuasion by asserting an acceptable argument $b$ in its knowledge base against $p$. Agent $\beta$ can either play an $Attack$ or a $Question$ move. If attack is possible, then agent $\alpha$ can either counter-attack or play the $Stop$ move. The same scenario continues until agent $\alpha$ plays $Stop$, and then agent $\beta$ plays a $Question$ Move. Agent $\alpha$ answers now the question by providing the missing assumptions, after which agent $\beta$ attacks and agent $\alpha$ can only accept since $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta \rhd a)$. It follows that $\mathcal{AR}(\cup CS \rhd a)$.*

*Let us now prove the $\Leftarrow$ direction: $\mathcal{AR}(\cup CS) \rhd a \Rightarrow \mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta) \rhd a$.*

*In the protocol, to have $\mathcal{AR}(\cup CS) \rhd a$ one of the two agents, say agent $\alpha$, puts forward the argument $a$ and the other, agent $\beta$, accepts it. On the one hand, to advance an argument, agent $\alpha$ plays the $Assert$ move (in the initial or question rules) or $Attack$ move (in the assertion or attack rules). In all these cases, we have: $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a$ and there is no partial acceptable argument attacking $a$ from $\mathcal{A}_\alpha \cup SC_\beta$. On the other hand, to accept an argument (in the assertion or attack rules), agent $\beta$ should check that $\mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a$, there is no other arguments changing the status of the persuasion topic, and there is no partial acceptable argument attacking $a$ from $\mathcal{A}_\beta \cup SC_\alpha$. Therefore we obtain: $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta \cup \mathcal{A}_\beta \cup CS_\alpha) \rhd a$. Because $CS_\alpha \subseteq \mathcal{A}_\alpha$ and $CS_\beta \subseteq \mathcal{A}_\beta$ we are done.* □

# Normative Multi-Agent Programs and Their Logics

Mehdi Dastani[1], Davide Grossi[2], John-Jules Ch. Meyer[1], and Nick Tinnemeier[1]

[1] Universiteit Utrecht
The Netherlands
[2] Computer Science and Communication
University of Luxembourg, Luxembourg

**Abstract.** Multi-agent systems are viewed as consisting of individual agents whose behaviors are regulated by an organization artefact. This paper presents a simplified version of a programming language that is designed to implement norm-based artefacts. Such artefacts are specified in terms of norms being enforced by monitoring, regimenting and sanctioning mechanisms. The syntax and operational semantics of the programming language are introduced and discussed. A logic is presented that can be used to specify and verify properties of programs developed in this language.

## 1 Introduction

In this paper, multi-agent systems are considered as consisting of individual agents that are autonomous and heterogenous. Autonomy implies that each individual agent pursues its own objectives and heterogeneity implies that the internal states and operations of individual agents may not be known to external entities [14, 7]. In order to achieve the overall objectives of such multi-agent systems, the observable/external behavior of individual agents and their interactions should be regulated/coordinated.

There are two main approaches to regulate the external behavior of individual agents. The first approach is based on coordination artefacts that are specified in terms of low-level coordination concepts such as synchronization of processes[12]. The second approach is motivated by organizational models, normative systems, and electronic institutions[13, 10, 7, 8]. In such an approach, norm-based artefacts are used to regulate the behavior of individual agents in terms of norms being enforced by monitoring, regimenting and sanctioning mechanisms. Generally speaking, the social and normative perspective is conceived as a way to make the development and maintenance of multi-agent systems easier to manage. A plethora of social concepts (e.g., roles, social structures, organizations, institutions, norms) has been introduced in multi-agent system methodologies (e.g. Gaia [14]), models (e.g. OperA [6], $\mathcal{M}$oise$^+$ [9], electronic institutions and frameworks (e.g. AMELI [7], $\mathcal{S}$-$\mathcal{M}$oise$^+$ [9]).

The main contribution of this paper is twofold. On the one hand, a simplified version of a programming language is presented that is designed to implement multi-agent systems in which the observable (external) behavior of individual agents is regulated by means of norm-based artefacts. Such artefacts are implemented in terms of social concepts such as norms and sanctions, monitor the actions performed by individual agents, evaluate their effects, and impose sanctions if necessary. On the other hand, we devise a logic to specify and verify properties of programs that implement norm-based artefacts.

In order to illustrate the idea of norm-based artefacts, consider the following simple example of a simulated train station where agents ought to buy a ticket before entering the platform or trains. To avoid the queue formation, agents are not checked individually before allowing them to enter the platform or trains. In this simulation, being on the platform without a ticket is considered as a violation and getting on the train without having a ticket is considered as a more severe violation. A norm-based artefact detects (all or some) violations by (all or some) agents and reacts on them by issuing a fine if the first violation occurs, for instance by charging the credit card of the defecting user, and a higher fine if the second violation occurs.

In this paper, we first briefly explain our idea of normative multi-agent systems and discuss two norm-based approaches to multi-agent systems, that is, ISLANDER/AMELI [7] and S-MOISE+ [9]. In section 3, we present the syntax and operational semantics of a programming language designed to implement normative multi-agent systems. This programming language allows the implementation of norm-based artefacts by providing programming constructs to represent norms and mechanisms to enforce them. In section 4, a logic is presented that can be used to specify and verify properties of norm-based artefacts implemented in the presented programming language. Finally, in section 5, we conclude the paper and discuss some future directions in this research area.

## 2 Norms and Multi-Agent Systems

Norms in multi-agent systems can be used to specify the standards of behavior that agents ought to follow to meet the overall objectives of the system. However, to develop a multi-agent system does not boil down to state a number of standards of behavior in the form of a set of norms, but rather to organize the system in such a way that those standards of behavior are actually followed by the agents. This can be achieved by regimentation [10] or enforcement mechanisms, e.g., [8].

When regimenting norms all agents' external actions leading to a violation of those norms are made impossible. Via regimentation (e.g., gates in train stations) the system prevents an agent from performing a forbidden action (e.g., entering a train platform without a ticket). However, regimentation drastically decreases agent autonomy. Instead, enforcement is based on the idea of responding after a violation of the norms has occurred. Such a response, which includes sanctions, aims to return the system to an acceptable/optimal state. Crucial for enforcement is that the actions that violate norms are observable by the system (e.g., fines can be issued only if the system can detect travelers entering the platform or trains without a ticket). Another advantage of having enforcement over regimentation is that allowing for violations contributes to the flexibility and autonomy of the agent's behavior [3]. These norms are often specified by means of concepts like permissions, obligations, and prohibitions.

In the literature of multi-agent systems related work can be found on electronic institutions. In particular, ISLANDER[7] is a formal framework for specifying norms in institutions, which is used in the AMELI platform [7] for executing electronic institutions based on norms provided in it. However, the key aspect of ISLANDER/AMELI is that norms can never be violated by agents. In other words, systems programmed via ISLANDER/AMELI make only use of regimentation in order to guarantee the norms to be actually followed. This is an aspect which our approach intends to relax guaranteeing higher autonomy to the agents, and higher flexibility to the system.

A work that is concerned with programming multiagent systems using (among others) normative concepts is also S-MOISE+, which is an organizational middleware that follows the Moise+ model[9]. This approach, like ours, builds on programming constructs investigated in social and organizational sciences. However, S-MOISE+ lacks formal operational semantics, which is instead the main contribution of the present paper to the development of programming languages form multi-agent systems. Besides, norms in S-MOISE+ typically lack monitoring and sanctioning mechanisms for their implementation which are, instead, the focus of our proposal. It should be noted that [11] advocates the use of artifacts to implement norm enforcement mechanisms. However, it is not explained how this can be done using those artifacts.

To summarize, ISLANDER/AMELI implements norm via full regimentation, while in S-MOISE+ violations are possible, although no specific system's response to violations is built in the framework. We deem these shortcomings to have a common root, namely the absence of a computational model of norms endowed with a suitable operational semantics. The present paper fills this gap along the same lines that have been followed for the operationalization of BDI notions in the APL-like agent programming languages [5, 4]. Finally, it should be noted that besides normative concepts MOISE+ and ISLANDER/AMELI also provide a variety of other social and organizational concepts. Since the focus of this paper is on the normative aspect, the above discussion is limited hereto. Future research will focus on other social and organizational concepts.

## 3 Programming Multi-Agent Systems with Norms

In this section, we present a programming language to facilitate the implementation of multi-agent systems with norms, i.e., to facilitate the implementation of norm-based artefacts that coordinate/regulate the behavior of participating individual agents. A normative multi-agent system (i.e., a norm-based artefact) is considered to contain two modules: an organization module that specifies norms and sanctions, and an environment module in which individual agents can perform actions. The individual agents are assumed to be implemented in a programming language, not necessarily known to the multi-agent system programmer, though the programmer is required to have the reference to the (executable) programs of each individual agent. It is also assumed that all actions that are performed by individual agents are observable to the multi-agent system (i.e., norm-based artefact). Note that the reference to the (executable) programs of individual agents are required such that multi-agent systems (i.e., normative artefact) can observe the actions generated by the agent programs. Finally, we assume that the effect of an individual agent's action in the external environment is determined by the program that implements the norm-based artefact (i.e., by the multi-agent system program). Most noticeably it is not assumed that the agents are able to reason about the norms of the system.

The programming language for normative multi-agent systems provides programming constructs to specify the effect of an agent's actions in the environment, norms, sanctions, and the initial state of the environment. Moreover, the programming language is based on a monitoring and a sanctioning mechanism that observes the actions performed by the agents, determines their effects in the shared environment, determines the violations caused by performing the actions, and possibly, imposes sanctions. A program in this language is the implementation of a norm-based artefact. As we assume that the norm-based artefacts determine the effects of external actions in the shared environment, the programming language should provide constructs to implement these effects. The effect of an agent's (external) actions is specified by a set of literals that should hold in the shared environment after the external action is performed by the agent. As external actions can have different effects when they are executed in different states of the shared environment, we add a set of literals that function as the pre-condition of those effect.

We consider norms as being represented by counts-as rules [13], which ascribe "institutional facts" (e.g. "a violation has occurred"), to "brute facts" (e.g. "an agent is on the train without ticket"). For example, a counts-as rule may express the norm "an agent on the train without ticket counts-as a violation". In our framework, brute facts constitute the environment shared by the agents, while institutional facts constitute the normative/institutional state of the multi-agent system. Institutional facts are used with the explicit aim of triggering system's reactions (e.g., sanctions). As showed in [8] counts-as rules can enjoy a rather classical logical behavior, and are here implemented as simple rules that relate brute and normative facts. In the presented programming language, we distinguish brute facts from normative (institutional) facts and assume two disjoint sets of propositions to denote these facts.

Brute and institutional facts constitute the (initial) state of the multi-agent system (i.e., the state of the norm-based artefact). Brute facts are initially set by the programmer by means of the initial state of the shared environment. These facts can change as individual agents perform actions in the shared environment. Normative facts are determined by applying counts-as rules in multi-agent states. The application of counts-as rules in subsequent states of a multi-agent system realizes a monitoring mechanism as it determines and detects norm violations during the execution of the multi-agent system.

Sanctions are also implemented as rules, but follow the opposite direction of counts-as rules. A sanction rule determines which brute facts will be brought about by the system as a consequence of the normative facts. Typically, such brute facts are sanctions, such as fines. Notice that in human systems sanctions are usually issued by specific agents (e.g. police agents). This is not the case in our computational setting, where sanctions necessarily follow the occurrence of a violation if the relevant sanction rule is in place (comparable to automatic traffic control and issuing tickets). It is important to stress, however, that this is not an intrinsic limitation of our approach. We do not aim at mimicking human institutions but rather providing the specification of computational systems.

### 3.1 Syntax.

In order to represent brute and institutional facts in our normative multi-agent systems programming language, we introduce two disjoint sets of propositions to denote these facts. The syntax of the normative multi-agent system programming language is presented below using the EBNF notation. In the following, we use `<b-prop>` and `<i-prop>` to be propositional formulae taken from two different disjoint sets of propositions. Moreover, we use `<ident>` to denote a string and `<int>` to denote an integer.

```
N-MAS_Prog   := "Agents: " (<agentName> <agentProg> [<nr>])+ ;
                "Facts: " <bruteFacts>
                "Effects: " <effects>
                "Counts-as rules: " <counts-as>
                "Sanction rules: " <sanctions>;
<agentName>  := <ident>;
<agentProg>  := <ident>;
<nr>         := <int>;
<bruteFacts> := <b-literals>;
<effects>    := ({<b-literals>} <actionName> {<b-literals>})+;
<counts-as>  := ( <literals> ⇒ <i-literals> )+;
<sanctions>  := ( <i-literals> ⇒ <b-literals>)+;
<actionName> := <ident>;
<b-literals> := <b-literal> {"," <b-literal>};
<i-literals> := <i-literal> {"," <i-literal>};
<literals>   := <literal> {"," <literal>};
<literal>    := <b-literal> | <i-literal>;
<b-literal>  := <b-prop> | "not" <b-prop>;
<i-literal>  := <i-prop> | "not" <i-prop>;
```

In order to illustrate the use of this programming language, consider the following underground station example.

```
Agents:          passenger   PassProg   1
Facts:           {-at_platform, -in_train, -ticket}
Effects:         {-at_platform} enter {at_platform},
                 {-ticket} buy_ticket {ticket},
                 {at_platform, -in_train} embark {-at_platform, in_train}
Counts_as rules: {at_platform , -ticket} ⇒ {viol_1},
                 {in_train , -ticket} ⇒ {viol_⊥}
Sanction rules:  {viol_1}  ⇒  {fined_10}
```

This program creates one agent called `passenger` whose (executable) specification is included in a file with the name `PassProg`. The `Facts`, which implement brute facts, determine the initial state of the shared environment. In this case, the agent is not at the platform (`-at_platform`) nor in the train (`-in_train`) and has no ticket (`-ticket`). The `Effects` indicate how the environment can advance in its computation. Each effect is of the form {`pre-condition`} `action` {`post-condition`}. The first effect, for instance, means that if the agent performs an `enter` action when not at the platform, the result is that the agent is on the platform (either with or without a ticket). Only those effects that are changed are thus listed in the post-condition. The `Counts_as rules` determine the normative effects for a given (brute and normative) state of the multi-agent system. The first rule, for example, states that being on the platform without having a ticket is a specific violation (marked by $\text{viol}_1$). The second rule marks states where agents are on a train without a ticket with the specifically designated literal $\text{viol}_\bot$. This literal is used to implement regimentation. The operational semantics of the language ensures that the designated literal $\text{viol}_\bot$ can never hold during any run of the system (see Definition 3). Intuitively, rules with $\text{viol}_\bot$ as consequence could be thought of as placing gates blocking an agent's action. Finally, the aim of `Sanction rules` is to determine the punishments that are imposed as a consequence of violations. In the example the violation of type $\text{viol}_1$ causes the sanction $\text{fined}_{10}$ (e.g., a 10 EUR fine).

Counts-as rules obey syntactic constraints. Let $l = (\Phi \Rightarrow \Psi)$ be a rule, we use $\texttt{cond}_l$ and $\texttt{cons}_l$ to indicate the condition $\Phi$ and consequent $\Psi$ of the rule $l$, respectively. We consider only sets of rules such that 1) they are finite; 2) they are such that each condition has exactly one associated consequence (i.e., all the consequences of a given conditions are packed in one single set `cons`); and 3) they are such that for counts-as rule $k, l$, if $\texttt{cons}_k \cup \texttt{cons}_l$ is inconsistent (i.e., contains $p$ and $\neg p$), then $\texttt{cond}_k \cup \texttt{cond}_l$ is also inconsistent. That is to say, rules trigger inconsistent conclusions only in different states. In the rest of this paper, sets of rules enjoying these three properties are denoted by $\mathbf{R}$.

### 3.2 Operational Semantics.

One way to define the semantics of this programming language is by means of operational semantics. Using such semantics, one needs to define the configuration (i.e., state) of normative multi-agent systems and the transitions that such configurations can undergo through transition rules. The state of a multi-agent system with norms consists of the state of the external environment, the normative state, and the states of individual agents.

**Definition 1.** *(Normative Multi-Agent System Configuration) Let $P_b$ and $P_n$ be two disjoint sets of literals denoting atomic brute and normative facts (including* $\mathrm{viol}_\perp$*), respectively. Let $A_i$ be the configuration of individual agent $i$. The configuration of a normative multi-agent system is defined as $\langle \mathcal{A}, \sigma_b, \sigma_n \rangle$ where $\mathcal{A} = \{A_1, \ldots, A_n\}$, $\sigma_b$ is a consistent set of literals from $P_b$ denoting the brute state of multi-agent system and $\sigma_n$ is a consistent set of literals from $P_n$ denoting the normative state of multi-agent system.*

The configuration of such a multi-agent system can change for various reasons, e.g., because individual agents perform actions in the external environment or because the external environment can have its own internal dynamics (the state of a clock changes independent of an individual agent's action). In operational semantics, transition rules specify how and when configurations can change, i.e., they specify which transition between configurations are allowed and when they can be derived. In this paper, we consider only the transition rules that specify the transition of multi-agent system configurations as a result of performing external actions by individual agents. Of course, individual agents can perform (internal) actions that modify only their own configurations and have no influence on the multi-agent system configuration. The transition rules to derive such transitions are out of the scope of this paper.

**Definition 2.** *(Transitions of Individual Agent's Actions) Let $A_i$ and $A_i'$ be configurations of individual agent $i$, and $\alpha(i)$ be an (observable) external action performed by agent $i$. Then, the following transition captures the execution of an external action by an agent.*

$$A_i \xrightarrow{\alpha(i)} A_i' : \textit{agent } i \textit{ can perform external action } \alpha$$

This transition indicates that an agent configuration can change by performing an external action. The performance of the external action is broadcasted to the multi-agent system level. Note that no assumption is made about the internals of individual agents as we do not present transition rules for deriving internal agent transitions (denoted as $A \longrightarrow A'$). The only assumption is that the action of the agent is observable. This is done by labeling the transition with the external action name.

Before presenting the transition rule specifying the possible transitions of the normative MAS configurations, the closure of a set of conditions under a set of (counts-as and sanction) rules needs to be defined. Given a set $\mathbf{R}$ of rules and a set $X$ of literals, we define the set of applicable rules in $X$ as $\mathtt{Appl}^{\mathbf{R}}(X) = \{\Phi \Rightarrow \Psi \mid X \models \Phi\}$. The closure of $X$ under $\mathbf{R}$, denoted as $\mathtt{Cl}^{\mathbf{R}}(X)$, is inductively defined as follows:

**B:** $\mathtt{Cl}_0^{\mathbf{R}}(X) = X \cup (\bigcup_{l \in \mathtt{Appl}^{\mathbf{R}}(X)} \mathtt{cons}_l)$
**S:** $\mathtt{Cl}_{n+1}^{\mathbf{R}}(X) = \mathtt{Cl}_n^{\mathbf{R}}(X) \cup (\bigcup_{l \in \mathtt{Appl}^{\mathbf{R}}(\mathtt{Cl}_n^{\mathbf{R}}(X))} \mathtt{cons}_l)$

Because of the properties of finiteness, consequence uniqueness and consistency of $\mathbf{R}$ one and only one finite number $m+1$ can always be found such that $\mathtt{Cl}^{\mathbf{R}}_{m+1}(X) = \mathtt{Cl}^{\mathbf{R}}_{m}(X)$ and $\mathtt{Cl}^{\mathbf{R}}_{m}(X) \neq \mathtt{Cl}^{\mathbf{R}}_{m-1}(X)$. Let such $m+1$ define the closure $X$ under $\mathbf{R}$: $\mathtt{Cl}^{\mathbf{R}}(X) = \mathtt{Cl}^{\mathbf{R}}_{m+1}(X)$. Note that the closure may become inconsistent due to the ill-defined set of counts-as rules. For example, the counts-as rule $p \Rightarrow -p$ (or the set of counts as rules $\{p \Rightarrow q\ ,\ q \Rightarrow -p\}$), where $p$ and $q$ are normative facts, may cause the normative state of a multi-agent system to become inconsistent.

We can now define a transition rule to derive transitions between normative multi-agent system configurations. In this transition rule, the function *up* determines the effect of action $\alpha(i)$ on the environment $\sigma_b$ based on its specification $(\Phi\ \alpha(i)\ \Phi')$ as follows:

$$up(\alpha(i), \sigma_b) = (\sigma_b \cup \Phi') \setminus (\{p \mid -p \in \Phi'\} \cup \{-p \mid p \in \Phi'\})$$

**Definition 3.** *(Transition Rule for Normative Multi-Agent Systems) Let $\mathbf{R_c}$ be the set of counts-as rules, $\mathbf{R_s}$ be the set of sanction rules, and $(\Phi\ \alpha(i)\ \Phi')$ be the specification of action $\alpha(i)$. The multi-agent transition rule for the derivation of normative multi-agent system transitions is defined as follows:*

$$\frac{\begin{array}{c} A_i \in \mathcal{A} \quad \& \quad A_i \overset{\alpha(i)}{\rightarrow} A'_i \quad \& \quad \sigma_b \models \Phi \quad \& \quad \sigma'_b = up(\alpha(i), \sigma_b) \\ \sigma'_n = \mathtt{Cl}^{\mathbf{R_c}}(\sigma'_b) \setminus \sigma'_b \quad \& \quad \sigma'_n \not\models \mathrm{viol}_\perp \quad \& \quad S = \mathtt{Cl}^{\mathbf{R_s}}(\sigma'_n) \setminus \sigma'_n \quad \& \quad \sigma'_b \cup S \not\models \perp \end{array}}{\langle \mathcal{A}, \sigma_b, \sigma_n \rangle \longrightarrow \langle \mathcal{A}', \sigma'_b \cup S, \sigma'_n \rangle}$$

*where $\mathcal{A}' = (\mathcal{A} \setminus \{A_i\}) \cup \{A'_i\}$ and* $\mathrm{viol}_\perp$ *is the designated literal for regimentation.*

This transition rule captures the effects of performing an external action by an individual agent on both external environments and the normative state of the MAS. First, the effect of $\alpha$ on $\sigma_b$ is computed. Then, the updated environment is used to determine the new normative state of the system by applying all counts-as rules to the new state of the external environments. Finally, possible sanctions are added to the new environment state by applying sanction rules to the new normative state of the system. In should be emphasized that other multi-agent transition rules, such as transition rules for communication actions, are not presented in this paper because the focus here is on how norms determine the effects of external actions.

Note that the external action of an agent can be executed only if it would not result in a state containing $\mathrm{viol}_\perp$. This captures exactly the regimentation of norms. Hence, once assumed that the initial normative state does not include $\mathrm{viol}_\perp$, it is easy to see that the system will never be in a $\mathrm{viol}_\perp$-state. It is important to note that when a normative state $\sigma'_n$ becomes inconsistent, the proposed transition rule cannot be applied because an inconsistent $\sigma'_n$ entails $viol_\perp$. Also, note that the condition $\sigma'_b \cup S \not\models \perp$ guarantees that the environment state never can become inconsistent. Finally, it should be emphasized that the normative state $\sigma'_b$ is not defined on $\sigma_n$ and is always computed anew.

## 4 Logic

In this section, we propose a logic to specify and verify liveness and safety properties of multi-agent system programs with norms. This logic, which is a variant of Propositional Dynamic Logic (PDL, see [2]), is in the spirit of [1] and rely on that work. It is important to note that the logic developed in [1] aims at specifying and verifying properties of single agents programmed in terms of beliefs, goals, and plans. Here we modify the logic and apply it to multi-agent system programs. We first introduce some preliminaries before presenting the logic.

### 4.1 Preliminaries

We show how the programming constructs can be used for grounding a logical semantics. Let $P$ denote the set of propositional variables used to describe brute and normative states of the system. It is assumed that each propositional variable in $P$ denotes either an institutional/normative or a brute state-of-affairs: $P = P_n \cup P_b$ and $P_n \cap P_b = \emptyset$. A state $s$ is represented as a pair $\langle \sigma_b, \sigma_n \rangle$ where $\sigma_b = \{(-)p_1, \dots, (-)p_n : p_i \in P_b\}$ is a consistent set of literals (i.e., for no $p \in P_b$ it is the case that $p \in \sigma_b$ and $-p \in \sigma_b$), and $\sigma_n$ is like $\sigma_b$ for $P_n$.

Rules are pairs of conditions and consequences $(\{(-)p_1, \dots, (-)p_n \mid (-)p_i \in X\}, \{(-)q_1, \dots, (-)q_k \mid (-)q_i \in Y\})$ with $X$ and $Y$ being either $\sigma_b$ or $\sigma_n$ when applied in state $\langle \sigma_b, \sigma_n \rangle$. Following [8], if $X = \sigma_b$ and $Y = \sigma_n$ then the rule is called *bridge counts-as rule*; if $X = Y = \sigma_n$ then the rule is an *institutional counts-as rule*; if $X = \sigma_n$ and $Y = \sigma_b$ then the rule is a *sanction rule*. Literals $p$'s and $q$'s are taken to be disjoint. Leaving technicalities aside, bridge counts-as rules connect brute states to normative/institutional ones, institutional counts-as rules connect institutional facts to institutional facts, and sanction rules connect normative states to brute ones.

Given a set $\mathbf{R}$ of rules, we say a state $s = \langle \sigma_b, \sigma_n \rangle$ to be $\mathbf{R}$-aligned if for all pairs $(\mathtt{cond}_k, \mathtt{cons}_k)$ in $\mathbf{R}$: if $\mathtt{cond}_k$ is satisfied either by $\sigma_b$ (in the case of a bridge counts-as rule) or by $\sigma_n$ (in the case of an institutional counts-as or a sanction rule), then $\mathtt{cons}_k$ is satisfied by $\sigma_n$ (in the case of a bridge or institutional counts-as rule) or by $\sigma_b$ (in the case of a sanction rule), respectively. States that are $\mathbf{R}$-aligned are states which instantiate the normative system specified by $\mathbf{R}$.

Let the set of agents' external actions $\mathtt{Ac}$ be the union $\bigcup_{i \in I} \mathtt{Ac}_i$ of the finite sets $\mathtt{Ac}_i$ of external actions of each agent $i$ in the set $I$. We denote external actions as $\alpha(i)$ where $\alpha \in \mathtt{Ac}_i$ and $i \in I$. We associate now with each $\alpha(i) \in \mathtt{Ac}_i$ a set of pre- and post-conditions $\{(-)p_1 \in \sigma_b, \dots, (-)p_n \in \sigma_b\}$, $\{(-)q_1 \in \sigma'_b, \dots, (-)q_k \in \sigma'_b\}$ (where $p$'s and $q$'s are not necessarily disjoint) when $\alpha(i)$ is executed in a state with brute facts set $\sigma_b$ which satisfies the pre-condition then the resulting state $s'$ has the brute facts set $\sigma'_b$ which satisfies the post-condition (including replacing $p$ with $-p$ if necessary to preserve consistency) and it is such that *the rest of* $\sigma'_b$ *is the same as* $\sigma_b$. Executing an action $\alpha(i)$ in different configurations may give different results. For each $\alpha(i)$, we denote the set of pre- and post-condition pairs $\{(\mathtt{prec}_1, \mathtt{post}_1), \dots, (\mathtt{prec}_m, \mathtt{post}_m)\}$ by $C_b(\alpha(i))$. We assume that $C_b(\alpha(i))$ is finite, that pre-conditions $\mathtt{prec}_k, \mathtt{prec}_l$ are mutually exclusive if $k \neq l$, and that each pre-condition has exactly one associated post-condition. We denote the set of all such pre- and post-conditions of all agents' external actions by $\mathbf{C}$.

Now everything is put into place to show how the execution of $\alpha(i)$ in a state with brute facts set $\sigma_b$ also univocally changes the normative facts set $\sigma_n$ by means of the applicable counts-as rules, and adds the resulting sanctions by means of the applicable sanction rules. If $\alpha(i)$ is executed in a state $\langle \sigma_b, \sigma_n \rangle$ with brute facts set $\sigma_b$, which satisfies the pre-conditions, then the resulting state $\langle \sigma'_b \cup S, \sigma'_n \rangle$ is such that $\sigma'_b$ satisfies the brute post-condition of $\alpha(i)$ (including replacing $p$ with $-p$ if necessary) and the rest of $\sigma'_b$ is the same of $\sigma_b$; $\sigma'_n$ is determined by the closure of $\sigma'_b$ with counts-as rules $\mathbf{R}_c$; sanctions $S$ are obtained via closure of $\sigma'_n$ with sanction rules $\mathbf{R}_s$.

### 4.2 Language

The language $L$ for talking about normative multi-agent system programs is just the language of PDL built out of a finite set of propositional variables $P \cup -P$ (i.e., the literals built from $P$), used to describe the system's normative and brute states, and a finite set $\mathtt{Ac}$ of agents' actions. Program expressions $\rho$ are built out of external actions $\alpha(i)$ as usual, and formulae $\phi$ of $L$ are closed under boolean connectives and modal operators:

$$\rho ::= \alpha(i) \mid \rho_1 \cup \rho_2 \mid \rho_1 ; \rho_2 \mid ?\phi \mid \rho^*$$
$$\phi ::= (-)p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle \rho \rangle \phi$$

with $\alpha(i) \in \mathtt{Ac}$ and $(-)p \in P \cup -P$. Connectives $\vee$ and $\rightarrow$, and the modal operator $[\rho]$ are defined as usual.

### 4.3 Semantics.

The language introduced above is interpreted on transition systems that generalize the operational semantics presented in the earlier section, in that they do not describe a particular program, but all possible programs —according to $\mathbf{C}$— generating transitions between all the $\mathbf{R_c}$ and $\mathbf{R_s}$-aligned states of the system. As a consequence, the class of transition systems we are about to define will need to be parameterized by the sets $\mathbf{C}$, $\mathbf{R_c}$ and $\mathbf{R_s}$.

A model is a structure $M = \langle S, \{R_{\alpha(i)}\}_{\alpha(i) \in \mathtt{Ac}}, V \rangle$ where:

- $S$ is a set of $\mathbf{R_c}$ and $\mathbf{R_s}$-aligned states.
- $V = (V_b, V_n)$ is the evaluation function consisting of brute and normative valuation functions $V_b$ and $V_n$ such that for $s = \langle \sigma_b, \sigma_n \rangle$, $V_b(s) = \sigma_b$ and $V_n(s) = \sigma_n$.
- $R_{\alpha(i)}$, for each $\alpha(i) \in \mathtt{Ac}$, is a relation on $S$ such that $(s, s') \in R_{\alpha(i)}$ iff for some $(\mathtt{prec}_k, \mathtt{post}_k) \in C(\alpha(i))$, $\mathtt{prec}_k(s)$ and $\mathtt{post}_k(s')$, i.e., for some pair of pre- and post-conditions of $\alpha(i)$, the pre-condition holds for $s$ and the corresponding post-condition holds for $s'$. Note that this implies two things.

First, an $\alpha(i)$ transition can only originate in a state $s$ which satisfies one of the pre-conditions for $\alpha(i)$. Second, since pre-conditions are mutually exclusive, every such $s$ satisfies exactly one pre-condition, and all $\alpha(i)$-successors of $s$ satisfy the matching post-condition.

Given the relations corresponding to agents' external actions in $M$, we can define sets of paths in the model corresponding to any PDL program expression $\rho$ in $M$. A set of paths $\tau(\rho) \subseteq (S \times S)^*$ is defined inductively:

- $\tau(\alpha(i)) = \{(s, s') : R_{\alpha(i)}(s, s')\}$
- $\tau(\phi?) = \{(s, s) : M, s \models \phi\}$
- $\tau(\rho_1 \cup \rho_2) = \{z : z \in \tau(\rho_1) \cup \tau(\rho_2)\}$
- $\tau(\rho_1; \rho_2) = \{z_1 \circ z_2 : z_1 \in \tau(\rho_1),\ z_2 \in \tau(\rho_2)\}$, where $\circ$ is concatenation of paths , such that $z_1 \circ z_2$ is only defined if $z_1$ ends in the state where $z_2$ starts
- $\tau(\rho^*)$ is the set of all paths consisting of zero or finitely many concatenations of paths in $\tau(\rho)$ (same condition on concatenation as above)

Constructs such as `If` $\phi$ `then` $\rho_1$ `else` $\rho_2$ and `while` $\phi$ `do` $\rho$ are defined as $(\phi?; \rho_1) \cup (\neg\phi?; \rho_2)$ and $(\phi?; \rho)^*; \neg\phi$, respectively. The satisfaction relation $\models$ is inductively defined as follows:

- $M, s \models (\text{-})p$ iff $(\text{-})p \in V_b(s)$ for $p \in P_b$
- $M, s \models (\text{-})p$ iff $(\text{-})p \in V_n(s)$ for $p \in P_n$
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$
- $M, s \models \phi \wedge \psi$ iff $M, s \models \phi$ and $M, s \models \psi$
- $M, s \models \langle\rho\rangle\phi$ iff there is a path in $\tau(\rho)$ starting in $s$ which ends in a state $s'$ such that $M, s' \models \phi$.
- $M, s \models [\rho]\phi$ iff for all paths $\tau(\rho)$ starting in $s$, the end state $s'$ of the path satisfies $M, s' \models \phi$.

Let the class of transition systems defined above be denoted $\mathbf{M}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$ where $\mathbf{C}$ is the set of pre- and post-conditions of external actions, $\mathbf{R_c}$ is the set of counts-as rules and $\mathbf{R_s}$ the set of sanction rules.

### 4.4 Axiomatics.

The axiomatics shows in what the logic presented differs w.r.t. standard PDL. In fact, it is a conservative extension of PDL with domain-specific axioms needed to axiomatize the behavior of normative multi-agent system programs.

For every pre- and post-condition pair $(\mathtt{prec}_i, \mathtt{post}_i)$ we describe states satisfying $\mathtt{prec}_i$ and states satisfying $\mathtt{post}_i$ by formulas of $L$. More formally, we define a formula $\mathit{tr}(X)$ corresponding to a pre- or post-condition $X$ as follows: $\mathit{tr}((\text{-})p) = (\text{-})p$ and $\mathit{tr}(\{\phi_1, \ldots, \phi_n\}) = \mathit{tr}(\phi_1) \wedge \ldots \wedge \mathit{tr}(\phi_n)$. This allows us to axiomatize pre- and post-conditions of actions. The conditions and consequences of counts-as rules and sanction rules can be defined in similar way as pre- and post-conditions of actions, respectively. The set of models $\mathbf{M}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$ is axiomatized as follows:

**PDL** Axioms and rules of PDL
**Ax Consistency** Consistency of literals: $\neg(p \wedge \neg p)$
**Ax Counts-as** For every rule $(\mathtt{cond}_k, \mathtt{cons}_k)$ in $\mathbf{R_c}$: $\mathit{tr}(\mathtt{cond}_k) \rightarrow \mathit{tr}(\mathtt{cons}_k)$
**Ax Sanction** For every rule $(\mathtt{viol}_k, \mathtt{sanc}_k)$ in $\mathbf{R_s}$: $\mathit{tr}(\mathtt{viol}_k) \rightarrow \mathit{tr}(\mathtt{sanc}_k)$
**Ax Regiment** $\mathrm{viol}_\perp \rightarrow \perp$
**Ax Frame** For every action $\alpha(i)$ and every pair of pre- and post-conditions $(\mathtt{prec}_j, \mathtt{post}_j)$ in $C(\alpha(i))$ and formula $\Phi$ built out of $P_b$ not containing any propositional variables occurring in $\mathtt{post}_j$:

$$\mathit{tr}(\mathtt{prec}_j) \wedge \Phi \rightarrow [\alpha(i)](\mathit{tr}(\mathtt{post_j}) \wedge \Phi)$$

This is a frame axiom for actions.
**Ax Non-Executability** For every action $\alpha(i)$, where all possible pre-conditions in $C(\alpha(i))$ are $\mathtt{prec}_1, \ldots, \mathtt{prec}_k$: $\neg\mathit{tr}(\mathtt{prec}_1) \wedge \ldots \wedge \neg\mathit{tr}(\mathtt{prec}_k) \rightarrow \neg\langle\alpha(i)\rangle\top$ where $\top$ is a tautology.
**Ax Executability** For every action $\alpha(i)$ and every pre-condition $\mathtt{prec}_j$ in $C(\alpha(i))$: $\mathit{tr}(\mathtt{prec}_j) \rightarrow \langle\alpha(i)\rangle\top$

Let us call the axiom system above $\mathbf{Ax}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$, where $\mathbf{C}$ is the set of brute pre- and post-conditions of atomic actions, $\mathbf{R_c}$ is the set of counts-as rules, and $\mathbf{R_s}$ is the set of sanction rules.

**Theorem 1.** *Axiomatics* $\mathbf{Ax}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$ *is sound and weakly complete for the class of models* $\mathbf{M}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$*.*

*Proof.* Soundness is proven as usual by induction on the length of derivations. We sketch the proof of completeness. It builds on the usual completeness proof of PDL via finite canonical models. Given a consistent formula $\phi$ to be proven satisfiable, such models are obtained via the Fischer-Ladner closure of the set of subformulae of the formula $\phi$ extended with all pre- and post-conditions of any action $\alpha(i)$ occurring in $\phi$. Let $FLC(\phi)$ denote such closure. The canonical model consists of all the maximal $\mathbf{Ax}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$-consistent subsets of $FLC(\phi)$. The accessibility relation and the valuation of the canonical model are defined like in PDL and the truth lemma follows in the standard way. It remains to be proven that the model satisfies the axioms. First, since the states in the model are maximal and consistent w.r.t. *Ax Counts-as*, *Ax Sanction*, *Ax Consistency*, and *AxRegiment*, they are $\mathbf{R_c}$- and $\mathbf{R_s}$-aligned, $\sigma_b$ and $\sigma_n$ are consistent, and no state is such that $\sigma_n \models \mathtt{viol}_\perp$. Second, it should be shown that the canonical model satisfies the pre- and post-conditions of the actions occurring in $\phi$ in that: a) no action $\alpha(i)$ is executable in a state $s$ if none of its preconditions are satisfied by $s$, and b) if they hold in $s$ then the corresponding post-conditions hold in $s'$ which is accessible by$R_{\alpha(i)}$ from $s$. As to a), if a state $s$ in the canonical model does not satisfy any of the preconditions of $\alpha(i)$ then, by *Ax Non-Executability* and the definition of the canonical accessibility relation, there is no $s'$ in the model such that $sR_{\alpha(i)}s'$. As to b), if a state $s$ in the canonical model satisfies one of the preconditions $\mathtt{prec_j}$ of $\alpha(i)$ then$\mathit{tr}(\mathtt{prec_j})$ belongs to $s$ and, by *Ax Frame*, $[\alpha(i)]\mathit{tr}(\mathtt{post_j})$ also do. Now, *Ax Executability* guarantees that there exists at least one $s'$ such that $sR_{\alpha(i)}s'$, and, for any $s'$ such that $sR_{\alpha(i)}s'$, by the definition of such canonical accessibility relation, $s'$ contains $\mathit{tr}(\mathtt{post_j})$ (otherwise it would not be the case that $sR_{\alpha(i)}s'$). On the other hand, for any literal $(\neg)p$ in $s$ not occurring in $tr(\texttt{post}_\texttt{j})$, its value cannot change from $s$ to $s'$ since, if it would, then for *Ax Frame* it would not be the case that $sR_{\alpha(i)}s'$, which is impossible. This concludes the proof.

### 4.5 Verification

To verify a normative multi-agent system program means, in our perspective, to check whether the program implementing the normative artefact is soundly designed w.r.t. the regimentation and sanctioning mechanisms it is supposed to realize or, to put it in more general terms, to check whether certain property holds in all (or some) states reachable by the execution traces of the multi-agent system program. In order to do this, we need to translate a multi-agent system program into a PDL program expression.

As explained in earlier sections, a multi-agent system program assumes a set of behaviors $A_1, \ldots, A_n$ of agents $1, \ldots, n$, each of which is a sequence of external actions (the agents actions observed from the multi-agent level), i.e., $A_i = \alpha_i^1; \alpha_i^2, \ldots$ where $\alpha_i^j \in Ac$. [1] Moreover, a multi-agent system program with norms consists of an initial set of brute facts, a set of counts-as rules and a set of sanction rules which together determine the initial state of the program. In this paper, we consider the execution of a multi-agent program as interleaved executions of the involved agents' behaviors started at the initial state.

Given $I$ as the set of agents' names and $A_i$ as the behavior of agent $i \in I$, the execution of a multi-agent program can be described as PDL expression $\bigcup$ *interleaved*$(\{A_i|i \ \in I\})$, where *interleaved*$(\{A_i|i \ \in I\})$ yields all possible interleavings of agents' behaviors, i.e., all possible interleavings of actions from sequences $A_i$. It is important to notice that $\bigcup$ *interleaved*$(\{A_i|i \ \in I\})$ corresponds to the set of computations sequences (execution traces) generated by the operational semantics.

The general verification problem can now be formulated as follows. Given a multi-agent system program with norms in a given initial state satisfying $\phi \in L$, the state reached after the execution of the program satisfies $\psi$, i.e.:

$$\phi \rightarrow \langle [ \bigcup \mathit{interleaved}(\{A_i|i \ \in I\})] \rangle \psi$$

In the above formulation, the modality $\langle[\ldots]\rangle$ is used to present both safety $[\ldots]$ and liveness $\langle\ldots\rangle$ properties. We briefly sketch a sample of such properties using again the multi-agent system program with norms which implements the train station example with one passenger agent (see Section 3).

**Sanction follows violation.** Entering without a ticket results in a fine, i.e.,

$$\neg\texttt{at\_platform} \wedge \neg\texttt{train} \wedge \neg\texttt{ticket} \rightarrow [\texttt{enter}](\texttt{viol}_1 \wedge \texttt{pay}_{10}).$$

[1] Note an agent's behavior can always be written as a (set of) sequence(s) of actions, which in turn can be written as a PDL expressions.

**Norm obedience avoids sanction.** Buying a ticket if you have none and entering the platform does not result in a fine, i.e.:

$$\neg\texttt{at\_platform} \wedge \neg\texttt{train} \rightarrow \langle \text{ If} \neg\texttt{ticket} \text{ then } \texttt{buy\_ticket}; \texttt{enter} \rangle (\texttt{at\_platform} \wedge \neg\texttt{pay}_{10}).$$

**Regimentation.** It is not possible for an agent to enter the platform and embark the train without a ticket, i.e.:

$$\neg\texttt{at\_platform} \wedge \neg\texttt{train} \wedge \neg\texttt{ticket} \rightarrow [\texttt{enter}; \texttt{embark}]\bot$$

Note that there is only one passenger agent involved in the example program. For this property, we assume that the passenger's behavior is $\texttt{enter}; \texttt{embark}$. Note also that:

$$\bigcup \mathit{interleaved}(\{\texttt{enter}; \texttt{embark}\}) = \texttt{enter}; \texttt{embark}.$$

Below is the proof of the regimentation property above with respect to the multi-agent system program with norms that implements the train station with one passenger.

*Proof.* First, axiom *Ax Frame* using the specification of the *enter* action (with pre-condition $\{\texttt{-at\_platform}\}$ and post-condition $\{\texttt{at\_platform}\}$) gives us
(1) $-\texttt{at\_platform} \wedge -\texttt{in\_train} \wedge -\texttt{ticket} \;\rightarrow$
$[\texttt{enter}]\ \texttt{at\_platform} \wedge -\texttt{in\_train} \wedge -\texttt{ticket}$
Moreover, axiom *Ax Frame* using the specification of the *embark* action (with pre-condition $\{\texttt{at\_platform, -in\_train}\}$ and post-condition $\{\texttt{-at\_platform, in\_train}\}$) gives us
(2) $\texttt{at\_platform} \wedge -\texttt{in\_train} \wedge -\texttt{ticket} \;\rightarrow$
$[\texttt{embark}]\ -\texttt{at\_platform} \wedge \texttt{in\_train} \wedge -\texttt{ticket}$
Also, axiom *Ax Counts-as* and the specification of the second counts-as rule of the program give us
(3) $\texttt{in\_train} \wedge -\texttt{ticket} \rightarrow \texttt{viol}_{\bot}$
And axiom *Ax Regiment* together with formula (3) gives us
(4) $\texttt{in\_train} \wedge -\texttt{ticket} \rightarrow \bot$
Now, using PDL axioms together with formula (1), (2), and (4) we get first
(5) $-\texttt{at\_platform} \wedge -\texttt{in\_train} \wedge -\texttt{ticket} \;\rightarrow [\texttt{enter}][\texttt{embark}]\ \bot$
and thus
(6) $-\texttt{at\_platform} \wedge -\texttt{in\_train} \wedge -\texttt{ticket} \;\rightarrow [\texttt{enter}; \texttt{embark}]\ \bot$. This completes the derivation.

## 5 Conclusions and Future Work

The paper has proposed a programming language for implementing multi-agent systems with norms. The programming language has been endowed with formal operational semantics, therefore formally grounding the use of certain social notions —eminently the notion of norm, regimentation and enforcement— as explicit programming constructs. A sound and complete logic has then been proposed which can be used for verifying properties of the multi-agent systems with norms implemented in the proposed programming language.

We have already implemented an interpreter for the programming language that facilitates the implementation of multi-agent systems without norms (see `http://www.cs.uu.nl/2apl/`). Currently, we are working to build an interpreter for the modified programming language. This interpreter can be used to execute programs that implement multi-agent systems with norms. Also, we are working on using the presented logic to devise a semi-automatic proof checker for verification properties of normative multi-agent programs.

We are aware that for a comprehensive treatment of normative multi-agent systems we need to extend our framework in many different ways. Future work aims at extending the programming language with constructs to support the implementation of a broader set of social concepts and structures (e.g., roles, power structure, task delegation, and information flow), and more complex forms of enforcement (e.g., policing agents) and norm types (e.g., norms with deadlines). Another extension of the work is the incorporation of the norm-awareness of agents in the design of the multi-agent system. We also aim at extending the framework to capture the role of norms and sanctions concerning the interaction between individual agents.

The approach in its present form concerns only closed multi-agent systems. Future work will also aim at relaxing this assumption providing similar formal semantics for open multi-agent systems. Finally, we have focused on the so-called 'ought-to-be' norms which pertain to socially preferable states. We intend to extend our programming framework with 'ought-to-do' norms pertaining to socially preferable actions.

## References

1. N. Alechina, M. Dastani, B. Logan, and J.-J.Ch Meyer. A logic of agent programs. In *Proc. AAAI 2007*, 2007.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, Cambridge, 2001.
3. C. Castelfranchi. Formalizing the informal?: Dynamic social order, bottom-up social control, and spontaneous normative relations. *JAL*, 1(1-2):47–92, 2004.
4. M. Dastani. 2apl: a practical agent programming language. *International Journal of Autonomous Agents and Multi-Agent Systems*, 16(3):214–248, 2008.
5. M. Dastani and J.-J. Meyer. A practical agent programming language. In *In Proc. of ProMAS'07*, 2008.
6. V. Dignum. *A Model for Organizational Interaction.* PhD thesis, Utrecht University, SIKS, 2003.
7. M. Esteva, J.A. Rodríguez-Aguilar, B. Rosell, and J.L. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Proc. of AAMAS 2004*, New York, US, July 2004.
8. D. Grossi. *Designing Invisible Handcuffs.* PhD thesis, Utrecht University, SIKS, 2007.

9. J. F. Hübner, J. S. Sichman, and O. Boissier. Moise+: Towards a structural functional and deontic model for mas organization. In *Proc. of AAMAS 2002*. ACM, July 2002.
10. A. J. I. Jones and M. Sergot. On the characterization of law and computer systems. In *Deontic Logic in Computer Science*. 1993.
11. Rosine Kitio, Olivier Boissier, Jomi Fred Hbner, and Alessandro Ricci. Organisational artifacts and agents for open multi-agent organisations: giving the power back to the agents. In *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870, pages 171–186. Springer, 2007.
12. A. Ricci, M. Viroli, and A. Omicini. "Give agents their artifacts": The A&A approach for engineering working environments in MAS. In *In proc. of AAMAS 2007*, Honolulu, Hawai'i, USA, 2007.
13. J. Searle. *The Construction of Social Reality*. Free, 1995.
14. F. Zambonelli, N. Jennings, and M. Wooldridge. Developing multiagent systems: the GAIA methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):317–370, 2003.

# Modal Logics for Preferences and Cooperation: Expressivity and Complexity

Cédric Dégremont and Lena Kurzen*

Universiteit van Amsterdam

**Abstract.** This paper studies expressivity and complexity of normal modal logics for reasoning about cooperation and preferences. We identify a class of local and global notions relevant for reasoning about cooperation of agents that have preferences. Many of these notions correspond to game- and social choice-theoretic concepts. We specify the expressive power required to express these notions by determining whether they are invariant under certain relevant operations on different classes of Kripke models and frames. A large class of known extended modal languages is specified and we show how the chosen notions can be expressed in fragments of this class. To determine how demanding reasoning about cooperation is in terms of computational complexity, we use known complexity results for extended modal logics and obtain for each local notion an upper bound on the complexity of modal logics expressing it.

## 1 Introduction

Cooperation of agents is a major issue in fields such as computer science, economics and philosophy. The conditions under which coalitions are formed occur in various situations involving multiple agents. A single airline company cannot afford the cost of an airport runway whereas a group of companies can. Generally, agents can form groups in order to share complementary resources or because as a group they can achieve better results than individually. Modal logic (ML) frameworks for reasoning about cooperation mostly focus on what coalitions can achieve. Coalition Logic (**CL**) [1] uses modalities of the form $[\mathtt{C}]\phi$ saying that "coalition $\mathtt{C}$ has a joint strategy to ensure that $\phi$". **CL** has neighborhood semantics but it has been shown how it can be simulated on Kripke models [2].

Another crucial concept for reasoning about interactive situations is that of *preferences*. It also received attention from modal logicians ([3] surveys). Recent works (e.g. [4, 5]) propose different mixtures of cooperation and preference logics for reasoning about cooperation. In such logics many concepts from *game theory* (GT) and *social choice theory* (SCT) are commonly encountered. Depending on the situations to be modelled, different bundles of notions are important. Ability

* Authors are supported by a GLoRiClass fellowship of the EU Commission (Research Training Fellowship MEST-CT-2005-020841) and by the Dutch Organization for Scientic Research (NWO), TACTICS project, grant number 612.000.525, respectively.

to express these notions – together with good computational behavior – make a logic appropriate for reasoning about the situations under consideration.

Rather than proposing a new logical framework, with specific expressivity and complexity, we identify how SCT and GT notions are demanding for MLs in terms of expressivity and complexity. We identify notions relevant for describing interactive situations and give satisfiability and validity invariance results as well as definability results for them, identifying the natural (extended) modal languages needed depending on the class of frames actually considered and the particular bundle of notions of interest. We draw some consequences about the complexity of reasoning about cooperation using ML. Our results apply to logics interpreted on Kripke structures using a (preference) relation for each agent and a relation for each coalition. The latter can be interpreted in various ways. The pair $(x, y)$ being in the relation for coalition $\mathtt{C}$ can e.g. mean:

- Coalition $\mathtt{C}$ considers $y$ as being at least as good as $x$.
- If the system is in state $x$, $\mathtt{C}$ would choose $y$ as the next state.
- $\mathtt{C}$ can submit a request such that if it is the first one received by the server while the state is in $x$, then the state of the system will change from $x$ to $y$.
- When the system is in state $x$, $\mathtt{C}$ considers it possible that it is in state $y$.

Interpreting the relation as the possibility to bring the system into a different state applies to scenarios where agents act sequentially (e.g. with a server treating requests in a "first-come, first-served" manner) rather than simultaneously (as in ATL or **CL**). In special cases - e.g. for turn-based [6, 1] frames - the approaches coincide. Still, the two approaches are first of all complementary. Our focus in this paper is on concepts bridging powers and preferences. The same analysis is possible for powers themselves in ATL-style. Both analyses can then be combined in an interesting way. Finally, an important alternative interpretation of the coalition relation is that of group preferences, in which case ATL models can simply be merged with the models we consider. We return to this in Sect. 2.

**Structure of this Paper.** Sect. 2 presents three classes of models of cooperative situations. Sect. 3 introduces local and global notions motivated by ideas from GT and SCT indicating local properties of a system and global properties that characterize classes of frames. Sect. 4 presents a large class of extended modal languages and background invariance results. In Sect. 5, we study the expressivity needed to express the local notions (to define the global properties) by giving invariance results for relevant operations and relations between models (frames). Sect. 6 completes this work by defining the notions in fragments of (extended) modal languages. We give complexity results for model checking and satisfiability for these languages and thereby give upper bounds for the complexity of logics that can express the introduced notions. Sect. 7 concludes.

## 2 The Models

Our aim is to study how demanding certain GT and SCT concepts are in terms of expressivity and complexity. This depends on the models chosen. We consider three classes of simple models that have many suitable interpretations. This gives our results additional significance. A *frame* refers to the relational part of a model. For simplicity, we introduce models and assume that the domain of the valuation are countable sets of propositional letters $\mathtt{PROP}$ and nominals $\mathtt{NOM}$. We focus on model theory and postpone discussion of formal languages to Sect. 4.

**Definition 1** **($\mathtt{N}$-$\mathtt{LTS}$).** *A* $\mathtt{N}$-$\mathtt{LTS}$ *(Labeled Transition Systems indexed by a finite set of agents* $\mathtt{N}$*) is of the form* $\langle W, \mathtt{N}, \{ \overset{\mathtt{C}}{\rightarrow} \mid \mathtt{C} \subseteq \mathtt{N}\}, \{ \leq_i \mid i \in \mathtt{N}\}, V\rangle$*, where* $W \neq \emptyset$*,* $\mathtt{N} = \{1, \ldots, n\}$ *for some* $n \in \mathbb{N}$*,* $\overset{\mathtt{C}}{\rightarrow} \subseteq W \times W$ *for each* $\mathtt{C} \subseteq \mathtt{N}$*,* $\leq_j \subseteq W \times W$ *for each* $j \in \mathtt{N}$*, and* $V : \mathtt{PROP} \cup \mathtt{NOM} \rightarrow \wp W$*,* $|V(i)| = 1$ *for each* $i \in \mathtt{NOM}$*.*

$W$ is the set of states, $\mathtt{N}$ a set of agents and $w \overset{\mathtt{C}}{\rightarrow} v$ says that coalition $\mathtt{C}$ can change the state of the system from $w$ into $v$. As mentioned, other interpretations are possible. $w \leq_i v$ means that $i$ finds the state $v$ at least as good as $w$. $w \in V(p)$ means that $p$ is true at $w$. Preferences are usually assumed to be total pre-orders ($\mathtt{TPO}$). Let $\mathtt{TPO}$-$\mathtt{N}$-$\mathtt{LTS}$ denote the class of $\mathtt{N}-\mathtt{LTS}$s in which for each $i \in \mathtt{N}$, $\leq_i$ is a $\mathtt{TPO}$. We also consider models with strict preferences as explicit primitives.

**Definition 2** **($\mathtt{S/TPO}-\mathtt{N}-\mathtt{LTS}$).** *Define* $\mathtt{S/TPO}-\mathtt{N}-\mathtt{LTS}$ *as models of the form* $\langle W, \mathtt{N}, \{ \overset{\mathtt{C}}{\rightarrow} \mid \mathtt{C} \subseteq \mathtt{N}\}, \{ \leq_i \mid i \in \mathtt{N}\}, \{ <_i \mid i \in \mathtt{N}\}, V\rangle$*,which extend* $\mathtt{TPO}-\mathtt{N}-\mathtt{LTS}$ *models by an additional relation* $<_i \subseteq W \times W$ *for each* $i \in \mathtt{N}$ *with the constraint that for each* $i \in \mathtt{N}$*,* $w <_i v$ *iff* $w \leq_i v$ and $v \not\leq_i w$*.*

Depending on the interpretation of $\overset{\mathtt{C}}{\rightarrow}$, it can be complemented or replaced by effectivity functions (**CL**) or more generally transition functions as in ATL. In the latter sense, powers of coalitions will in general not reduce to relations on states. We leave an analysis of powers in such settings aside for now. There would be two ways to go: drawing on the model-theory of neighborhood semantics [7] or on a normal simulation of **CL** [2]. Generally, the expressive power might depend on whether coalitional powers are taken as primitives or computed from individual powers. A last comment: in our analysis of the expressivity required by certain local notions, we check how the choice of models affects invariance results. We now turn to the notions playing the central role in the paper.

## 3 The Notions

Reasoning about cooperative interaction considers what coalitions of agents can achieve and what individuals prefer. Using these elements, more elaborated notions can be built. We consider natural counterparts of SCT and GT notions and are interested in local notions i.e. properties of a particular state in a particular system, i.e. properties of pointed models $\mathcal{M}, w$. But also in global notions, which are properties of classes of systems: we are interested in the class of frames a property characterizes. W.r.t to content, apart from notions describing only coalitional powers or preferences, we consider stability and effectivity concepts.

**Power of Coalitions.** We now present some interesting notions about coalitional power. Recall that $w \overset{\mathtt{C}}{\rightarrow} v$ can e.g. mean "$\mathtt{C}$ can achieve $v$ at $w$".

**Local Notions.** Interesting properties of coalitional power involve the relation between the powers of different groups ($PowL3$) and the contribution of individuals to a group's power, e.g. an agent is needed to achieve something ($PowL2$).

- *PowL*1. Coalition C can achieve a state where $p$ is true. $\exists x(w \xrightarrow{C} x \wedge P(x))$
- *PowL*2. Only groups with $i$ can achieve $p$-states. $\bigwedge_{\mathtt{C}\subseteq\mathtt{N}\setminus i}(\forall x(w \xrightarrow{\mathtt{C}} x \to \neg P(x)))$
- *PowL*3. Coalition C can force every state that coalition D can force. $\forall x(w \xrightarrow{\mathtt{D}} x \to w \xrightarrow{\mathtt{C}} x)$

**Global Notions.** *PowG*1 says that each coalition can achieve exactly one result, while *PowG*3 expresses coalition monotonicity, it says that if a coalition can achieve some result, then so can every superset of that coalition. In many situation, decision making in groups can only be achieved by a majority (*PowG*2). *PowG*4 (*G*5) exemplify consistency requirements between powers of non-overlapping coalitons, we find mathematically natural.

- *PowG*1. In any state each coalition can achieve exactly one state. $\bigwedge_{C\subseteq\mathtt{N}} \forall x \exists y(x \xrightarrow{C} y \wedge \forall z(x \xrightarrow{C} z \to z = y))$
- *PowG*2. Only coalitions containing a majority of N can achieve something. $\forall x(\bigwedge_{\mathtt{C}\subseteq\mathtt{N},|\mathtt{C}|<\frac{|N|}{2}}(\neg\exists y(w \xrightarrow{\mathtt{C}} y)))$
- *PowG*3. Coalition monotonicity, i.e. if for C and D, $\mathtt{C} \subseteq \mathtt{D}$, then $R_{\mathtt{C}} \subseteq R_{\mathtt{D}}$. $\forall x(\bigwedge_{\mathtt{C}\subseteq\mathtt{N}} \bigwedge_{\mathtt{D}\subseteq\mathtt{N},\mathtt{C}\subseteq\mathtt{D}}(\forall y(x \xrightarrow{\mathtt{C}} y \to x \xrightarrow{\mathtt{D}} y)))$
- *PowG*4. If C can achieve something, then subsets of its complement cannot. $\forall x \bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(((\exists y(x \xrightarrow{\mathtt{C}} y)) \to \bigwedge_{\mathtt{D}\subseteq\mathtt{N}\setminus\mathtt{C}} \neg\exists z(x \xrightarrow{\mathtt{D}} z)))$
- *PowG*5. —, then subsets of its complement cannot achieve something C cannot. $\forall x \bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(((\exists y(x \xrightarrow{\mathtt{C}} y)) \to \bigwedge_{\mathtt{D}\subseteq\mathtt{N}\setminus\mathtt{C}} \forall z(x \xrightarrow{\mathtt{D}} z \to x \xrightarrow{\mathtt{C}} z)))$

**Preferences.** What do agents prefer? What are suitable global constraints on preferences? $w \leq_i v$ means "$i$ finds $v$ *at least as good* (a.l.a.g.) as $w$". We write $w <_i v$ for $w \leq_i v \wedge \neg(v \leq_i w)$, meaning that "*$i$ strictly prefers* $v$ over $w$".

**Local Notions.** First of all, we can distinguish between strict and nonstrict preferences. The most basic preference relation that we consider is that of being a.l.a.g. We can also look at the relation "at least as bad" (a.l.a.b) (*PrefL*4). Agents' preferences over states can also be seen as being based on preferences over propositions [8]. *PrefL*8 (*PrefL*10) says the truth of a given proposition is a sufficient (necessary) condition for an agent to prefer some state.

- *PrefL*1. There is a state $i$ finds a.l.a.g. where $p$ holds. $\exists x(w \leq_i x \wedge P(x))$
- *PrefL*2. There is a $p$-state that $i$ strictly prefers. $\exists x(w <_i x \wedge P(x))$
- *PrefL*3.There is a state that all agents find a.l.a.g and that at least one strictly prefers. $\exists x(\bigwedge_{i\in\mathtt{N}}(w \leq_i x) \wedge \bigvee_{j\in\mathtt{N}} w <_j x)$
- *PrefL*4. There is a state that $i$ finds a.l.a.b. where p holds. $\exists x(x \leq_i w \wedge P(x))$
- *PrefL*5. There is a state that $i$ finds strictly worse where $p$ is true. $\exists x(x <_i w \wedge P(x))$
- *PrefL*6. $i$ finds a state a.l.a.g. iff $j$ does. $\forall x(w \leq_i x \leftrightarrow w \leq_j x)$
- *PrefL*7. There is a state only $i$ finds a.l.a.g. $\exists x(w \leq_i x \wedge \bigwedge_{j\in\mathtt{N}\setminus\{i\}} \neg(w \leq_j x))$
- *PrefL*8. $i$ finds every $p$-state a.l.a.g. $\forall x(P(x) \to w \leq_i x)$
- *PrefL*9. $i$ strictly prefers every $p$-state. $\forall x(P(x) \to w <_i x)$
- *PrefL*10. $i$ considers only $p$-states to be a.l.a.g. $\forall x(w \leq_i x \to P(x))$
- *PrefL*11. $i$ strictly prefers only $p$-states. $\forall x(w <_i x \to P(x))$

**Global Notions.** Capturing the intuitive idea of preferences requires several conditions for the preference relation: reflexivity, transitivity and completeness (trichotomy for strict preferences). Sometimes, it can also be appropriate to say that for each alternative there is exactly one that is at least as good ($PrefG8$).

- $PrefG1$. "at least as good as" is reflexive. $\forall x(\bigwedge_{i\in\mathtt{N}}(x \leq_i x))$
- $PrefG2$. — transitive. $\forall x\forall y\forall z(\bigwedge_{i\in\mathtt{N}}((x \leq_i y \wedge y \leq_i z) \rightarrow x \leq_i z))$
- $PrefG3$. — complete. $\forall x\forall y(\bigwedge_{i\in\mathtt{N}}(x \leq_i y \vee y \leq_i x))$
- $PrefG4$. — a total pre-order. (Conjunction of the two previous formulas.)
- $PrefG5$. "strictly better than" is transitive. $\forall x\forall y\forall z((\bigwedge_{i\in\mathtt{N}}(x <_i y \wedge y <_i z) \rightarrow x <_i z)))$
- $PrefG6$. "strictly better than" is trichotomous. $\forall x\forall y(\bigwedge_{i\in\mathtt{N}}(x <_i y \vee y <_i x \vee x = y))$
- $PrefG7$. — a strict total order.Conjunction of the previous two formulas.
- $PrefG8$. Determinacy for "at least as good as", i.e. exactly one successor. $\forall x(\bigwedge_{i\in\mathtt{N}}(\exists y(w \leq_i y \wedge \forall z(x \leq_i z \rightarrow z = y))))$

So far, we focussed on preferences of individuals. A natural question in SCT is how to aggregate individual preferences into group preferences. We can address this question by interpreting $\overset{\mathtt{C}}{\rightarrow}$ as a preference relation for each $\mathtt{C} \subseteq \mathtt{N}$.

- $PrefG9$. $\mathtt{C}$ finds a state a.l.a.g. as the current one iff all its members do. $\forall x\forall y(\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(x \overset{\mathtt{C}}{\rightarrow} y \leftrightarrow \bigwedge_{i\in\mathtt{C}} x \leq_i y))$
- $PrefG10$. — iff at least one member does. $\forall x\forall y(\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(x \overset{\mathtt{C}}{\rightarrow} y \leftrightarrow \bigvee_{i\in\mathtt{C}} x \leq_i y))$
- $PrefG11$. — iff most members do.$\forall x\forall y(\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(x \overset{\mathtt{C}}{\rightarrow} y \leftrightarrow \bigvee_{\mathtt{D}\subseteq\mathtt{C},|\mathtt{D}|>\frac{|\mathtt{C}|}{2}}(\bigwedge_{i\in\mathtt{D}} x \leq_i y)))$

**Combining preceding concepts.** We start with the conceptually and historically important SCT notion of a *dictator*. $d$ is a dictator if the group's preferences mimic $d$'s preferences. Interpreting $\overset{\mathtt{C}}{\rightarrow}$ as achievement relation, we get an even stronger notion: groups can only *do* what $d$ likes. A *local* dictator is a dictator who controls one state in the system and a *dictator* controls all states.

**Definition 3 (Local Dictatorship).** *i is a* weak (strong) local dictator *at w iff any group prefers v at w only if for i, v is a.l.a.g. as (strictly better than) w.*

We now introduce combinations of powers and preferences. The first notion says that coalition $\mathtt{C}$ can do something useful for $i$ (in some cases giving $i$ an incentive to join) and the third notion characterizes situations in which a unanimously desired state remains unachievable. We start with **Local Notions**.

- $PPL1$. $\mathtt{C}$ can achieve a state that $i$ finds at least as good as the current one. $\exists x(w \overset{\mathtt{C}}{\rightarrow} x \wedge w \leq_i x)$
- $PPL2$. $\mathtt{C}$ can achieve a state that all $i \in \mathtt{D}$ find a.l.a.g. as the current one. $\exists x(w \overset{\mathtt{C}}{\rightarrow} x \wedge \bigwedge_{i\in\mathtt{D}} w \leq_i x)$
- $PPL3$. There is a state that all agents prefers but no coalition can achieve it. $\exists x((\bigwedge_{i\in\mathtt{N}} w \leq_i x) \wedge \bigwedge_{\mathtt{C}\subseteq\mathtt{N}} \neg(w \overset{\mathtt{C}}{\rightarrow} x))$
- $PPL4$. $\mathtt{C}$ can achieve all states that agent $i$ finds a.l.a.g. as the current one. $\forall x(w \leq_i x \rightarrow w \overset{\mathtt{C}}{\rightarrow} x)$

- *PPL*5. $\mathtt{C}$ can achieve all states that $i$ strictly prefers. $\forall x(w <_i x \to w \xrightarrow{\mathtt{C}} x)$
- *PPL*6. $i$ is a weak local dictator. $\forall x(w \xrightarrow{\mathtt{C}} x \to w \leq_i x)$
- *PPL*7. $i$ is a strong local dictator. $\forall x(w \xrightarrow{\mathtt{C}} x \to w <_i x)$

**Global Notions**. *PPG*1 is a natural constraint on coalitional power: a group can achieve a state iff it is good for all members - otherwise they would not take part in the collective action. *PPG*3 is a condition of Arrow's impossibility theorem. *PPG*4 reflects individual rationality: don't join a group if you don't gain anything. It can be generalized to every sub-coalition or weakened to "not joining if you lose something" (cf. core of a coalitional game [9] (Def. 268.3)). *PPG*5 applies to systems where an agent is indispensable to achieve anything: a unique capitalist in a production economy or a unique server are typical examples.

- *PPG*1.Coalitions can only achieve states that all its members consider at least as good as the current one. $\forall x \forall y \bigwedge_{\mathtt{C} \subseteq \mathtt{N}}(x \xrightarrow{\mathtt{C}} y \to \bigwedge_{i \in \mathtt{C}}(x \leq_i y))$
- *PPG*2. One agent is a weak local dictator in every state (*dictator*). $\bigvee_{i \in \mathtt{N}} \forall x \forall y(x \xrightarrow{\mathtt{C}} y \to x \leq_i y)$
- *PPG*3. There is no *dictator*. $\neg(\bigvee_{i \in \mathtt{N}} \forall x \forall y(x \xrightarrow{\mathtt{C}} y \to x \leq_i y))$
- *PPG*4. If $i$ can achieve some state $i$ strictly prefers then for any $\mathtt{C}$ containing $i$: if $\mathtt{C} \setminus i$ cannot achieve some state but $\mathtt{C}$ can, then $i$ strictly prefers that state. $\bigwedge_{i \in \mathtt{N}} \forall x(\exists y(x \xrightarrow{\{i\}} y \wedge x <_i y) \to \bigwedge_{\mathtt{C} \subseteq \mathtt{N}, i \in \mathtt{C}}(\forall z(x \xrightarrow{\mathtt{C}} z \wedge \neg(x \xrightarrow{\mathtt{C} \setminus \{i\}} z)) \to x <_i z))$
- *PPG*5. Only groups with $i$ can achieve something. $\forall x \bigwedge_{\mathtt{C} \subseteq \mathtt{N} \setminus \{i\}} \neg \exists y(x \xrightarrow{\mathtt{C}} y)$
- *PPG*6. In each state, there is some $i$ such coalitions with $i$ can achieve exactly the same states as they can without $i$. $\forall x(\bigvee_{i \in \mathtt{N}} \bigwedge_{\mathtt{C} \subseteq \mathtt{N}, i \in \mathtt{C}} \forall y(x \xrightarrow{\mathtt{C}} y \leftrightarrow x \xrightarrow{\mathtt{C} \setminus \{i\}} y))$
- *PPG*7. For any agent, there is some state in which coalitions not containing this agent cannot achieve any state. $\bigwedge_{i \in \mathtt{N}} \exists x(\bigwedge_{\mathtt{C} \subseteq \mathtt{N}, i \in \mathtt{C}} \neg \exists y(x \xrightarrow{\mathtt{C}} y))$

**Efficiency and Stability Notions.** In our setting, it is natural to interpret the state space as possible social states or allocations of goods. A criterion from welfare economics to distinguish "good" from "bad" states is that of *efficiency*: if we can change the allocation or social state and make an agent happier without making anyone less happy then we are using resources more efficiently and it is socially desirable to do so. E.g. *PrefL*3 in this respect means that the current state is not efficient: there is a state that is a *Pareto-improvement* of it. Importing the notion of Pareto-efficiency into our framework is straightforward.

**Definition 4 (Pareto-efficiency).** *A state is weakly (strongly)* Pareto-*efficient iff there is no state that everyone strictly prefers (finds a.l.a.g). A state is* Pareto-*efficient iff there is no state such that everyone considers it to be at least as good and at least one agent thinks it is strictly better.*

GT equilibrium concepts characterize stable states: given what others are doing, I don't have an incentive to do something that makes us leave this stable state. Generalizing, a system is in a stable state if nobody has an incentive to change its current state. We can think of strategy profiles in a strategic game as assigning roles to the agents. Two profiles $x = (s^*_{-i}, s^*_i), y$ are related by $\xrightarrow{\{i\}}$ iff $i$ can unilaterally change role (strategy) to $s'_i$ in the next round of the game and $y = (s^*_{-i}, s'_i)$. E.g. the stability of a state where an agent provides the public good on his own depends on whether he cares enough about it to provide it on his own. A state is stable iff there is no strictly preferred state that an agent can achieve alone. Since the idea relates to *Nash* equilibria (see [9]), we use the names *Nash-stability*, and *Nash-cooperation stability* for its group version.

**Definition 5 (Nash-stability).** *A state is (strongly)* Nash-*stable iff there is no state that an agent $i$ strictly prefers (finds a.l.a.g.) and that $i$ can achieve alone. It is (strongly)* Nash-*cooperation stable iff there is no state $v$ and coalition such* $\mathtt{C}$ *that every $i \in \mathtt{C}$ strictly prefers $v$ (finds $v$ a.l.a.g.) and* $\mathtt{C}$ *can achieve $v$.*

**Local Notions**

- $EF1$.The current state is weakly *Pareto*-efficient. $\neg\exists x(\bigwedge_{i\in\mathtt{N}}(w <_i x))$
- $EF2$. The current state is *Pareto*-efficient. $\neg\exists x((\bigwedge_{i\in\mathtt{N}} w \leq_i x) \wedge \bigvee_{j\in\mathtt{N}} w <_i x)$
- $EF3$. The current state is strongly *Pareto*-efficient. $\neg\exists x(\bigwedge_{i\in\mathtt{N}} w \leq_i x)$
- $ST1$. The current state is *Nash* stable. $\neg\exists x(\bigvee_{i\in\mathtt{N}}(w \overset{\{i\}}{\to} x \wedge w <_i x))$
- $ST2$. The current state is strongly *Nash* stable. $\neg\exists x(\bigvee_{i\in\mathtt{N}}(w \overset{\{i\}}{\to} x \wedge w \leq_i x))$
- $ST3$. — *Nash*-cooperation stable. $\neg\exists x(\bigvee_{C\subseteq\mathtt{N}}(w \overset{\mathtt{C}}{\to} x \wedge \bigwedge_{i\in\mathtt{C}} w <_i x))$
- $ST4$. — strongly *Nash*-cooperation stable. $\neg\exists x(\bigvee_{C\subseteq\mathtt{N}}(w \overset{\mathtt{C}}{\to} x \wedge \bigwedge_{i\in\mathtt{C}} w \leq_i x))$

## 4 Modal Languages and their Expressivity

As will be clear from invariance results of next sections, Basic Modal Language will generally be too weak for reasoning about cooperation. However, any notion expressible in the FO correspondence language is expressible in the hybrid language $\mathcal{H}(\mathtt{E}, @, \downarrow)$ [10]. Amongst temporal logics, boolean modal logics and the various hybrid logics, there are well-understood fragments. We introduce all these **Extended Modal Languages** at once as a "super" logic.

**Syntax.** The syntax of this "super" logic is recursively defined as follows:

$\alpha ::= \ \leq_j \ \mid \mathtt{C} \mid v \mid \alpha^{-1} \mid ?\phi \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha \cap \alpha \mid \overline{\alpha}$

$\phi ::= \ p \mid i \mid x \mid \neg\phi \mid \phi \wedge \phi \mid \langle\alpha\rangle\phi \mid \mathtt{E}\phi \mid @_i\phi \mid @_x\phi \mid \ \downarrow x.\phi \mid \ \llbracket \ \alpha \ \rrbracket \ \phi \mid$

where $j \in \mathtt{N}$, $\mathtt{C} \in \wp(\mathtt{N}) - \{\emptyset\}$, $p$ ranges over $\mathtt{PROP}$, $i$ ranges over $\mathtt{NOM}$ and $x \in \mathtt{SVAR}$, for $\mathtt{SVAR}$ being a countable set of variables.

**Semantics.** Valuation maps propositional letters to subsets of the domain and nominals to singleton subsets. Given a $\mathtt{N}-\mathtt{LTS}$, a program $\alpha$ is interpreted as a relation as indicated on the left. Formulas are interpreted together with an assignment $g : \mathtt{SVAR} \to W$ as indicated (mostly) on the right. We skip booleans.

| | | | |
|---|---|---|---|
| $\mathcal{M}, w, g \Vdash i$ | iff $w \in V(i)$ | $\mathcal{M}, w, g \Vdash x$ | iff $w = g(x)$ |
| $R_{\leq_\mathtt{i}}$ | $= \leq_i$ | $\mathcal{M}, w, g \Vdash \langle\alpha\rangle\phi$ | iff $\exists v : wR_\alpha v$ and $\mathcal{M}, v, g \Vdash \phi$ |
| $R_C$ | $= \overset{C}{\to}$ | $\mathcal{M}, w, g, \Vdash \mathtt{E}\phi$ | iff $\exists v \ \in W \ \mathcal{M}, v, g \Vdash \phi$ |
| $R_{\beta^{-1}}$ | $= \{(v, w) \mid wR_\beta v\}$ | $\mathcal{M}, w, g, \Vdash @_i\phi$ | iff $\mathcal{M}, v, g \Vdash \phi$ where $V(i) = \{v\}$ |
| $R_{\beta\cup\gamma}$ | $= R_\beta \cup R_\gamma$ | $\mathcal{M}, w, g, \Vdash @_x\phi$ | iff $\mathcal{M}, g(x), g \Vdash \phi$ |
| $R_{\beta\cap\gamma}$ | $= R_\beta \cap R_\gamma$ | $\mathcal{M}, w, g, \Vdash \downarrow x.\phi$ | iff $\mathcal{M}, w, g[x := w] \Vdash \phi$ |
| $R_{\overline{\beta}}$ | $= (W \times W) - R_\beta$ | $\mathcal{M}, w, g \Vdash \llbracket \ \alpha \ \rrbracket \ \phi$ | iff $wR_\alpha v$ whenever $\mathcal{M}, v, g \Vdash \phi$ |

**Expressivity.** The least expressive modal language we consider is $\mathcal{L}(\mathtt{N})$, which is of similarity type $\langle(\mathtt{C})_{\mathtt{C}\subseteq\mathtt{N}}, (\leq_i)_{i\in\mathtt{N}}\rangle$. Its natural extensions go along two lines: adding program constructs and new operators. $\mathcal{L}(\mathtt{N},\cap,i)$ e.g. refers to the logic with language: $\alpha ::= \ \leq_{\mathtt{j}} \mid \mathtt{C} \mid \alpha\cap\alpha \quad \phi ::= \ p \mid i \mid \neg\phi \mid \phi\wedge\phi \mid \langle\alpha\rangle\phi$. As language inclusion implies expressivity inclusion (indicated by "$\leq$"), we only indicate (some) non-obvious facts of inclusions in this space of modal languages.

**1.** $\mathcal{L}(\mathtt{N},\cup,\ ;\ ,?)\leq\mathcal{L}(\mathtt{N})$. **2.** $\mathcal{L}(\mathtt{N},@,i)\leq\mathcal{L}(\mathtt{N},\mathtt{E},i)$. **3.** $\mathcal{L}(\mathtt{N},\cap)\leq\mathcal{L}(\mathtt{N},\downarrow,@,x)$.
**4.** $\mathcal{L}(\mathtt{N},\square\ \square)\leq\mathcal{L}(\mathtt{N},\text{-})$. **5.** $\mathcal{L}(\mathtt{N},\text{-})\leq\mathcal{L}(\mathtt{N},\downarrow,\mathtt{E},x)$. **6.** $\mathcal{L}(\mathtt{N},\ ^{-1})\leq\mathcal{L}(\mathtt{N},\downarrow,\mathtt{E},x)$.
**7.** $\mathcal{L}(\mathtt{N},\mathtt{E})\leq\mathcal{L}(\mathtt{N},\text{-})$.

Expressivity of MLs is usually characterized by invariance results. Definitions and background results follow. We first introduce some relations between models. Let $\tau$ be a finite modal similarity type with only binary relations. Let $\mathcal{M} = \langle W,(R_k)_{k\in\tau},V\rangle$ and $\mathcal{M}' = \langle W',(R'_k)_{k\in\tau},V'\rangle$ be models of similarity type $\tau$.

**Definition 6 (Bisimulations).** *A bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ is a non-empty binary relation $Z\subseteq W\times W'$ fulfilling the following conditions:*
`AtomicHarmony` *For every $p\in\mathtt{PROP}$, $wZw'$ implies $w\in V(p)$ iff $w'\in V'(p)$.*
`Forth` $\forall k\in\tau$ *, if $wZw'$ & $R_kwv$ then $\exists v'\in W'$ s.t. $R'_kw'v'$ & $vZv'$.*
`Back` $\forall k\in\tau$ *, if $wZw'$ & $R'_kw'v'$ then $\exists v\in W$ s.t. $R_kwv$ & $vZv'$.*

In a nutshell $\cap$-Bisimulations (resp. CBisimulations) require that `Back` and `Forth` also hold for the intersection (resp. the converse) of the relations. $\mathcal{H}$-Bisimulations extend `AtomicHarmony` to nominals. TBisimulations ($\mathcal{H}(@)$-bisimulations) are total [1] bisimulations (resp. total $\mathcal{H}$-Bisimulations). $\mathcal{H}(\mathtt{E})$-Bisimulations are $\mathcal{H}$-Bisimulations matching states "with the same name". See [10] for details. We now define bounded morphisms, generated subframes and disjoint unions.

**Definition 7 (BM).** *$f: W\to W'$ is a bounded morphism from $\mathcal{M}$ to $\mathcal{M}'$ iff:*
`AtomicHarmony` *For every $p\in\mathtt{PROP}$, $w\in V(p)$ iff $f(w)\in V'(p)$.*
$\mathtt{R-homomorphism}$ $\forall k\in\tau$ *, if $R_kwv$ then $R'f(w)f(v)$.*
`Back` $\forall k\in\tau$ *, if $R'_kf(w)v'$ then $\exists v\in W$ s.t. $f(v)=v'$ and $R_kwv$.*

**Definition 8 (Generated Submodel).** *We say that that $\mathcal{M}'$ is a generated submodel (GSM) of $\mathcal{M}$ iff $W'\subseteq W$, $\forall k\in\tau$ , $R'_k = R_k\cap(W'\times W')$, $\forall p\in\mathtt{PROP}$ , $V'(p) = V(p)\cap(W'\times W')$ and if $w\in W'$ and $Rwv$ then $v\in W'$.*

**Definition 9 (Disjoint Unions).** *Let $(\mathcal{M}_j)_{j\in J}$ be a collection of models with disjoint domains. Define their disjoint union $\biguplus_j\mathcal{M}_j = \langle W,R,V\rangle$ as the union of their domains and relations, and define for each $p\in\mathtt{PROP}$, $V(p) := \bigcup_j V_j(p)$.*

**Definition 10 (Invariance).** *A property of pointed models $\Phi(X,y)$ is* invariant under $\lambda$-Bisimulations *iff whenever there exists a $\lambda$-bisimulation $Z$ between $\mathcal{M}$ and $\mathcal{M}'$ such that $(w,w')\in Z$, then $\Phi(\mathcal{M},w')$ holds iff $\Phi(\mathcal{M}',w')$ holds. Invariance for other operations is defined similarly.*

[1] $Z\subseteq W\times W'$ is *total* iff $\forall w\in W\ \exists w'\in W'\ .wZw'$ & $\forall w'\in W'\ \exists w\in W\ .wZw'$.

We now consider closure conditions. First, we consider bounded morphic images (`BMI`) of frames. `BM` on frames are obtained by dropping `AtomicHarmony` in Def. 7. A class of frames is closed under `BMI` iff it is closed under *surjective* `BM`. Next, we consider closure under generated subframes (`GSF`) – the frame-analogue to `GSM` (cf. Def. 8). We also check if properties *reflect* `GSF`. A property $\phi$ *reflects* `GSF` if whenever for every frame $\mathcal{F}$, it holds that every `GSF` of $\mathcal{F}$ has property $\phi$, then so does $\mathcal{F}$. We also consider closure under taking disjoint unions (`DU`) of frames, which are defined in the obvious way. Moreover, we look at closure under images of bisimulation systems [10], which are families of partial isomorphisms.

**Definition 11 (Bisimulation System).** *A bisimulation system from a frame $\mathcal{F}$ to a frame $\mathcal{F}'$ is a function $\mathcal{Z} : \wp W' \to \wp(W \times W')$ that assigns to each $Y \subseteq W'$ a total bisimulation $\mathcal{Z}(Y) \subseteq W \times W'$ such that for each $y \in Y$:*

1. *There is exactly one $w \in W$ such that $(w, y) \in \mathcal{Z}(Y)$.*
2. *If $(w, y), (w, w') \in \mathcal{Z}(Y)$, then $w' = y$.*

**Background results.** We indicate three classical characterization results. For details see [11, 10]. Let $\phi(x)$ be a formula of the FO correspondence language with at most one free variable. [12] proved that $\phi(x)$ is invariant under bisimulations iff $\phi(x)$ is equivalent to the standard translation of a modal formula. While [13, 14] proved that $\phi(x)$ is invariant under taking generated submodels iff $\phi(x)$ is equivalent to the standard translation of a formula of $\mathcal{L}(\mathtt{N}, \downarrow, @, x)$. On the level of frames [15] proved that a FO definable class of frames is modally definable iff it is closed under taking `BMI`, `GSF`, disjoint unions and reflects ultrafilter extensions.

The reader might now like to see immediately how the notions can be defined in extended modal languages and go directly to Sect. 6. Of course, the choice of the languages is only justified once we have determined the required expressive power both to express the local notions and to define the class of frames corresponding to the global ones. Thus we start by doing so in the next section.

## 5 Invariance and Closure Results

We start with satisfiability invariance results for the classes of pointed models defined in Sect. 2. Then we turn to closure results for classes of frames defined by global notions. A "Y" in a cell means that the row notion is invariant under the column operation. The number in the columns refer to representative proofs for these results that can be found in a technical report [16]. **Overview of the Results for the General Case.**

| | Bis | CBis | $\cap$-Bis | TBis | $\mathcal{H}$-Bis | $\mathcal{H}(@)$-Bis | $\mathcal{H}(\mathtt{E})$-Bis | BM | GSM | DU |
|---|---|---|---|---|---|---|---|---|---|---|
| $[PowL1]$ | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| $[PowL2]$ | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| $[PowL3]$ | N | N | N | N | N | N | N | N | Y | Y |
| $[PrefL1]$ | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| $[PrefL2]$ | N | N | Y | N | N | N | N | N | Y | Y |
| $[PrefL3]$ | N | N | N | N | N | N | N | N | Y | Y |
| $[PrefL4]$ | N | Y | N | N | N | N | N | N | N(2) | Y |
| $[PrefL5]$ | N | N | N | N | N | N | N | N | N | Y |
| $[PrefL6]$ | N | N | N | N | N | N | N | N | Y | Y |

|  | Bis | CBis | $\cap$-Bis | TBis | $\mathcal{H}$-Bis | $\mathcal{H}(@)$-Bis | $\mathcal{H}(\mathtt{E})$-Bis | BM | GSM | DU |
|---|---|---|---|---|---|---|---|---|---|---|
| [$PrefL7$] | N | N | N | N | N | N | N | N | Y | Y |
| [$PrefL8$] | N | N | N | N | N | N | N | N | N | N |
| [$PrefL9$] | N | N | N | N | N | N | N | N | N | N |
| [$PrefL10$] | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| [$PrefL11$] | N | N | N | N | N | N | N | N | Y | Y |
| [$PPL1$] | N | N | Y | N | N | N | N | N | Y | Y |
| [$PPL2$] | N | N | Y | N | N | N | N | N | Y | Y |
| [$PPL3$] | N | N | N | N | N | N | N | N | Y | Y |
| [$PPL4$] | N | N | N | N | N | N | N | N | Y | Y |
| [$PPL5$] | N | N | N | N | N | N | N | N | Y | Y |
| [$PPL6$] | N | N | N | N | N | N | N | Y | Y | Y |
| [$PPL7$] | N | N | N | N | N | N | N | N | Y | Y |
| [$EF1$] | N | N | N | N | N | N | N | N | Y | Y |
| [$EF2$] | N | N | N | N | N | N | N | N | Y | Y |
| [$EF3$] | N | N | Y | N | N | N | N | N | Y | Y |
| [$ST1$] | N | N | N | N | N | N | N | N | Y | Y |
| [$ST2$] | N | N | N(1) | N | N | N | N | N | Y | Y |
| [$ST3$] | N | N | N | N | N | N | N | N | Y | Y |
| [$ST4$] | N | N | Y | N | N | N | N | N | Y | Y |

**Comments**. Most of our notions are not bisimulation-invariant. Basic modal language [2] is thus not expressive enough to describe our local notions (without restrictions on the class of frames). Invariance under `BM` often fails; some failures are due to intersections of relations, but as $\cap$-Bis also fails quite often, this cannot be the only reason. By contrast invariance under `GSM` generally holds; it fails for properties with backward looking features. This is good news for expressivity: we can expect definability in the hybrid language with $\downarrow$-binder [3] . But not for computability, since the satisfiability problem of the bounded fragment is undecidable. Finally, the results are the same for hybrid and basic bisimulations. No suprise: roughly speaking, at the level of local satisfaction, to exploit the expressive power of nominals, the notions would have to refer explicitly to some state. Here are two representative results.

**Results Overview for the Total Pre-orders (`TPO`) Case**. This table shows rows that differ from the general case. Entries that differ are in boldface.

|  | Bis | CBis | $\cap$-Bis | TBis | $\mathcal{H}$-Bis | $\mathcal{H}(@)$-Bis | $\mathcal{H}(\mathtt{E})$-Bis | BM | GSM | DU |
|---|---|---|---|---|---|---|---|---|---|---|
| [$PrefL8$] | N | **Y** | N | N | N | N | N | N | N | **Y*** |
| [$PrefL9$] | N | **Y** | N | N | N | N | N | N | N | **Y*** |
| [$ST2$] | N | N | **Y** | N | N | N | N | N | Y | Y* |

**Comments.** Except for disjoint union (`DU`), the restriction to the `TPO` case brings only slight benefits. * marks trivial invariance: the only `DU` of models that is complete is the trivial one: mapping a model to itself.

[2] of similarity type $\langle\{\ \xrightarrow{\mathtt{C}}\ \mid \mathtt{C} \subseteq \mathtt{N}\}, \{\ \leq_i\ \mid i \in \mathtt{N}\}\rangle$

[3] [14, 13] have proved that all notions definable in the first-order correspondence language that are invariant under `GSM` are equivalent to a formula of the bounded fragment, i.e. of the hybrid language with $\downarrow$-binder (which are notational variants).

**Overview of the Results for the TPO Case with Strict Preferences.** The following table contains the rows that differ from the ones in the table for total preorders without strict preference relation.

| | Bis | CBis | $\cap$-Bis | TBis | $\mathcal{H}$-Bis | $\mathcal{H}(@)$-Bis | $\mathcal{H}(\mathtt{E})$-Bis | BM | GSM | DU |
|---|---|---|---|---|---|---|---|---|---|---|
| $[PrefL2]$ | **Y** | **Y** | Y | **Y** | **Y** | **Y** | **Y** | **Y** | Y | Y |
| $[PrefL3]$ | N | N | N | N | N | N | N | **Y** | Y | Y |
| $[PrefL5]$ | N | **Y** | N | N | N | N | N | N | N | Y |
| $[PrefL6]$ | N | N | N | N | N | N | N | **Y** | Y | Y |
| $[PrefL7]$ | N | N | N | N | N | N | N | **Y** | Y | Y |
| $[PrefL11]$ | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** (4) | Y | Y |
| $[PPL7]$ | N | N | N | N | N | N | N | **Y** | Y | Y |
| $[EF1]$ | N | N | **Y** | N | N | N | N | N | Y | Y |
| $[EF2]$ | N | N | **Y** | N | N | N | N | **Y** | Y | Y |
| $[ST1]$ | N | N | **Y** | N | N | N | N | N | Y | Y |
| $[ST3]$ | N | N | **Y** | N | N | N | N | N | Y | Y |

**Comments.** The failures of invariance under GSM are still present, reflecting the fact that we do not have converse relations. By contrast, $PrefL11$ and $PrefL2$ are now invariant under bisimulation and a simple boolean modal logic with intersection seems to have the right expressive power to talk about efficiency and stability notions, since all of them are now invariant under $\cap$-Bisimulations. We now check if the properties define class of frames that are closed under the different operations introduced. The tables read off as in the previous section. **Closure Results for class of frames defined by global properties.**

| | BMI | GSF | DU | refl.GSF | BisSysIm | | BMI | GSF | DU | refl.GSF | BisSysIm | | BMI | GSF | DU | refl.GSF | BisSysIm |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $PowG1$ | Y | Y | Y | Y | Y | $PrefG4$ | Y | Y | N | N | Y | $PPG1$ | Y | Y | Y | Y | Y |
| $PowG2$ | Y | Y | Y | Y | Y | $PrefG5$ | N | Y | Y | Y | Y | $PPG2$ | Y | Y | N | N | Y |
| $PowG3$ | Y | Y | Y | Y | Y | $PrefG6$ | N | Y | N | N | Y | $PPG3$ | N | N | N | Y | N? |
| $PowG4$ | Y | Y | Y | Y | Y | $PrefG7$ | N | Y | N | N | Y | $PPG4$ | N | Y | Y | Y | Y |
| $PowG5$ | Y | Y | Y | Y | Y | $PrefG8$ | N | Y | Y | Y | Y | $PPG5$ | Y | Y | Y | Y | Y |
| $PrefG1$ | Y | Y | Y | Y | Y | $PrefG9$ | N | Y | Y | Y | Y | $PPG6$ | Y | Y | Y | Y | Y |
| $PrefG2$ | Y | Y | Y | Y | Y | $PrefG10$ | Y | Y | Y | Y | Y | $PPG7$ | Y | N | Y | Y | Y |
| $PrefG3$ | Y | Y | N | N | Y | $PrefG11$ | N(5) | Y | Y | Y | Y | | | | | | |

At the frame level, ML is a fragment of Monadic Second Order Logic. That it does better at this level is thus not only an artifact of the chosen notions.

## 6 Modal Definability

The previous model-theoretic results give us information about possible definability results. However, let us be more constructive and give formulas that indeed do the job: be it for local-satisfaction or frame-definability aims. Correspondence proofs can be found in the technical report. We indicate the least expressive language we found still being able to express the property under consideration. Another useful criterion is that of the computational complexity of the logic, i.e. of its satisfiability problem (SAT) and model checking problem (*MC*). Since we lack the space to discuss these issues in depth here is how we bridge our expressivity and complexity results: for each local (resp. global) notion, find the least expressive logic that is still able to express it locally (resp. define the class of frames corresponding to it) and take the complexity of this logic as an *upper bound.* Due to space restrictions we only indicate these upper bounds and references for them. The technical report has more details. We assume the reader to be familiar PSPACE and EXPTIME [17]. $\Pi^0_1$-complete problems [18] are undecidable but co-recursively enumerable (e.g. $\mathbb{N}\times\mathbb{N}$ tiling [19]).

**Defining Local Notions**

| | Local Formula | Best Language | SAT | *MC* |
|---|---|---|---|---|
| $PowL1$ | $\langle\mathtt{C}\rangle p$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[20] | P[?] |
| $PowL2$ | $\bigwedge_{\mathtt{C}\not\supseteq}[\mathtt{C}]\neg p$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[21] | P[?] |
| $PowL3$ | $\downarrow x.[\mathtt{D}]\downarrow y.@_x\langle\mathtt{C}\rangle y$ (6) | $\mathcal{L}(\mathtt{N},\downarrow,@,x)$ | (7) EXPTIME [22] | PSPACE[23] |
| $PrefL1$ | $\langle\leq_i\rangle$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[20] | P[?] |
| $PrefL2$ | $\downarrow x.\langle\leq_i\rangle(p \wedge [\leq_i]\neg x)$ | $\mathcal{L}(\mathtt{N},\downarrow,x)$ | EXPTIME[22] | PSPACE[23] |
| $PrefL3$ | $\downarrow x.\langle\bigcap_{i\in N}\leq_i\rangle(\bigvee_{j\in\mathtt{N}}[\leq_j]\neg x)$ | $\mathcal{L}(\mathtt{N},\downarrow,\cap,x)$[10] | $\Pi^0_1$ | PSPACE |
| $PrefL4$ | $\langle{\leq_i}^{-1}\rangle p$ | $\mathcal{L}(\mathtt{N},\downarrow,@,x)$ | PSPACE | PSPACE[23] |
| $PrefL5$ | $\downarrow x.\langle{\leq_i}^{-1}\rangle(p \wedge [{\leq_i}^{-1}]\neg x)$ | $\mathcal{L}(\mathtt{N},\downarrow,{}^{-1},x)$ | $\Pi^0_1$ [24] | PSPACE |
| $PrefL6$ | $[(\leq_i \cap\overline{\leq_j})\cup(\leq_j \cap\overline{\leq_i})]\bot$ | $\mathcal{L}(\mathtt{N},\text{-},\cap)$ | EXPTIME[25, sec.5] | P[26] |
| $PrefL7$ | $\langle\leq_i \cap(\bigcap_{j\in\mathtt{N}-\{i\}}\overline{\leq_j})\rangle\top$ | $\mathcal{L}(\mathtt{N},\text{-},\cap)$ | EXPTIME[25, sec.5] | P[26] |
| $PrefL8$ | $[\![\ \leq_i\ ]\!]p$ | $\mathcal{L}(\mathtt{N},[\![\ \ ]\!])$ | EXPTIME[25, sec.5] | P[26] |
| $PrefL9$ | $\downarrow x.\mathtt{A}\downarrow y.(\neg\langle\leq_i\rangle x \wedge @_x\langle\leq_i\rangle y)$ | $\mathcal{L}(\mathtt{N},\downarrow,@,x,\mathtt{E})$ | $\Pi^0_1$ [24] | PSPACE[27] |
| $PrefL10$ | $[\leq_i]p$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[21] | P[?] |
| $PrefL11$ | $\downarrow x.[\leq_i]([\leq_i]\neg x \rightarrow p)$ | $\mathcal{L}(\downarrow,x)$ | EXPTIME[22] | PSPACE[23] |
| $PPL1$ | $\langle\mathtt{C}\cap\leq_i\rangle\top$ | $\mathcal{L}(\mathtt{N},\cap)$ | PSPACE [28] | P[26] |
| $PPL2$ | $\langle\mathtt{C}\cap(\bigcap_{i\in\mathtt{D}}\leq_i)\rangle\top$ | $\mathcal{L}(\mathtt{N},\text{-},\cap)$ | EXPTIME[25, sec.5] | P[26] |
| $PPL3$ | $\langle(\bigcap_{i\in\mathtt{N}}\leq_i)\cap(\bigcup_{\mathtt{C}\subseteq\mathtt{N}}\xrightarrow{\mathtt{C}})\rangle\top$ | $\mathcal{L}(\mathtt{N},\text{-},\cap)$ | EXPTIME[25, sec.5] | P[26] |
| $PPL4$ | $[\overline{\mathtt{C}}\cap\leq_i]\bot$ | $\mathcal{L}(\mathtt{N},\text{-},\cap)$ | EXPTIME[25, sec.5] | P[26] |
| $PPL5$ | $\downarrow x.[\overline{\mathtt{C}}\cap\leq_i]\langle\leq_i\rangle x$ | $\mathcal{L}(\mathtt{N}\downarrow,,\text{-},\cap,x)$ | $\Pi^0_1$ [24] | PSPACE |
| $PPL6$ | $\bigvee_{\mathtt{C}\subseteq\mathtt{N}}[\mathtt{C}\cap\overline{\leq_i}]\bot$ | $\mathcal{L}(\mathtt{N},\text{-},\cap)$ | EXPTIME | P[26] |
| $PPL7$ | $\downarrow x.[\mathtt{C}]\downarrow.y.(\neg\langle\leq_i\rangle x \wedge @_x\langle\leq_i\rangle y)$ | $\mathcal{L}(\mathtt{N},\downarrow,@,x)$ | $\Pi^0_1$[22] | PSPACE[23] |
| $EF1$ | $\downarrow x.[\bigcap_{i\in\mathtt{N}}\leq_i]\bigvee_{i\in\mathtt{N}}\langle\leq_i\rangle x$ | $\mathcal{L}(\mathtt{N},\downarrow,\cap)$ | $\Pi^0_1[10,22]$ | PSPACE |
| $EF2$ | $\neg\downarrow x.\langle\bigcap_{i\in N}\leq_i\rangle(\bigvee_{j\in\mathtt{N}}[\leq_j]\neg x)$ | $\mathcal{L}(\mathtt{N},\downarrow,\cap)$ | $\Pi^0_1[10,22]$ | PSPACE |
| $EF3$ | $[\bigcap_{i\in\mathtt{N}}\leq_i]\bot$ | $\mathcal{L}(\mathtt{N},\cap)$ | PSPACE [28] | P[26] |
| $ST1$ | $\bigwedge_{i\in\mathtt{N}}\downarrow x.[i\cap\leq_i]\langle\leq_i\rangle x$ | $\mathcal{L}(\mathtt{N},\downarrow,\cap)$ | $\Pi^0_1$[10] | PSPACE |
| $ST2$ | $\bigwedge_{i\in\mathtt{N}}[i\cap\leq_i]\bot$ | $\mathcal{L}(\mathtt{N},\cap)$ | PSPACE [28] | P[26] |
| $ST3$ | $\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}\downarrow x.[\mathtt{C}\cap(\bigcap_{i\in\mathtt{C}}\leq_i)]\bigvee_{j\in\mathtt{C}}\langle\leq_j\rangle x$ | $\mathcal{L}(\mathtt{N},\downarrow,\cap)$ | $\Pi^0_1$[22] | PSPACE |
| $ST4$ | $\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}[\mathtt{C}\cap(\bigcap_{i\in\mathtt{C}}\leq_i)]\bot$ | $\mathcal{L}(\mathtt{N},\cap)$ | PSPACE [28] | P[26] |

## 6.1 Defining Global Notions

First of all, we define what it means for a formula to be valid on a class of frames.

**Definition 12 (Validity on a class of frames).** *We say that a formula $\phi$ is valid on a class of frames $\mathtt{F}$ iff for any frame $\mathcal{F} \in \mathtt{F}$ and any model $\mathcal{M}$ based on $\mathcal{F}$, at all states $w$ in $Dom(\mathcal{F})$, $\mathcal{M}, w \Vdash \phi$. We write $\mathtt{F} \Vdash \phi$.*

Modal definability has again two sides: We can look for a formula $\phi$ such that $\mathcal{M}, w \Vdash \phi$ iff $\mathcal{M}, w$ has some property, or such that $\mathtt{F} \Vdash \phi$ iff $\mathtt{F}$ has the property.

| | Axiom | Best Language | SAT | $MC$ |
|---|---|---|---|---|
| $PowG1$ | $\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(((\langle\mathtt{C}\rangle\phi \rightarrow [\mathtt{C}]\phi) \wedge \langle\mathtt{C}\rangle\top)$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[20] | P[?] |
| $PowG2$ | $\bigwedge_{\mathtt{C}:\lvert\mathtt{C}\rvert<\lvert\mathtt{N}\rvert/2}[\mathtt{C}]\bot$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[21] | P[?] |
| $PowG3$ | $\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(\langle\mathtt{C}\rangle\phi \rightarrow [\mathtt{C}]\phi)$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[20] | P[?] |
| $PowG4$ | $\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}\bigwedge_{\mathtt{D}\supseteq\mathtt{C}}(\langle\mathtt{C}\rangle\phi \rightarrow \langle\mathtt{D}\rangle\phi)$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[21] | P[?] |
| $PowG5$ | $\langle\mathtt{C}\rangle\top \rightarrow \bigwedge_{\mathtt{D}:\mathtt{C}\cap\mathtt{D}=\emptyset}(\langle\mathtt{D}\rangle\phi \rightarrow \langle\mathtt{C}\rangle\phi)$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[20] | P[?] |
| $PrefG1$ | $\phi \rightarrow \langle\leq_i\rangle\phi$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[21] | P[?] |
| $PrefG2$ | $\langle\leq_i\rangle\langle\leq_i\rangle\phi \rightarrow \langle\leq_i\rangle\phi$ | $\mathcal{L}(\mathtt{N})$ | PSPACE[20] | P[?] |
| $PrefG3$ | $p \wedge \mathtt{E}q \;\rightarrow\; (\mathtt{E}(p \wedge \langle\leq_i\rangle q) \vee \mathtt{E}(q \wedge \langle\leq_i\rangle p))$ | $\mathcal{L}(\mathtt{N},\mathtt{E})$ | EXPTIME[29] | P[26] |
| $PrefG4$ | Conjunction of the 3 previous axioms | $\mathcal{L}(\mathtt{N},\mathtt{E})$ | EXPTIME[30] | P[26] |
| $PrefG5$ | see below | $\mathcal{L}(\mathtt{N})$ | PSPACE[21] | P |
| $PrefG6$ | $\bigwedge_{i\in\mathtt{N}}(@_j\langle\leq_i\rangle k \vee @_k j \vee @_k\langle\leq_i\rangle j)$ | $\mathcal{L}(\mathtt{N},@,i)$ | PSPACE[31] | P[23] |
| $PrefG7$ | $[PrefG5] \wedge [PrefG6] \wedge (\bigwedge_{i\in\mathtt{N}}(j \rightarrow \neg\langle\leq_j\rangle j))$ | $\mathcal{L}(\mathtt{N},@,i)$ | PSPACE[31] | P[23] |
| $PrefG8$ | $\bigwedge_{i\in\mathtt{N}}\;(((\langle\leq_i\rangle\phi \rightarrow [\leq_i]\phi) \wedge \langle\leq_i\rangle\top)$ | $\mathcal{L}(\mathtt{N})$ | PSPACE | P |
| $PrefG9$ | $\langle\mathtt{C}\rangle i \leftrightarrow \bigwedge_{i\in\mathtt{C}}\langle\leq_i\rangle i$ | $\mathcal{L}(\mathtt{N},i)$ | PSPACE[31] | P[23] |
| $PrefG10$ | $\langle\mathtt{C}\rangle p \leftrightarrow \bigvee_{i\in\mathtt{C}}\langle\leq_i\rangle p$ | $\mathcal{L}(\mathtt{N})$ | PSPACE | P |
| $PrefG11$ | $\langle\mathtt{C}\rangle i \leftrightarrow \bigvee_{\mathtt{D}\subseteq\mathtt{C}\&\lvert\mathtt{D}\rvert>\frac{\lvert\mathtt{C}\rvert}{2}}(\bigwedge_{i\in\mathtt{D}}\langle\leq_i\rangle i)$ | $\mathcal{L}(\mathtt{N},i)$ | PSPACE[31] | P[23] |
| $PPG1$ | $\langle\mathtt{C}\rangle\phi \rightarrow \bigwedge_{i\in\mathtt{N}}\langle\leq_i\rangle p$ | $\mathcal{L}(\mathtt{N})$ | PSPACE | P |
| $PPG2$ | $\bigvee_{i\in\mathtt{N}}\mathtt{A}\bigwedge_{\mathtt{C}\subseteq\mathtt{N}}(\langle\mathtt{C}\rangle\phi \rightarrow \langle\leq_i\rangle\phi)$ | $\mathcal{L}(\mathtt{N},\mathtt{E})$ | EXPTIME[29] | P |
| $PPG3$ | $\bigwedge_{i\in\mathtt{N}}\bigvee_{\mathtt{C}\subseteq\mathtt{N}}(\overline{\leq_i}\cup\leq_i)\langle\overline{\leq_i}\cap\mathtt{C}\rangle\top$ | $\mathcal{L}(\text{-},\cap,\cup)$ | EXPTIME | P[26] |
| $PPG4$ | see below | $\mathcal{L}(\mathtt{N},i)$ | PSPACE[31] | P[23] |
| $PPG5$ | $\bigwedge_{\mathtt{C}\not\supseteq\{i\}}[\overset{\mathtt{C}}{\rightarrow}]\bot$ | $\mathcal{L}(\mathtt{N})$ | PSPACE | P |
| $PPG6$ | $\langle\mathtt{C}\rangle\phi \rightarrow \bigvee_{\mathtt{D}\subset\mathtt{C}}\langle\mathtt{D}\rangle\phi$ | $\mathcal{L}(\mathtt{N})$ | PSPACE | P |
| $PPG7$ | $\bigwedge_{i\in\mathtt{N}}\mathtt{E}\bigwedge_{\mathtt{C}\not\supseteq\{i\}}[\mathtt{C}]\bot$ | $\mathcal{L}(\mathtt{N},\mathtt{E},i)$ | EXPTIME[31] | P[23] |

$$\bigwedge_{i\in\mathtt{N}} (p \wedge \langle\leq_i\rangle(q \wedge \neg\langle\leq_i\rangle p \wedge \langle\leq_i\rangle(r \wedge \neg\langle\leq_i\rangle q))) \;\rightarrow\; p \wedge \langle\leq_i\rangle(r \wedge \neg\langle\leq_i\rangle p) \;\; (AxPrefG5)$$

$$[p \wedge \langle\{i\}\rangle q \wedge \langle\leq_i\rangle(q \wedge \langle\leq_i\rangle\neg p)] \;\rightarrow\; \bigwedge_{\{i\}\subseteq\mathtt{C}\subseteq\mathtt{N}} [(\langle\mathtt{C}\rangle r \wedge \bigwedge_{\mathtt{D}\subseteq\mathtt{C}-\{i\}} \neg\langle\mathtt{D}\rangle r) \;\rightarrow\; \langle\leq_i\rangle(r \wedge \neg\langle\leq_i\rangle p)] \; (PPG4)$$

## 7 Conclusion

We identified a set of natural notions for reasoning about cooperation: local notions giving properties of a state of a given system and global notions defining a class of frames. We provided satisfiability (resp. validity) invariance results for these notions for a large class of operations and relations between models (resp.

frames). We also gave explicit definability results and observed that defining frames for cooperation logics does not seem too demanding in terms of expressive power, as most of the notions considered are definable in the basic modal language. On the other hand, our results show that many local notions of interest call for modal logics for which satisfaction is not invariant under bounded morphisms. However, as long as we avoid converse modalities, interesting reasoning about cooperation can be done within GSM-invariant modal languages. Though this fact does not directly lead to a nice upper bound on the complexity of the logic's SAT (nor to its decidability), our definability results show that most of the considered notions can (invidually) be expressed in MLs in EXPTIME. Moreover, the notions for which expressibility in logics with decidable SAT was not found all require to express the idea of a "strict" improvement (e.g. Nash-stable, Pareto-efficient). By contrast, strong notions of stability and efficiency (EF3, ST2, ST4) are expressible in logics with SAT in PSPACE. So, we could say that "expressing strictness" – and thus "weak" notions – are dangerous, while "strong" notions (looking only at the non-strict preference relation) are safe.

Based on our current work, the following lines seem worth exploring:

- Since dealing with real coalitional powers is probably more natural using neighborhood semantics, it will be useful to do the same work for modal logics of the **CL**-type or of the type of one of its normal simulations [2].
- It would be interesting to obtain similar invariance results and upper bounds on the complexity of the logics needed to encode *concrete arguments* from SCT and (cooperative) GT, thus addressing the complexity of *actual reasoning* about cooperative situations.
- In order to obtain a complete picture of the complexity of reasoning about cooperation, we need a procedure to assess the lower bound (LB) of the complexity of modal logics that can express some notion. As an example: a way to go could be to take a hardness result for the problem of determining whether a profile of strategies is a pure Nash-equilibrium of a given game (with respect to some reasonable and qualitative encoding of games) as a LB on the model-checking complexity of a logic than can express this notion.

**Acknowledgments.** The authors are very grateful to Johan van Benthem, Eric Pacuit, Natasha Alechina, Peter van Emde Boas, Balder ten Cate, Ulle Endriss and the anonymous referees for their useful comments.

## References

1. Pauly, M.: A modal logic for coalitional power in games. JLC **12**(1) (2002) 149–166
2. Broersen, J., Herzig, A., Troquard, N.: Normal Coalition Logic and its conformant extension. In Samet, D., ed.: TARK'07, PUL (2007) 91–101
3. Girard, P.: Modal Logic for Preference Change. PhD thesis, Stanford (2008)
4. Kurzen, L.: Logics for Cooperation, Actions and Preferences. Master's thesis, Universiteit van Amsterdam, the Netherlands (2007)
5. Ågotnes, T., Dunne, P.E., van der Hoek, W., Wooldridge, M.: Logics for coalitional games. In: LORI '07, Beijing, China (2007) to appear.

6. Goranko, V.: Coalition games and alternating temporal logics. In: TARK '01, San Francisco, CA, USA, Morgan Kaufmann (2001) 259–272
7. Hansen, H.H., Kupke, C., Pacuit, E.: Bisimulations for neighbourhood structures. In: Proc. of 2nd Conference on Algebra and Coalgebra in CS. LNCS (2007)
8. de Jongh, D., Liu, F.: Optimality, belief and preference. In Artemov, S., Parikh, R., eds.: Proc. of the Workshop on Rationality and Knowledge, ESSLLI (2006)
9. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT Press
10. Cate, B.: Model theory for extended modal languages. PhD thesis, University of Amsterdam (2005) ILLC Dissertation Series DS-2005-01.
11. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. CUP (2001)
12. van Benthem, J.: Modal Logic and Classical Logic. Bibliopolis, Napoli (1983)
13. Areces, C., Blackburn, P., Marx, M.: Hybrid logics: characterization, interpolation and complexity. The Journal of Symbolic Logic **66**(3) (2001) 977–1010
14. Feferman, S.: Persistent and invariant formulas for outer extensions. Compositio Mathematica **20** (1969) 29–52
15. Goldblatt, R.I., Thomason, S.K.: Axiomatic classes in propositional modal logic. In Crossley, J.N., ed.: Algebra and Logic: Papers 14th Summer Research Inst. of the Australian Math. Soc. Volume 450 of LNM. 163–173
16. Dégremont, C., Kurzen, L.: Modal logics for preferences and cooperation: Expressivity and complexity. Technical report, ILLC, University of Amsterdam (2008) http://staff.science.uva.nl/ cdegremo/.
17. Papadimitriou, C.M.: Computational complexity. Addison-Wesley, MA (1994)
18. Odifreddi, P.: Classical Recursion Theory. Number 125 in Studies in Logic and the Foundations of Mathematics. North-Holland (1989)
19. Harel, D.: Recurring dominoes: making the highly undecidable highly understandable. In: Topics in the theory of computation, Elsevier (1985) 51–71
20. Ladner, R.E.: The computational complexity of provability in systems of modal propositional logic. SIAM J. Comput. **6**(3) (1977) 467–480
21. Halpern, J.Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence **54**(3) (1992) 319–379
22. ten Cate, B., Franceschet, M.: On the complexity of hybrid logics with binders. In Ong, L., ed.: Proc. of CSL 2005. Volume 3634 of LNCS., Springer (2005) 339–354
23. Franceschet, M., de Rijke, M.: Model checking for hybrid logics. In: Proceedings of the Workshop Methods for Modalities. (2003)
24. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem, Berlin (1997)
25. Lutz, C., Sattler, U.: The complexity of reasoning with boolean modal logics. In Wolter, Wansing, de Rijke, Zakharyaschev, eds.: AiML, WS (2000) 329–348
26. Lange, M.: Model checking pdl with all extras. J. Applied Logic **4**(1) (2006) 39–49
27. Franceschet, M., de Rijke, M.: Model checking hybrid logics (with an application to semistructured data). J. Applied Logic **4**(3) (2006) 279–304
28. Donini, F.M., Lenzerini, M., Nardi, D., Nutt, W.: The complexity of concept languages. In: KR. (1991) 151–162
29. Spaan, E.: Complexity of modal logics. PhD thesis, ILLC Amsterdam (1993)
30. Hemaspaandra, E.: The price of universality. NDJFL **37**(2) (1996) 174–203
31. Areces, C., Blackburn, P., Marx, M.: A road-map on complexity for hybrid logics. In Flum, Rodríguez-Artalejo, eds.: CSL. Number 1683 in LNCS (1999) 307–321

# Simulation and information: quantifying over epistemic events

Hans van Ditmarsch[*], Tim French[†]

**Abstract**

We introduce a multi-agent logic of knowledge with time where $F\varphi$ stands for 'there is an informative event after which $\varphi$'. Formula $F\varphi$ is true in a model iff it is true in all its refinements (i.e., 'atoms' and 'back' are satisfied; the dual of simulation). The logic is 'almost' normal, and positive knowledge is preserved. The meaning of $F\varphi$ is also "after the agents become aware of new factual information, $\varphi$ is true," and on finite models it is also "there is an event model $(\mathsf{M}, \mathsf{s})$ after which $\varphi$." The former provides a correspondence with bisimulation quantifiers in a setting with epistemic operators.

## 1 Introduction

If you know where you are and you know what's going to happen, you *want* to know where you will end up. But it can also be that you know where you are and know where you would like to end up, and that you *want* to know how to make that happen. Or you might *want* to know where you can end up in the first place, disregarding how that may be brought about. In the setting of logics for information update [3, 12, 11], knowledge of where you are and where you end up is formalized in multi-agent epistemic logic and semantically represented by a pointed multi-agent Kripke model, and knowledge about what's going to happen is formalized as a dynamic modal operation that is interpreted as a relation between such Kripke models. The standard focus in dynamic epistemic logic was on the first of the three issues above: precision about a specific information update and precision about the effects of that update. In this contribution we focus on the other two issues instead. As this is partly about what may happen after *any* event, this concerns quantification over events. Our work is a further generalization of works such as [8, 2] and our presentation of future event operators as temporal is motivated by works such as [9] linking temporal epistemic logic to dynamic epistemic logic.

We introduce a very succinct logic of future events: the multi-agent logic of knowledge with (only) an operation $G\varphi$ that stands for '$\varphi$ holds after all informative events' — the diamond version $F\varphi$ stands for 'there is an informative event after which $\varphi$.' The semantics of $G\varphi$ employs the notion of simulation [1]. We demonstrate that this is useful notion for informative event by a number of technical results for this logic—the logic is 'almost' normal, positive knowledge is preserved—and by a number of equivalence results for alternative semantics: $F\varphi$ also means "there is an event model $(\mathsf{M}, \mathsf{s})$ after which $\varphi$," and it also means "after the agents become aware of new factual information, $\varphi$ is true." The last provides a correspondence with bisimulation quantifiers [13, 7] in a setting with epistemic operators, as in [5].

For standard notions such as epistemic model, epistemic state, bisimulation, simulation, refinement, and event model, and for standard abbreviations and other conventions, we refer

[*] Computer Science, University of Otago, New Zealand & IRIT, France; hans@cs.otago.ac.nz.

[†] Computer Science and Software Engineering, University of Western Australia; tim@csse.uwa.edu.au.

to the appendix. Throughout our contribution, the set of agents $A$ is finite and the set of atoms $P$ is (infinitely) enumerable.

## 2 Simulation and information

In *future event logic* one can express what informative events can be expected in a given information state. The language and the semantics of future event logic are as follows.

**Definition 1 (Language $\mathcal{L}_{fel}$)** Given agents $A$ and atoms $P$, the language $\mathcal{L}_{fel}$ is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \land \varphi) \mid K_a\varphi \mid G\varphi$$

where $a \in A$ and $p \in P$.

We write $F\varphi$ for $\neg G \neg \varphi$. We propose a dynamic epistemic modal way to interpret temporal operators. This means that our future is the *computable future*: $F\varphi$ is true now, iff there is an (unspecified) informative event after which $\varphi$ is true.

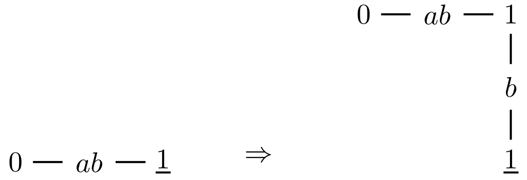
In the semantics for $G\varphi$, now to follow, we use the structural notion of *refinement*. A bisimulation satisfies **atoms**, **forth** and **back**, a simulation **atoms** and **forth**, and a refinement **atoms** and **back**. Refinement is therefore the dual of simulation: if $(M, s)$ is a simulation of $(M', s')$, then $(M', s')$ is a refinement of $(M, s)$ (and we write $(M', s') \underline{\rightarrow} (M', s')$ and $(M, s) \underline{\leftarrow} (M', s')$, respectively). See the appendix for more details.

**Definition 2 (Semantics of future event logic)** Assume an epistemic model $M = (S, R, V)$. The interpretation of $\varphi \in \mathcal{L}_{fel}$ is defined by induction.

$$\begin{array}{lll}
M, s \models p & \text{iff} & s \in V_p \\
M, s \models \neg\varphi & \text{iff} & M, s \not\models \varphi \\
M, s \models \varphi \land \psi & \text{iff} & M, s \models \varphi \text{ and } M, s \models \psi \\
M, s \models K_a\varphi & \text{iff} & \text{for all } t \in S : (s, t) \in R_a \text{ implies } M, t \models \varphi \\
M, s \models G\varphi & \text{iff} & \text{for all } (M', s') \underline{\rightarrow} (M, s) : M', s' \models \varphi
\end{array}$$

In other words, $G\varphi$ is true in an epistemic state iff $\varphi$ is true in all of its *refinements*. Note the 'wrong direction' in the definition: the future epistemic state simulates the current epistemic state. Typical model operations that produce a refinement are: blowing up the model (to a bisimilar model) such as adding copies that are indistinguishable from the current model and one another for some agent(s), removing states, and removing pairs of the accessibility relation for an agent. Validity in a model, and validity, are defined as usual. For $\{s \mid M, s \models \varphi\}$ we write $[\![\varphi]\!]_M$.

**Example 3** Given are two agents that are uncertain about the value of a fact $p$, and where this is common knowledge, and where $p$ is true. We assume both accessibility relations are equivalence relations, and that the epistemic operators model the agents' knowledge. An informative event is possible after which $a$ knows that $p$ but $b$ does not know that: $M, 1 \models F(K_a p \land \neg K_b K_a p)$. In the figure, $(M, 1)$ is the structure on the left, and its refinement validating the postcondition is on the right.



**Proposition 4** Some elementary validities are:

1. $\models G(\varphi \to \psi) \leftrightarrow (G\varphi \to G\psi)$
2. $\models G\varphi \to \varphi$
3. $\models G\varphi \to GG\varphi$
4. $\models \varphi$ implies $\models G\varphi$
5. $\models K_a G\varphi \to GK_a\varphi$

**Proof**

1. Obvious.
2. A model is a refinement of itself; this corresponds to the trivial event 'announce true'.
3. Consider the diamond version $FF\varphi \to F\varphi$. The relational composition of two simulations is again a simulation.
4. Obvious.
5. Consider the diamond version. Choose an accessible state in a refinement of a model. By **back**, this accessibility step can also be taken in the initial model.

Proposition 4 makes clear that $G$ comes close to being a normal modal operator. But it is not a normal modal logic: the validities of the logic are not closed under uniform substitution of atomic variables for other formulas. For example, given some atom $p$, $p \to Gp$ is valid, but $\neg Kp \to G\neg Kp$ is not valid. A countermodel of the latter is the typical two-state situation where there is uncertainty about the value of $p$ and where $p$ is true. In that case, restriction to the $p$-state ('public announcement of $p$') makes it known. Another countermodel is provided by the example above, for the knowledge of agent $b$.

A standard check for our bold claim that $G$ formalizes a notion of informative event is that

**Proposition 5** Bisimilar epistemic states have the same logical theory.

**Proof** This is not completely trivial, because bisimilarity is with respect to the epistemic operators, whereas the same logical theory is with respect to the epistemic operators and the temporal operator. Both can be established easily by the observation that if an epistemic state is a refinement of one of two given bisimilar epistemic states, it is also a refinement of the other epistemic state, because the relational composition of a simulation relation and a bisimulation relation is a simulation relation. The inductive case $G\varphi$ of the proof is:

Assume $\mathfrak{R} : (M, s) \underline{\leftrightarrow} (M', s')$, and let $M, s \models G\varphi$. To show that $M', s' \models G\varphi$, let $(M'', s'')$ be such that $\mathfrak{R}' : (M'', s'') \underline{\rightarrow} (M', s')$. We now have that $\mathfrak{R}' \circ \mathfrak{R}^{-1} : (M'', s'') \underline{\rightarrow} (M, s)$. From that and $M, s \models G\varphi$ follows $M'', s'' \models \varphi$.

The positive formulas are those in the inductively defined fragment

$$\varphi ::= p|\neg p|\varphi \vee \varphi|\varphi \wedge \varphi|K_a\varphi|G\varphi.$$

The preserved formulas are those for which

$$\varphi \rightarrow G\varphi \text{ is valid.}$$

I.e., they preserve truth under model refinement as long as the refinement includes an image for the actual state; the better known setting is model restriction. The first real corroboration that the temporal operators formalize a notion of informative event is that they model *growth of information* in the sense that positive knowledge does not get lost:

**Proposition 6** Positive formulas preserve truth under refinement of models.

**Proof** Elementary.

Further corroboration that the temporal operators are quantifying over informative events is provided by the observation that *a restricted modal product is a refinement of a model* if the valuations of the states in that model are preserved under the product operation. This entails that the execution of an event model in an epistemic state is a refinement of that epistemic state. This we will now address.

## 3 Quantifying over event models

An informative update is the execution of an event model in an epistemic state. We consider event models for the epistemic language $\mathcal{L}_{el}$.

**Proposition 7** An informative update is a refinement.

**Proof** Let $(\mathsf{M}, \mathsf{s}) = ((\mathsf{S}, \mathsf{R}, \mathsf{pre}), \mathsf{s})$ be an event model for language $\mathcal{L}_{el}$. Let $(M, s) = ((S, R, V), s)$ be an epistemic state and suppose $M, s \models \mathsf{pre}(\mathsf{s})$. Then $\mathfrak{R}(t, \mathsf{t}) = t$ is a simulation between $((M \otimes \mathsf{M}), (s, \mathsf{s}))$ and $(M, s)$; below we assume that $(M \otimes \mathsf{M}) = (S', R', V')$.

- **atoms**: if $(t, \mathsf{t}) \in V'(p)$ then $t \in V(p)$;
- **forth**: let $((t, \mathsf{t}), (t', \mathsf{t}')) \in R'_a$; then $(t, t') \in R_a$.

Subject to the restrictions that we also have common knowledge in the epistemic language (language $\mathcal{L}^C_{el}$) and that the epistemic models are finite, the fit is exact: refinements are informative updates.

**Proposition 8** (On finite epistemic models, given common knowledge) A refinement is an informative update.

**Proof** Given are a finite epistemic state $((S, R, V), s)$ and a refinement $((S', R', V'), s')$ of that model (according to refinement $\mathfrak{R}$). Consider the event model that is isomorphic to that refinement (according to isomorphism $\mathfrak{I}$). Instead of valuations for states $t$, this event model has preconditions for events $\mathfrak{I}(t)$. We want the preconditions only to be satisfied in states $s$ such that $(s, t) \in \mathfrak{R}$—this we cannot guarantee, but we can come close enough. In a finite model, states can be distinguished from all other (except bisimilar) states by employing the characteristic formulas $\delta_{((S,R,V),s)}$. (Characteristic formulas satisfy the property that truth in the structure equals entailment from the formula: $M, s \models \varphi$ iff $\delta_{(M,s)} \models \varphi$. Finite models have characteristic formulas in $\mathcal{L}_{el}^{C}$ [4].) These are the preconditions we need. Given a state $s \in S$:

$$\mathsf{pre}(\mathfrak{I}(t)) = \bigvee_{(s,t)\in\mathfrak{R}} \delta_{((S,R,V),s)}$$

This may give us pairs $(s, \mathfrak{I}(t))$ with $(s,t) \notin \mathfrak{R}$, but in that case $s$ will be bisimilar to some $s'$ satisfying the same characteristic formula and such that $(s', t) \in \mathfrak{R}$. Of course, the composition of the total bisimulation on $(S, R, V)$ with the refinement relation$\mathfrak{R}$ will also be a refinement relation. Without loss of generality we assume that $\mathfrak{R}$ is maximal in the sense that it is a fixed-point of composition with that total bisimulation. This makes the structure of the proof clearer.

We now show that the restricted modal product $((S'', R'', V''), (s, s'))$ resulting from executing the event model $((S', R', \mathsf{pre}), s')$ in the given epistemic state $(S, R, V), s)$ is bisimilar to its refinement $((S', R', V'), s')$. The bisimulation $\mathfrak{R}'$ is as follows: all pairs $(t, \mathfrak{I}(u))$ in the restricted modal product are bisimilar to the state $u \in S'$ of which their second argument of the pair is the isomorphic image:

$$\mathfrak{R}'(t, \mathfrak{I}(u)) = u$$

Condition **atoms** is obvious, as refinement satisfies **atoms**. Condition **forth** is also obvious: if $((t, \mathfrak{I}(u)), (t', \mathfrak{I}(u'))) \in R''_a$, then by definition of the modal product $(\mathfrak{I}(u), \mathfrak{I}(u')) \in R'_a$, so $(u, u') \in R'_a$ (and, indeed, $((t', \mathfrak{I}(u')), u') \in \mathfrak{R}'$ by definition). Condition **back** is not obvious but also holds. Let $(u, u') \in R'_a$ and $((t, \mathfrak{I}(u)), u) \in \mathfrak{R}'$. There must be a $t' \in S$ (modulo bisimilarity) such that $(t', u') \in \mathfrak{R}$ so that $(t', \mathfrak{I}(u'))$ is in the modal product. We now have that from $(u, u') \in R'_a$ follows $(\mathfrak{I}(u), \mathfrak{I}(u')) \in R'_a$, and we also have that from $(u, u') \in R'_a$ follows $(t, t') \in R_a$ (as $\mathfrak{R}$ is a refinement). From $(t, t') \in R_a$ and $(\mathfrak{I}(u), \mathfrak{I}(u')) \in R'_a$ follows by definition the requested $(t, \mathfrak{I}(u)), (t', \mathfrak{I}(u')) \in R''_a$.

**Tim's addition** To remove the reliance on common knowledge in the previous lemma, we give the following short technical result:

**Lemma 9** For every finite multi-agent epistemic ($S5_m$) model $M = (S, R, V)$, for every world $t \in S$ we can find a formula $\varphi_t$ such that for all $u \in S$, $M, u \models \varphi_t$ if and only if $M, u$ is bisimilar to $M, t$.

**Proof** To show this it is sufficient to show that the relation

$$\mathfrak{R} = \{(v, w) |\ M, v \text{ and } M, w \text{ agree on the interpretation of all } S5_m \text{ formulae}\},$$

is a bisimulation. Clearly it satisfies **atoms**. To show that it satisfies **forth**, suppose $(v, w) \in \mathfrak{R}$ nd let $i \leq m$, and let $v^*$ be an $i$-successor of the world $v$. We note that the world $w$ has finitely many $i$-successors, $w_1, \ldots, w_k$. Suppose for contradiction that none of these worlds were related to $v^*$ by $\mathfrak{R}$. Thus for each such world $w_i$, there is a formula $\tau_i$ such that $M, v^* \not\models \tau_i$ and $M, w \models \tau_i$. However then we have that $M, v \models \neg K_i \left(\bigvee_{i=1}^{k} \tau_i\right)$, but $M, w \models K_i \left(\bigvee_{i=1}^{k} \tau_i\right)$, contradicting the fact that $(v, w) \in \mathfrak{R}$. Therefore, for every $i \leq m$, every $i$ successor of $v$ is related by $\mathfrak{R}$ to some $i$-successor of $w$. The property **back** can be shown symmetrically, so $\mathfrak{R}$ is a bisimulation.

The lemma now follows, since for every $u \in S$ where $M, u \not\!\underline{\leftrightarrow} M, t$, there is some formula $\delta_t^u$ such that $M, t \models \delta_t^u$ and $M, u \not\models \delta_t^u$. We define $\varphi_t = \bigwedge\{\delta_t^u \mid u \in S \text{ and } M, u \not\!\underline{\leftrightarrow} M, t\}$. Since bisimilar states agree on the interpretation of all $S5_m$ formulas, for all $u \in S$, $M, u \models \varphi_t$ if and only if $M, u \underline{\leftrightarrow} M, t$.

We note that this proof is independent of the logic $S5_m$ and applies equally well to other modal logics. **end of Tim's addition**

We emphasize that the notion of event model relative to a language allows for *infinite* event models, unlike in a logic with an inductively defined language including (finite!) event models. That is to come next. This will also allow us to compare our proposal with a known method [2] for quantifying over events.

**Definition 10 ([2])** The language $\mathcal{L}_{aeml}$ of arbitrary event model logic is the language $\mathcal{L}_{fel}$ of future event logic with an additional inductive construct $[\mathsf{M}, \mathsf{s}]\varphi$.

We can view $[\mathsf{M}, \mathsf{s}]\varphi$ as an inductive construct, because, given the (enumerable) set of event model frames, $[\mathsf{M}, \mathsf{s}]$ can be seen as an operation on $|\mathcal{D}(\mathsf{M})|$ arguments of type formula (similar to automata-PDL). These arguments are the preconditions of the events in the event model. The language $\mathcal{L}_{aeml}$ can also be seen as extension with construct $G\varphi$ of the language $\mathcal{L}_{eml}$ for event model logic shown in the appendix.

To distinguish future event logic from logics with the same language but other semantics for $G\varphi$, we also write $\models_{\underline{\leftarrow}}$ instead of $\models$ for the forcing relation in future event logic; we (always) write $\models_{\otimes}$ for the forcing relation in arbitrary event model logic.

For the semantics of $G\varphi$ in terms of event models we need to restrict the preconditions of their events to $G$-free formulas, i.e. $\mathcal{L}_{eml}$ formulas. This is to avoid circularity in the definition, as $G\varphi$ could itself be a precondition of such an event. An event model is $G$-free iff all preconditions of its events are $G$-free.

**Definition 11 (Semantics of arbitrary event model logic)** Where the preconditions of events in any $\mathsf{M}$ are $G$-free.

$$M, s \models_{\otimes} G\varphi \quad \text{iff} \quad \text{for all } G\text{-free } (\mathsf{M}, \mathsf{s}) : M, s \models_{\otimes} [\mathsf{M}, \mathsf{s}]\varphi$$

There are refinements of epistemic models that cannot be seen as the result of executing an event model. This is because event models (in the language) have by definition a finite domain. For example, given a finite epistemic model $(M, s)$, consider its unwinding as an infinite tree (representing the bisimulation class). This is a refinement of $(M, s)$. But the result of executing a finite event model in a finite epistemic model cannot be an infinite tree. Of course, that tree is bisimilar to the initial epistemic state so can be seen in another sense as the result of execution the trivial event. But:

Because of the restriction to $G$-free preconditions in event models, we will still not get precise correspondence between the two semantics. The crux is that there are more epistemic distinctions in models than can be enumerated by epistemic formulas, see [2] for a similar matter. (However, we do not have a counterexample.)

Restricted to the class of finite epistemic models we still have that:

**Proposition 12** Let $M$ be finite. Then: $M, s \models_{\underline{\leftrightarrow}} \varphi$ iff $M, s \models_{\otimes} \varphi$.

**Proof** Directly from Propositions 8 and 7.

# 4 Bisimulation and information

Instead of validating $F\varphi$ in some $(M, s)$ by finding a refinement of $(M, s)$, we can equivalently find a *model restriction of a bisimilar epistemic state*. This alternative semantics $\models_{\underline{\leftrightarrow}}$ is interesting because of a relationship with bisimulation quantifiers [13], for which many theoretical results are known; and it is also interesting because it shows that every informative update is equivalent to public announcement of factual information 'of which the agents may not have been aware'.

**Definition 13** Below, $S'$ is the domain of $M'$, and $S''$ is such that $s \in S''$:

$$M, s \models_{\underline{\leftrightarrow}} G\varphi \quad \text{iff} \quad \text{for all } (M', s')\underline{\leftrightarrow}(M, s) \text{ and for all } S'' \subseteq S' : M'|S'', s' \models_{\underline{\leftrightarrow}} \varphi$$

On first thought it might seem that there are more refinements of a given model than domain restrictions of bisimilar models. In a refinement we can both restrict the domain (remove states) and remove links between states (delete pairs of the accessibility relation for an agent). But removing links between states can also be seen as a domain restriction on a even larger bisimilar model.

**Proposition 14** $M, s \models_{\underline{\leftarrow}} \varphi$ iff $M, s \models_{\underline{\leftrightarrow}} \varphi$.

**Proof** This can be shown by induction on the complexity of formulas. As $\models_{\underline{\leftarrow}}$ and $\models_{\underline{\leftrightarrow}}$ agree on the interpretations of atoms and all operators except $G$, it is sufficient to show that given ($M, s \models_{\underline{\leftarrow}} \varphi$ iff $M, s \models_{\underline{\leftrightarrow}} \varphi$) we have ($M, s \models_{\underline{\leftarrow}} G\varphi$ iff $M, s \models_{\underline{\leftrightarrow}} G\varphi$). From left to right the latter is trivial, because the refinements of $(M, s)$ include the bisimulations of $(M, s)$. For the direction from right to left, it suffices to show that any refinement $(M', s')$ of model $(M, s)$ is the restriction of a model $(M'', s'')$ that is bisimilar to $(M, s)$. This model is constructed as follows:

Let $M = (S, R, V)$, $M' = (S', R', V')$, and suppose that the refinement relation is $\mathfrak{R}$. Consider $(M'', s'') = ((S \oplus S', R'', V''), (s', 1))$, where for all agents $a \in A$

$$\begin{array}{lll}
((s,0),(t,0)) \in R''_a & \text{iff} & (s,t) \in R_a \\
((s',1),(t',1)) \in R''_a & \text{iff} & (s',t') \in R'_a \\
((s',1),(t,0)) \in R''_a & \text{iff} & \exists s \in S : (s,s') \in \mathfrak{R} \text{ and } (s,t) \in R_a
\end{array}$$

We can then define the relation $\mathfrak{R}'$ between $(M, s)$ and $(M'', (s, 0))$ as follows:

$$\begin{array}{lll}
(s,(s',1)) \in \mathfrak{R}' & \text{iff} & (s,s') \in \mathfrak{R} \\
(s,(s,0)) \in \mathfrak{R}' & \text{iff} & s \in S
\end{array}$$

This relation $\mathfrak{R}'$ is a bisimulation: it *still* satisfies **back** since the states of $S$ added to $M'$ also satisfy **back**: any relation between them copied their relation in the original $M$. But it now also satisfies **forth**:

If $(s,(s,0)) \in \mathfrak{R}'$ and $(s,t) \in R_a$ then by definition of the first clause of $\mathfrak{R}'$ we have $(t,(t,0)) \in \mathfrak{R}'$ and, trivially by the definition of $R''_a$ we have $((s,0),(t,0)) \in R''_a$. If $(s,(s',1)) \in \mathfrak{R}'$ and $(s,t) \in R_a$ then we have (as before) $(t,(t,0)) \in \mathfrak{R}'$ and $((s',1),(t,0)) \in R''_a$. The latter holds because of the *third* clause in the definition of $R''_a$.

Since $M''|(S' \times \{1\})$ is isomorphic to $M'$ this concludes the proof.

We proceed by explaining the stated relation of this semantics with bisimulation quantifiers.

## 5 Bisimulation quantifiers

Suppose that apart from the atoms in $P$ we had an additional, reserved, atom $r$. The future temporal operator can be seen as (existential) bisimulation quantification over $r$. (See the appendix—withheld from this abstract—for bisimulation quantifier semantics.) This relation becomes clear if we consider the restricted bisimulation version of the semantics for $F$:

> First choose a bisimilar epistemic state, then do a model restriction in that epistemic state that contains the actual state.

Given the class of models also valuing $r$ we can replace this by

> First choose a $P$-bisimilar epistemic state (but where the valuation of $r$ may vary wildly), then do a model restriction in that epistemic state that contains the actual state.

Of course we can match the variation in the valuation of $r$, as long as it contains the actual state, with that model restriction so we get

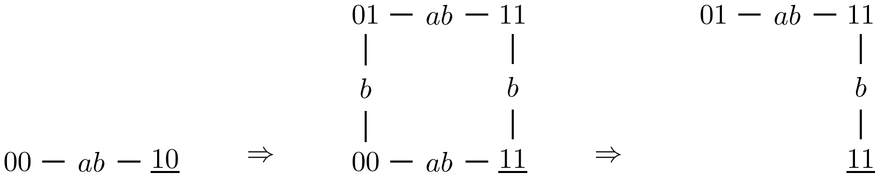
> First choose a $P$-bisimilar epistemic state, then do a model restriction to the $r$-states in that epistemic state, on condition that it contains the actual state.

The part 'choose a $P$-bisimilar epistemic state' of this informal description is the semantics of a existential bisimulation quantification.

**Definition 15** Where $V'$ is the valuation of $M'$.

$$M, s \models_{\forall r} G\varphi \quad \text{iff} \quad \text{for all } (M', s') \underline{\leftrightarrow}_P (M, s) : s' \in V'(r) \text{ implies } M'|r, s' \models_{\forall r} \varphi$$

**Example 16** For an example, consider again the model with common uncertainty about the value of an atom $p$ for agents $a$ and $b$, where $p$ is true. We now operate on models that also value the atom $r$, in the figure this is the value of the second digit: note that $r$ is *not* part of the logical language! Given the bisimulation quantification, the initial value of $r$ does not matter. In this model the formula $F(K_a p \wedge \neg K_a K_b p)$ is true. The first transition is to a model that is bisimilar with respect to $p$ only. The second transition is a restriction to the states where $r$ is true.



**Proposition 17** $M, s \models_{\underline{\leftrightarrow}} \varphi$ iff $M, s \models_{\forall r} \varphi$

**Corollary 18** On finite models and given common knowledge in the language, the four different semantics for $G$ correspond. (I.e. $\models_{\underline{\leftrightarrow}}$, $\models_{\underline{\leftarrow}}$, $\models_{\otimes}$, and $\models_{\forall r}$.)

Note that the extra atom $r$ does not disturb these results. As yet it is mere surplus luggage that we're carrying along towards the next section where it will become more meaningful. Our fourth perspective of bisimulation quantifier semantics is useful for theoretical and for practical reasons. A theoretical consequence is that

**Proposition 19** Future event logic is decidable.

**Proof** Consider some $\varphi \in \mathcal{L}_{fel}$. Replace all occurrences of $G$ in $\varphi$ by $\forall r[r]$. It is decidable whether $\varphi^{\forall r}$ is satisfiable. (The decidability of bisimulation quantified modal logics can be generalized to multi-agent logics. Note that it also holds for specific model classes such as $\mathcal{KD}45$, $\mathcal{S}5$ and the modal $\mu$-calculus; see [10].)

This is a useful result. If we add dynamic event model operators to future event logic (the language $\mathcal{L}_{aeml}$) we obtain arbitrary event model logic (see Definition 10). The restriction of this arbitrary event model logic to events that are public announcements is the logic $APAL$ investigated in [2]. For that logic, the satisfiability problem is undecidable (see [6]). That result also motivated this current investigation, because it promised more decidable logics.

However, we may note that the translation given (replace all occurrences of $G$ in $\varphi$ by $\forall r[r]$) is an accurate translation for *all* logics that are closed under bisimulation quantifiers and announcement. From a recent result of van Benthem and Ikegami [10] we know that the modal $\mu$-calculus is also closed under products with event models. Since future event logic and arbitrary event model logic agree on the interpretation of $G\varphi$ over finite models (Proposition 12), we can conclude that the satisfiability problem for $\mathcal{L}_{aeml}$ restricted to finite models is reducible to the satisfiability problem for the $\mu$-calculus, and hence decidable.

Note that the $G$-operator in arbitrary event model logic is interpreted differently (see Definition 11), and it is unknown whether this logic is decidable.

Our current perspective also provides us with additional modelling insight, namely that every informative update corresponds to the public announcement of an atomic fact. Kind of. What kind of? So far, it is unclear how to interpret this new perspective: we compare semantics with respect to model classes for different sets of atomic propositions; we did not add the fresh atom $r$ to the logical language $\mathcal{L}_{fel}$. Here is where some trouble seems to start. If we merely add $r$ as a formula to the language, but, e.g., rule out $K_a r$, we cannot truly interpret a $r$-restriction of a model as a public announcement: what use is an announcement of $r$ if we cannot express that an agent $a$ knows $r$ after its announcement? But if we add $r$ as just another propositional variable to the base clause of our inductive language definition, we run into trouble of a different kind: an existential bisimulation quantification means that the value of $p$ is scrambled. Even with the restriction that the value of $r$ remains unchanged in the actual state, we may now still have that an agent $a$ knew $r$ before an event, but has forgotten it afterwards, or vice versa. This is highly undesirable!

**Example 20** In the previous example, we have that initially agent $b$ knows that $r$ is false: $K_b \neg r$, but after the update he apparently has forgotten that: $\neg K_b \neg r$. For another example: $K_a r \rightarrow F \neg K_a r$ would be a validity.

A technical solution to this dilemma, that at least makes the public announcement clear, is to

> replace all occurrences of $G$ in formulas by occurrences of $\forall r[r]$,

where $\forall r$ is universal bisimulation over $r$ and where $[r]$ stands for public announcement of $r$. Public announcement is a singleton event model, accessible to all agents, where there precondition of the event is the formuma between brackets, in this case: $r$. If we also allow $r$ as formula, we can now interpret formulas of form $[r]\varphi$ in the usual sense for such events.

For example, in our running example it is initially true that $\exists r\langle r\rangle(K_a p \wedge \neg K_a K_b p)$, as this is the translation of $F(K_a p \wedge \neg K_a K_b p)$. (For $\neg[\varphi]\neg\psi$ we write $\langle\varphi\rangle\psi$.)

But the real solution to this seeming dilemma is to consider an existential bisimulation as 'the agents become aware of an additional fact', about which uncertainty is possible. From a modelling point of view this means that, before the bisimulation operation, the value of $r$ should be 'no care', in other words, 'the agents are unaware of $r$', the bisimulation quantification itself then means 'the agents become aware of $r$'. This is now in the proper sense that we move to a bisimilar model except for atom $r$, and (unlike before!) without the restriction that $r$ should remain true in the actual state, because maybe it was false in the first place. And after that it should be possible for them to know that $r$, or know that $\neg r$: they are now aware of their uncertainty about $r$. Of course after that, there might be other facts the agents might become aware of. If we merely add $r$ to the base of the inductive language definition we cannot express this. We need one more step. That final step we will now set in the next section.

## 6 Becoming aware of factual information

First, we add more structure: For each epistemic model $M$, the set of atoms $P$ is the disjoint union of a set of *relevant facts* $P_r(M)$ and a set of *irrelevant facts* $P_i(M)$. The set of relevant facts is typically finite. Then, in a given model, the interpretation of formulas containing irrelevant facts is undefined, unless they are bound by a bisimulation quantifier: we can only interpret irrelevant facts *after* they have *become* relevant to the agents. The bisimulation quantifier 'makes a fact relevant': its interpretation involves removing it from the set of irrelevant facts and adding it to the set of relevant facts. The result of this is that the value of irrelevant facts in any model is now truly 'don't care' from the perspective of the agents. But they can still reason about the consequences of new facts after they were to become relevant, i.e., after the agents were to become aware of those facts.

**Definition 21** The language $\mathcal{L}_{qel}$ of quantified event logic is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid [\mathsf{M},\mathsf{s}]\varphi \mid \forall p\varphi$$

where $a \in A$ and $p \in P$.

For the dual $\exists p\varphi$, read, "(there exists a fact $p$ such that) after the agents have become aware of $p$, $\varphi$." We emphasize that by 'becoming aware of $p$' we do not mean 'learning that $p$ is true'. In the information state resulting from becoming aware of $p$, the agents may know that $p$ is true, or that $p$ is false, or have any epistemic uncertainty about its value, e.g., they may not know whether $p$ is true, or one agent may know but not another, etc.

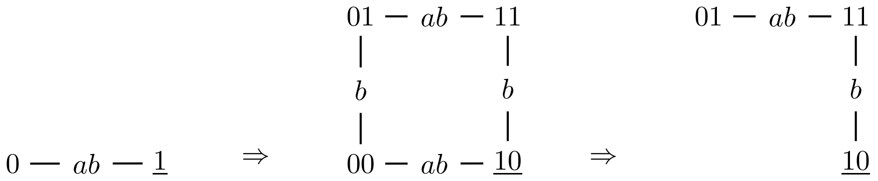
**Definition 22 (Semantics)** Assume an epistemic model $M = (S, R, V)$ for atoms $P = P_r(M) \cup P_i(M)$. The interpretation of $\varphi \in \mathcal{L}_{qel}$ is defined by induction. We only give the clauses for relevant atoms $p$ and for $\forall p\varphi$. The interpretation of irrelevant atoms is undefined. In the clause for $\forall p\varphi$ it is required that $(M', s')$ is such that $P_r(M') = P_r(M) + p$ and $P_i(M') = P_i(M) - p$.

$$\begin{array}{lll} M, s \models p & \text{iff} \quad s \in V_p & \text{where } p \in P_r(M) \\ M, s \models \forall p\varphi & \text{iff} \quad \text{for all } (M', s') \leftrightarroweq_{P-p} (M, s) : M', s' \models \varphi & \text{where } p \in P_i(M) \end{array}$$

We have not explored this version in greater detail yet. Unlike the logics with temporal operators and the proposal with a reserved atom $r$ for bisimulation quantification, agents may in this logic become aware of several different facts.

We think this logic may help modellers construct epistemic models in steps. In this logic, if we say that agent $a$ is uncertain about $p$, and we represent this in the two-state epistemic model, this now means that the agent *only* is uncertain about $p$. The value of other atoms in that epistemic state is 'don't care': information on an additional fact $q$ might become available later, we then 'simply' construct a $p$-but-not-$q$ bisimulation of this current epistemic state that represents the agents' current knowledge, that includes $q$. This is exactly the $\exists q$-operation! We close this section with a suitable illustration of this.

**Example 23** Initially the agents are only uncertain about $p$. Then, they become aware of $q$: in fact, $a$ knows the value of $q$ but $b$ doesn't. Finally, it is announced that $p \vee q$. In the resulting state, $a$ knows that $p$ but $b$ does not know that. Initially, the formula $\exists r \langle p \vee q \rangle (K_a p \wedge \neg K_b K_a p)$ is true. Observe that the bisimulation quantification is in this example different from the subsequent announcement. We now cannot announce the value of an atom, but only that of a more complex formula (well, a disjunction, but it could have been an epistemic formula as well).



# 7 Further research

We are currently investigating the axiomatization of these logics, their model checking complexities (relative to different model classes, such as $S5$), and expressivity issues.

**Acknowledgements:** The authors gratefully acknowledge the suggestions and insights from the anonymous LOFT referees. Hans van Ditmarsch acknowledges support of the Netherlands Institute of Advanced Study where he was Lorentz Fellow in 2008.

# References

[1] P. Aczel. *Non-Well-Founded Sets.* CSLI Publications, Stanford, CA, 1988. CSLI Lecture Notes 14.

[2] P. Balbiani, A. Baltag, H.P. van Ditmarsch, A. Herzig, T. Hoshi, and T. De Lima. What can we achieve by arbitrary announcements? A dynamic take on Fitch's knowability. In D. Samet, editor, *Proceedings of TARK XI*, Louvain-la-Neuve, Belgium, 2007. Presses Universitaires de Louvain.

[3] A. Baltag and L.S. Moss. Logics for epistemic programs. *Synthese*, 139:165–224, 2004. Knowledge, Rationality & Action 1–60.

[4] J. Barwise and L.S. Moss. *Vicious Circles.* CSLI Publications, Stanford, 1996.

[5] T. French. *Bisimulation quantifiers for modal logic.* PhD thesis, University of Western Australia, 2006.

[6] T. French and H.P. van Ditmarsch. Undecidability for arbitrary public announcement logic. To appear in Proceedings of Advances in Modal Logic 2008, Nancy, 2008.

[7] M. Hollenberg. *Logic and bisimulation.* PhD thesis, University of Utrecht, 1998.

[8] T. Hoshi. The logic of communication graphs for group communication and the dynamic epistemic logic with a future operator. Philosophy Department, Stanford University, 2006.

[9] J.F.A.K. van Benthem, J.D. Gerbrandy, and E. Pacuit. Merging frameworks for interaction: DEL and ETL. In D. Samet, editor, *Proceedings of TARK 2007*, pages 72–81, 2007.

[10] J.F.A.K. van Benthem and D. Ikegami. Modal fixed-point logic and changing models. In A. Avron, N. Dershowitz, and A. Rabinovich, editors, *Pillars of Computer Science*, volume 4800 of *Lecture Notes in Computer Science*, pages 146–165. Springer, 2008. Also available as ILLC Prepublication Series PP-2008-19.

[11] J.F.A.K. van Benthem, J. van Eijck, and B.P. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[12] H.P. van Ditmarsch, W. van der Hoek, and B.P. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library.* Springer, 2007.

[13] A. Visser. Bisimulations, model descriptions and propositional quantifiers, 1996. Logic Group Preprint Series 161, Department of Philosophy, Utrecht University.

## Appendix of technical terms

**Structural notions** Assume a finite set of agents $A$ and a countably infinite set of atoms $P$.

**Definition 24 (Structures)** An *epistemic model* $M = (S, R, V)$ consists of a *domain* $S$ of (factual) *states* (or 'worlds'), *accessibility* $R : A \rightarrow \mathcal{P}(S \times S)$, and a *valuation* $V : P \rightarrow \mathcal{P}(S)$. For $s \in S$, $(M, s)$ is an *epistemic state* (also known as a pointed Kripke model).

For $R(a)$ we write $R_a$; accessibility $R$ can be seen as a set of relations $R_a$, and $V$ as a set of valuations $V(p)$. Given two states $s, s'$ in the domain, $R_a(s, s')$ means that in state $s$ agent $a$ considers $s'$ a possibility. We adopt the standard rules for omission of parentheses in formulas, and we also delete them in representations of structures such as $(M, s)$ whenever convenient and unambiguous. (For $B \subseteq A$, write $R(B)$ (or $R_B$) for $(\bigcup_{a \in A} R(a))^*$. This accessibility relation is used to interpret common knowledge among agents in $B$, except in the reference to characteristic formulas not otherwise used in this contribution: $M, s \models C_B\varphi$ iff for all $t$: $(s, t) \in R_B$ implies $M, t \models \varphi$.)

**Definition 25 (Bisimulation, simulation, refinement)** Let two models $M = (S, R, V)$ and $M' = (S', R', V')$ be given. A non-empty relation $\mathfrak{R} \subseteq S \times S'$ is a bisimulation, iff for all $s \in S$ and $s' \in S'$ with $(s, s') \in \mathfrak{R}$:

**atoms** $s \in V(p)$ iff $s' \in V'(p)$ for all $p \in P'$

**forth** for all $a \in A$ and all $t \in S$, if $R_a(s,t)$, then there is a $t' \in S'$ such that $R_a(s',t')$ and $(t,t') \in \mathfrak{R}$

**back** for all $a \in A$ and all $t' \in S'$, if $R_a(s',t')$, then there is a $t \in S$ such that $R_a(s,t)$ and $(t,t') \in \mathfrak{R}$

We write $(M,s)\underline{\leftrightarrow}(M',s')$, iff there is a bisimulation restricted to $P'$ between $M$ and $M'$ linking $s$ and $s'$. Then we call $(M,s)$ and $(M',s')$ bisimilar. We also say that $(M,s)$ is similar to $(M',s')$ and vice versa.

A relation that satisfies **atoms** and **forth** is a *simulation*, and in that case $(M',s')$ is a *simulation* of $(M,s)$, and $(M,s)$ is a *refinement* of $(M',s)$, and we write $(M,s)\underline{\rightarrow}(M',s')$ (or $(M',s')\underline{\leftarrow}(M,s)$).

A bisimulation (simulation) that satisfies atoms for a subset $P' \subseteq P$ is a $P'$-bisimulation ($P'$-simulation); we write $(M,s)\underline{\leftrightarrow}_{P'}(M',s')$ ($(M,s)\underline{\rightarrow}_{P'}(M',s')$), etc.

**Standard language notions** The languages of propositional logic ($\mathcal{L}_{pl}$) and of epistemic logic ($\mathcal{L}_{el}$) — $a \in A$, $p \in P$, $B \subseteq A$.

$$\mathcal{L}_{pl}: \qquad \varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi)$$

$$\mathcal{L}_{el}: \qquad \varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi$$

$$\mathcal{L}_{el}^{C}: \qquad \varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid C_B\varphi$$

Standard abbreviations include: $\varphi \vee \psi$ iff $\neg(\neg\varphi \wedge \neg\psi)$; $\varphi \rightarrow \psi$ iff $\neg\varphi \vee \psi$, $\hat{K}_a\varphi$ iff $\neg K_a \neg\varphi$.

**Event model logic** All the following are simultaneously defined:

**Definition 26** Language $\mathcal{L}_{eml}$ of event model logic:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a\varphi \mid [\mathsf{M},\mathsf{s}]\varphi$$

**Definition 27 (Event model)** An *event model* for a finite set of agents $A$ and a language $\mathcal{L}$ is a triple $\mathsf{M} = (\mathsf{S}, \mathsf{R}, \mathsf{pre})$ where

- *domain* $\mathsf{S}$ is a finite non-empty set of events,
- $\mathsf{R} : A \rightarrow \mathcal{P}(\mathsf{S} \times \mathsf{S})$ assigns an *accessibility relation* to each agent,
- $\mathsf{pre} : \mathsf{S} \rightarrow \mathcal{L}$ assigns to each event a *precondition*,

A pair $(\mathsf{M},\mathsf{s})$ with a distinguished actual event $\mathsf{s} \in \mathsf{S}$ is called an *epistemic event*. An epistemic event with a singleton domain, accessible to all agents, and identity postcondition, is a *public announcement*.

**Definition 28 (Semantics of event model logic)** Let a model $(M,s)$ with $M = (S,R,V)$ be given. Let $a \in A$, $B \subseteq A$, and $\varphi, \psi \in \mathcal{L}$.

$$(M,s) \models [\mathsf{M},\mathsf{s}]\varphi \quad \text{iff} \quad (M,s) \models \mathsf{pre}(\mathsf{s}) \text{ implies } (M \otimes \mathsf{M}, (s,\mathsf{s})) \models \varphi$$

**Definition 29 (Execution of an event model)** Given are an epistemic model $M = (S, R, V)$, a state $s \in S$, an event model $\mathsf{M} = (\mathsf{S}, \mathsf{R}, \mathsf{pre})$, and an event $\mathsf{s} \in \mathsf{S}$ with $(M, s) \models \mathsf{pre}(\mathsf{s})$. The result of executing $(\mathsf{M}, \mathsf{s})$ in $(M, s)$ is the model $(M \otimes \mathsf{M}, (s, \mathsf{s})) = ((S', R', V'), (s, \mathsf{s}))$ where

- $S' = \{(t, \mathsf{t}) \mid (M, t) \models \mathsf{pre}(\mathsf{t})\}$,
- $R'(a) = \{((t, \mathsf{t}), (u, \mathsf{u})) \mid (t, \mathsf{t}), (u, \mathsf{u}) \in S'$ and $(t, u) \in R(a)$ and $(\mathsf{t}, \mathsf{u}) \in \mathsf{R}(a)\}$,
- $V'(p) = \{(t, \mathsf{t}) \mid (M, t) \models p\}$.

**Bisimulation quantifiers and bisimulation quantified epistemic logic** The language and semantics are as follows.

**Definition 30 (Bisimulation quantified epistemic logic)** Bisimulation quantified epistemic logic augments epistemic logic by additionally allowing formulas of the kind $\forall p \varphi$ in the recursive definition, where $p$ is an atom of $P$, and $\varphi$ is a formula. This is the language $\mathcal{L}_{bqel}$.

Given an epistemic model $M = (S, R, V)$ and a state $s \in S$ we say:

$$M, s \models \forall p \varphi \quad \text{iff} \quad \text{for every epistemic model } (M', s') \leftrightarroows_{P \setminus \{p\}} (M, s) : M', s' \models \varphi.$$

# What do we accept after an announcement?

Andreas Herzig[1], Tiago de Lima[2], and Emiliano Lorini[1]

[1] IRIT, Toulouse, France
Andreas.Herzig@irit.fr
Emiliano.Lorini@irit.fr
[2] Eindhoven University of Technology, The Netherlands
T.d.Lima@tue.nl

**Abstract.** In this work we continue the work initiated in [1], in which a logic of individual and collective acceptance was introduced. Our aim in this paper is to investigate the extension of the logic of acceptance by *public announcements* of formulas. The function of public announcements is to diminish the space of possible worlds accepted by agents and sets of agents while functioning as members of a given group, team, organization, institution, etc., $x$. If a set of agents $C$ ends up with an empty set of worlds that they accept while functioning as members of $x$, then the agents in $C$ do not identify themselves any longer with $x$. In such a situation the agents in $C$ should have the possibility to join $x$ again.
To that aim we discuss at the end of the paper an operation which consists of an agent (or set of agents) joining a given group, team, organization, institution, etc.

## 1 Introduction

The concept of *collective acceptance* has been studied in social philosophy in opposition to group attitudes such as *common belief* and *common knowledge* that are popular in artificial intelligence and theoretical computer science [2, 3]. As suggested in [4], the main difference between collective acceptance and common belief (or common knowledge) is that a collective acceptance by a set of agents $C$ is based on the fact that the agents in $C$ identify and recognize themselves as members of the same *social context*, such as a group, team, organization, institution, etc. Common belief (and common knowledge) does not necessarily entail this aspect of mutual recognition and identification with respect to a social context. In this sense, according to [4, 5], collective acceptance rather than common belief is the more appropriate concept to characterize a proper notion of *group belief*. For example, in the context of the organization Greenpeace the agents in a set $C$ (collectively) accept that their mission is to protect the Earth *qua* members of Greenpeace. The state of acceptance *qua* members of Greenpeace is the kind of acceptance the agents in $C$ are committed to when they are functioning together as members of Greenpeace.

It has been emphasized that a similar distinction between acceptance and belief exists at the *individual* level. While an agent's belief that $p$ is an attitude of the agent constitutively aimed at the truth of $p$, an agent's acceptance is not necessarily connected to the actual truth of the proposition. In order to better distinguish these two notions, it has been suggested in [6] that while an agent's beliefs are not subject to the agent's will, its acceptances are voluntary; while its beliefs aim at truth, its acceptances are sensitive to pragmatic considerations; while its beliefs are shaped by evidence, its acceptances need not be; finally, while its beliefs are context-independent, its acceptances might depend on context. Often the acceptances of an agent depend on social contexts, that is, while identifying itself as a member of a group (or team, organization, institution, etc.) an agent reasons and accepts things *qua* member of this group. In these situations it may happen that the agent's acceptances are in conflict with its beliefs. For instance, a lawyer who is trying to defend a client in a murder case accepts *qua* lawyer that the client is innocent, even he believes the contrary.

The aim of this paper is to continue the work initiated in [1, 7]. There, a logic of individual and collective acceptance was introduced.[3] One of the notable features of that logic is that the accessibility relation associated to the acceptance operator is not necessarily serial: an empty set of possible worlds associated to a group $C$ in a context $x$ just means that $C$ does not identify itself with $x$.

Our aim here is to investigate the extension of the logic of acceptance by *public announcements* of formulas, noted $x!\psi$. Modal operators of type $[x!\varphi]$ are intended to express that the members of a certain group, team, organization, institution, etc., $x$ learn that $\varphi$ is true in that institution in such a way that their acceptances, *qua* members of $x$, are updated. The function of public announcements is to diminish the space of possible worlds accepted by agents and groups of agents. It might also happen that a given set of agents $C$ ends up with an empty set of possible worlds that they accept while functioning as members of a certain social context $x$. As we have said, this means that $C$ quits $x$: the agents in $C$ do not identify themselves any longer with $x$. In such a situation $C$ should have the possibility to join $x$ again. To that aim we discuss at the end of the paper an operation which consists of an agent (or set of agents) joining a given social context $x$.

The main contribution of this paper is to extend the logic presented in [1] to public announcements and show that, differently from common belief and common knowledge, reduction axioms can be given. As usual, the addition of these axioms to the Hilbert axiomatics of the logic of acceptance provides a complete axiomatization of the logic of acceptance and announcements.

The paper is organized as follows. In Section 2 we present the syntax and semantics of acceptance logic together with its axiomatization. In Section 3 we extend it with announcements, and show that our extension also allows for reduction axioms and thereby a complete axiomatization. In Section 4 we formalize an example which illustrates the dynamics of acceptance based on announcements. In Section 5 we briefly discuss the operation which consists of an agent (or set of agents) joining a social context. This section is not intended to provide a solution

[3] This logic has some similarities with the logic of *group belief* that we have developed in [8, 9].

to the logical characterization of this social phenomenon though. In Section 6 we draw conclusions.

## 2 The Logic of Acceptance $\mathcal{AL}$

We now present a variant of the *Acceptance Logic* ($\mathcal{AL}$) that was introduced in [1]. $\mathcal{AL}$ enables expressing that certain agents identify themselves as members of a social context $x$ and, reasoning about what agents and groups of agents accept while functioning together as members of a certain social context. The axioms of $\mathcal{AL}$ clarify the relationships between individual acceptance (acceptances of individual agents) and collective acceptance (acceptances of groups of agents).

### 2.1 Syntax

The syntactic primitives of $\mathcal{AL}$ are the following: a finite non-empty set of agents $AGT$; a countable set of atomic formulas $ATM$; and a finite set of labels $CTXT$ denoting social contexts such as groups, teams, organizations, institutions, etc. The language $\mathcal{L}_{\mathcal{AL}}$ of the logic $\mathcal{AL}$ is given by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathcal{A}_{C:x}\varphi$$

where $p$ ranges over $ATM$, $C$ ranges over $2^{AGT}$, and $x$ ranges over $CTXT$. The formula $\mathcal{A}_{C:x}\varphi$ reads "the agents in $C$ accept that $\varphi$ while functioning together as members of $x$". We write $i{:}x$ instead of $\{i\}:x$.

For example, $\mathcal{A}_{C:Greenpeace}$*protectEarth* expresses that the agents in $C$ accept that the mission of Greenpeace is to protect the Earth while functioning as activists in the context of Greenpeace; and $\mathcal{A}_{i:Catholic}$*PopeInfallibility* expresses that agent $i$ accepts that the Pope is infallible while functioning as a Catholic in the context of the Catholic Church.

The intuition is that in two different contexts the same agent may accept contradictory propositions. For example, while functioning as a Catholic, agent $i$ accepts that killing is forbidden, and while functioning as a soldier $i$ accepts that killing is allowed. The CEO of Airbus accepts that Airbus is in good health while functioning as a member of Airbus Industries, and privately accepts the contrary.

The classical boolean connectives $\wedge$, $\rightarrow$, $\leftrightarrow$, $\top$ (tautology) and $\bot$ (contradiction) are defined from $\vee$ and $\neg$ in the usual manner.

The formula $\mathcal{A}_{C:x}\bot$ has to be read "agents in $C$ are not functioning together as members of $x$", because we assume that functioning as a group member is, at least in this minimal sense, a rational activity. Conversely, $\neg\mathcal{A}_{C:x}\bot$ has to be read "agents in $C$ are functioning together as members of $x$". Thus, $\neg\mathcal{A}_{C:x}\bot \wedge \mathcal{A}_{C:x}\varphi$ stands for "agents in $C$ are functioning together as members of $x$ and they accept that $\varphi$ while functioning together as members of $x$" or simply "agents in $C$ accept that $\varphi$ *qua* members of $x$". This is a case of *group acceptance.* For the individual case, formula $\neg\mathcal{A}_{i:x}\bot \wedge \mathcal{A}_{i:x}\varphi$ has to be read "agent $i$ accepts that $\varphi$ *qua* member of $x$". This is a case of *individual acceptance.*

### 2.2 Semantics and Axiomatization

We use a standard possible worlds semantics. Let the set of all couples of non-empty subsets of agents and social contexts be

$$\Delta = \big\{ C{:}x \ : \ C \in 2^{AGT} \text{ and } x \in CTXT \big\}.$$

An *acceptance model* is a triple $\mathcal{M} = \langle W, \mathscr{A}, \mathscr{V} \rangle$ where:

- $W$ is a non-empty set of possible worlds;
- $\mathscr{A} : \Delta \to W \times W$ maps every $C{:}x \in \Delta$ to a relation $\mathscr{A}_{C:x}$ between possible worlds in $W$; and
- $\mathscr{V} : ATM \to 2^W$ is valuation function associating a set of possible worlds $\mathscr{V}(p) \subseteq W$ to each atomic formula $p$ of $ATM$.

We write $\mathscr{A}_{C:x}(w)$ for the set $\{w' : \langle w, w' \rangle \in \mathscr{A}_{C:x}\}$. $\mathscr{A}_{C:x}(w)$ is the set of worlds that is accepted by the agents in $C$ while functioning together as members of $x$.

Given $\mathcal{M} = \langle W, \mathscr{A}, \mathscr{V} \rangle$ and $w \in W$, the couple $\langle \mathcal{M}, w \rangle$ is a *pointed acceptance model.* The satisfaction relation $\models$ between formulas of $\mathcal{L}_{\mathcal{AL}}$ and pointed acceptance models $\langle \mathcal{M}, w \rangle$ is defined as usual for atomic propositions, negation and disjunction. The satisfaction relation for acceptance operators is the following:

$$\mathcal{M}, w \models \mathcal{A}_{C:x}\varphi \quad \text{iff} \quad \mathcal{M}, w' \models \varphi \text{ for every } w' \in \mathscr{A}_{C:x}(w)$$

Validity of a formula $\varphi$ (noted $\models_{\mathcal{AL}} \varphi$) is defined as usual.

The axiomatization of $\mathcal{AL}$ is given in Fig. 1. As usual, the K-principles are the axioms and inference rules of the basic modal logic K.

| | |
|---|---|
| **(K)** | All K-principles for the operators $\mathcal{A}_{C:x}$ |
| **(4*)** | $\mathcal{A}_{C:x}\varphi \to \mathcal{A}_{B:y}\mathcal{A}_{C:x}\varphi$ if $B \subseteq C$ |
| **(5*)** | $\neg\mathcal{A}_{C:x}\varphi \to \mathcal{A}_{B:y}\neg\mathcal{A}_{C:x}\varphi$ if $B \subseteq C$ |
| **(Inc)** | $(\neg\mathcal{A}_{C:x}\bot \wedge \mathcal{A}_{C:x}\varphi) \to \mathcal{A}_{B:x}\varphi$ if $B \subseteq C$ |
| **(Unanim)** | $\mathcal{A}_{C:x}(\bigwedge_{i\in C} \mathcal{A}_{i:x}\varphi \to \varphi)$ |

**Fig. 1.** Axiomatization of $\mathcal{AL}$.

Axioms **4*** and **5*** are introspection axioms: when the agents in a set $C$ function together as members of $x$, then, for all $y \in CTXT$ and $B$ such that $B \subseteq C$, the agents in $B$ have access to all the facts that are accepted (or that are not accepted) by the agents in $C$. In particular, if the agents in $C$ (do not) accept that $\varphi$ while functioning together as members of $x$ then, while functioning together as members of $y$, the agents of every subset $B$ of $C$ accept that agents in $C$ (do not) accept that $\varphi$.

*Example 1.* Suppose that three agents $i, j, k$, while functioning together as members of the UK trade union, accept that their mission is to try to increase teachers' wages, but they do not accept *qua* members of the trade union that their mission is to try to increase railway workers' wages: $\mathcal{A}_{\{i,j,k\}:Union} IncreaseTeacherWage$

and $\neg\mathcal{A}_{\{i,j,k\}:Union} IncreaseRailwayWage$. By axiom **4*** we infer that, while functioning as a UK citizen, $i$ accepts that $i, j, k$ accept that their mission is to try to increase teachers' wages, while functioning together as members of the trade union: $\mathcal{A}_{i:UK}\mathcal{A}_{\{i,j,k\}:Union} IncreaseTeacherWage$. By Axiom **5*** we infer that, while functioning as a UK citizen, $i$ accepts that $i, j, k$ do not accept, *qua* members of the trade union, that their mission is to try to increase railway workers' wages: $\mathcal{A}_{i:UK}\neg\mathcal{A}_{\{i,j,k\}:Union} IncreaseRailwayWage$.

Axiom **Inc** says that, if the agents in $C$ accept that $\varphi$ *qua* members of $x$ then every subset $B$ of $C$ accepts $\varphi$ while functioning together as members of $x$. This means that things accepted by the agents in $C$ *qua* members of a certain social context $x$ are necessarily accepted by agents in all of $C$'s subsets with respect to the same context $x$. Axiom **Inc** describes the *top down* process leading from $C$'s collective acceptance to the individual acceptances of $C$'s members.[4]

*Example 2.* Imagine three agents $i$, $j$, $k$ that, *qua* players of the game Clue, accept that someone called Mrs. Red, has been killed:

$$\neg\mathcal{A}_{\{i,j,k\}:Clue}\bot \wedge \mathcal{A}_{\{i,j,k\}:Clue} killedMrsRed.$$

By axiom **Inc** we infer that also the two agents $i, j$, while functioning as Clue players, accept that someone called Mrs. Red has been killed:

$$\mathcal{A}_{\{i,j\}:Clue} killedMrsRed.$$

Axiom **Unanim** expresses a unanimity principle according to which the agents in $C$, while functioning together as members of $x$, accept that if each of them individually accepts that $\varphi$ while functioning as member of $x$, then $\varphi$ is the case. This axiom describes the *bottom up* process leading from individual acceptances of the members of $C$ to the collective acceptance of the group $C$.

In order to make our axioms valid we impose the following constraints on acceptance models, for any world $w \in W$, context $x, y \in CTXT$, and coalitions $C, B \in 2^{AGT}$ such that $B \subseteq C$:

**(S.1)** if $w' \in \mathscr{A}_{B:y}(w)$ then $\mathscr{A}_{C:x}(w') = \mathscr{A}_{C:x}(w)$;
**(S.2)** if $\mathscr{A}_{C:x}(w) \neq \emptyset$ then $\mathscr{A}_{B:x}(w) \subseteq \mathscr{A}_{C:x}(w)$;
**(S.3)** if $w' \in \mathscr{A}_{C:x}(w)$ then $w' \in \bigcup_{i\in C} \mathscr{A}_{i:x}(w')$.

Axioms **4*** and **5*** together correspond to the constraint **S.1**; axiom **Inc** corresponds to **S.2**, and axiom **Unanim** to **S.3** (in the sense of correspondence theory). As all our axioms are in the Sahlqvist class we obtain straightforwardly:

**Theorem 1.** *The axiomatization of $\mathcal{AL}$ of Fig. 1 is sound and complete w.r.t. the class of $\mathcal{AL}$ models satisfying constraints **S.1**, **S.2**, and **S.3**.*

[4] Note that the more general

$$(\neg\mathcal{A}_{C:x}\bot \wedge \mathcal{A}_{C:x}\varphi) \rightarrow \mathcal{A}_{B:y}\varphi \text{ if } B \subseteq C$$

would lead to unwanted consequences: the group of Catholics' acceptance *qua* members of the Catholic church that the Pope is infallible does not entail that Catholics privately accept that the Pope is infallible.

Also note that for $B \subseteq C$, neither $\mathcal{A}_{C:x}\bot \rightarrow \mathcal{A}_{B:x}\bot$ nor $\mathcal{A}_{B:x}\bot \rightarrow \mathcal{A}_{C:x}\bot$ should hold.

*Proof.* It is a routine task to check that all the axioms of the logic $\mathcal{AL}$ correspond to their semantic counterparts. It is routine, too, to check that all $\mathcal{AL}$ axioms are in the Sahlqvist class, for which a general completeness result exists [10].

*Example 3.* It follows from axioms **4***, **5*** and **Inc** that if $B \subseteq C$ then $\models_{\mathcal{AL}} \mathcal{A}_{B:y}\mathcal{A}_{C:x}\varphi \leftrightarrow \mathcal{A}_{B:y}\bot \vee \mathcal{A}_{C:x}\varphi$ and $\models_{\mathcal{AL}} \mathcal{A}_{B:y}\neg\mathcal{A}_{C:x}\varphi \leftrightarrow \mathcal{A}_{B:y}\bot \vee \neg\mathcal{A}_{C:x}\varphi$. We also have $\models_{\mathcal{AL}} \mathcal{A}_{C:x}(\mathcal{A}_{C:x}\varphi \rightarrow \varphi)$.

# 3 The Logic of Acceptance and Public Announcements $\mathcal{ALA}$

In its nature, acceptance comes by communication: if a group accepts that one of its members $i$ accepts that $\varphi$ then this is often the result of a speech act performed by $i$. Acceptance is therefore closely related to the notion of commitment that has been studied in agent communication languages [11–13].

In this paper we study the combination of acceptance logic $\mathcal{AL}$ with a rather simple communicative act, viz. public announcements as defined in public announcement logic ($\mathcal{PAL}$) [14]. Basically, when $\psi$ is publicly announced then all agents learn that $\psi$ is true. Our truth condition is that of Kooi [15], that is slightly different from the standard one in public announcement logic: it does not require announcements to be truthful.

## 3.1 Language and Models

The language $\mathcal{L}_{\mathcal{ALA}}$ of acceptance logic with announcements ($\mathcal{ALA}$) extends $\mathcal{L}_{\mathcal{AL}}$ by modal formulas of the form $[x!\psi]\varphi$. Such formulas are read "$\varphi$ holds after the public announcement of $\psi$ in context $x$". Modal operators of type $[x!\varphi]$ are intended to express that the members of a certain group, team, organization, institution, etc., $x$ learn that $\varphi$ is true in that institution in such a way that their acceptances, *qua* members of $x$, are updated.

The announcement $x!\mathcal{A}_{i:x}\psi$ is an *event.* It approximates $i$'s *action* of announcing that $\psi$ in context $x$. (This is an assertion in speech act theory and in Walton and Krabbe's dialogue games [16].)

It is worth noting that when $x$ denotes an institution, events of type $x!\psi$ can be used to describe the event of issuing or promulgating a certain norm $\psi$ (e.g. obligation, permission) within the context of the institution $x$.[5]

Formulas of $\mathcal{L}_{\mathcal{ALA}}$ are interpreted in pointed acceptance models. The satisfaction relation $\models$ of Section 2 is extended by the following clause:

$$\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\varphi \quad \text{iff} \quad \langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \varphi$$

with

[5] For a logical characterization of the act of *proclaiming* or *promulgating* a norm, see also [17].

- $\mathscr{A}^{x!\psi}_{C:y}(w) = \mathscr{A}_{C:y}(w)$, for all $C{:}y \in \Delta$, $w \in W$ and $y \neq x$;
- $\mathscr{A}^{x!\psi}_{C:y}(w) = \mathscr{A}_{C:y}(w) \cap ||\psi||_{\mathcal{M}}$, for all $C{:}y \in \Delta$, $w \in W$ and $y = x$;

where as usual $||\psi||_{\mathcal{M}} = \{w \ : \ \mathcal{M}, w \models \psi\}$ is the extension of $\psi$ in $\mathcal{M}$, i.e. the set of worlds where $\psi$ is true. Thus, in a way similar to [15], the agents take into account the announcement of $\psi$ in the social context $x$ and modify their acceptances *qua* members of $x$ by eliminating all arrows leading to $\neg\psi$ worlds (instead of eliminating the worlds themselves, as in $\mathcal{PAL}$). On the contrary, when $x$ and $y$ are different, the accessibility relations associated to the acceptances *qua* members of $y$ are not modified, after the announcement of $\psi$ in the social context $x$.

Validity of a formula $\varphi$ (noted $\models_{\mathcal{ALA}} \varphi$) is defined as before. For example, $\models_{\mathcal{ALA}} [x!p]\mathcal{A}_{C:x}p$, and $\models_{\mathcal{ALA}} \mathcal{A}_{C:x}\neg p \rightarrow [x!p]\mathcal{A}_{C:x}\bot$. The latter means that coalition $C$ quits all social contexts within which $C$'s acceptances are inconsistent with what is announced.

Note that contrarily to standard common knowledge and belief, the modified accessibility relations for acceptances are not computed from the modified accessibility relations for individuals, but are first-class citizens here: they are changed 'on their own'.

**Proposition 1.** *If $\mathcal{M}$ is an acceptance model then $\mathcal{M}^{x!\psi}$ is an acceptance model.*

*Proof.* We show that $\mathcal{M}^{x!\psi} = \langle W, \mathscr{A}^{x!\psi}, \mathscr{V} \rangle$ satisfies **S.1**, **S.2** and **S.3**. In what follows let $B \subseteq C$.

**(S.1):** Let $w_2 \in \mathscr{A}^{x!\psi}_{B:y}(w_1)$. If the latter is true then $w_2 \in \mathscr{A}_{B:y}(w_1)$, which implies $\mathscr{A}_{C:x}(w_2) = \mathscr{A}_{C:x}(w_1)$, because $\mathcal{M}$ respects **S.1**.
Now, we show that $\mathscr{A}^{x!\psi}_{C:x}(w_2) \subseteq \mathscr{A}^{x!\psi}_{C:x}(w_1)$. Consider a possible world $w_3 \in \mathscr{A}^{x!\psi}_{C:x}(w_2)$. This means that $w_3 \in \mathscr{A}_{C:x}(w_2) \cap ||\psi||_{\mathcal{M}}$. Then, in particular, $w_3 \in \mathscr{A}_{C:x}(w_2)$, which implies $w_3 \in \mathscr{A}^{x!\psi}_{C:x}(w_1)$, because $\mathscr{A}_{C:x}(w_2) = \mathscr{A}_{C:x}(w_1)$.
By using an analogous argument, we show that $\mathscr{A}^{x!\psi}_{C:x}(w_1) \subseteq \mathscr{A}^{x!\psi}_{C:x}(w_2)$.

**(S.2):** Let $\mathscr{A}^{x!\psi}_{C:x}(w_1) \neq \emptyset$ and $w_2 \in \mathscr{A}^{x!\psi}_{B:x}(w_1)$. We show that $w_2 \in \mathscr{A}^{x!\psi}_{C:x}(w_1)$. The hypothesis implies $w_2 \in \mathscr{A}_{B:x}(w_1) \cap ||\psi||_{\mathcal{M}}$. Then, in particular, $w_2 \in \mathscr{A}_{B:x}(w_1)$. Also note that the hypothesis implies $\mathscr{A}_{C:x}(w_1) \neq \emptyset$. Then, $w_2 \in \mathscr{A}_{C:x}(w_1)$, because $\mathcal{M}$ respects **S.2**. We conclude that $w_2 \in \mathscr{A}_{C:x}(w_1) \cap ||\psi||_{\mathcal{M}}$. The latter is true if and only if $w_2 \in \mathscr{A}^{x!\psi}_{C:x}(w_1)$.

**(S.3):** Let $w_2 \in \mathscr{A}^{x!\psi}_{C:x}(w_1)$. We show that $w_2 \in \mathscr{A}^{x!\psi}_{i:x}(w_2)$ for some $i \in C$. The hypothesis is equivalent to $w_2 \in \mathscr{A}_{C:x}(w_1) \cap ||\psi||_{\mathcal{M}}$. Then, in particular, $w_2 \in \mathscr{A}_{C:x}(w_1)$, which implies $w_2 \in \mathscr{A}_{i:x}(w_2)$ for some $i \in C$, because $\mathcal{M}$ respects **S.3**. Then, $w_2 \in \mathscr{A}_{i:x}(w_2) \cap ||\psi||_{\mathcal{M}}$ for some $i \in C$. The latter is true if and only if $w_2 \in \mathscr{A}^{x!\psi}_{i:x}(w_2)$ for some $i \in C$.

□

### 3.2 Reduction Axioms and Completeness

Just as in dynamic epistemic logics without common belief, $\mathcal{ALA}$ has reduction axioms for all cases (invididual and collective acceptance). This contrasts with logics having the common belief operator, for which such axioms do not exist [18].

**Proposition 2.** *The following equivalences are $\mathcal{ALA}$ valid.*

**(R.1)** $[x!\psi]p \leftrightarrow p$
**(R.2)** $[x!\psi]\neg\varphi \leftrightarrow \neg[x!\psi]\varphi$
**(R.3)** $[x!\psi](\varphi_1 \wedge \varphi_2) \leftrightarrow [x!\psi]\varphi_1 \wedge [\psi!\varphi]_2$
**(R.4)** $[x!\psi]\mathcal{A}_{C:y}\varphi \leftrightarrow \mathcal{A}_{C:y}[x!\psi]\varphi$ *(if $y \neq x$)*
**(R.5)** $[x!\psi]\mathcal{A}_{C:y}\varphi \leftrightarrow \mathcal{A}_{C:y}(\psi \to [x!\psi]\varphi)$ *(if $y = x$)*

*Proof.* **(R.1):**
$\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]p$
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle \models p$
iff $w \in \mathscr{V}(p)$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models p$.

**(R.2):**
$\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\neg\varphi$
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \neg\varphi$
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \not\models \varphi$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \not\models [x!\psi]\varphi$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models \neg[x!\psi]\varphi$.

**(R.3):**
$\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi](\varphi_1 \wedge \varphi_2)$
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \varphi_1 \wedge \varphi_2$
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \varphi_1$ and $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \varphi_2$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\varphi_1$ and $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\varphi_2$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\varphi_1 \wedge [x!\psi]\varphi_2$.

**(R.4):** We show that the equivalent $\neg[x!\psi]\mathcal{A}_{C:y}\varphi \leftrightarrow \neg\mathcal{A}_{C:y}[x!\psi]\varphi$ is valid.
$\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models \neg[x!\psi]\mathcal{A}_{C:y}\varphi$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\neg\mathcal{A}_{C:y}\varphi$, by **(R.3)**,
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \neg\mathcal{A}_{C:y}\varphi$
iff there is $w' \in \mathscr{A}^{x!\psi}_{C:y}(w)$ such that $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w' \models \neg\varphi$
iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle \models \neg\varphi$, because $y \neq x$,
iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}, \mathscr{V}\rangle \models [x!\psi]\neg\varphi$
iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \neg[x!\psi]\varphi$, by **(R.3)**,
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \neg\mathcal{A}_{C:y}[x!\psi]\varphi$.

**(R.5):** We show that $\neg[x!\psi]\mathcal{A}_{C:y}\varphi \leftrightarrow \neg\mathcal{A}_{C:y}(\psi \to [x!\psi]\varphi)$ is valid.
$\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models \neg[x!\psi]\mathcal{A}_{C:y}\varphi$
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle, w \models [x!\psi]\neg\mathcal{A}_{C:y}\varphi$, by **(R.3)**,
iff $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w \models \neg\mathcal{A}_{C:y}\varphi$
iff there is $w' \in \mathscr{A}^{x!\psi}_{C:y}(w)$ such that $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle, w' \models \neg\varphi$
iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \psi$ and $\langle W, \mathscr{A}^{x!\psi}, \mathscr{V}\rangle \models \neg\varphi$, because $y = x$,
iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \psi$ and $\langle W, \mathscr{A}, \mathscr{V}\rangle \models [x!\psi]\neg\varphi$

iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \psi \wedge [x!\psi]\neg\varphi$
iff there is $w' \in \mathscr{A}_{C:y}(w)$ such that $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \psi \wedge \neg[x!\psi]\varphi$, by **(R.3)**,
iff $\langle W, \mathscr{A}, \mathscr{V}\rangle \models \neg\mathcal{A}_{C:y}(\psi \to [x!\psi]\varphi)$.

□

These equivalences are called reduction axioms because they allow to rewrite every formula by successively eliminating the announcement operators, ending up with a formula that contains none.

**Theorem 2.** *For every $\mathcal{ALA}$ formula there is an equivalent $\mathcal{AL}$ formula.*

*Proof.* The proof goes just as for public announcement logic: each of the above $\mathcal{ALA}$ valid equivalences **R.2**-**R.5**, when applied from the left to the right, yields a simpler formula, where 'simpler' roughly speaking means that the announcement operator is pushed inwards. Once the announcement operator attains an atom it is eliminated by the first equivalence **R.1**. □

**Theorem 3.** *The formulas that are valid in $\mathcal{ALA}$ models are completely axiomatized by the axioms and inference rules of $\mathcal{AL}$ together with the reduction axioms of Proposition 2.*

*Proof.* This a straightforward consequence of Theorem 1 and Theorem 2.

Here are some examples of reductions.

*Example 4.* The formula $[x!p]\mathcal{A}_{C:x}p$ is successively rewritten as follows:
$[x!p]\mathcal{A}_{C:x}p$
$\mathcal{A}_{C:x}(p \to [x!p])$ by **R.5**
$\mathcal{A}_{C:x}(p \to p)$ by **R.1**
The latter is a theorem of every normal modal logic (and therefore also of acceptance logic $\mathcal{AL}$). It follows that the initial formula is valid, too.

*Example 5.* The formula $\mathcal{A}_{i:x}\neg p \to [x!p]\mathcal{A}_{i:x}\bot$ is rewritten as follows:
$\mathcal{A}_{i:x}\neg p \to [x!p]\mathcal{A}_{i:x}\bot$
$\mathcal{A}_{i:x}\neg p \to \mathcal{A}_{i:x}(p \to [x!p]\bot)$ by **R.5**
$\mathcal{A}_{i:x}\neg p \to \mathcal{A}_{i:x}(p \to \bot)$ by **R.1**
The latter is a theorem of every normal modal logic (and therefore also of acceptance logic $\mathcal{AL}$).

*Example 6.* The formula $[x!(\mathcal{A}_{i:x}p)]\mathcal{A}_{C:x}\mathcal{A}_{i:x}p$ is rewritten as follows:
$[x!(\mathcal{A}_{i:x}p)]\mathcal{A}_{C:x}\mathcal{A}_{i:x}p$
$\mathcal{A}_{C:x}(\mathcal{A}_{i:x}p \to [x!(\mathcal{A}_{i:x}p)]\mathcal{A}_{i:x}p)$ by **R.5**
$\mathcal{A}_{C:x}(\mathcal{A}_{i:x}p \to \mathcal{A}_{i:x}(\mathcal{A}_{i:x}p \to [x!(\mathcal{A}_{i:x}p)]p))$ by **R.5**
$\mathcal{A}_{C:x}(\mathcal{A}_{i:x}p \to \mathcal{A}_{i:x}(\mathcal{A}_{i:x}p \to p))$ by **R.1**
The latter is an $\mathcal{AL}$ theorem (because $\mathcal{A}_{i:x}(\mathcal{A}_{i:x}p \to p)$ is an $\mathcal{AL}$ theorem, cf. Example 3 of Section 2). It follows that the initial formula is valid, too.

### 3.3 Discussion

The reduction axiom **R.5** is an intuitive property of collective acceptance. This is due to the fact that, differently from the standard notions of common belief and common knowledge, collective acceptance entails an aspect of mutual identification and recognition with respect to a group.

Consider the left to right direction of the reduction axiom **R.5**. When the agents in a set $C$ identify themselves with a group $x$ and recognize each other as members of this group, they accept certain rules and principles to stand for the the rules and principles of the group. That is, the agents in $C$ *share a common body* of rules and principles. Among these shared rules and principles, there are the rules and principles which describe how the world should evolve when an announcement occurs. They govern how the acceptance of the agents in the group will be changed after an announcement. Suppose that a certain fact $\psi$ is publicly announced. After this announcement, the agents in $C$ accept $\varphi$, while identifying themselves with a group $x$ and recognizing each other as members of this group: $[x!\psi]\mathcal{A}_{C:x}\varphi$. This collective acceptance of the agents in $C$ is not created from scratch after the announcement of $\psi$. On the contrary, the creation of this acceptance depends on what the agents in $C$ accepted (before the announcement) as a principle of group $x$. In particular, the creation of $C$'s acceptance that $\varphi$ rests on the fact that, before $\psi$ is announced, the agents in $C$, while identifying themselves and recognizing each other as members of $x$, accept a principle saying that "if $\psi$ is true then, after $\psi$ is announced in $x$, $\varphi$ will be true": $\mathcal{A}_{C:x}(\psi \rightarrow [x!\psi]\varphi)$.

For example, imagine that the agents in a set $C$ identify themselves and recognize each other as members of the Lilliputian pacifist movement. Let $\psi$ denote the proposition "the government of Lilliput has decided to attack the neighboring nation of Blefuscu".[6] After $\psi$ is publicly announced the agents in $C$ accept that $\varphi$ = "they should start to protest against the Lilliput government", while functioning as members of the Lilliputian pacifist movement: $[LilliputPacifist!\psi]\mathcal{A}_{C:LilliputPacifist}\varphi$. This implies that (before the announcement) the agents in $C$, while identifying themselves and recognizing each other as members of the Lilliputian pacifist movement, accept a principle saying that "if $\psi$ is true then, after $\psi$ is announced, $\varphi$ will be true": $\mathcal{A}_{C:LilliputPacifist}(\psi \rightarrow [LilliputPacifist!\psi]\varphi)$. That is, the creation of $C$'s acceptance to protest against the Lilliput government depends on the fact that, before the announcement, the agents in $C$ accept to protest against the Lilliput government in case it will announce its decision to attack the neighboring nation of Blefuscu. This means that $C$'s acceptance to protest depends on the fact that, before the announcement, the agents in $C$ accept a principle which specifies what to do in case the Lilliput government will manifest its intention to attack Blefuscu.

It is worth noting that this situation for the logic of acceptance contrasts with the logic of common belief, where no reduction axioms for the common belief operator exist [18]. Intuitively speaking, this means that, differently from

[6] Lilliput and Blefuscu are the two fictional nations, permanently at war, that appear in the novel "Gulliver's Travels" by Jonathan Swift.

collective acceptance, the common belief of a set of agents $C$ may appear ‘out of the blue’: it was not foreseeable by the agents in $C$ that a common belief would ‘pop up’.

## 4 An Example

Until now we only considered that group acceptances emerge from consensus, by admitting axiom **Unanim**. One can go further and also consider other kinds of group acceptances, as shown in the next example. The example is inspired by Pettit [19].

*Example 7.* Imagine a three-member court which has to make a judgment on whether a defendant is liable (noted $l$) for a breach of contract. The three judges $i, j$ and $k$ accept a majority rule to decide on the issue. That is, $i, j$ and $k$, while functioning as members of the court, accept that if the majority of them accepts that the defendant is liable (resp. not liable), then the defendant is liable (resp. not liable). Formally, for any $B$ such that $B \subseteq \{i, j, k\}$ and $|B| = 2$ we have:

$$\textbf{(Maj)} \quad \mathcal{A}_{\{i,j,k\}:court}\Big(\bigwedge_{i \in B} \mathcal{A}_{i:court} l \to l\Big) \qquad \mathcal{A}_{\{i,j,k\}:court}\Big(\bigwedge_{i \in B} \mathcal{A}_{i:court} \neg l \to \neg l\Big)$$

Given the previous majority rule, we can prove that: after the announcement that both $i$ and $j$ accept $l$ (the defendant is liable) while functioning as members of the court, the agents in $\{i, j, k\}$ accept $l$ while functioning together as members of the court. Indeed, from the previous majority rule we can derive the formula $[court!\mathcal{A}_{i:court} l \wedge \mathcal{A}_{j:court} l]\mathcal{A}_{\{i,j,k\}:court} l$. To prove this, it is sufficient to note that, by means of the reduction axioms, the formula $[court!\mathcal{A}_{i:court} l \wedge \mathcal{A}_{j:court} l]\mathcal{A}_{\{i,j,k\}:court} l$ is successively rewritten as follows:
$\mathcal{A}_{\{i,j,k\}:court}((\mathcal{A}_{i:court} l \wedge \mathcal{A}_{j:court} l) \to [court!\mathcal{A}_{i:court} l \wedge \mathcal{A}_{j:court} l]l)$ by **R.5**
$\mathcal{A}_{\{i,j,k\}:court}((\mathcal{A}_{i:court} l \wedge \mathcal{A}_{j:court} l) \to l)$ by **R.1**
The latter is entailed by the majority rule **Maj**.

In the previous example, we have considered a majority rule as a principle which is responsible for the creation of collective acceptances from individual acceptances. This is stronger than the basic axiom of unanimity (**Unanim**) of $\mathcal{AL}$. One can imagine other kinds of rules. For instance, one can consider social contexts with leaders (see also [7]). In such contexts, one can formalize the rule according to which everything that the leaders accept is universally accepted in the social context. Let the set of leaders of $x$ be $L_x \in 2^{AGT}$. Then one can formalize that everything that the leaders accept is universally accepted in the social context by:

$$\textbf{(Leader)} \quad \mathcal{A}_{C:x}(\mathcal{A}_{L_x:x}\varphi \to \varphi)$$

## 5 Adding Retractions to $\mathcal{ALA}$: Some General Insights

According to our semantics, $\mathcal{A}_{i:x}\neg p \to [x!p]\mathcal{A}_{i:x}\bot$ is an $\mathcal{ALA}$ theorem (cf. Example 5). In words, when $p$ is publicly announced then $i$ quits all contexts $x$

where he accepted $p$: agent $i$ is no longer part of the institution, is kicked out of the group, etc. In $\mathcal{ALA}$ there is no means for $i$ to get out of that situation and re-integrate context $x$. At the present stage, our logic of acceptance does not include an operation which consists of an agent (or set of agents) joining a certain social context.

Semantically, what we need is the opposite of the previous model restrictions: an operation of adding arrows labelled by $i{:}x$ to the model. Syntactically, what we need is a new form of announcements $i{\leftarrow}C{:}x$ and corresponding modal operators of type $[i{\leftarrow}C{:}x]$, meaning that agent $i$ adopts $C$'s acceptances in context $x$. In terms of Kripke models, the accessibility relation $\mathscr{A}_{i:x}$ is identified with $\mathscr{A}_{C:x}$. This kind operation of adding arrows is reminiscent of the logic of preference upgrade of van Benthem and Liu [20], and the logic of granting and revoking permissions of Pucella and Weissman [21].[7] More intuitively, $i{\leftarrow}C{:}x$ represents the operation of agent $i$'s joining the social context $x$ by adopting the acceptances of group $C$ of members of $x$. After this operation, agent $i$ should start to function again as members of $x$.

Other kinds of retraction operations can be devised. For example, one might want to consider the operation of creating a supergroup $D$ of a given group $C$, where $D$ takes over all of $C$'s acceptances. The logical form of such an operation might be expressed by the operator $[D{:=}C:x]$. This operation should allow in particular to express that the agents in $D$ start to function as members of $x$ (*i.e.* to move from $\mathcal{A}_{D:x}\bot$ to $\neg\mathcal{A}_{D:x}\bot$), by taking over all acceptances of the agents in the subgroup $C$.

We are currently working on the technical issue of providing a semantic characterization and axiomatics of the previous operations $i{\leftarrow}C{:}x$ and $D{:=}C:x$ and corresponding modal operators $[i{\leftarrow}C{:}x]$ and $[D{:=}C:x]$.

## 6 Conclusion

In this paper we continued the studies initiated in [1], where the logic $\mathcal{AL}$, intended to formalize group (and individual) acceptances, was proposed. Here we extend $\mathcal{AL}$ by public announcements. The public announcement of $\psi$ is an event that results in all agents learning that $\psi$ is true. The public announcement of $\mathcal{A}_{C:x}\psi$ can be understood as a speech act. It simulates the announcement made by the group $C$ itself, that they accept $\psi$ while functioning as members of $x$. Therefore, as seen in Example 7, public announcements can be used to reason about the acceptances of agents when they express their own acceptances to each other. For instance, in that particular example we saw that a public announcement makes one of the agents quit the group, since he learns that the acceptances of the other agents are contrary to his own acceptances in the same context. As noted in Section 3.1, when the social context $x$ denotes an institution, announcements of the form $x!\psi$ can be used to describe the event of issuing or promulgating a certain norm $\psi$ (e.g. obligation, permission) within the context of the institution $x$.

[7] See [22] for a systemic study of these operators.

We also provide a complete axiomatization for the logic of acceptances and announcements $\mathcal{ALA}$. As well as for epistemic logic with public announcements, the axiomatization given for $\mathcal{ALA}$ uses reduction axioms. In $\mathcal{ALA}$, group acceptances are related to individual acceptances, but they are not computed from them. It contrasts with epistemic logics where the concept of common knowledge (or common belief) is completely defined in terms of individual knowledge (or belief). Due to this difference, it is possible to have reduction axioms for group acceptances, while it is known to be impossible for common knowledge. Still, in Section 3.3 we argue that this is an intuitive feature of group acceptances.

## References

1. Gaudou, B., Longin, D., Lorini, E., Tummolini, L.: Anchoring institutions in agents' attitudes: Towards a logical framework for autonomous multi-agent systems. In Padgham, L., Parkes, D.C., eds.: Proc. of AAMAS 2008, ACM Press (2008) 728–735
2. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. The MIT Press, Cambridge (1995)
3. Lewis, D.K.: Convention: a philosophical study. Harvard University Press, Cambridge (1969)
4. Gilbert, M.: On Social Facts. Routledge, London and New York (1989)
5. Tuomela, R.: The Philosophy of Sociality. Oxford University Press, Oxford (2007)
6. Hakli, P.: Group beliefs and the distinction between belief and acceptance. Cognitive Systems Research (7) (2006) 286–297
7. Lorini, E., Longin, D.: A logical approach to institutional dynamics: from acceptances to norms via legislators. In Brewka, G., Lang, J., eds.: Proc. of KR 2008, AAAI Press (forthcoming)
8. Gaudou, B., Herzig, A., Longin, D.: Grounding and the expression of belief . In: Proc. of KR 2006, AAAI Press (2006) 211–229
9. Gaudou, B., Herzig, A., Longin, D., Nickles, M.: A new semantics for the FIPA agent communication language based on social attitudes. In Brewka, G., Coradeschi, S., Perini, A., Traverso, P., eds.: Proc. of ECAI 2006, IOS Press (2006) 245–249
10. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001)
11. Fornara, N., Colombetti, M.: Operational specification of a commitment-based agent communication language. In Castelfranchi, C., Johnson, W.L., eds.: Proc. of AAMAS 2002, Bologna, ACM Press (2002) 535–542
12. Verdicchio, M., Colombetti, M.: A Logical Model of Social Commitment for Agent Communication. In: Proc. of AAMAS 2003, ACM (2003) 528–535
13. Singh, M.P.: Agent communication languages: Rethinking the principles. IEEE Computer **31**(12) (December 1998) 40–47
14. Plaza, J.: Logics of public communications. In Emrich, M.L., Hadzikadic, M., Pfeifer, M.S., Ras, Z.W., eds.: Proc. of ISMIS 1989. (1989) 201–216
15. Kooi, B.: Expressivity and completeness for public update logic via reduction axioms. Journal of Applied Non-Classical Logics **17**(2) (2007) 231–253
16. Walton, D.N., Krabbe, E.C.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New-York Press (1995)

17. Gelati, J., Rotolo, A., Sartor, G., Governatori, G.: Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. Artificial Intelligence and Law **12(1-2)** (2004) 53–81
18. Kooi, B., Van Benthem, J.: Reduction axioms for epistemic actions. In Schmidt, R., Pratt-Hartmann, I., Reynolds, M., Wansing, H., eds.: Proc. AiML 2004, King's College Publications (2004) 197–211
19. Pettit, P.: Deliberative democracy and the discursive dilemma. Philosophical Issues **11** (2001) 268–99
20. van Benthem, J., Liu, F.: Dynamic logic of preference upgrade. Journal of Applied Non-Classical Logics **17**(2) (2007) 157–182
21. Pucella, R., Weissman, V.: Reasoning about dynamic policies. In: Proc. of FOSSACS 2004. Number 2987 in LNCS, Springer Verlag (2004)
22. Aucher, G., Balbiani, P., Farias Del Cerro, L., Herzig, A.: Global and local graph modifiers. In: Methods for Modalities 5 (M4M-5). ENTCS, Elsevier (2007)

# From trust in information sources to trust in communication systems: an analysis in modal logic

Emiliano Lorini and Robert Demolombe

Institut de Recherche en Informatique de Toulouse (IRIT), France
{lorini,demolombe}@irit.fr

**Abstract.** We present a logical analysis of trust that integrates in the definition of trust: the truster's goal and the truster's belief that the trustee has the right properties (powers, abilities, dispositions) to ensure that the goal will be achieved. The second part of the paper is focused on the specific domain of trust in information sources and communication systems. We provide an analysis of the properties of information sources (validity, completeness, sincerity, competence, vigilance and cooperativity) and communication systems (availability and privacy) and, we discuss their relationships with trust.

## 1 Introduction

Future computer applications such as the semantic Web [1], e-business and e-commerce [11], Web services [20] will be open distributed systems in which the many constituent components are agents spread throughout a network in a decentralized manner. These agents will interact between them in flexible ways in order to achieve their design objectives and to accomplish the tasks which are delegated to them by the human users. Some of them will directly interact and communicate with the human users. During the system's lifetime, these agents will need to manage and deal with trust. They will need to automatically make trust judgments in order to assess the trustworthiness of other (software and human) agents while, for example, exchanging money for a service, giving access to a certain information, choosing between conflicting sources of information. They will also need to understand how trust can be induced in a human user in order to support his interaction with the system and to motivate him to use the application. Consequently, these agents will need to understand the components and the determinants of the user's trust in the system.

Thus, to realize all their potential, future computer applications will require the development of sophisticated formal and computational models of trust. These models must provide clear definitions of the relevant concepts related to trust and safe reasoning rules which can be exploited by the agents for assessing the trustworthiness of a given target. Moreover, these models of trust must be cognitively plausible, so that they can be directly exploited by the agents during their interactions with the human user in order to induce him to trust the system and the underlying Information and Communication Technology (ICT) infrastructure. With cognitively plausible models of trust, we mean models in which the main cognitive constituents of trust as a mental attitude are identified (e.g. beliefs, goals).

This paper follows our previous works [17] with the objective of developing a general formal model of trust which meets the previous desiderata. It is worth noting that it is not our aim to propose a model of trust based on statistics about past interactions with a given target and reputational information. In particular, the present paper focuses on an issue that we have neglected up to now: the issue of trust in information sources and communication systems. We think that this issue is very relevant for future computer applications such as the semantic Web, e-business and Web services. For example, in a typical scenario of e-business, trust in information sources has a strong influence on an agent's decision to buy, or to sale, a specific kind of stocks. Indeed, to take such a decision an agent has several types of information sources to consult in order to predict the future evolution of the stock value. These information sources may be banks, companies, consultants, *etc.* and the agent may believe that some of these information sources have a good competence but are not necessarily sincere, others are reluctant to inform about bad news, others are competent but are not necessarily informed at the right moment, *etc*. In a typical scenario of Web services, an agent might want to make a credit card transaction by means of a certain online payment system. In this case, the agent's trust in the communication system has a strong influence on the agent's decision to exploit it for the credit card transaction. In particular, the agent's trust in the online payment system is supported by the agent's belief that the online payment system will ensure the privacy of the credit card number from potential intruders.

The paper is organized as follows. We start with a presentation of a modal logic which enables reasoning about actions, beliefs and goals of agents (Section 2). This logic will be used during the paper for formalizing the relevant concepts of our model of trust. Then, a general definition of trust is presented (Section 3). Section 4 is focused on the formal characterization of the main properties of an information source: validity, completeness, sincerity, competence, vigilance and cooperativity. In Section 5 we show that these properties are epistemic supports for trust in information sources. In section 6 we provide an analysis of communication systems. We define two fundamental properties of communication systems: availability and privacy. Then, in Section 7, we show that these properties are epistemic supports for an agent's trust in a communication system. We conclude with a discussion of some related works and we show some directions for future works.

## 2 A modal logic of beliefs, goals and actions

We present in this section the multimodal logic $\mathcal{L}$ that we use in the paper to formalize the relevant concepts of our model of trust. $\mathcal{L}$ combines the expressiveness of a dynamic logic [13] with the expressiveness of a logic of agents' mental attitudes [7].

### 2.1 Syntax and semantics

The syntactic primitives of the logic $\mathcal{L}$ are the following: a nonempty finite set of agents $AGT = \{i, j, \ldots\}$; a nonempty finite set of atomic actions $AT = \{a, b, \ldots\}$; a finite set of atomic formulas $\Pi = \{p, q, \ldots\}$. $LIT$ is the set of literals which includes all atomic formulas and their negations, that is, $LIT = \{p, \neg p | p \in \Pi\}$. We note $P, Q, \ldots$ the elements in $LIT$. We also introduce specific actions of the form

$inf_j(P)$ denoting the action of informing agent $j$ that $P$ is true. We call them informative actions. The set $INFO$ of informative actions is defined as follows: $INFO = \{inf_j(P)|j \in AGT,\ P \in LIT\}$. Since the set $\Pi$ is finite, the set $INFO$ is finite as well. The set $ACT$ of complex actions is given by the union of the set of atomic actions and the set of informative actions, that is: $ACT = AT \cup INFO$. We note $\alpha, \beta, \ldots$ the elements in $ACT$. The language of $\mathcal{L}$ is the set of formulas defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid After_{i:\alpha}\varphi \mid Does_{i:\alpha}\varphi \mid Bel_i\varphi \mid Goal_i\varphi$$

where $p$ ranges over $\Pi$, $\alpha$ ranges over $ACT$ and $i$ ranges over $AGT$. The operators of our logic have the following intuitive meaning. $Bel_i\varphi$: the agent $i$ believes that $\varphi$; $After_{i:\alpha}\varphi$: after agent $i$ does $\alpha$, it is the case that $\varphi$ ($After_{i:\alpha}\bot$ is read: agent $i$ cannot do action $\alpha$); $Does_{i:\alpha}\varphi$: agent $i$ is going to do $\alpha$ and $\varphi$ will be true afterward ($Does_{i:\alpha}\top$ is read: agent $i$ is going to do $\alpha$); $Goal_i\varphi$: the agent $i$ wants that $\varphi$ holds. The following abbreviations are given: $Can_i(\alpha) \overset{\text{def}}{=} \neg After_{i:\alpha}\bot$;
$Int_i(\alpha) \overset{\text{def}}{=} Goal_i Does_{i:\alpha}\top$;
$Inf_{i,j}(P) \overset{\text{def}}{=} Does_{i:inf_j(P)}\top$.
$Can_i(\alpha)$ stands for: agent $i$ can do action $\alpha$ (i.e. $i$ has the capacity to do $\alpha$). $Int_i(\alpha)$ stands for: agent $i$ intends to do $\alpha$. Finally $Inf_{i,j}(P)$ stands for: $i$ informs $j$ that $P$ is true. Models of the logic $\mathcal{L}$ are tuples $M = \langle W, R, D, B, G, V\rangle$ defined as follows.

- $W$ is a non empty set of possible worlds or states.
- $R : AGT \times ACT \longrightarrow W \times W$ maps every agent $i$ and action $\alpha$ to a relation $R_{i:\alpha}$ between possible worlds in $W$. Given a world $w \in W$, if $(w, w') \in R_{i:\alpha}$ then $w'$ is a world which can be reached from $w$ through the occurrence of agent $i$'s action $\alpha$.
- $D : AGT \times ACT \longrightarrow W \times W$ maps every agent $i$ and action $\alpha$ to a relation $D_{i:\alpha}$ between possible worlds in $W$. Given a world $w \in W$, if $(w, w') \in D_{i:\alpha}$ then $w'$ is the unique actual *next* world of $w$ which will be reached from $w$ through the occurrence of agent $i$'s action $\alpha$.
- $B : AGT \longrightarrow W \times W$ maps every agent $i$ to a serial, transitive and euclidean relation $B_i$ between possible worlds in $W$. Given a world $w \in W$, if $(w, w') \in B_i$ then $w'$ is a world which is compatible with agent $i$'s beliefs at $w$.
- $G : AGT \longrightarrow W \times W$ maps every agent $i$ to a serial relation $G_i$ between possible worlds in $W$. Given a world $w \in W$, if $(w, w') \in G_i$ then $w'$ is a world which is compatible with agent $i$'s goals at $w$.
- $V : W \longrightarrow 2^{\Pi}$ is a truth assignment which associates each world $w$ with the set $V(w)$ of atomic propositions true in $w$.

Given a model $M$, a world $w$ and a formula $\varphi$, we write $M, w \models \varphi$ to mean that $\varphi$ is true at world $w$ in $M$, under the basic semantics. The rules defining the truth conditions of formulas are just standard for atomic formulas, negation and disjunction. The following are the remaining truth conditions for $After_{i:\alpha}\varphi$, $Does_{i:\alpha}\varphi$, $Bel_i\varphi$ and $Goal_i\varphi$.

- $M, w \models After_{i:\alpha}\varphi$ iff $M, w' \models \varphi$ for all $w'$ such that $(w, w') \in R_{i:\alpha}$
- $M, w \models Does_{i:\alpha}\varphi$ iff $\exists w'$ such that $(w, w') \in D_{i:\alpha}$ and $M, w' \models \varphi$
- $M, w \models Bel_i\varphi$ iff $M, w' \models \varphi$ for all $w'$ such that $(w, w') \in B_i$
- $M, w \models Goal_i\varphi$ iff $M, w' \models \varphi$ for all $w'$ such that $(w, w') \in G_i$

The following section is devoted to illustrate the additional semantic constraints over $\mathcal{L}$ models and the corresponding axiomatization of the logic $\mathcal{L}$.

### 2.2 Axiomatization

Operators for actions of type $After_{i:\alpha}$ and $Does_{i:\alpha}$ are supposed to be normal modal operators satisfying the axioms and rules of inference of system $K$. Operators for belief of type $Bel_i$ are supposed to be $KD45$ normal modal operators, whilst operators for goal of type $Goal_i$ are supposed to be $KD$ normal modal operators. Thus, we make assumptions about positive and negative introspection for beliefs and we suppose that an agent have no inconsistent beliefs or conflicting goals.

We add the following constraint over every relation $D_{i:\alpha}$ and every relation $D_{j:\beta}$ of all $\mathcal{L}$ models. For every $i, j \in AGT$, $\alpha, \beta \in ACT$ and $w \in W$:

S1 if $(w, w') \in D_{i:\alpha}$ and $(w, w'') \in D_{j:\beta}$ then $w' = w''$

Constraint $S1$ says that if $w'$ is the *next* world of $w$ which is reachable from $w$ through the occurrence of agent $i$'s action $\alpha$ and $w''$ is also the *next* world of $w$ which is reachable from $w$ through the occurrence of agent $j$'s action $\beta$, then $w'$ and $w''$ denote the same world. Indeed, we suppose that every world can only have one *next* world. The semantic constraint $S1$ corresponds to the following axiom.

**Alt**$_{Act}$ $Does_{i:\alpha}\varphi \rightarrow \neg Does_{j:\beta}\neg\varphi$

Axiom **Alt**$_{Act}$ says that: if $i$ is going to do $\alpha$ and $\varphi$ will be true afterward, then it cannot be the case that $j$ is going to do $\beta$ and $\neg\varphi$ will be true afterward.

We also suppose that the world is never static in our framework, that is, we suppose that for every world $w$ there exists some agent $i$ and action $\alpha$ such that $i$ is going to perform $\alpha$ at $w$. Formally, for every $w \in W$ we have that:

S2 $\exists i \in AGT, \exists \alpha \in ACT, \exists w' \in W$ such that $(w, w') \in D_{i:\alpha}$

The semantic constraint $S2$ corresponds to the following axiom of our logic.

**Active** $\bigvee_{i \in AGT, \alpha \in ACT} Does_{i:\alpha}\top$

Axiom **Active** ensures that for every world $w$ there is a *next* world of $w$ which is reachable from $w$ by the occurrence of some action of some agent. This is the reason why the operator $X$ for *next* of LTL (linear temporal logic) can be defined as follows:[1]

$$X\varphi \overset{\text{def}}{=} \bigvee_{i \in AGT, \alpha \in ACT} Does_{i:\alpha}\varphi$$

The following relationship is supposed between every relation $D_{i:\alpha}$ and the corresponding relation $R_{i:\alpha}$ of all $\mathcal{L}$ models. For every $i \in AGT$, $\alpha \in ACT$ and $w \in W$:

S3 if $(w, w') \in D_{i:\alpha}$ then $(w, w') \in R_{i:\alpha}$

The constraint $S3$ says that if $w'$ is the *next* world of $w$ which is reachable from $w$ through the occurrence of agent $i$'s action $\alpha$, then $w'$ is a world which is *possibly* reachable from $w$ through the occurrence of agent $i$'s action $\alpha$. The semantic constraint $S3$ corresponds to the following axiom **Inc**$_{Act,PAct}$.

[1] Note that $X$ satisfies the standard property $X\varphi \leftrightarrow \neg X\neg\varphi$ (i.e. $\varphi$ will be true in the next state iff $\neg\varphi$ will not be true in the next state).

**Inc**$_{Act,PAct}$ $Does_{i:\alpha}\varphi \rightarrow \neg After_{i:\alpha}\neg\varphi$

According to **Inc**$_{Act,PAct}$, if $i$ is going to do $\alpha$ and $\varphi$ will be true afterward, then it is not the case that $\neg\varphi$ will be true after $i$ does $\alpha$. The following axioms relates intentions with actions.

**IntAct1** $(Int_i(\alpha) \wedge Can_i(\alpha)) \rightarrow Does_{i:\alpha}\top$
**IntAct2** $Does_{i:\alpha}\top \rightarrow Int_i(\alpha)$

According to **IntAct1**, if $i$ has the intention to do action $\alpha$ and has the capacity to do $\alpha$, then $i$ is going to do $\alpha$. According to **IntAct2**, an agent is going to do action $\alpha$ only if he has the intention to do $\alpha$. In this sense we suppose that an agent's *doing* is by definition intentional. Similar axioms have been studied in [18] in which a logical model of the relationships between intention and action performance is proposed. **IntAct1** and **IntAct2** correspond to the following semantic constraints over $\mathcal{L}$ models. For every $i \in AGT$, $\alpha \in ACT$ and $w \in W$:

S4 if $\forall(w,w') \in G_i, \exists w''$ such that $(w',w'') \in D_{i:\alpha}$ and $\exists v$ such that $(w,v) \in R_{i:\alpha}$ then $\exists v'$ such that $(w,v') \in D_{i:\alpha}$

S5 if $\exists v'$ such that $(w,v') \in D_{i:\alpha}$ then $\forall(w,w') \in G_i, \exists w''$ such that $(w',w'') \in D_{i:\alpha}$

As far as informative actions are concerned, we assume that they are always executable, *i.e.*an agent $i$ can always inform another agent $j$ about a fact $P$. Formally:

**CanInf** $Can_i(inf_j(P))$

Axiom **CanInf** corresponds to the following semantic constraint over $\mathcal{L}$ models. For every $i \in AGT$, $inf_j(P) \in INFO$ and $w \in W$:

S6 $\exists w'$ such that $(w,w') \in R_{i:inf_j(P)}$

We also suppose that goals and beliefs must be compatible, that is, if an agent has the goal that $\varphi$ then, he cannot believe that $\neg\varphi$. Indeed, the notion of goal we characterize here is a notion of an agent's *chosen goal*, i.e. a goal that an agent decides to pursue. As some authors have stressed (e.g. [3]), a rational agent cannot decide to pursue a certain state of affairs $\varphi$, if he believes that $\neg\varphi$. Thus, for any $i \in AGT$ and $w \in W$ the following semantic constraint over $\mathcal{L}$ models is supposed:

S7 $\exists w'$ such that $(w,w') \in B_i$ and $(w,w') \in G_i$

The constraint $S7$ corresponds to the following axiom **WR** (*weak realism*) of our logic.

**WR** $Goal_i\varphi \rightarrow \neg Bel_i\neg\varphi$

In this work we assume positive and negative introspection over (chosen) goals, that is:

**PIntrGoal** $Goal_i\varphi \rightarrow Bel_iGoal_i\varphi$
**NIntrGoal** $\neg Goal_i\varphi \rightarrow Bel_i\neg Goal_i\varphi$

Axioms **PIntrGoal** and **NIntrGoal** correspond to the following semantic constraints over $\mathcal{L}$ models. For any $i \in AGT$ and $w \in W$:

S8 if $(w, w') \in B_i$ then $\forall v$, if $(w, v) \in G_i$ then $(w', v) \in G_i$
S9 if $(w, w') \in B_i$ then $\forall v$, if $(w', v) \in G_i$ then $(w, v) \in G_i$

We suppose that agents satisfy the property of *no forgetting* (**NF**)[2], that is, if an agent $i$ believes that after agent $j$ does $\alpha$, it is the case that $\varphi$, and agent $i$ does not believe that $j$ cannot do action $\alpha$, then after agent $j$ does $\alpha$, $i$ believes that $\varphi$.

**NF** $(Bel_i After_{j:\alpha}\varphi \wedge \neg Bel_i \neg Can_j(\alpha)) \rightarrow After_{j:\alpha} Bel_i \varphi$

Axiom **NF** corresponds to the following semantic constraint over $\mathcal{L}$ models. For any $i, j \in AGT$, $\alpha \in ACT$, and $w \in W$:

S10 if $(w, w') \in R_{j:\alpha} \circ B_i$ and $\exists v$ such that $(w, v) \in B_i \circ R_{j:\alpha}$ then $(w, w') \in B_i \circ R_{j:\alpha}$

where $\circ$ is the standard composition operator between two binary relations. In accepting the axiom **NF**, we suppose that events are always uninformative, that is, $i$ should not forget anything about the particular effects of $j$'s action $\alpha$ that starts at a world $w$. What an agent $i$ believes at a world $w'$, only depends on what $i$ believed at the previous world $w$ and on the action which has occurred and which was responsible for the transition from $w$ to $w'$. Besides, the axiom **NF** relies on an additional assumption of complete and correct information. It is supposed that $j$'s action $\alpha$ occurs if and only if every agent is informed of this fact. Hence all action occurrences are supposed to be public.

We also have specific properties for informative actions. We suppose that if an agent $i$ is informed (resp. not informed) by another $j$ that some fact $P$ is true then $i$ is aware of being informed (resp. not being informed) by $j$.

**PIntrInf** $Inf_{j,i}(P) \rightarrow Bel_i Inf_{j,i}(P)$
**NIntrInf** $\neg Inf_{j,i}(P) \rightarrow Bel_i \neg Inf_{j,i}(P)$

Axioms **PIntrInf** and **NIntrInf** correspond to the following semantic constraints over $\mathcal{L}$ models. For any $i, j \in AGT$, $inf_i(P) \in INFO$, and $w \in W$:

S11 if $\exists w'$ sutch that $(w, w') \in D_{j:inf_i(P)}$ then
$\forall (w, v) \in B_i, \exists w''$ such that $(v, w'') \in D_{j:inf_i(P)}$
S12 if $\exists w', w''$ such that $(w, w') \in B_i$ and $(w', w'') \in D_{j:inf_i(P)}$ then $\exists v$ sutch that $(w, v) \in D_{j:inf_i(P)}$

We call $\mathcal{L}$ the logic axiomatized by the axioms and rules of inference presented above. We write $\vdash \varphi$ if formula $\varphi$ is a theorem of $\mathcal{L}$ (i.e. $\varphi$ is the derivable from the axioms and rules of inference of the logic $\mathcal{L}$). We write $\models \varphi$ if $\varphi$ is *valid* in all $\mathcal{L}$ models, i.e. $M, w \models \varphi$ for every $\mathcal{L}$ model $M$ and world $w$ in $M$. Finally, we say that $\varphi$ is *satisfiable* if there exists a $\mathcal{L}$ model $M$ and world $w$ in $M$ such that $M, w \models \varphi$. We can prove that the logic $\mathcal{L}$ is *sound* and *complete* with respect to the class of $\mathcal{L}$ models. Namely:

**Theorem 1** $\vdash \varphi$ *if and only if* $\models \varphi$.

**Proof 1** *It is a routine task to check that all the axioms of the logic $\mathcal{L}$ correspond to their semantic counterparts. It is routine, too, to check that all of our axioms are in the Sahlqvist class, for which a general completeness result exists [2].*

[2] See also [10, 22] for a discussion of this property.

## 3 A general definition of trust

In this work trust is conceived as a complex configuration of mental states in which there is both a motivational component and an epistemic component. More precisely, we assume that an agent $i$'s trust in agent $j$ necessarily involves a goal of the truster: if agent $i$ trusts agent $j$ then, necessarily, $i$ trusts $j$ with respect to some of his goals. The core of trust is a belief of the truster about some properties of the trustee, that is, if agent $i$ trusts agent $j$ then necessarily $i$ trusts $j$ because $i$ has some goal and believes that $j$ has the right properties to ensure that such a goal will be achieved. The concept of trust formalized in this work is similar to the concept of trust defined by Castelfranchi & Falcone [5]. We agree with them that trust should not be seen as an unitary and simplistic notion as other models implicitly suppose. For instance, there are computational models of trust in which trust is conceived as an expectation of the truster about a successful performance of the trustee sustained by the repeated direct interactions with the trustee (under the assumption that iterated experiences of success strengthen the truster's confidence). More sophisticated models of social trust have been developed in which reputational information is added to information obtained via direct interaction (e.g. [14, 21]). All these models are in our view over-simplified since they do not consider the beliefs supporting the truster's evaluation of the trustee.

On this point we agree with Castelfranchi & Falcone on the fact that trust is based on the truster's *evaluation* of specific properties of the trustee (e.g. abilities, competencies, dispositions, etc.) and of the environment in which the trustee is going to act, which are relevant for the achievement of a goal of the truster. From this perspective, trust is nothing more than the truster's belief about some relevant properties of the trustee with respect to a given goal. [3] The following is the concept of trust as an *evaluation* that interests us in this paper.

**Definition 1** ***TRUST IN THE TRUSTEE'S ACTION.*** *$i$ trusts $j$ to do $\alpha$ with regard to his goal that $\varphi$ if and only if $i$ wants $\varphi$ to be true and $i$ believes that:*[4]

1. *$j$, by doing $\alpha$, will ensure that $\varphi$ AND*
2. *$j$ has the capacity to do $\alpha$ AND*
3. *$j$ intends to do $\alpha$*

The formal translation of Definition 1 is: [5]

$$Trust(i, j, \alpha, \varphi) \overset{\text{def}}{=} Goal_i X\varphi \wedge Bel_i(After_{j:\alpha}\varphi \wedge Can_j(\alpha) \wedge Int_j(\alpha))$$

In our logic the conditions $Can_j(\alpha)$ and $Int_j(\alpha)$ together are equivalent to $Does_{j:\alpha}\top$ (by axioms $\textbf{Inc}_{Act,PAct}$, **IntAct1** and **IntAct2**), so the definition of trust in the trustee's action can be simplified as follows:

[3] In this paper we do not consider a related notion of *decision to trust*, that is, the truster's decision to bet and wager on the trustee and to rely on her for the accomplishment of a given task. For a distinction between trust as an *evaluation* and trust as a *decision*, see [5, 19].

[4] In the present paper we only focus on *full trust* involving a *certain belief* of the truster. In order to extend the present analysis to forms of *partial trust*, a notion of *graded belief* (i.e. uncertain belief) or *graded trust*, as in [6], is needed.

[5] Notice that positive and negative introspection of trust follows from similar properties for beliefs and goals.

$$Trust(i,j,\alpha,\varphi) \overset{\text{def}}{=} Goal_i X\varphi \wedge Bel_i(After_{j:\alpha}\varphi \wedge Does_{j:\alpha}\top)$$

$Trust(i,j,\alpha,\varphi)$ is meant to stand for: $i$ trusts $j$ to do $\alpha$ with regard to his goal that $\varphi$.[6]

*Example 1.* The two agents $i$ and $j$ are making a transaction in Internet. After having paid $j$, $i$ trusts $j$ to send him a certain product with regard to his goal of having the product in the next state:

$$Trust(i,j,send,HasProduct(i)).$$

This means that $i$ wants to have the product in the next state:

$$Goal_i X\,HasProduct(i).$$

Moreover, according to $i$'s beliefs, $j$, by sending him the product, will ensure that he will have the product in the next state, and $j$ is going to send the product:

$$Bel_i(After_{j:send}HasProduct(i) \wedge Does_{j:send}\top).$$

The following theorem highlights the fact that if $i$ trusts $j$ to do $\alpha$ with regard to his goal that $\varphi$ then $i$ has a positive expectation that $\varphi$ will be true in the next state.

**Theorem 2** *Let $i,j \in AGT$ and $\alpha \in ACT$. Then:*
$\vdash Trust(i,j,\alpha,\varphi) \rightarrow Bel_i X\varphi$

In our view *trust in the trustee's action* must be distinguished from *trust in the trustee's inaction*. The former is focused on the domain of gains (goal achievements) whereas the latter is focused on the domain of losses (goal frustrations). That is, in the former case the truster believes that the trustee is in condition to *further* the achievement of his goals, and she will do that; in the latter case the truster believes that the trustee is in condition to *endanger* the achievement of his goals, but she will not do that. Trust in the trustee's inaction is based on the fact that, by doing some action $\alpha$, agent $j$ can prevent $i$ to reach his goal. In that case $i$ expects that $j$ will not intend to do $\alpha$.

**Definition 2** ***TRUST IN THE TRUSTEE'S INACTION****. $i$ trusts $j$ not to do $\alpha$ with regard to his goal $\varphi$ if and only if i wants $\varphi$ to be true and i believes that:*

1. *$j$, by doing $\alpha$, will ensure that $\neg\varphi$ AND*
2. *$j$ has the capacity to do $\alpha$ AND*
3. *$j$ does not intend to do $\alpha$*

The formal definition of trust in the trustee's inaction is given by the following abbreviation.

$$Trust(i,j,\neg\alpha,\varphi) \overset{\text{def}}{=} Goal_i X\varphi \wedge Bel_i(After_{j:\alpha}\neg\varphi \wedge Can_j(\alpha) \wedge \neg Int_j(\alpha))$$

$Trust(i,j,\neg\alpha,\varphi)$ stands for: $i$ trusts $j$ not to do $\alpha$ with regard to his goal that $\varphi$.

*Example 2.* Agent $j$ is the webmaster of a public access website with financial information. Agent $i$ is a regular reader of this website and he trusts $j$ not to restrict the access to the website with regard to his goal of having free access to the website:

$$Trust(i,j,\neg restrict,freeAccess(i)).$$

This means that, $i$ has the goal of having free access to the website in the next state:

[6] The meaning of the goal $\varphi$ may be that sometimes in the future $\psi$ holds.

$$Goal_i X freeAccess(i).$$

Moreover, according to $i$'s beliefs, $j$ has the capacity to restrict the access to the website and, by restricting the access to the website, $j$ will ensure that $i$ will not have free access to the website in the next state, but $j$ does not intend to restrict the access:

$$Bel_i(After_{j:restrict}\neg freeAccess(i) \wedge Can_j(restrict) \wedge \neg Int_j(restrict)).$$

In this situation, $i$'s trust in $j$ is based on $i$'s belief that $j$ is in condition to restrict the access to the website, but she does not have the intention to do this.

Note that, differently from agent $i$'s trust in agent $j$'s action, agent $i$'s trust in agent $j$'s inaction with respect to the goal that $\varphi$ does not entail $i$'s positive expectation that $\varphi$ will be true. Indeed, $Trust(i, j, \neg\alpha, \varphi) \wedge \neg Bel_i X\varphi$ is satisfiable in our logic. The intuitive reason is that $\neg\varphi$ may be the effect of another action than $j : \alpha$.

In the following sections 4 and 5 we will study the properties of information sources and show how these properties can be evaluated by the truster in order to assess the trustworthiness of an information source.

## 4 Basic properties of an information source

We suppose that the properties of an information source can be defined in terms of the relationships between three facts: an information source $j$ informs an agent $i$ that a certain fact $P$ is true; an information source $j$ believes that $P$ is true; the fact $P$ is true. These properties are all expressed in a conditional form. Since we have three facts that can be related in a similar conditional form, a systematic analysis of these relationships leads to six different properties of information sources.

**Definition 3** ***INFORMATION SOURCE VALIDITY***. *Agent $j$ is a valid information source about $P$ with regard to $i$ if and only if, after $j$ does the action of informing $i$ about $P$, it is the case that $P$.*

Formally: $Valid(j, i, P) \overset{\text{def}}{=} After_{j:inf_i(P)}P$

Note $After_{j:inf_i(P)}P$ can also be read in an explicit conditional form: if $j$ does the action $inf_i(P)$, then $P$ is true after the action has been done.

**Definition 4** ***INFORMATION SOURCE COMPLETENESS***. *Agent $j$ is a complete information source about $P$ with regard to $i$ if and only if, if $P$ is true then $j$ does the action of informing $i$ about $P$.*

Formally: $Compl(j, i, P) \overset{\text{def}}{=} P \rightarrow Inf_{j,i}(P)$

**Definition 5** ***INFORMATION SOURCE SINCERITY***. *Agent $j$ is a sincere information source about $P$ with regard to $i$ if and only if, if $j$ does the action of informing $i$ about $P$ then $j$ believes that $P$.*

Formally: $Sinc(j, i, P) \overset{\text{def}}{=} Inf_{j,i}(P) \rightarrow Bel_j P$

**Definition 6** ***INFORMATION SOURCE COMPETENCE***. *Agent $j$ is a competent information source about $P$ if and only if, if $j$ believes $P$ then $P$ is true.*

Formally: $Compet(j, P) \overset{\text{def}}{=} Bel_j P \rightarrow P$

**Definition 7** ***INFORMATION SOURCE VIGILANCE***. *Agent $j$ is a vigilant information source about $P$ if and only if, if $P$ is true then $j$ believes $P$.*

Formally: $Vigil(j,P) \overset{\text{def}}{=} P \rightarrow Bel_j P$

**Definition 8** ***INFORMATION SOURCE COOPERATIVITY***. *Agent $j$ is a cooperative information source about $P$ with regard to $i$ if and only if, if $j$ believes that $P$ then $j$ informs $i$ about $P$.* [7]

Formally: $Coop(j,i,P) \overset{\text{def}}{=} Bel_j P \rightarrow Inf_{j,i}(P)$

It is worth noting that the previous properties of information sources are not independent. For instance, as the following theorem 3 shows, validity can be derived from sincerity and competence and, completeness from vigilance and cooperativity.

**Theorem 3** *Let $i, j \in AGT$ and $inf_i(P) \in$ INFO, then:*

1. $\vdash After_{j:inf_i(P)}(Compet(j,P) \wedge Sinc(j,i,P)) \rightarrow Valid(j,i,P)$
2. $\vdash (Vigil(j,P) \wedge Coop(j,i,P)) \rightarrow Compl(j,i,P)$

Note that in theorem 3.1, the derivation of $Valid(j,i,P)$ requires $j$'s competence and sincerity at the instant where the action $inf_i(P)$ has been done. This is the reason why we have $After_{j:inf_i(P)}(Compet(j,P) \wedge Sinc(j,i,P))$ in the antecedent.

*Example 3.* Consider an example in the field of stocks and bonds market. The agent BUG is the Bank of Union of Groenland. Sue Naive (SN) and Very Wise (VW) are two BUG's customers. BUG plays the role of an information source for the customers, for instance for the facts $p$: "it is recommended to buy MicroHard stocks", and $q$: "Microhard stocks are dropping". SN believes that BUG is sincere with regard to her about $p$ and BUG is competent about $p$, because SN believes that BUG wants to help its customers and BUG has a long experience in the domain. SN also believes that BUG is cooperative with regard to her about $q$ because $q$ is a relevant information for customers in order to make decisions. VW too believes that BUG is competent about $p$. But VW does not believe that BUG is sincere with regard to him about $p$. Indeed, VW believes that BUG wants that VW buys Microhard stocks, even if this is not profitable for VW. This example is formally represented by the following formula:

$$Bel_{SN} Sinc(BUG, SN, p) \wedge Bel_{SN} Compet(BUG, p) \wedge Bel_{SN} Coop(BUG, SN, q) \wedge$$
$$Bel_{VW} Compet(BUG, p) \wedge \neg Bel_{VW} Sinc(BUG, VW, p).$$

## 5 Trust in information sources

We conceive trust in information sources as a specific instance of the general notion of trust in the trustee's action defined in section 3. In our view, the relevant aspect of trust in information sources is the content of the truster's goal. In particular, we suppose that an agent $i$ trusts the information source $j$ to inform him whether the fact $P$ is true only if $i$ has the *epistemic goal* of knowing whether $P$ is true and believes that, due to the information transmitted by $j$, he will achieve this goal. In this sense, trust in information sources is characterized by an epistemic goal of the truster and an informative action of the trustee. The concept of epistemic goal can be defined from the following standard definitions of *knowing that* (i.e. as having the correct belief that something is the case) and *knowing whether*:

[7] This definition of cooperativity does not exclude that $i$ does not want to be informed about $P$, like in spamming.

$$K_i\varphi \overset{\text{def}}{=} Bel_i\varphi \wedge \varphi \quad KW_i\varphi \overset{\text{def}}{=} K_i\varphi \vee K_i\neg\varphi$$

where $K_i\varphi$ stands for "agent $i$ knows that $\varphi$ is true" and, $KW_i\varphi$ stands for "$i$ knows whether $\varphi$ is true". An *epistemic goal* of an agent $i$ is $i$'s goal of knowing the truth value of a certain formula. Formally, $Goal_iKW_i\varphi$ denotes $i$'s epistemic goal of knowing whether $\varphi$ is true now; $Goal_iX\ KW_i\varphi$ denotes $i$'s epistemic goal of knowing whether $\varphi$ is true in the next state.

Our aim in this section of the paper is to investigate the relationships between trust in information sources and the properties of information sources defined in section 4. The following theorem 4 highlights the relationship between trust in information sources and the properties of validity and completeness of information sources. It says that: if $i$ believes that $j$ is a valid information source about $p$ and $\neg p$ with regard to $i$ and that $j$ is a complete information source about $p$ and $\neg p$ with regard to $i$ and, $i$ has the epistemic goal of knowing whether $p$ is true then, either $i$ trusts the information source $j$ to inform him that $p$ is true or $i$ trusts the information source $j$ to inform him that $\neg p$ is true (with regard to his epistemic goal of knowing whether $p$ is true).

**Theorem 4** *Let $i, j \in AGT$ and $inf_i(p), inf_i(\neg p) \in$ INFO, then:*
$\vdash (Bel_i(Valid(j,i,p) \wedge Valid(j,i,\neg p))\wedge\ Bel_i(Compl(j,i,p) \wedge Compl(j,i,\neg p))\wedge$
$Goal_iX\ KW_ip) \rightarrow (Trust(i,j,inf_i(p),KW_ip)\vee\ Trust(i,j,inf_i(\neg p),KW_ip))$

The reason why in the consequent we have a disjunction (instead of a conjunction) is that $p$ is either true or false. Then, $j$ may inform $i$ either about the truth of $p$ or about the truth of $\neg p$. From theorems 3.1 and 3.2, similar theorems can be proved by substituting $Valid(j,i,p)$ with $After_{j:inf_i(p)}(Compet(j,p) \wedge Sinc(j,i,p))$, $Valid(j,i,\neg p)$ with $After_{j:inf_i(\neg p)}$
$(Compet(j,\neg p) \wedge Sinc(j,i,\neg p))$, $Compl(j,i,p)$ with $Vigil(j,p) \wedge Coop(j,i,p)$ and, $Compl(j,i,\neg p)$ with $Vigil(j,\neg p) \wedge Coop(j,i,\neg p)$ in the antecedent of theorem 4.

The following theorem 5 is a specific instantiation of theorem 2. It says that: if $i$ trusts the information source $j$ to inform him that $p$ or $i$ trusts the information source $j$ to inform him that $\neg p$ with regard to his goal of knowing whether $p$ is true, then $i$ believes that in the next state he will achieve his goal of knowing whether $p$ is true.

**Theorem 5** *Let $i, j \in AGT$ and $inf_i(p), inf_i(\neg p) \in$ INFO, then:*
$\vdash (Trust(i,j,inf_i(p),KW_ip)\vee\ Trust(i,j,inf_i(\neg p),KW_ip)) \rightarrow Bel_iX\ KW_ip$

*Example 4.* Let us consider again the example of stocks and bonds market. SN has the epistemic goal of knowing whether $q$ ("Microhard stocks are dropping") is true:

$$Goal_{SN}X\ KW_{SN}q.$$

SN believes that BUG is a valid information source with regard to her both about $q$ and about $\neg q$ and that BUG is a complete information source with regard to her both about $q$ and about $\neg q$:

$$Bel_{SN}(Valid(BUG,SN,q) \wedge Valid(BUG,SN,\neg q))\wedge$$
$$Bel_{SN}(Compl(BUG,SN,q) \wedge Compl(BUG,SN,\neg q)).$$

Then, by theorem 4, we can infer that either SN trusts the information source BUG to inform her that $q$ is true or SN trusts the information source BUG to inform her that $\neg q$ is true (with regard to her epistemic goal of knowing whether $q$ is true):

$$Trust(SN,BUG,inf_{SN}(q),KW_{SN}q) \vee Trust(SN,BUG,inf_{SN}(\neg q),KW_{SN}q).$$

Finally, by theorem 5, we can infer that SN believes that in the next state she will achieve her goal of knowing whether $q$ is true:

$$Bel_{SN} X\ KW_{SN} q.$$

In the following two sections 6 and 7 we will shift the focus of analysis from information sources to communication systems. We will study some important properties of communication systems and show how these properties can be evaluated by the truster in order to assess the trustworthiness of a communication system.

## 6 Basic properties of a communication system

We suppose that the fundamental properties of a communication system $j$ can be defined in terms of two facts: the communication system $j$ satisfies an agent $i$'s goal that a certain information will be transmitted to another agent $z$ or, the communication system $j$ satisfies an agent $i$'s goal that a certain information will not be transmitted to another agent $z$. In the former case we say that the communication system $j$ is available to $i$ to transmit the information. In the latter case we say that the communication system $j$ ensures to $i$ the privacy of the information. For simplification, we ignore in this work other properties of communication systems like authentication and integrity.

**Definition 9** ***COMMUNICATION SYSTEM AVAILABILITY***. *The communication system $j$ is available to agent $i$ to transmit the information $P$ to agent $z$ if and only if, if $j$ believes that $i$ wants that $j$ informs $z$ about $P$ then $j$ informs $z$ about $P$.*

Formally: $Avail(j, i, z, P) \overset{\text{def}}{=} Bel_j Goal_i Inf_{j,z}(P) \rightarrow Inf_{j,z}(P)$

**Definition 10** ***COMMUNICATION SYSTEM PRIVACY***. *The communication system $j$ ensures to agent $i$ the privacy of information $P$ from agent $z$ if and only if, if $j$ believes that $i$ does not want that $j$ informs $z$ about $P$ then, $j$ does not inform $z$ about $P$.*

Formally: $Priv(j, i, z, P) \overset{\text{def}}{=} Bel_j Goal_i \neg Inf_{j,z}(P) \rightarrow \neg Inf_{j,z}(P)$

*Example 5.* Let us consider an example in the field of Web services. An agent called Bill decides to use a Hotel Booking Service (HBS) in Internet in order to book a double room at the Hotel Colosseum (HC) in Rome. Bill's decision is affected by two beliefs of Bill, the belief that HBS ensures the privacy from a potential intruder of the information $r$: "Bill's credit card number is 01234567891234", the belief that HBS is available to inform HC about $s$: "Bill has made an online reservation". According to our definitions, this example is formally represented by:

$$Bel_{Bill} Priv(HBS, Bill, intruder, r) \wedge Bel_{Bill} Avail(HBS, Bill, HC, s).$$

## 7 Trust in communication systems

We conceive trust in communication systems as a specific instance of the notion of trust defined in section 3. On the one hand, we suppose that an agent $i$ trusts the communication system $j$ to inform agent $z$ about $P$ with regard to his goal that $z$ believes $P$ (*trust in a communication system's action*) if and only if, $i$ has the goal that $z$ believes $P$ and $i$ believes that, due to the information transmitted by $j$ to $z$, $z$ will believe $P$. On the other hand, we suppose that an agent $i$ trusts the communication system $j$ not to inform agent $z$ about $P$ with regard to his goal that $z$ does not believe $P$ (*trust in a* *communication system's inaction*) if and only if, $i$ has the goal that $z$ does not believe $P$, $i$ believes that, by informing $z$ about $P$, $j$ will ensure that $z$ believes $P$ but, $i$ believes that $j$ does not intend to inform $z$ about $P$. In this sense, $i$'s trust in the communication system $j$'s action (resp. inaction) is characterized by $i$'s goal that a certain information will be transmitted (resp. will not be transmitted) to another agent $z$ so that $z$ will have access (resp. will not have access) to this information.

Our aim in this section of the paper is to investigate the relationships between trust in communication systems and the two properties of communication systems defined in section 6. The following theorem 6 highlights the relationship between trust in a communication system's action and the availability of the communication system. It says that: if $i$ has the goal that in the next state $z$ will believe $P$, $i$ believes that $j$ is available to inform $z$ about $P$, $i$ believes that $j$ believes that $i$ wants $j$ to inform $z$ about $P$ and, $i$ believes that $z$ believes that $j$ is a valid information source about $P$ then, $i$ trusts $j$ to inform $z$ about $P$ with regard to his goal that $z$ will believe $P$.

**Theorem 6** *Let $i, j, z \in AGT$ and $inf_z(P) \in INFO$, then:*
$\vdash (Goal_i X Bel_z P \wedge Bel_i Avail(j, i, z, P) \wedge\ Bel_i Bel_j Goal_i Inf_{j,z}(P) \wedge$
$Bel_i Bel_z Valid(j, z, P)) \rightarrow Trust(i, j, inf_z(P), Bel_z P)$

*Example 6.* Let us consider again the example of the Hotel Booking Service (HBS). Bill has the goal that the receptionist at the Hotel Colosseum (HC) believes that $s$ ("Bill has made an online reservation"):

$$Goal_{Bill} X Bel_{HC} s.$$

Bill believes that HBS is available to inform the HC's receptionist about $s$ and that HBS believes that Bill wants HBS to inform the HC's receptionist about $s$:

$$Bel_{Bill} Avail(HBS, Bill, HC, s) \wedge Bel_{Bill} Bel_{HBS} Goal_{Bill} Inf_{HBS,HC}(s).$$

Bill also believes that the HC's receptionist believes that HBS is a valid information source about $s$:

$$Bel_{Bill} Bel_{HC} Valid(HBS, HC, s).$$

Then, from theorem 6, we can infer that Bill trusts HBS to inform the HC's receptionist about $s$ with regard to his goal that the HC's receptionist will believe $s$:

$$Trust(Bill, HBS, inf_{HC}(s), Bel_{HC} s).$$

The following theorem 7 highlights the relationship between trust in a communication system's inaction and the fact that the communication system ensures privacy. It says that: if $i$ has the goal that $z$ does not believe $P$, $i$ believes that $j$ ensures the privacy of information $P$ from agent $z$, $i$ believes that $j$ believes that $i$ wants $j$ not to inform $z$ about $P$ and, $i$ believes that $z$ believes that $j$ is a valid information source about $P$ then, $i$ trusts $j$ not to inform $z$ about $P$ with regard to his goal that $z$ does not believe $P$.

**Theorem 7** *Let $i, j, z \in AGT$ and $inf_z(P) \in INFO$, then:*
$\vdash (Goal_i X \neg Bel_z P \wedge Bel_i Priv(j, i, z, P) \wedge Bel_i Bel_j Goal_i \neg Inf_{j,z}(P) \wedge$
$Bel_i Bel_z Valid(j, z, P)) \rightarrow Trust(i, j, \neg inf_z(P), \neg Bel_z P)$

*Example 7.* In this version of the scenario of the Hotel Booking Service (HBS) Bill has the goal that a potential intruder will not have access to the information $r$ ("Bill's credit card number is 01234567891234"):

$$Goal_{Bill} X \neg Bel_{intruder} r.$$

Bill believes that HBS ensures the privacy of information $r$ from potential intruders and that HBS believes that Bill wants HBS not to inform a potential intruder about $r$: $Bel_{Bill}Priv(HBS, Bill, intruder, r) \wedge Bel_{Bill}Bel_{HBS}Goal_{Bill}\neg Inf_{HBS,intruder}(r)$. Finally, Bill believes that every potential intruder believes that HBS is a valid information source about credit card numbers:

$$Bel_{Bill}Bel_{intruder}Valid(HBS, intruder, r).$$

Then, from theorem 7, we can infer that Bill trusts HBS not to inform a potential intruder about $r$ with regard to his goal that a potential intruder will not believe $r$:

$$Trust(Bill, HBS, \neg inf_{intruder}(r), \neg Bel_{intruder}r).$$

## 8 Related works

Several logical models of trust in information sources have been proposed in the recent literature [16, 15, 9, 8]. Some of them take the concept of trust as a primitive [16], whereas others reduce trust to a kind of belief of the truster [15, 9]. For instance, in [15] trust is characterized on the basis of two kinds of beliefs of the truster: the truster's belief that a certain rule or regularity applies to the trustee (called "rule belief"), and the truster's belief that the rule or regularity is going to be followed by the trustee (called "conformity belief"). Nevertheless, all these models ignore the motivational aspect of trust in information sources. Moreover, all these models do not consider the epistemic supports for this form of trust. In the present paper both aspects of trust in information sources have been taken into account. On the one hand, we have modeled the truster's epistemic goal of knowing whether a certain fact is true. On the other hand, we have modeled the properties of information sources such as sincerity, validity, competence, etc. and shown that some of them are epistemic supports for an agent's trust in an information source, that is, they are sufficient conditions for trusting an information source to inform whether a certain fact is true (theorem 4). As far as communication systems are concerned, there are several logical models in the literature which deal with the properties of a communication system such as privacy, confidentiality, availability, integrity, authentication (*e.g.*[4]). Nevertheless, there is still no formal analysis of the relationships between these properties of a communication system and trust in the communication system. In this work such relationships have been clarified (theorems 6 and 7).

## 9 Conclusion

We have presented in a modal logical framework a model that integrates in the definition of trust: the truster's goal, the trustee's action that ensures the achievement of the truster's goal, and the trustee's ability and intention to do this action. In the same logical framework we have defined several properties of information sources (validity, completeness, sincerity, competence, vigilance and cooperativity) and discussed their relationships with an agent's trust in an information source. In the last part of the paper, we have investigated some properties of communication systems (availability and privacy) and discussed their relationships with an agent's trust in a communication system. It has to be noted that, due to the complexity of the concepts involved in our analysis of trust, we had to accept strong simplifications. For instance, in the definitions of the properties of information sources and communication systems entailment is formalized by a material implication, while some form of conditional might be more adequate. Our future work will be devoted to refine this aspect of the proposed logical formalization of trust.

## References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001.
3. M. Bratman. *Intentions, plans, and practical reason*. Harvard University Press, 1987.
4. M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.
5. C. Castelfranchi and R. Falcone. Social trust: A cognitive approach. In C. Castelfranchi and Y. H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer, 2001.
6. C. Castelfranchi, R. Falcone and E. Lorini. A non reductionist approach to trust. In J. Golbeck, editor, *Computing with Social Trust and Reputation*. Springer, 2008. In press.
7. P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.
8. M. Dastani, A. Herzig, J. Hulstijn, and L. van der Torre. Inferring trust. In *Proc. of Fifth Workshop on Computational Logic in Multi-agent Systems (CLIMA V)*, pages 144–160. Springer, 2004.
9. R. Demolombe. To trust information sources: a proposal for a modal logical framework. In C. Castelfranchi and Y. H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 111–124. Kluwer, 2001.
10. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.
11. M. Fasli. *Agent Technology for E-commerce*. Wiley & Sons, Chichester, 2007.
12. J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, The Netherlands, 1999.
13. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, 2000.
14. T. G. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Journal of Autonomous Agent and Multi-Agent Systems*, 13:119–154, 2006.
15. A. J. I. Jones. On the concept of trust. *Decision Support Systems*, 33(3):225–232, 2002.
16. C. J. Liau. Belief, information acquisition, and trust in multi-agent systems: a modal logic formulation. *Artificial Intelligence*, 149:31–60, 2003.
17. E. Lorini and R. Demolombe. Trust and norms in the context of computer security. In *Proc. of the Ninth International Conference on Deontic Logic in Computer Science (DEON'08)*, volume 5076 of *LNAI*, pages 50–64. Springer Verlag, 2008.
18. E. Lorini and A. Herzig. A logic of intention and attempt. *Synthese*, 163(1):45–77.
19. S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, Scotland, 1994.
20. S. A. Mcllraith, T. C. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53, 2001.
21. J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In *Proc. of AAMAS'02*, pages 475–482. ACM Press, 2001.
22. R. B. Scherl and H. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144:1–39, 2003.

# Pre-processing Techniques For Anytime Coalition Structure Generation Algorithms

Tomasz Michalak, Andrew Dowell, Peter McBurney and Michael Wooldridge

Department of Computer Science,
The University of Liverpool, UK
{tomasz, adowell, mcburney, mjw}@ liv.ac.uk

**Abstract.** [1] This paper is concerned with optimal coalition structure generation in multi-agent systems. For characteristic function game representations, we propose a pre-processing technique, presented in the form of filter rules, that reduces the intractability of the coalition structure generation problem by identifying coalitions which cannot belong to any optimal structure. These filter rules can be incorporated into many potential anytime coalition structure generation algorithms but we test the effectiveness of these filter rules in the sequential application of the distributed coalition value calculation algorithm (DCVC) [1] and the anytime coalition structure generation algorithm of Rahwan *et al.* (RCSG) [2]. The distributed DCVC algorithm provides an input to the centralised RCSG algorithm and we show that for both normal and uniform distributions of coalition values, the proposed filter rules reduce the size of this input by a considerable amount. For example, in a system of 20 agents, fewer than 5% of coalition values have to be input, compared to more than 90% when filter rules are not employed. Furthermore, for a normal distribution, the running time of the RCSG algorithm exponentially accelerates as a consequence of the significantly reduced input size. This pre-processing technique bridges the gap between the distributed DCVC and centralised RCSG algorithms and is a natural benchmark to develop a distributed CSG algorithm.

## 1 Background and Problem Definition

In multi-agent systems (MAS), coalition formation occurs when distinct autonomous agents group together to achieve something more efficiently than they could accomplish individually. One of the main challenges in co-operative MAS is to determine which exhaustive division of agents into disjoint coalitions (referred to as a *coalition structure* ($CS$) from now on) maximizes the total payoff to the system. This complex research issue is referred to as the *coalition structure generation* ($CSG$) problem. To this end, coalition formation is often studied using *characteristic function game* ($CFG$) representations which consist of a set of agents $A$ and a characteristic function $v$, which assigns a numerical value to every feasible coalition $C \subseteq A$, reflecting the effectiveness of the co-operation of the agents within each coalition. In this convenient but simplified representation, it is assumed that the performance of any one coalition is independent

[1] The authors are grateful for financial support received from the UK EPSRC through the project *Market-Based Control of Complex Computational Systems* (GR/T10657/01).

from other co-existing coalitions in the system. In other words, the value of a coalition $C$ in a structure $CS$ has the same value as it does in another distinct structure $CS'$. Evidently, the $CFG$ representation is a special case of the more general *partition function game* ($PFG$) representation in which the value of any coalition depends on other co-operational arrangements between the agents in the whole system [3].

Since the number of structures increases exponentially as the number of agents increases linearly (for example, there are $190,899,322$ possible structures for 14 agents compared to 1.3 billion structures for 15 agents) then the problem of computing an optimal CS becomes intractable, even for a moderate number of agents. Consequently, much of the research into the CSG problem has focused on circumventing this computational intractability. Proposed solutions to circumvent this intractability can be divided into three broad categories: (i) limiting the size of coalitions that can be formed [4], (ii) reducing the number of structures that need to be searched at the expense of accuracy [5, 6] and (iii) proposing algorithms which take the advantage of the CFG representation to vastly reduce the time taken to generate an optimal CS. Since we are interested in generating optimal solutions in an unconstrained domain, in this paper, we will focus exclusively on the third approach.

There are two general classes of CSG algorithms. Yen [7] proposed an algorithm based on Dynamic Programming (DP) paradigm. The advantage of this approach is that it outputs an optimal structure without comparing every possible structure. However, one disadvantage is that it only outputs an optimal after it has completed its entire execution meaning such paradigms are not appropriate when the time required to return an optimal solution is longer than the time available to the agents. To circumvent these problems, Rahwan *et al.* [2] proposed an anytime CSG algorithm which divides the space of all structures (denoted $\Pi$ from now on) into sub-spaces consisting of coalition structures which are identical w.r.t. the sizes of the coalitions involved. Using this representation, and taking, as input, all feasible coalition values, this algorithm uses statistical information, computed from the coalition values input, to determine which of the sub-spaces are 'promising', *i.e.* which of them may contain an optimal structure. The algorithm then searches these 'promising' subspaces, once again, using the statistical information to avoid generating structures which cannot be optimal. This methodology exploits the fact that in CFGs, the value of a coalition $C$ in a structure $CS$ has the same value as it does in every other distinct structure $CS'$. Since it is possible to utilize statistical data computed from coalition values to reason about the values of coalition structures, in this paper, we propose a number of pre-processing techniques. These are represented in the form of filter rules which have the following syntax: *condition* $\rightarrow$ *action*, with the interpretation being that all coalition values which meet the requirements of the condition cannot belong to an optimal structure and so an appropriate action is performed.

Typically, such actions involve filtering coalition values from the input, or filtering all structures containing these coalitions from the search-space. Filtering coalition values from the input is important for two reasons. Firstly, it reduces the number of coalition values an individual agent needs to transfer if the coalition value calculation process is distributed as in the DCVC algorithm of Rahwan and Jennings [1]. Secondly, it automatically reduces the search space as fewer coalition structures can be created from the input. To test the effectiveness of our approach, we compare the sequential operation of the DCVC and Rahwan *et al.* anytime algorithm both with and without the filter rules. Following the MAS literature, we focus on normal and uniform distributions of coalition values [2] and show that our filter rules:

- always significantly reduce the size of input (from 90% to 5% and 3% for normal and uniform distributions, respectively), and;
- exponentially reduce the time needed to search promising subspaces for a normal distribution whereas they do not affect the performance of the anytime CSG algorithm for a uniform distribution.

## 2 A pre-processing approach to solve the CSG problem

Let $A = \{1, \ldots, n\}$ be the set of all agents in the system. Since more than one structure can maximize the value of a system, the output of the CSG process may consist of a set of optimal coalition structures, denoted by $\{CS^*\}$. As of now, the CSG literature for CFG representations has exclusively focused on finding a single optimal coalition structure, denoted by $CS^* \subseteq \{CS^*\}$ [7,2]. Usually, the choice of $CS^* \subseteq \{CS^*\}$ for $|\{CS^*\}| \geq 2$ is made in an *ad hoc* manner, *e.g.* the optimal coalition structure output is the first CS with maximal value which the algorithm encounters. However, there may be other factors which, although not displayed in the characteristic function, can be used to determine if one structure is better than the other and so we consider the $CSG(\{CS^*\})$ problem from now onward. We will denote the set of all feasible coalitions that can be created in the system by $\mathcal{F}$ and the corresponding set of all coalition values by $V(\mathcal{F})$.

The most important property of CFG representation is that, as opposed to the general PFG representation, the value of any coalition is independent from the formation of other distinct coalitions in the system. In other words, in a system of $n = |A|$ agents, for any coalition $C \subseteq A$, the value of this coalition $v(C)$ is the same in every possible structure $CS$ where $C \in CS$. Thus, if it can be shown that the value $v(C)$ is too small for $C$ to be in any optimal structure of the system then, clearly, any structure containing $C$ cannot be in the optimal set and can be disregarded. Similarly, if it is proven that the combined value of a group of disjoint coalitions it too small for this group to be in any optimal structure then any structure simultaneously containing every single one of these coalitions can be disregarded. The above discussion can be summarized in the following lemma which holds under the CFG representation:

**Lemma 1.** *For any non-trivial*[2] *coalition* $C$ *that can be divided into* $k$ *disjoint sub-coalitions* $C_1, \ldots, C_k$ *where* $C_1 \cup C_2 \ldots \cup C_k = C$*; if* $v(C_1) + \ldots v(C_k) > v(C)$ *then* $C \notin CS^*$ *and so* $\forall CS : C \in CS, CS \notin \{CS^*\}$.

*Proof.* Consider any coalition structure $CS$ containing coalition $C$. If this structure is the optimal structure (belongs to $\{CS^*\}$) then no other structure can have a value

[2] All coalitions of more than two agents will sometimes be referred to as *non-trivial coalitions*. In contrast, singletons, *i.e.* agents acting on their own, will sometimes be referred to as *trivial coalitions*.

greater then this. However, if $C$ can be divided into $k$ sub-coalitions $C_1, \ldots, C_k$ where $C_1 \cup C_2 \ldots \cup C_k = C$ and $|C_i| \geq 1 \; \forall i = 1, \ldots, k$ such that $v(C_1) + \ldots v(C_k) > v(C)$ then clearly the structure $CS' = CS \setminus \{C\} \cup \{C_1 \cup \ldots \cup C_k\}$ has a value greater than $CS$. Therefore $CS$, where $CS$ is any structure containing coalition $C$, cannot be an optimal structure and $\forall CS : C \in CS, CS$ does not belong to $\{CS^*\}$.

For example, in a system of 5 agents $A = \{a_1, a_2, a_3, a_4, a_5\}$ if the value of the coalition $\{a_1, a_2\}$ is less than the sum of the values of the individual coalitions $\{a_1\}$ and $\{a_2\}$ then it is clear that any structure $CS_a = \{a_1, a_2\} \cup CS'$, where $CS'$ is a specific structure of the agents $\{a_3, a_4, a_5\}$, must have value less than the structure $CS_b = \{a_1\}, \{a_2\} \cup CS'$. Consequently, any structure $CS$ which contains coalition $\{a_1, a_2\}$ cannot be optimal and all such structures can be disregarded.

Most existing CSG algorithms take advantage of the above fundamental characteristic of the CFG setting. For example, in the anytime CSG algorithm of Rahwan *et al.* described in Section 1 some subspaces of coalition structures are pruned away from the search space before the search process has begun. Using statistical information obtained from the input of coalition values, it is *a priori* decided if certain types of coalition structures cannot be optimal. Similarly, in the process of searching promising subspaces, certain search directions are *a priori* identified as those that cannot lead to an improved outcome. Examples include dynamic programming (DP) CSG algorithms [7, 8] in which it is evaluated whether a decomposition a coalition $C$ into exactly two smaller coalitions of all the agents in $C$ would be profitable. In this spirit, the improved dynamic programming (IDP) algorithm presented in [**?**] considers more dissociations than just the dissociation of a coalition into two disjoint coalitions. This approach, which is yet another application of Lemma 1, turns out to be more efficient in terms of time and memory cost than the conventional DP algorithms.

Consider a system of 5 agents where a value of a non-trivial coalition $C := \{a_1, a_2, a_3, a_4\}$ is 7. In order to prove that $C$ cannot be in an optimal CS, it is not always necessary to show that any partition of this coalition has a combined value greater than 7. It may also be sufficient to show that the values of a strict subset of $k$ disjoint coalitions as in Lemma 1 are greater than the value of $v(C)$, for instance, it may be the case that $v(\{a_1\}) + v(\{a_3, a_4\}) \geq v(\{a_1, a_2, a_3, a_4\})$. Following this intuition, we can relax some of the assumptions in Lemma 1 and propose Lemma 2:

**Lemma 2.** *For any non-trivial coalition $C$ that can be divided into $k$ disjoint sub-coalitions $C_1, \ldots, C_k$ where $C_1 \cup C_2 \ldots \cup C_k = C$; if $\sum_{i=1}^{j} v(C_i) > v(C)$ where $j < k$ then $C \notin CS^*$ and so $\forall CS : C \in CS$, $CS$ does not belong to $\{CS^*\}$.*

Theoretically, every feasible non-trivial coalition could be decomposed into all possible combinations of sub-coalitions, however, this is a completely inefficient approach. After all, such a decomposition of the grand coalition yields $\Pi$. In the remainder of this paper, we will show that an appropriate application of both Lemmas 1 and 2 may still considerably speed up the CSG process in the state-of-the-art CSG algorithm.

Firstly, we will extend Lemma 1 so that it can be applied not only to a particular coalition but to collections of coalitions which have been grouped together w.r.t. some criteria. One natural criterion to group coalitions is size. Let all coalitions of the same size be grouped together in $|A|$ sets $\mathcal{C}_1, \ldots, \mathcal{C}_i, \ldots, \mathcal{C}_{|A|}$ where $\mathcal{C}_i$ denotes the set of all coalitions of size $i$. For example, for $A = \{1, \ldots, 5\}$ there will be $|A| = 5$ sets $\mathcal{C}_1, \ldots, \mathcal{C}_5$ where $\mathcal{C}_1$ contains all coalitions of size 1, $\mathcal{C}_2$ all coalitions of size 2, *etc*. Additionally, in this example, suppose that the coalitions with smallest values in $\mathcal{C}_1$ and $\mathcal{C}_2$ are $\{a_1\}$ and $\{a_2, a_3\}$, respectively. This means that any decomposition of coalition of size 3 will not have a smaller value than $v(\{a_1\}) + v(\{a_2, a_3\})$. Consequently, if any coalition from $\mathcal{C}_3$ has a value smaller than $v(\{1\}) + v(\{2, 3\})$ then we can disregard this coalition as, following Lemma 1, it cannot be in an optimal structure. We extend Lemma 1 as follows:

**Lemma 3.** *Let $\mathcal{C}_i$ denote the (complete) set of all coalitions of size $i$. For any set $Z_i \subseteq \mathcal{C}_i$ of coalitions of size $i$ and for a particular integer partition $p$ of $i$, $i_1, \ldots, i_k$ such that $i_1 + \ldots + i_k = i$, if the sum of the lowest coalition values in sets $\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_k}$ (denoted $d_i(p)$) is strictly greater than the maximum value in set $Z_i$ then no coalition from set $Z_i$ can be in an optimal structure. More formally, if $d_i(p) := \sum_{i=1}^{k} \min \mathcal{C}_{i_k} > \max Z_i$ then $\forall CS : C \in CS$ and $C \in Z_i$ it holds that $CS \notin \{CS^*\}$.*

*Proof.* Suppose that coalition $C_i$ is the coalition with the biggest value in set $Z_i \subseteq \mathcal{C}_i$ and coalitions $C_{i_l}, \ldots, C_{i_k}$ are the coalitions with the smallest value in lists $i_1, \ldots, i_k$ respectively. Now consider any coalition structure $CS$ which contains coalition $C_i$. If $CS$ is an optimal structure no other structure can have value greater than this. Now, consider structure $CS' = CS \setminus \{C_i\} \cup \{C_{i_1} \cup \ldots \cup C_{i_k}\}$ where $C_{i_1}, \ldots, C_{i_k}$ are all disjoint and $C_{i_1} \cup \ldots \cup C_{i_k} = C_i$. If the sum of the smallest values in sets $\mathcal{C}_{i_l}, \ldots, \mathcal{C}_{i_k}$ is greater than the biggest value in set $Z_i \subseteq \mathcal{C}_i$ then clearly for all coalitions $C_{i_l}, \ldots, C_{i_k}$ in $\mathcal{C}_{i_l}, \ldots, \mathcal{C}_{i_k}$ respectively, and any $C_i \in Z_i \subseteq \mathcal{C}_i$, $v(C_{i_1}) + \ldots + v(C_{i_k}) > v(C_i)$ and so $v(CS') > v(CS)$. Therefore, following Theorem 1, for any partition of $i$, $i_1, \ldots, i_k$ such that $i_1 + \ldots + i_k = i$, if the sum of the minimum values in sets $i_1, \ldots, i_k$ is greater than the maximum value in set $Z_i \subseteq \mathcal{C}_i$ then no coalition in $Z_i \subseteq \mathcal{C}_i$ can be in an optimal coalition structure and so no $CS$ which contains a coalition in $Z_i \subseteq \mathcal{C}_i$ can belong to $\{CS^*\}$.

For any set containing non-trivial coalitions of size $i$, it is possible to compute the set of all integer partitions of value $i$. Denote such a set of integer partitions by $P(i)$. Furthermore, for each $p' \in P(i)$ it is possible to compute a value $d_i(p')$ as in Lemma 3. Now, following the same lemma, we can compare every coalition $C$ of size $i$ with $d_i(p')$ and immediately disregard those for which $v(C) < d_i(p')$. In fact, there is no need to apply Lemma 3 to all partitions in $P(i)$ but only to the partition $p'' \in P(i)$ such that $d_i(p'')$ is maximal in $P(i)$. Clearly, if for a coalition $C$ of size $i$ it holds that $v(C) < d_i(p')$ it also holds that $v(C) < d_i(p') \leq d_i(p'')$. Such a maximal value will be referred to as the *domination value* of coalitions of size $i$. More formally:

**Definition 1.** *For any set containing coalitions of size $i$ and every partition $p \in P(i)$,* domination value *$\tilde{d}_i$ is the highest value of $d_i(p)$ for all $p \in P$, or $\tilde{d}_i = \max_{p \in P} d_i(p)$.*

Following Lemma 1, if a value of any coalitions $C$ of size $i$ is less than the domination value $\tilde{d}_i$ then there exists a dissociation of $C$ with a greater value than $v(C)$. In this instance, $C$ cannot be in any optimal structure. This is one of the properties exploited in the filter rules presented in the next section.

## 3 Filter Rules

In this section, we propose filter rules that can be applied both while calculating the values of all the coalitions in $\mathcal{F}$ (thus, generating $V(\mathcal{F})$) and while searching through the (sub-)spaces of coalition structures. We will refer to coalitions for which it was determined that they cannot belong to any optimal structure as *not-promising*. All the other coalitions will be called *promising*. We denote both of these disjoint sets by $\mathcal{F}_{np} \subseteq \mathcal{F}$ and $\mathcal{F}_p \subseteq \mathcal{F}$, respectively. Initially, before the filter rules are applied, it is assumed that all coalitions are promising, *i.e.* $\mathcal{F}_p = \mathcal{F}$. Furthermore, we will refer to a subset of coalition values $Z \subseteq V(\mathcal{F}_p)$ $(Z \subseteq V(\mathcal{F}_{np}))$ as promising (not promising).

### 3.1 Filter rules for input calculation

While calculating all coalition values in the input, Lemma 2 can be applied to highlight those coalitions that are not promising. However, dissociating every coalition into all potential sub-coalition combinations is usually inefficient for coalitions of greater size. It should be left to a system designer to decide into how many sub-coalitions a currently computed coalition should be disaggregated. The natural and possibly most efficient choice is to consider the singleton partition of a coalition, *i.e.* the decomposition of the coalition into all of the individual agents who participate in this coalition. Clearly, in the process of computing the value of a given coalition, all agents who co-operate in this coalition must be known and such a disaggregation can be easily performed. Therefore, following Lemma 2 we propose the first filter rule:

**FR1** For any non-trivial coalition $C$ that can be divided into $k$ disjoint singleton sub-coalitions $C_1, \ldots, C_k$ where $C_1 \cup C_2 \ldots \cup C_k = C$ and $\forall i = 1, \ldots, k$ $|C_i| = 1$; if it is the case that either (i) $v(C_1) + \ldots v(C_k) > v(C)$ or (ii) $\sum_{i=1}^{j} v(C_i) > v(C)$ where $j < k$ then all such coalitions $C \in \mathcal{F}_{np}$.

Naturally, we can relax the constraint that $|C_i| = 1$ to $|C_i| \leq s$ where $1 < s < |A|$ depending on into how many partitions the system designer wishes to divide coalition $C$. The computational cost of applying **FR1** should be balanced against potential gains.

*Example 1.* Consider a 5 agent system $A = \{a_1, a_2, a_3, a_4, a_5\}$ with the following coalition values: $v(C) = 14$ for $C = A$; $\forall |C| = 1$ $v(C) = 3$; $\forall |C| = 2$ $v(C) = 5$; $\forall |C| = 3$ $v(C) = 10$;, and $\forall |C| = 4$ $v(C) = 12$. Observe that the sum of any two coalitions of size 1 is strictly greater than the value of any coalition of size 2. Furthermore, the value of the grand coalition is smaller than the value of a non-cooperative coalition structure. Thus, **FR 1** filters out all coalitions of size two as well as the grand coalition. However, following Lemma 1 the system designer may consider all dissociations of a coalition into exactly two disjoint sub-coalitions in the same way as in the DP algorithm presented in [7]. Since the value of any coalition of size 4 is smaller than the value of any coalition of size 1 added to the value of a disjoint coalition of size 3, then coalitions of size 4 are not promising.

Now, consider the domination value. Lemma 3 immediately yields the following filter rule:

**FR 2** For any subset $Z_s \subseteq \mathcal{C}_s$ of coalitions of size $s$, if $\tilde{d_s} > \max Z_s$ then all coalition from $Z_s$ are not promising. More formally: $\forall Z_s \subseteq \mathcal{C}_s$, if $\tilde{d_s} > \max Z_s$ then $Z_s \in \mathcal{F}_{np}$ and $Z_s \notin \mathcal{F}_{p}.$

**FR2** can be applied by the system designer to every coalition $C$ of size $i$ immediately after $v(C)$ has been calculated, *i.e.* in the same manner as **FR1**. Alternatively, if $|Z_i| \geq 2$, the value of $\max Z_i$ can be recorded while calculating coalition values in this subset and **FR2** can be applied after this process has been finished. We illustrate the functioning of **FR2** with the following example:

*Example 2.* Suppose the coalition values for four agents $A = \{a_1, a_2, a_3, a_4\}$ are as follows: $\forall |C| = 1, v(C) \in \langle 4, 7\rangle, \forall |C| = 2, v(C) \in \langle 5, 7\rangle, \forall |C| = 3, v(C) \in \langle 7, 11.5\rangle$ and $v(A) = 11$.[3] The domination values for the coalitions of size 2 to 4 are computed as follows: $\tilde{d_2} = \min S_1 + \min S_1 = 4+4 = 8$, $\tilde{d_3}, = \max\{3 \times \min S_1, \min S_1 + \min S_2 \} = 12$, $\tilde{d_4}, = \max\{ 4 \times \min S_1, 2 \times \min S_1 + \min S_3, 2 \times \min S_2 \} = 16$, Observe that both $\tilde{d_2} > \max S_2$ and $\tilde{d_4} > \max S_2 = v(A)$. Intuitively, this means that for every coalition of size 2 and 4, there exists a dissociation of this coalition into subcoalitions which have value greater than the value of the coalition and so following previous reasoning, no coalition of size 2 or 4 can be in an optimal structure and no structure containing these coalitions can be in an optimal set.

### 3.2 Filter rules for search of coalition structure space

As mentioned in the Introduction, the anytime algorithm of Rahwan *et al.* divides $\Pi$ into sub-spaces containing structures which are identical w.r.t. size of the coalitions involved. In a system of $|A|$ agents let $S^* := \{m_1, \ldots, m_k\}$ denote the (currently) most promising subspace, where $m_1, \ldots, m_k$ represent sizes of the coalition involved ($\sum_{i=1}^{k} m_i = |A|$ and $k \geq 2$). To search $S^*$, the values of all the coalition structures belonging to this sub-space should be computed unless it is proven beforehand that they cannot belong to $\{CS^*\}$.[4] The general form of a coalition structure in $S^*$ is $\{C_1, \ldots, C_k\}$ such that all of $C_1, \ldots, C_k$ are disjoint and $\forall i = 1, \ldots, k\ C_i \in \mathcal{C}_{m_i}$. Let $CS_N^*$ denote the coalition structure with the highest value found thus far. Rahwan *et al.* propose a filter rule that, based on the statistical information gathered about the coalition value input, avoids those structures which cannot be optimal. For $k \geq 3$ and $k-1 \geq l \geq 1$ it holds that:

**B&B** If $\sum_{i=1}^{l} v(C_i) + \sum_{i=l+1}^{k} \max \mathcal{C}_{m_i} \leq v(CS_N^*)$ then no structures in $S^*$ can be optimal, to which simultaneously belong all of $C_1, \ldots, C_l$.

This filter rule ensures that, for a particular structure under consideration $\{C_1, \ldots, C_k\}$, if the combined value of first $1 \leq l \leq n-1$ coalitions ($\sum_{i=1}^{l} v(C_i)$) plus the value of the sum of maximum values of coalitions in the remaining sets $\mathcal{C}_{m_{l+1}}$, ..., $\mathcal{C}_{m_k}$ is less

[3] For example, the notation $v(C) \in \langle 4, 7\rangle$ means that $v(C)$ can be any real value higher than or equal to 4 and lower than or equal to 7 and that the minimal value of all such coalitions is 4 and the maximal is 7.

[4] Or unless it is proven that an optimal coalition structure in this sub-space has been found.

than the current optimum value $v(CS^*_N)$ then no structures to which all of $C_1, \ldots, C_l$ simultaneously belong will be considered in the optimal CSG process.[5]

*Example 3.* For $|A| = 9$ agents let $S^* := \{1, 2, 2, 2, 2\}$ be the (currently) most promising subspace, $CS^*_N = 28$, $v(\{a_1\}) + v(\{a_2, a_3\}) = 9$, $v(\{a_4, a_5\}) = 4$, $v(\{a_4, a_6\}) = 6$ and $\max \mathcal{C}_2 = 7$. Following **B&B**, since $v(\{a_1\}) + v(\{a_2, a_3\}) + v(\{a_4, a_5\}) + \max \mathcal{C}_2 + \max \mathcal{C}_2 < CS^*_N$ then no structures in $S^*$ can be optimal, to which simultaneously belong all of $\{1\}$, $\{2, 3\}$ and $\{4, 5\}$. These structures are: $\{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}, \{a_6, a_7\}, \{a_8, a_9\}\}$, $\{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}, \{a_6, a_8\}, \{a_7, a_9\}\}$, and $\{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}, \{a_6, a_9\}, \{a_7, a_8\}\}$. Thus, in this example branch-and-bound rule saves on calculation time by avoiding calculations that lead to three structures which cannot be optimal. Considering $\{a_4, a_6\}$, since $v(\{a_1\}) + v(\{a_2, a_3\}) + v(\{a_4, a_6\}) + \max \mathcal{C}_2 + \max \mathcal{C}_2 > CS^*_N$ then condition in **B&B** does not hold.

The above branch-and-bound technique is based on basic statistical information collected about $\mathcal{C}_m$, namely the maximum value of coalitions in this set ($\max \mathcal{C}_m$). Assuming that this information is known about some subsets $Z_m \subseteq \mathcal{C}_m$ for $m = m_1, \ldots, m_k$ then the filter rule above can be generalized as follows:

**FR 3** If $\sum_{i=1}^{k} \max Z_{m_i} < v(CS^*_N)$, where $k \geq 2$, then no structures in $S^*$ can be optimal, in which simultaneously $C_1 \in Z_{m_1}, \ldots$ and $C_k \in Z_{m_k}$.[6]

*Example 4.* In the system of 9 agents let $S^* := \{1, 2, 2, 4\}$ be the (currently) most promising subspace, $CS^*_N = 25$, $\max\{Z_1 := \{\{a_1\}, \{a_2\}, \{a_3\}\}\} = 4$, $\max\{Z_2 := \{\{a_2, a_3\}, \{a_4, a_5\}, \{a_6, a_7\}, \{a_8, a_9\}, \{a_1, a_3\}, \{a_1, a_2\}\}\} = 6$ and $\max\{Z_4 := \mathcal{C}_4\} = 7$. Following **FR3**, since $\max Z_1 + 2 \times \max Z_2 + \max Z_4 < CS^*_N$ then no structures in $S^*$ containing all of $C_1, C_2, C_3, C_4$ such that $C_1 \in Z_1$, $C_2 \in Z_2$, $C_3 \in Z_2$, and $C_4 \in Z_4$ can be optimal. In this example **FR3** saves on calculation time by avoiding $\{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}, \{a_6, a_7, a_8, a_9\}\}$, $\{\{a_2\}, \{a_1, a_3\}, \{a_4, a_5\}, \{a_6, a_7, a_8, a_9\}\}$, and $\{\{a_1, a_2\}, \{a_3\}$,$\{a_4, a_5\}, \{a_6, a_7, a_8, a_9\}\}$.Note that there are certain combinations of $C_1 \in Z_1, C_2 \in Z_2, C_3 \in Z_2, C_4 \in Z_4$ for which $\{C_1, C_2, C_3, C_4\}$ is not a coalition structure. For instance, any combination $\{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}, \{a_1, a_2, a_3, a_4\}\}$ is neither exhaustive nor disjoint. Although, we are not interested in such combinations, it can be observed that the inequality in **FR3** holds for them as well.

## 4 Application

To test the effectiveness of the filter rules we employ them in the state-of-the-art distributed coalition value calculation algorithm of Rahwan and Jennings [1] (DCVC from

[5] Note that Rahwan *et al.* focused on CSG($CS^*$) problem so that in their branch-and-bound filter rule there is weak inequality sign '$\leq$'. To use this filter rule in the solution of CSG($\{CS^*\}$) problem the sign " $<$ " should be assumed. This holds for the other inequality conditions in this paper.

[6] Note that certain combination of $C_1 \in Z_{m_1}, \ldots, C_k \in Z_{m_k}$ are neither disjoint nor exhaustive, *i.e.* they are not proper coalition structures as assumed in this paper. We leave them aside as not being relevant to our analysis. See Example 4.

now on) and in the state-of-the-art anytime CSG algorithm of Rahwan *et al.* [2] (RCSG from now on).

Apart from the decentralised design, the key advantage of DCVC is that calculated coalition values in $V(\mathcal{F})$ are ordered in a unique way ensuring that only $V(\mathcal{F})$ and not $\mathcal{F}$ must be kept in memory. Coalition values structured in the same way as in DCVC are used as an input to RCSG. However, it is not trivial to connect both algorithms as the former one is distributed whereas the latter one is centralised. This means that, at the moment when calculations in DCVC are complete, every agent knows only a fraction of $V(\mathcal{F})$ needed in RCSG. Thus, a resource-consuming data transfer has to take place.

An application of any filter rules always requires to balance computational costs with potential gains. We will show that filter rules **FR1**, **FR2** and **FR3** can be applied at a relatively low computational cost, and using them results in:

1. A substantial decrease the number of coalitions to be transferred; and
2. An increase the efficiency of the promising structure sub-space search in RCSG.

### 4.1 Application of FR1, FR2 and FR3 in DCVC

In DCVC, the space of all coalitions is represented as a set of lists $L_1, \ldots, L_{|A|}$ where list $L_i$ contains all coalitions of size $i$. Within each list, the coalitions are ordered w.r.t. the agents they are made of. This ordering ensures that the agent composition of every coalition in each list can be derived from the place in the list it occupies. In terms of the notation introduced in Section 2 we may write $L_i = \overrightarrow{\mathcal{C}_i}$, *i.e.* every list is an ordered set of the values in $\mathcal{C}_i$. Such lists for a system of 6 agents $a_1, a_2, ..., a_6$, all with equal computational capabilities, are presented in Figure 1.[7]

All lists are divided into disjoint segments proportional to the agents' computational capabilities. Usually every agent is assigned two segments, one in the upper and one in the lower part of the list. We will denote both segments assigned to agent $a_i$ in list $L_m$ by $\mathcal{L}_m^U(a_i)$ and $\mathcal{L}_m^L(a_i)$, respectively. As not all the list are exactly divisible by $|A|$, variable $\alpha$, depicted in Figure 1, is used to distribute 'left over' coalitions as equally as possible between agents. 'Left over' coalition values assigned to agent $a_i$ will be denoted by $\mathcal{L}_m^O(a_i)$. Overall, the above methods to distribute $V(\mathcal{F})$ have the advantage that all agent computations are finished (almost) simultaneously, even when some coalitions require more arithmetic operations than others and when agents have different processing capabilities. The allocation process is described in detail in [1].

After the allocation of segments has been performed, agents sequentially compute values in lists starting from $L_1$. Since there are $|A|$ coalitions in $L_1$, each agents is assigned exactly one value from this list to calculate. Let us assume that every agent transmits this value to all the other agents in the system. By doing so, agents are ready to apply **FR1** while computing coalition values in $L_2$. Additionally, they are able to efficiently compute the value of a structure consisting of every promising coalition $C$ and singletons, *i.e.* $\{C, \{j\} : j \in A \backslash C\}$.

[7] Note that in Figure 1 numbers in lists are a shorthand notation representing agents. It should be emphasized that every list contains coalition values from $V(\mathcal{F})$ and not coalitions from $\mathcal{F}$ themselves. Thus, the representation of lists $L_1, ..., L_{|A|}$ in Figure 1 should be read not as coalitions of agents from $\mathcal{F}$ but as their numerical values from $V(\mathcal{F})$.

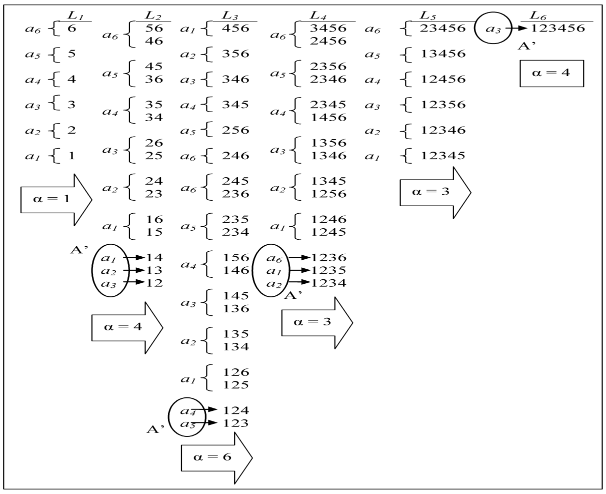


Figure 1: Division of $V(\mathcal{F})$ in a system of 6 agents

For example, in Figure 1, while computing the values of coalition $\{a_4, a_5\}$ and $\{a_3, a_6\}$ in $\mathcal{L}_2^U(a_5)$, agent $a_5$ applies **FR1** and compares: $v(\{a_4\}) + v(\{a_5\})$ with $v(\{a_4, a_5\})$ and $v(\{a_3\})+v(\{a_6\})$ with $v(\{a_4, a_6\})$. Furthermore, this agent calculates the values of $\{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4, a_5\}, \{a_6\}\}$ and $\{\{a_1\}, \{a_2\}$,$\{a_3, a_6\}$ ,$\{a_4\}, \{a_5\}\}$. Let $CS_N^*(a_i)$ denote a coalition structure with the highest value that agent $a_i$ found in its assigned segments.

In DCVC, when the computational capabilities of all the agents are symmetric, each of them receives an (almost) equal fraction of $\mathcal{F}$ to calculate $V(\mathcal{F})$ (as in Figure 1). The total number of values assigned to every agent is equal to either $\left\lfloor \left(2^{|A|} - 1\right) /|A| \right\rfloor$ or $\left\lfloor \left(2^{|A|} - 1\right) /|A| \right\rfloor + 1$, depending on the distribution of 'left-over' coalitions. For example, for 20 agents this amounts to either 52428 or 52429 and for 30 agents to either 53687091 or 53687092. Clearly, transferring such numbers of coalition values from DCVC to RCSG is a potential bottleneck in the system performance. The application of **FR1** helps to reduce the transfer load but its effectiveness depends on the values of the singletons. In some systems, these values might be lower than a value of any coalition. In such a case, following Lemma 2, **FR1** can be extended to consider other partitions of a coalition $C$ whose value is being computed. However, dissociations other than into singletons may considerably increase computational cost and should be weighted against potential gains. Consequently, we propose another approach that can significantly reduce the transfer load. Since RCSG searches only through promising subspaces of $\mathit{\Pi}$, then only some lists of coalition values have to be transferred. This means that the procedure to evaluate whether a given subspace is promising should be removed from the RCSG and incorporated into DCVC.

In RCSG a subspace $S$ is considered to be promising if its upper bound $UB_S := \sum_{\forall m_i \in S} maxL_{m_i}$ is no smaller than the lower bound of the entire system $LB := \max\{v(CS_N^*), \max\{Avg_S\}\}$, where $Avg_S = \sum_{\forall m_i \in S} avgL_{m_i}$ is proven to be the average value of every subspace. In DCVC, where calculations are distributed among all of the agents, statistics such as maximum, minimum and average coalition values cannot be computed without an exchange of information among agents. However, if agents record maximum, minimum and average values in every segment they are assigned to calculate then after all calculations are (almost simultaneously) finished every agent $a_i$ can broadcast this statistical data along with $CS^*_N(a_i)$. Using this knowledge, the agents can calculate, for every list $L_m$, the values of $maxL_m$, $avgL_m$, and $minL_m$. Both the maximum and average values are then used, by the agents, to determine which sub-spaces are promising. Consequently, by exchanging some basic information about segments the agents can prune the search space at the end of DCVC in the same way as the centre would do it in RCSG.

The subspace with the maximum upper bound is the most promising, *i.e.* $S^* = \arg\max_S UB_S$, and is searched first. It might happen that the unique $CS^*$ (*i.e.* when $|\{CS^*\}| = 1$) is found in this sub-space. Thus, in order to reduce potential transfer load, we assume that only lists needed to construct coalition structures in $S^*$ are initially passed on from DCVC to RCSG.

In the process of transmitting the lists which are required to generate all structures in $S^*$, both **FR2** and a version of **FR3** are applied. Both filter rules can reduce the transfer load even further. Before the transmission begins, agents calculate the domination value $\tilde{d}_m$ for every list $L_m : m \in S^*$ using the relevant statistics.[8] Segments assigned to agent $a_i$ which meet the condition $\max \mathcal{L}_m^{U/L/O}(a_i) < \tilde{d}_m$ are filtered out as not promising. Furthermore, for all segments in which $\max \mathcal{L}_m^{U/L/O}(a_i) \geq \tilde{d}_m$, it is determined if individual coalition values within these segments, not filtered by **FR1** are not smaller than $\tilde{d}_m$. Individual coalitions with such values become not promising. The version of **FR2** that concerns segments will be denoted as **FR2a** whereas the version that concerns individual coalitions as **FR2b.**

Although **FR3** has been constructed to increase efficiency of searching $\Pi$, a version of this filter rule can also be applied before data transfer. Let the most promising subspace be $S^* := \{m_1, \ldots, m_k\}$. Since every agent $a_1$ knows $\max \mathcal{L}_m^{U/L/O}(a_i)$ and $\max L_m$ for all $m = m_1, ..., m_k$ the following version of **FR3** (denoted as **FR3a**) can be applied to decide whether either $\mathcal{L}^U_{m_j}(a_i)$, $\mathcal{L}^L_{m_j}(a_i)$ or $\mathcal{L}^O_{m_j}(a_i)$, where $m_j \in S$, can be filtered out as not promising:

$$\sum_{m \in S \setminus \{m_j\}} \max L_m + \max \mathcal{L}_{m_j}^{U/L/O}(a_i) < v(CS^*_N). \tag{1}$$

If the segment cannot be filtered out as not promising then the above filter rule can be applied to every individual coalition $C$ within this segment not filtered out by **FR1** or **FR2a/b**. In such a situation, $\max \mathcal{L}_{m_j}^{U/L/O}(a_i)$ in formula (1) can be replaced with $v(C)$, where $v(C) \in \mathcal{L}_{m_j}^{U/L/O}(a_i)$. and this version of **FR3** is denoted **FR3b**. Finally,

[8] We leave it to the discretion of the system designer as to the exact architecture of domination value calculations. At the end of DCVC, agents can either calculate all needed domination values individually or distribute calculations among themselves in such a way that the most efficient agents calculate domination values for the lists of coalitions with the highest cardinality and *vice versa.*

after all the filter rules have been applied, the agents transfer only those values which are promising.

To summarize, in order to reduce the transfer load from DCVC to RCSG we propose the following extension of the former algorithm:

Step 1: Agents exchange among themselves values of singleton coalitions in $L_1$. While calculating value of any non-trivial coalition $C$ they: (1.1) apply **FR1**; (1.2) compute $CS^*_N(a_i)$ and (1.3) Update and store the statistics about their segments $\mathcal{L}_m^{U/L/O}$;

Step 2: After calculating the values in all of their segments, agents exchange among themselves the segments' statistics and $CS^*_N(a_i)$. Using this information, they determine the most promising sub-space $S^*$ and domination values $\tilde{d}_m$ for every list $L_m : m \in S^*$ that needs to be transmitted;

Step 3: Just before transmission takes place each agent applies both **FR2a** and **FR3a** to segments $\mathcal{L}_m^{U/L/O}$, where $m \in S^*$. If a segment is not filtered out then **FR2b** and **FR3b** are applied again to individual coalitions within this segment that have not been filtered out by **FR1**. Only the values of promising coalitions are transferred.[9]

Finally, it should be observed that even if the coalition value calculation algorithm was not distributed, then **FR1**, **FR2b** and **FR3b** can be still be applied to determine the not promising coalitions. In the simulations we will show that it considerably reduces the running time of RCSG.

### 4.2 Application of FR3 in RCSG

The strength of RCSG in searching the promising sub-spaces is that it avoids creating both invalid and repeated coalition structures as described in [9]. Suppose that $S^* := \{m_1, \ldots, m_k\}$ is the most promising subspace in a system of $|A|$ agents and that the value of the particular partial structure $\{C_{m_1}, ..., C_{m_l}\}$, where $l < k$, has already been calculated. Let $\overrightarrow{A'}$ be the ordered set of agents which do not belong to any coalition in this partial structure, *i.e.* $\overrightarrow{A'} := \overrightarrow{A} \backslash \{C_{m_1}, ..., C_{m_l}\}$. The method proposed by Rahwan *et al.* in [9] cycles through all $m_{l+1}-$element combination from $\overrightarrow{A'}$ to construct all feasible coalitions $C_{m_{l+1}}$. The cycle is designed in such a way that all $m_{l+1}-$combinations which must contain the agent in $\overrightarrow{A'}$ with the lowest index are considered first and all $m_{l+1}-$combinations which must contain the second agent in $\overrightarrow{A'}$ but not the first one are considered in the next step and so on. As explained in Subsection 3.2, after constructing every $C_{m_{l+1}}$ RCSG applies **B&B** filter rule to decide whether any partial structure $\{C_{m_1}, ..., C_{m_l}, C_{m_{l+1}}\}$ is promising. A version of **FR3** can be used to identify groups rather than only individual partial structures. To this end, agents in DCVC have to gather additional information about some particular segments in lists $L_1, ..., L_{|A|}$. Let $\overrightarrow{Z}_m(i)$ denote a segment of list $L_m$ such that $a_i$ is the first agent in every coalition in $\overrightarrow{Z}_m(i) \subseteq L_m$. For example, referring to Figure

[9] To indicate the position of each promising coalition value and maintain the list structure we propose to transmit a characteristic bit vector alongside the stream of values. In such a vector 1 indicates value of a promising coalition and 0 a not promising one.

1, $\overrightarrow{Z}_3(2)$ is $\{\{2,5,6\},\{2,4,6\},\{2,4,5\},\{2,3,6\},\{2,3,5\},\{2,3,4\}\}$. Assume that, while calculating coalition values in DCVC, the agents record $\max \overrightarrow{Z}_m(i)$ for every combination of $m = 1, ..., |A|$ and $a_i \in A$. In such a case, before cycling through all $m_{l+1}-$combinations which contain the agent in $\overrightarrow{A'}$ with the lowest index, the following version of **FR3** (denoted as **FR3c**) can be applied, by a centre in RCSG. If

$$\sum_{i=1}^{l} v(C_i) + \max \overrightarrow{Z}_{l+1}(i) + \sum_{i=l+2}^{k} \max L_{m_i} < v(CS_N^*),$$

then any partial structure $\{C_{m_1}, ..., C_{m_l}, C_{m_{l+1}}\}$ such that $C_{m_{l+1}} \in \overrightarrow{Z}_{l+1}$ is not promising. **FR3c** is a generalization of RCSG **B&B** as it is applicable to groups rather than individual coalitions.

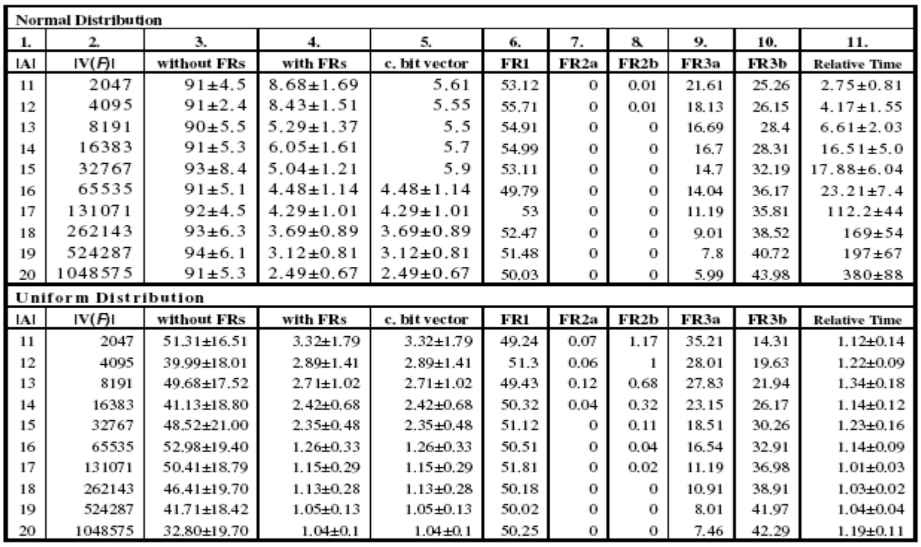
### 4.3 Numerical Simulations

In the numerical simulations we compare the sequential execution of DCVC and RCSG with and without filter rules. The following assumptions are imposed: (i) the calculation of any particular coalition values in DCVC takes no time; and (ii) any data transfers are instantaneous. We focus on the percentage of $V(\mathcal{F})$ which does not have to be transmitted from DCVC to RCSG due to applied filter rules. Additionally, we demonstrate that, for a normal distribution of coalition values, filter rules considerably improve the performance of the CSG process in RCSG.

Following [2, 5], we evaluate the algorithms under two different assumptions as to the probability distributions of coalition values:[10] $(ND)$ Normal: $v(\mathbf{C}) = max(0, |C| \times p)$, where $p \in N(\mu = 1, \sigma = 0.1)$; and $(UD)$ Uniform: $v(\mathbf{C}) = max(0, |C| \times p)$, where $p \in U(a, b)$, where $a = 0$ and $b = 1$. For a system of $|A| = 11 \ldots 20$ agents, we ran both versions of the algorithms 25 times and reported the results within a $95\%$ confidence interval. The algorithms were implemented in MATLAB.

The results are presented in Table 1. Column 2 shows the number of all coalition values $|\mathcal{F}|$ and Column 3 percentage of $V(\mathcal{F})$ that needs to be transmitted from DCVC to RCSG without filter rules. Column 4 shows the actual percentage of $V(\mathcal{F})$ transferred when filter rules are applied. Column 5 contains the size of the characteristic bit vector (as a percentage of $V(\mathcal{F})$) that must be transferred simultaneously with the values in Column 4 (see Footnote 8). The next five columns present the individual contribution of filter rules **FR1**, **FR2a**, **FR2b**, **FR3a** and **FR3b** to the reduction of the overall transfer load between Columns 3 and 4. The last column shows the running time of RCSG based on the unfiltered input and without **FR3c** divided by the running time of this algorithm based on filtered input and with **FR3c**.

[10] We omit the sub- and super-additive cases as their solution is straightforward.



| 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. |
|---|---|---|---|---|---|---|---|---|---|---|
| **Normal Distribution** | | | | | | | | | | |
| \|A\| | $\|V(F)\|$ | without FRs | with FRs | c. bit vector | FR1 | FR2a | FR2b | FR3a | FR3b | Relative Time |
| 11 | 2047 | 91±4.5 | 8.68±1.69 | 5.61 | 53.12 | 0 | 0.01 | 21.61 | 25.26 | 2.75±0.81 |
| 12 | 4095 | 91±2.4 | 8.43±1.51 | 5.55 | 55.71 | 0 | 0.01 | 18.13 | 26.15 | 4.17±1.55 |
| 13 | 8191 | 90±5.5 | 5.29±1.37 | 5.5 | 54.91 | 0 | 0 | 16.69 | 28.4 | 6.61±2.03 |
| 14 | 16383 | 91±5.3 | 6.05±1.61 | 5.7 | 54.99 | 0 | 0 | 16.7 | 28.31 | 16.51±5.0 |
| 15 | 32767 | 93±8.4 | 5.04±1.21 | 5.9 | 53.11 | 0 | 0 | 14.7 | 32.19 | 17.88±6.04 |
| 16 | 65535 | 91±5.1 | 4.48±1.14 | 4.48±1.14 | 49.79 | 0 | 0 | 14.04 | 36.17 | 23.21±7.4 |
| 17 | 131071 | 92±4.5 | 4.29±1.01 | 4.29±1.01 | 53 | 0 | 0 | 11.19 | 35.81 | 112.2±44 |
| 18 | 262143 | 93±6.3 | 3.69±0.89 | 3.69±0.89 | 52.47 | 0 | 0 | 9.01 | 38.52 | 169±54 |
| 19 | 524287 | 94±6.1 | 3.12±0.81 | 3.12±0.81 | 51.48 | 0 | 0 | 7.8 | 40.72 | 197±67 |
| 20 | 1048575 | 91±5.3 | 2.49±0.67 | 2.49±0.67 | 50.03 | 0 | 0 | 5.99 | 43.98 | 380±88 |
| **Uniform Distribution** | | | | | | | | | | |
| \|A\| | $\|V(F)\|$ | without FRs | with FRs | c. bit vector | FR1 | FR2a | FR2b | FR3a | FR3b | Relative Time |
| 11 | 2047 | 51.31±16.51 | 3.32±1.79 | 3.32±1.79 | 49.24 | 0.07 | 1.17 | 35.21 | 14.31 | 1.12±0.14 |
| 12 | 4095 | 39.99±18.01 | 2.89±1.41 | 2.89±1.41 | 51.3 | 0.06 | 1 | 28.01 | 19.63 | 1.22±0.09 |
| 13 | 8191 | 49.68±17.52 | 2.71±1.02 | 2.71±1.02 | 49.43 | 0.12 | 0.68 | 27.83 | 21.94 | 1.34±0.18 |
| 14 | 16383 | 41.13±18.80 | 2.42±0.68 | 2.42±0.68 | 50.32 | 0.04 | 0.32 | 23.15 | 26.17 | 1.14±0.12 |
| 15 | 32767 | 48.52±21.00 | 2.35±0.48 | 2.35±0.48 | 51.12 | 0 | 0.11 | 18.51 | 30.26 | 1.23±0.16 |
| 16 | 65535 | 52.98±19.40 | 1.26±0.33 | 1.26±0.33 | 50.51 | 0 | 0.04 | 16.54 | 32.91 | 1.14±0.09 |
| 17 | 131071 | 50.41±18.79 | 1.15±0.29 | 1.15±0.29 | 51.81 | 0 | 0.02 | 11.19 | 36.98 | 1.01±0.03 |
| 18 | 262143 | 46.41±19.70 | 1.13±0.28 | 1.13±0.28 | 50.18 | 0 | 0 | 10.91 | 38.91 | 1.03±0.02 |
| 19 | 524287 | 41.71±18.42 | 1.05±0.13 | 1.05±0.13 | 50.02 | 0 | 0 | 8.01 | 41.97 | 1.04±0.04 |
| 20 | 1048575 | 32.80±19.70 | 1.04±0.1 | 1.04±0.1 | 50.25 | 0 | 0 | 7.46 | 42.29 | 1.19±0.11 |

Table 1: Simulation results (all values, except for columns 1,2, and 11 are expressed in $\%$)

Consider $ND$ first. Without filter rules, about $90\%$ of $V(\mathcal{F})$ needs to be transmitted from DCVC to RCSG (Column 3). Filter rules reduce this number to about $9\% + 5.5\%$ for $n = 11$ and to around $2.5\%$+$2.5\%$ for $n = 20$ (Columns 4 and 5).[11] **FR1** is the most successful in filtering coalitions accounting for about $50\%$ of them. Consequently, both **FR2a** and **FR2b**, based on the same assumption, cannot offer any significant improvement upon this result. However, if the singleton coalitions have relatively low values, then **FR1** would not perform so well, and **FR2a/b** would become more profitable. A decreasing percentage of coalition values are ruled out by rule **FR3a**. The intuition behind the decreasing effectiveness of this filter rule is as follows. The higher the value of $|A|$, the longer the segments become. Consequently, there is a greater probability that the (randomly-drawn) extreme values in these segments are similar to each other. This reduces the effectiveness of filter rules based on segments' maxima. In contrast, **FR3b**, which focuses on the particular coalition values, has increasing success. We expect that for $n > 25$ only **FR1** and **FR3b** should be used. In conclusion, the combination of the filtered input and the application of **FR3c** in RCSG results in an exponentially faster performance of this algorithm.

For the $UD$ case, only about $50\%$ of $V(\mathcal{F})$ needs to be transmitted from DCVC to RCSG (Column 3) but filter rules are able to reduce this number: from about $2 \times (3.32\% \pm 1.79\%)$ for $n = 11$ to around $2 \times (1.04\% + 0.1\%)$ for $n = 20$. Analysis of the effectiveness of individual filter rules is similar to the $ND$ case. However, in contrast to $ND$, the combination of the filtered input and the application of **FR3c** does

[11] Note that for $n > 15$, it is more efficient to transfer the exact location of every coalition (in the form of an integer) that does not satisfy the filter rules rather than transfer the characteristic bit function since the former method requires less data.

not significantly outperform the standard RCSG. This is caused by the very nature of the uniform-distribution. Intuitively, since coalition values in any lists are dispersed neither reduced input nor **FR3c** perform much better than **B&B** in standard RCSG. However, our filter rules still prove its effectiveness by achieving large reduction in the transfer load.

## 5 Conclusions

In this paper we have discussed a number of techniques designed to increase the efficiency of CSG algorithms. In particular, we developed filter rules which can, for CFG representations, eliminate a significant proportion of not promising coalitions from the space of all coalition values. Such a (structured) space of coalition values acts as input to, among others, the state-of-the-art CSG algorithm of Rahwan *et al.* Although this algorithm has been demonstrated to be very effective, its particular drawback is its centralised nature. This is especially challenging as there exist already a very efficient algorithm for distributed coalition value calculation; this means that after the DCVC algorithm has been employed, all of the agents have to transmit all the values they have computed to a single entity. In this paper, we demonstrated that our proposed filter rules are extremely effective in reducing the size of this coalition value input to the CSG algorithm. Consequently, we showed how to efficiently bridge the gap between the decentralised DCVC and the centralised RCSG.

A natural follow up to our work, that we plan to explore, is to develop a distributed CSG algorithm that can be relatively easily constructed from the analysis in this paper.

## References

1. Rahwan, T., Jennings, N.: An algorithm for distributing coalitional value calculations among cooperating agents. Artificial Inteligence **(8-9)**(171) (2007) 535–567
2. Rahwan, T., Ramchurn, S., Dang, V., Giovannucci, A., Jennings, N.: Anytime optimal coalition structure generation. In: Proceedings of AAAI, Vancouver, Canada (2007) 1184–1190
3. Michalak, T., Dowell, A., McBurney, P., Wooldridge, M.: Optimal coalition structure generation in partition function games. In: Proceedings of ECAI 2008, Patras, Greece (2008)
4. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence **1**(101) (1998) 165–200
5. Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohme, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence **1-2**(111) (1999) 209–238
6. Dang, V., Jennings, N.: Generating coalition structures with finite bound from the optimal guarantees. In: Proceedings of AAMAS, New York, USA (2004)
7. Yeh, D.Y.: A dynamic programming approach to the complete. BIT **4**(26) (1986) 467–474
8. M.Rothkopf, Pekec, A., Harstad, R.: Computationally manageable combinatorial auctions. Management Science **8**(44) (1998) 1131–1147
9. Rahwan, T., Ramchurn, S., Dang, V., Giovannucci, A., Jennings, N.: Near-optimal anytime coalition structure generation. In: Proceedings of IJCAI, Hyderabad, India (2007)

# Cognitive Artifacts for Intelligent Agents in MAS: Exploiting Relevant Information Residing in Environments

Michele Piunti[1,2] and Alessandro Ricci[2]

[1] Institute of Cognitive Sciences and Technologies, ISTC-CNR, Rome, Italy.
[2] Alma Mater Studiorum, Università degli studi di Bologna, DEIS, Cesena, Italy.
{michele.piunti,a.ricci}@unibo.it

**Abstract.** Besides using languages and direct communication, humans adopt various kind of *artifacts* as effective means to represent and share knowledge, and finally support knowledge-based cooperation in complex work environments. Similarly to the human case, we argue that an analogous concept can be effective also in the context of cognitive multi-agent systems (MAS). Based on previous work on artifact-based environment and A&A conceptual framework, in this paper we investigate the use of a special kind of artifacts, cognitive artifacts, as computational entities designed to store, process and make available those information which is relevant for agents to coordinate their cooperative and distributed activities. After introducing the main concepts, we discuss some of the practical benefits of the approach through an experiment based on CARTAGO and *Jason* technologies, respectively a platform for developing artifact-based environments for MAS and for programming cognitive BDI-based agents, comparing different interaction styles for teams of goal-directed agents engaged in distributed cooperative works.

## 1 Introduction

According to conceptual frameworks such as Activity Theory and Distributed / Situated Cognition, in human cooperative activities *artifacts* play a fundamental role, deeply affecting the way in which people solve problems and perform individual / cooperative tasks, manage and share knowledge, cooperate and coordinate their work. More generally, artifacts are a way to conceive, structure and organise the *environment* where humans live and work. Recent agent literature highlighted the important role that the notion of environment can play in designing and engineering Multi-Agent Systems (MAS). The environment is thus conceived as the suitable locus to encapsulate services and functionalities in order to improve agent interaction, coordination, and cooperation [13]. In this view, a notion of artifact has been recently introduced by the A&A conceptual model as first-class entity to structure and organise agent computational environments, representing resources and tools[3] that agents may want to use to support

[3] In this context we consider the terms *artifacts* and *tools* as synonyms.

their activities, analogously to artifacts in the human case [8]. For knowledge sharing and coordination, *coordination artifacts*—i.e. calendars, programmable blackboards, schedulers, etc.—have been introduced in MAS to enable and mediate agent-agent interaction and communication. Besides communication models, there are many traits relating cognitive models of agency to A&A. From a cognitive perspective, artifacts can be viewed as external entities that agents may exploit to improve their repertoire of actions. For knowledge representation in particular, *cognitive artifacts*—defined by Norman as those artifacts that maintain, display or operate upon information in order to serve a representational function [7]—are essential tools for reasoning, heavily affecting human behavior. Once an agent uses an artifact, he is both relying on its functionalities and delegating part of his purposive activities on artifact functioning. In this view, artifacts play the role of suitable services that agent can exploit to externalise activities thus easing their computational burdens [4].

An additional aspect relating cognitive agents to artifacts concerns the nature of open systems, exacting agents with the ability to discover and learn affordances of resources which may not be known at design time. This may concern the ability to learn and co-use environmental and distributed entities in terms of subjective experiences and mental states (i.e. to achieve shared goals). In this paper we discuss the abilities of agents to exploit artifacts with representational functionalities, i.e. cognitive artifacts. This, we guess, has manifold benefits, in particular in making it more effective to organise and share distributed knowledge, reasoning about distributed work, retrieving information which is relevant with respect of the ongoing tasks, finally improving the coordination of agents activities. First we discuss the role played by cognitive artifacts upon the A&A model and CARTAGO technology (Section 2). Then, to provide some evidence of the overall idea, we describe the design model through the RoomsWorld scenario (Section 3), discussing an experiment (Section 4) where specific kind of cognitive artifacts, i.e. *log*s, are exploited to support the distributed work and interaction of a team of goal-oriented agents. CARTAGO [11] and *Jason* [1] are exploited as integrated technologies to implement respectively the artifact-based environment and the cognitive agents.

## 2 Cognitive Artifacts in Multi-Agent Systems

The notion of artifact in MAS has been introduced with the A&A conceptual model as first-class abstraction along with agents when modeling, designing, and developing MAS [8, 11]. The main inspiration of A&A comes from Activity Theory [6] along with Distributed Cognition and other movements inside cognitive science. These approaches promote the notion of artifacts and tools to play a pivotal (mediation) role in coping with the scaling up of complexity, in particular when social activities and interactions are concerned. In this view, the use of tools is an evolutionary accumulation and transmission of social knowledge, habits, memes, which not only influences the nature of external behavior, but also the mental functioning of individuals.

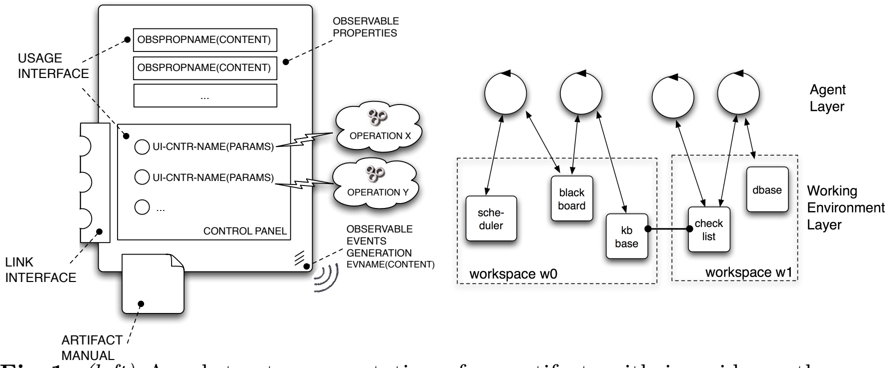


**Fig. 1.** *(left)* An abstract representation of an artifact, with in evidence the usage interface, with operation controls and observable properties, the manual and the link interface, used to link together artifacts. *(right)* An overview of a MAS according to A&A, with agents working with artifacts distributed in multiple workspaces.

By analogy, the basic idea of A&A is to define a notion of *artifact* in MAS as first-class computational entities representing resources and tools that agent can exploit to perform their tasks, in particular social tasks. Differently from agents, which can be described as autonomous, pro- active (goal-oriented) entities encapsulating the control of their behaviour, artifacts are a non-autonomous, *function-oriented* computational objects, i.e., encapsulating some kind of function (service) to be exploited by agents. The model of interaction between agents and artifacts is based on *use* and *observation*: agents use artifacts through their *usage interface*, which is a control interface, and gets their output by perceiving observable events generated with artifact functioning and observable properties which are part of the artifact state. Basically, A&A approach enriches the basic view of multi-agent systems as sets of agents communicating and cooperating solely through speech-act based message passing and ACL towards a view in which agents live into shared (distributed) computational environments which are part of MAS design, composed by dynamic sets of artifacts, created/disposed by agents themselves. Besides the notion artifact, A&A introduces the notion of *workspace* as logic container of agents and artifacts. Workspaces can be used to structure the overall sets of entities, defining a topology of MAS environment. In this view a MAS is conceived as a set of (distributed) workspaces, containing dynamic sets of agents working together by both directly communicating and sharing / co-using artifacts (Fig. 1 Right).

## 2.1 The Role of Cognitive Artifacts

As remarked in [7], cognitive artifacts shape the way by which individuals interact with their surroundings, shaping external activities results, in turns, in shaping internal ones. Besides the specific functionalities, Norman discussed the benefits of cognitive artifacts under two different aspects. From an individual point of view, artifacts basically make it possible to:

- Enhance agent cognitive capabilities, either by saving internal computational load or being merely memory enhancer, for instance creating observable cues

in order to highlight relevant information; Besides, agents are not forced to contemporaneously communicate within message passing but they can asynchronously interact according to emerging dynamics (*situated cognition*);
- Change the way a task gets done, by changing the actions required to the individuals for achieving a goal. In so doing cognitive artifacts can help individual agents to eliminate decision points, creating more rational decision alternatives and reducing the *fan-out* of a decision tree [5].

From a system perspective, cognitive artifacts can be useful to:

- Distribute the actions across time (*pre-computation*), by allowing to store information and enable information processing that can be performed even before the actual operation usage;
- Distribute the actions across agents (*distributed cognition*), asynchronously distributing tasks across agents, allowing social coordination and shared activities;

Among the artifacts cognitively used by humans we here mention activity lists or logs—useful to maintain and manage a shared list of task to do—and whiteboards—useful to maintain shared knowledge and provide suitable operation to access and update it. Analogously to the human case, we refer to *cognitive artifacts* for MAS those artifacts that are designed to maintain, make it observable, or operate information in order to serve a representational function for agents with respect to their environment and the work inside it. Hence, from a agent viewpoint, artifacts can be cognitively used once their representational contents can be mapped and translated into reasoning processes. These relations from cognitive agents to cognitive artifacts are, at least, bidirectional. On the one side artifact interface descriptions can be matched with agents epistemic (beliefs) and motivational (goals) mental states. On the other side, the external events coming from a cognitive artifact can be integrated at an architectural level by automatically promoting such events as "relevant" signals to be addressed to the reasoning processes.

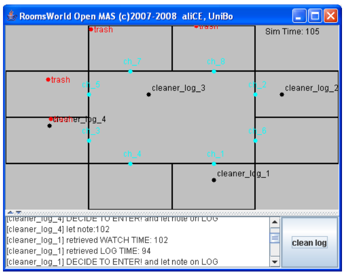
A first obvious outcome in interacting through artifacts in MAS concerns knowledge sharing, as a complimentary approach with respect to direct communication based on ACL. For this purpose, cognitive artifacts provide a natural support for time and space uncoupled communication, being then particularly useful in loose coupled and open systems, where agents need to cooperate without necessarily being in the same temporal and spatial context.

## 2.2 Cognitive Agents using CARTAGO Artifacts

By adopting a functional view, artifacts can be viewed as devices playing the role of targets for agents activities, or serving as means for goal-oriented agents to achieve their goals. To this end, the A&A meta-model defines a series of general properties by which artifact functionalities are defined in terms of *operations*, which can be triggered by agents via artifact's *usage interface* (Fig. 1 Left). Analogously to usage interface of artifacts in the real world, an artifact usage interface is composed by a set of *operation controls* that agents can use to trigger and control operation execution. Each operation control is identified by a label (typically equal to the operation name to be triggered) and a list of input parameters. An *operation* is the basic unit upon which artifact functionality is structured. The execution of an operation can result both in changes in the artifact's inner (i.e., non-observable) state, and in the generation of a stream of *observable events* that can be perceived by agents that are using or simply observing the artifact. Besides the operation controls, the usage interface might contain also a set of *observable properties*, i.e. properties whose dynamic values can be observed by agents without necessarily acting upon the artifact operations. Similarly to artifacts in the human case, in A&A each artifact is meant to be equipped with a "manual" that is intended to provide the description of the artifact's functions and usage interfaces. Finally, artifacts can be linked together (by agents) through a *link interface*, so to enable inter-artifact interactions and artifact compositionality.

Whereas A&A provide a conceptual underpinning on the agents and artifacts approach, CARTAGO [11] puts in practice the enabling technology to realise artifact-based MAS. CARTAGO offers a suitable integration technology supporting the implementation and the deployment of distributed, heterogeneous and open systems, where agents dwelling in different agent platforms can autonomously join and work together in distributed and shared workspaces. Current CARTAGO distribution[4] offers bridge mechanisms to be integrated by some well-known agent platforms (namely *Jason*, *Jadex*, 2APL). To allow cognitive agents to play in CARTAGO environments we consider basic sensory-motor aspects of a cognitive system. On the one hand agents have been equipped with mechanisms to interact with artifacts (*effectors*); on the other hand they also have been enabled to perceive events generated by artifacts during their operations (*sensors*). In this view, the integration approach has been realized at the language level, i.e. the set of artifact-related actions have been added to the repertoire of natively available actions. Therefore, the bridge mechanism introduced the notion of *agent body* as that part of an agent conceptually belonging to a workspace (once the agent is inside it) and containing those sensory-motor capabilities to interact with artifacts. Fig. 2 (Right) shows basic actions integrated in the body allowing agents to interact with CARTAGO artifact-based environments. In particular, the described actions make it possible for an agent to: join, move, and leave workspaces (1-3); use an artifact by acting on its control interface and perceive events generated by artifacts (4-7); observe artifact properties (8). As for the syntax, a pseudo-code first-order logic-like is adopted, while semantics is described informally. `use` is the basic action for triggering the execution of an operation, specifying operation control name and parameters, and optionally a sensor name. Sensors are conceived as a kind of body's *perceptual memory* to keep track of observable events generated by artifacts, possibly applying filters and specific kinds of "buffering" policies. If no sensor is specified,

[4] CARTAGO is available as an open source technology at: `http://www.alice.unibo.it/cartago`



```
[cleaner_log_4] DECIDE TO ENTER! and let note on LOG
[cleaner_log_4] let note:102
[cleaner_log_1] retrieved WATCH TIME: 102
[cleaner_log_1] retrieved LOG TIME: 94
[cleaner_log_1] DECIDE TO ENTER! and let note on LOG
```

```
(1) joinWorkspace(WName,-Node)
(2) moveToWorkspace(WName,-Node)
(3) quitWorkspace(WName)
----------------------------------------------------
(4) use(AName,UICntrlName(Params),?SName,?Timeout,?Filter)
(5) sense(SName,?Filter,?Timeout,+Perception)
----------------------------------------------------
(6) focus(AName,?SName,?Filter)
(7) stopFocussing(AName)
----------------------------------------------------
(8) observeProperty(AName,?SName,?PFilter,+Property)

   (?) optional parameters; (-/+) in/out parameters
```

**Fig. 2.** (*Left*) The RoomsWorld scenario. Agents are engaged in cleaning thrash items spread over eight rooms relying on artifact based facilities of watch and logs. (*Right*) Basic set of actions integrating agent platforms (*Jason* in this case) and CARTAGO.

all the observable events generated by the artifact executing the operation are signalled as internal events to the agent. Otherwise, events collected in sensors can be retrieved by using the `sense` primitive, specifying a template (which functions as a filter) and possibly a timeout, indicating how long the current course of action can be suspended waiting for the event to be detected by the sensor. The basic support for artifact observation is provided by `focus` action, which makes it possible to continuously be aware of the observable events generated by the specified artifact. Finally, `observeProperty` is provided to observe and get the value of a given observable property of an artifact. In this case no sensors or percepts are involved: the value is directly bound to the variable specified within the operation. It's worth remarking that, differently from `use`, `focus` and `observeProperty` do not cause the execution of any operation or computational work by the (observed) artifact. A more detailed description of the model—including aspects not essential for this paper—can be found here [11, 10] and in CARTAGO manual. In order to provide a practical example showing cognitive artifacts in practice, in the next sections we discuss the RoomsWorld scenario, where teams of goal-oriented agents exploit cognitive artifacts to perform their cooperative tasks.

# 3 Benefits of Cognitive Artifacts in practice: the RoomsWorld experience

Built with CARTAGO technology, the RoomsWorld scenario realises an open system where heterogeneous agents have the possibility to join, test and interact with an artifact-based workspace. The workspace is composed by a number of virtual rooms separated by walls and doors (See Fig. 2 Left). Once a room is entered, agents should achieve the goal to find and clean trash objects which may appear in the rooms with arbitrary frequencies. It is worth noting that, given its structure, the global environment is partially observable. Rooms are visitable one at a time and, to locate trash items, an agent has to enter and then perform some epistemic action. Once they get the percept of a trash, agents reach their location and adopt a "clean" goal. For convenience, a global *environment* artifact is used to hold physical consistence to the system and supporting

```
public class Log extends Artifact {
   private LinkedList<String> notes;
   @OBSPROPERTY int lastnote;

   @OPERATION void init(String n, location l){
     name=n; loc= l;
     notes = new LinkedList<String>();
   }
   @OPERATION void putNote( Object ts ){
         notes.addLast(timestamp.toString());
         lastnote =
             ((Integer)timestamp).intValue();
   }
 }
```

**Table 1.** CARTAGO implementation for the Log Artifact.

agents in their epistemic activities. The environment artifact is used here just as a convenient way to represent and implement in the simulation the physical environment: for instance, the location of the trash items are provided in the form of symbolic percepts which are assumed to contain the exact location of the trash. In experiment series we engaged agents pursuing different strategies for achieving their goals. The "normal cleaners" simply look for trash exploring the rooms based on personal experience. Agents in this case act as if they were alone in the environment, by autonomously updating relevant information without any cooperation by other member of the team.

The second teams of agents use a message based strategy. "Messenger cleaners" exploit explicit messages to coordinate themselves. Once a given room has been cleaned, a messenger shares this information by broadcasting a message to the overall society, thus indicating the room identifier and the exact time at which the clean activity has been finished. For simplicity, messengers have been implemented using the same agent platform (*Jason* in this case), thus exploiting their native infrastructure to process messages.

To support the third strategy, a set of particular cognitive artifacts are deployed in the workspace to supply agents with additional information. In the example described here, agents have the possibility to use *log* artifacts which are physically placed at the entrance of each room. For each Log, the `@OPERATION void putNote(Object ts)` allows agents to store the time-stamp at which a given clean activity has been performed (Table 1). Accordingly Logs expose `@OBSPROPERTY int lastnote` as an observable property, namely the last registered time-stamp, by which agents can infer the last performed clean activity for that room. Logs make it possible uncoupled interaction between who wants to share a given information (informing agent) and who may exploit this information (reading agent) and allow agents to be focused on those particular knowledge which is relevant to achieve the current tasks. In addition, agents using logs also exploit a *watch* artifact, functioning as a timer which provides them with a symbolic record of the ongoing simulated time. Log strategy assumes that, before entering a given a room, an agent retrieves the actual time from the watch and then lets a time record in the related log. In the next sections we detail the design model for *Jason* agents implementing the Log strategy.

### 3.1 Agent's Knowledge Model

Our proposed agent model uses a systematic design structure for beliefs. In particular two different kinds of beliefs have been employed for agents. The first set of beliefs indicates symbolic references to objects (i.e. system entities, workspaces, identifiers, registered CARTAGO sensors etc.). Belonging to this class, a `nRooms(Nr)` belief is updated once the agent joins the environment, `Nr` indicating the current number of rooms. Besides, agent's cognitive process relies on a second class of beliefs, assumed to be significant for practical reasoning. Two sets of pivotal belief set belongs to this class. The former belief set is composed by facts on the form `cleaned(n,t)` containing the state of the various rooms (`n` indicating a room identifier between 1 and `Nr`, `t` containing the time-stamp at which this information has been retrieved). The latter belief set is related to the room the agent has decided to clean (`targetRoom(n)`). We consider theses belief sets as "salient" facts, namely the information which is *relevant* for agents for achieving their goals and thus for selecting and committing the intention to clean a given room. The notion of relevance concerns here those information required to agents for ruling over deliberation and means-end reasoning. On these basis we refer to this class of relevant beliefs as *goal-supporting beliefs*[5]. Differently from the first defined class of beliefs, goal-supporting beliefs are dynamic: during agents' purposive activities they need to be updated in order to reflect world changes. Notice that information about the state of each room can be considered certain only at a given time `t`, afterwards the activities of other agents or the rise of new trash items would have modified the room state.

## 3.2 Cleaner Agent using Logs

For simplicity here we describe a cutout of the code for the `Cleaner` agent implementing the log strategy. Agents are implemented in *Jason* notation [6]. The initial goal `join` allows the agent to register himself to the RoomsWorld workspace and to join the `environment` artifact. The agent here uses the `getNRooms` operation retrieving the number of rooms `n_rooms(Nr)` where `Nr` unifies with the number of rooms signalled by the environment.

```
+!join
  <- cartago.joinWorkspace("RoomsWorld",
               "localhost:4010");
     !locate_artifacts;
     ?myName(N); ?mySensor(S);
     ?artifactBel(environment, Env);
     cartago.use(Env, join(N), S);
     cartago.sense(S, Percept, "joined");
     cartago.use(Env, getNRooms, S);
     cartago.sense(S, n_rooms(Nr), "n_rooms");
     +nRooms(Nr); !explore.
```

```
+!explore
  <- -targetLog(_);
     ?nRooms(N); ?mySensor(S);
     roomsworld.randomInt(N,Rid);
     +targetLog(Rid+1);
     ?artifactBel(watch, IDWatch);
     cartago.observeProperty(IDWatch,
                       currentime(Wt) );
     !analizeBel(X+1,Wt).
```

Notice that a particular plan is then called to retrieve and store the identifiers belonging to the artifacts that will be used during the task. Once retrieved through the use of `cartago.lookupArtifact` operation, the artifacts identifiers are stored in a belief set in the form `+artifactBel(artifactName, artifactId)` providing a knowledge repository associating each artifact name to the corresponding ID.

[5] We are grateful to Cristiano Castelfranchi and Fabio Paglieri for pointing out to our attention this particular relation, in cognitive agents, between *relevant* beliefs and goal processing[3]. As explained in section Section 4, we'll exploit the number of belief update pursued by agents upon these relevant facts as a measure for the computational load of their cognitive processes.

[6] *Jason* is an agent language for goal-oriented, BDI-like agents based on AgentSpeak(L) [1]. The entire code of RoomsWorld experiment, along with agents implementation using alternative platforms, is available on CARTAGO web site.

```
+!locate_artifacts
   <- cartago.lookupArtifact("watch", Wid);
      +artifactBel(watch, Wid);
```

The following `explore` goal is the starting point for agent's purposive activity, by which the agent randomly select a room `Rid` and prepare a new intention to explore in it. The action `cartago.use(W, whatTime, S)` executed upon the *watch* artifact allows the agent to retrieve the actual time: the agent observes the watch artifact, getting the content of the `time(Wt)` percept ($W_T$ unifies with the percept content signalled by the watch)[7].

To filter out worth intentions and thus prevent exploring places which are recently visited, the randomly selected room is then compared with the relevant belief base that refers to the acknowledged state of the rooms. So far, if the difference between the remembered time and the time at which the target room has been previously cleaned is greater than a given threshold $D$, the agent "commits" to the intention to go toward the room and then try to use the related log (i.e. by updating relevant belief `targetRoom(N)`). Otherwise, the agent abandons the current plan and readopt the explore goal to generate a different intention. Notice here that agent is using his *relevant* information to make a decision about his next course of actions.

```
+!analizeBel(N,Wt) : not cleaned(N,Lt) | (cleaned(N,Lt) & day(D) & (D< T-Wt))
   <- -cleaned(N,_); -targetRoom(_); +targetRoom(N); !observeLog(N).
+!analizeBel(N,Wt) <- !explore.
```

So far, the agent observes the log artifact corresponding to the selected room and reads the last recorded time-stamp. This is done by executing the `observeProperty` action and perceiving the `lastnote` value, carrying to the agent the relevant information about the last time-stamp. We here focus on an interesting difference with respect to the strategies employed by other agent teams. Whereas normal agents have to autonomously update the knowledge about the problem domain and messenger agents have to continuously process incoming messages, here the log agent finds information by simply observing the log artifact on which he is interested. Relevant information left by some other agent at the end of a previous cleaning activity has been stored and collected within the artifact that makes it available for the overall society. This approach directly enables uncoupled interactions—mediated by the log—among different agents during their practical behavior.

```
+!observeLog(N) : targetCh(N) & myRoom(MR)
   <- roomsworld.goTo(MR, log(N));
      -targetCh(N);
        [ ... retrieve artifacts ID IDLog and IDWatch ... ]
      cartago.observeProperty(IDLog, lastnote(LogT) );
      cartago.observeProperty(IDWatch, currentime(Wt) );
      !decide(N, Wt, LogT).
```

Once the agent has observed both information about log's last note $Log_T$ and the information about the current time $W_T$ provided by the watch, he can *decide* what to do next. Notice that the agent is here deciding upon an updated and situated information (the actual state retrieved from the `log` corresponds to the last known state of the room, stored by some other agent of the group). If the difference between the actual time $W_T$ and the time-stamp retrieved in the log $Log_T$ is greater than a given threshold (say, a day's length `day(D)`), the agent maintains his intention of entering the room $N$, otherwise he reconsiders his intentions and adopt a new `explore` goal to select a new room. It is worth

[7] The `cartago.*` actions refer to CARTAGO basic primitives (see Section 2.2), while `roomsworld.*` refer to the library of internal actions defined to operate within the RoomsWorld scenario.

```
+!decide(N, CurrentTime, LastTime)
  : (day(D) & (D< CurrentTime-LastTime))
    | (LastTime=0)
  <- !log("DECIDE TO ENTER! Put a note on Log");
     [ ... retrieve sensor S and Log name LogN ...]
     cartago.lookupArtifact(LogN, IDLog);
     cartago.use(IDLog, putNote(CurrentTime), S);
     !log("let note:", CurrentTime); !go(N).
```

```
+!decide(N, CurrentTime, LastTime)
  <- !log("RECONSIDERED INTENTIONS:EXPLORING!") ;
     -cleaned(N,_); +cleaned(N,LastTime);
     -targetRoom(_);
     !explore.
```

noting that, within the decide plan, the agent updates either the state of his belief or the state of the log. Indeed once the agent has decided to enter the room, he puts a note on the log, either anticipating the information about that room to be cleaned or preventing follower agents of the society to waste their resources going to explore the same room. Otherwise the agent needs to update the relevant beliefs and thus reconsider his intentions: in this case room $N$ will be stored as 'cleaned' at time `LastTime` through the belief update `+cleaned(N,LastTime)`. The following action `go(N)` allows the agent to enter the selected room $N$. After having updated the beliefs about target room and actual room, it's time for the agent to search for trash items: `roomsworld.epistemicAction(N)` is an internal action used to perceive the objects in sight from the environment artifact. The epistemic action, in turn, encapsulates the use of the environment artifact and simulates agent's perceptive activities transforming volatile percepts (coming from CARTAGO sensors) into agent beliefs.

```
+!go(N): not myRoom(N)
   <- .print("enter in room ", N);
      ?myRoom(MR); roomsworld.goTo(MR, room(N));
      -targetRoom(_); -myRoom(_); +myRoom(N);
      roomsworld.epistemicAction(N); !clean(N).
+!go(N) <- !explore.
```

If trash is found in the room, the epistemic action adds beliefs in the form `trash(N,X,Y)` where $N$ is the room ID, and $X, Y$ are the coordinate of the discovered trash item. Once the agent has located trash items (if any), it can reach them with the action `roomsworld.goTo(X,Y)` and clean exploiting the `cartago.use(E, clean(Na, X, Y), S)` operation upon the environment `E`. Accordingly, the agent updates the beliefs referring to the state of the room. Otherwise, if the epistemic action has returned no items, the agent drops the goal, reconsider the current intention exploring another room.

Finally, the action to clean the trash item is realized calling the `cartago.use(Env, clean(Na, X, Y), S)` operation upon the `Env` artifact. The agent indicates his own name and the coordinates of the trash to clean (which is then returned as a percept). It is worth noting that the `clean/3` operation may fail, due to the non-determinism of the environment or due to the execution of concurrent cleaning actions performed by two or more agents on the same trash item a the same time.

## 3.3 Two Interaction Styles from Agents to Artifacts

As detailed above, agent specification introduces two main interaction styles. We here describe these interaction approaches indicating them as respectively *general purpose* and *special purpose* styles. The former programming approach is to directly use the basic actions (`use` and `sense`) provided by CARTAGO. The use of these general purpose primitives (described in Section 2.2) guarantees agents to control artifacts by exploiting their interfaces. However, interactions between agents and artifacts can be arbitrarily complex, sometimes requiring lower level control phases during execution of complex and composite actions, i.e. interleaved sequences of use operations. An alternative interaction approach may thus suggest to design special-purpose actions, allowing the developer to interact at a higher level of abstraction. Special-purpose actions encapsulate a given activity in coarse-grained actions and can be used in a goal-oriented fashion which in turn may transparently control the course of interaction from agents to artifacts. In the RoomsWorld case, this approach has been used in the above mentioned `epistemicAction/1`, allowing to encapsulate in a single internal action all the perceptive activities, thus simplifying the course from composite percepts related to the sensory activities to the symbolic beliefs referred to the problem domain. Special-purpose actions are assumed to possibly handle exceptions, failures and manage low-level details. This is what Norman refers as "gulf of execution and evaluations" of actions upon artifacts, by which an individual may manage the interaction i.e. evaluating mismatches between his internal expectations and the actual course of observable events [7].

# 4 **RoomsWorld** Experiment

To evaluate the different interaction strategies for cleaner agents in RoomsWorld described in Section 3, we ran sequences of trials measuring performances for different teams belonging to the different strategies. RoomsWorld was set to 8 rooms and the total amount of trash items contemporaneously present is limited to 4, whilst not more than a trash is generated for each room (Fig. 2). Each reported experiment consists of repeated runs for each team, using different randomly generated initial conditions and distribution of trash items.

To evaluate strategy effectiveness we defined further metrics relying on the performances of each team. What we are interested in is a quantitative evaluation of the tradeoff between computational costs spent by agents to update their beliefs and the absolute performances in terms of achieved goals (i.e., agent's score). This allows us to give an account about the agents reasoning processes in terms of their computational load. For each trial we defined team's *goal effectiveness* in terms of cleaned items (i.e. agent's score) and in function of elapsed time. Besides, for each agent typology, we focused on the particular belief set which is involved in updating relevant information used to support reasoning and thus to perform the task. In more detail, we define the *belief change cost* as the aggregate of belief change operations performed by all the agents of the team upon their relevant beliefs (i.e. the sum of all modifications of agent's relevant belief base during all the trials). Given the structure of relevant belief set as it has been defined in Subsection 3.2, two kinds of updates entail an increase of belief change cost:

1. The state of a given room $N$ may be remembered by agents as cleaned at a certain instant of $Time$ through the addition of relevant `+cleaned(N,Time)` facts. Once an agent has become aware of the new state of a given room $N$ (either because of a message sent by another agent of the team, or because the agent has retrieved some log report, or because he autonomously discovers the state of a given room) he immediately updates the relative

facts. According to the definition, revising the relevant belief base with a new `cleaned(N,T)` fact entails a *belief change cost* of +1.

2. An additional cost is accounted by updating the the belief on `targetRoom(n)`. In so doing, the agent is going reconsider his intention selecting an alternative course of action. In turn, no more applicable plans will match the precondition on the room identifier. This entails the agent to reconsider his plans, selecting a new intention and exploring a new room.

As an example of cost counting, the following cutout shows how the messenger agents are supposed to spend two units of *belief change cost*. Sender agents update the `cleaned(N,T)` belief once a room has been cleaned (*right*) and have to broadcast this information to the overall society; accordingly, a receiver agent updates his beliefs about the state of that room once a message is received (*left*).

```
+!clean(N) : not trash(N,_,_)
  <-  ?artifactBel(watch, W);
      cartago.observeProperty(W, sim_time(Wt));
      .broadcast(tell, cleaned(room(N, Wt)));
      -cleaned(N,_); +cleaned(N,Wt);
      roomsworld.goTo(N, log(N));
      !explore.
```

```
+cleaned(room(N,T))[source(Ag)] : true
  <-  -cleaned(room(N,_))[source(_)];
      -cleaned(N,_);
      +cleaned(N,T);
```

Then, we define the *cost effectiveness* ratio for a team in terms of the total amount of achieved task (agent's score) divided by the *belief change cost*. Namely, *cost effectiveness* represents the unit of achieved goals for each belief change.

Conducted experiments considered the above defined metrics and used the threshold $D$ to 12 and 6 units of simulated time respectively for experiment of two and four agents for each team. Because of the random distributions of items, the previously defined metrics present a fluctuating course before converging. Hence the length of each individual trial has been set sufficiently large to become stable and, after the analysis of the courses of experiments, it has been set to 250 units of simulated time. Fig. 3 and Fig. 4 show the performance of the different teams by averaging progresses of their effectiveness (on the left) and their cost ratio (on the right). In particular Fig. 3 refers for teams composed by two agents, while Fig. 4 to teams of four agents.

### 4.1 Experiment Discussion

Experiments put in evidence log artifacts functioning as cognitive resources for agents (a belief base enhancer in this case), clearly enabling agents to ease their epistemic activities. Anyhow, on the basis of the defined metrics, the results show some noticeable payoffs for the various strategies. In the case of teams composed by two agents, messenger agents attain the best performance in terms of *goal effectiveness* (Fig. 3, left). Even if *goal effectiveness* for messenger and log agents approximatively converge to the same value (both reach a goal effectiveness of about 0.13 cleaned items on each elapsed time unit), messenger seems to tackle a shorter transitory phase. This evidence clearly comes from the fact that messengers can suddenly achieve and maintain an updated knowledge of the global states of the various rooms. Indeed, on each achieved goal, message

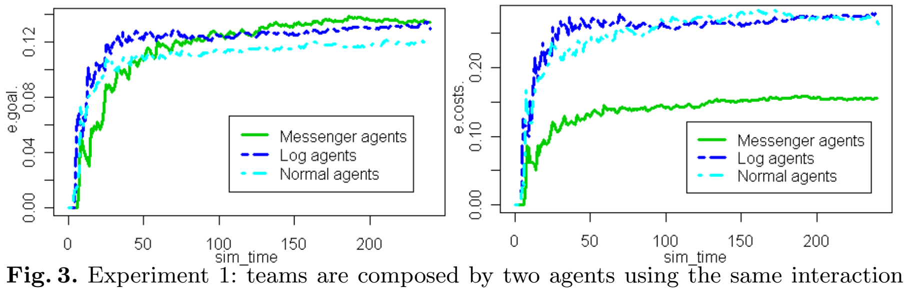


**Fig. 3.** Experiment 1: teams are composed by two agents using the same interaction strategy. Agents performances are measured in terms of *goal effectiveness (left)* referred to the amount of achieved goals, and *cost effectiveness (right)* referred to the computational load related to the update processing of relevant beliefs.

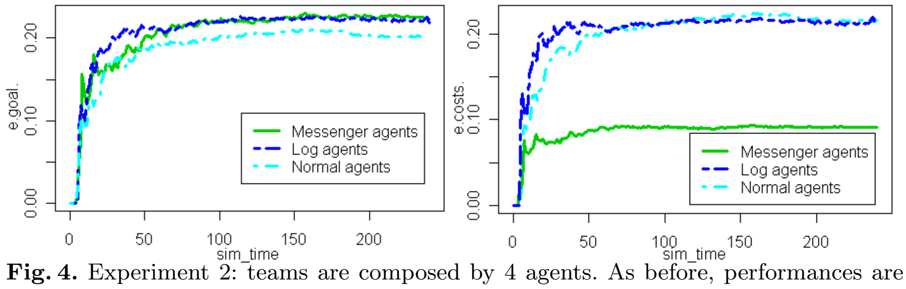


**Fig. 4.** Experiment 2: teams are composed by 4 agents. As before, performances are measured in terms of *goal effectiveness (left)*, and *costs effectiveness (right)*.

exchange allows agents to handle a more complete knowledge of environments. By augmenting the team members the global performance in terms of achieved goals for the various teams reach an higher value and messenger superiority in terms of achieved goals becomes less evident (Fig. 4, left). In this case, societies of numerous agents are better prone to make relevant information available to the overall group. Accordingly, agents using artifacts are more effective to cooperatively update the logs so to reflect a more updated state of the rooms. Hence, for teams composed by four agents, the *goal effectiveness* for messenger and log agents approximatively converge to the same value (both the teams reach a global goal effectiveness of about 0.22 cleaned items on each elapsed time unit). Besides, the log agents outperform normal agents in each condition: whereas normal agents waste time and resources randomly looking for trash in rooms that have just been cleaned, log agents are smarter in recognising rooms requiring services, thus better balancing their payoff between exploration and exploitation.

Performance result inverted when considering the *cost effectiveness*: in this case the winning team is the one composed by normal agents, due to the lower amount of belief update performed upon the relevant facts and to the lower frequency of intention reconsideration. As Fig. 3 (right) shows, for each internal belief update the teams composed by two normal agents achieve 0.25 goals (0.27 for logging agents and 0.15 for messenger agents). Similar results have been collected for the teams composed by 4 agents (Fig. 4, right), where messenger agents attain a very poor performance (whilst normal and log agents achieve 0.22 cleaned trash items for each belief update, messenger agents reach the value of 0.09, due to the significant increase of message passing).

Balancing the performances for each selected metrics in each environmental condition, we here enlighten the effectiveness offered by the communication strategy mediated by log artifacts. The advantages of exploiting distributed logs to externalise information which is relevant for coordinating team activities are more evident the more the team is numerous. Logs give a considerable contribute both in terms of propagation and synchronization of the information. Even more logs provide a informational support in uncertain, transitory conditions (i.e. when agents need to adjust behavior given environment partial observability). Although broadcasting messages to the overall society allows agents to maintain an updated knowledge of the global environment, this requires agents to waste their computational resources to continually process messages which are not strictly relevant for the ongoing task. On the contrary, log agents locally exploit information concerning their *actual* purposes, fully exploiting logs as suitable belief base enhancer. Differently from other approaches exploiting mediated interactions as shared memories and blackboards, logs are here conceived with the aim to improve *situated cognition*: agents can exploit logs to attain only the local information which is relevant to achieve the actual goal. Furthermore, the information coming from distant rooms is not relevant for agents because does not affect their ongoing intentions. This allows agents to deal with situated information, and may not require agents to pay attention (nor spend computational resources) to information which is useless with respect of the ongoing tasks. It is worth remarking that logs store the information which remains available even beyond their use. This has a pivotal importance in the context of open systems, where different agents may asynchronously operate, with interleaved presence, in specific tasks. Agents using logs are deliberatively modifying their environment with the aim to coordinate with other agents of the society and let them know the actual state of affairs. In so doing, the global behavior of the society is governed by the *emerging contents* of the distributed logs, which are cooperatively updated by agents during their activities.

# 5 Conclusions and Related Works

In this paper we discussed the role that cognitive artifacts, as artifacts with specific representational functionalities, can play in multi-agent systems, analogously to the human case, in supporting efficiently the cooperation of intelligent agents, who can share information by cooperatively updating the state of the artifacts, thus externalizing belief revision activities and easing their computational burdens. This is in agreement with many other research works recently appeared in literature (see [14] for a survey), remarking the role that the environment could play in designing complex MAS, as a suitable place where to encapsulate functionalities and services useful for agent interaction, coordination and cooperation. Among these, few works are specifically about cognitive notion agency, in particular about high-level environment models specifically conceived to support and promote goal-oriented behavior of cognitive agents, and related agent reasoning techniques. A first one is Brahms [12], a multi-agent programming language and platform to develop and simulate multi-agent models of human and machine behavior, based on a theory of work practice and situated cognition. Our approach shares many points with Brahms, starting from a common reference conceptual background based on conceptual frameworks such as Activity Theory. Differently from Brahms, our primary context is not modeling and simulation, but agent-oriented software development. A further work is GOLEM [2], that introduces a platform for modeling situated cognitive agents in distributed environments by declaratively describing the representation of the environment in a logic-based form. GOLEM models (physical) environments in terms of *containers* where agents based on the KGP model and *objects* are situated. Besides sharing the same modeling perspective—which can be traced back to our early works on artifact-based coordination in MAS [9], we here investigate the cognitive use of artifacts, focussing in particular on the role of cognitive artifacts.

# References

1. R. Bordini and J. Hübner. BDI agent programming in AgentSpeak using Jason. In F. Toni and P. Torroni, editors, *CLIMA VI*, volume 3900 of *LNAI*, pages 143–164. Springer, Mar. 2006.
2. S. Bromuri and K. Stathis. Situating Cognitive Agents in GOLEM. In *Engineering Environment-Mediated Multiagent Systems (EEMMAS'07).* LNCS Springer, 2007.
3. C. Castelfranchi and F. Paglieri. The role of beliefs in goal dynamics: Prolegomena to a constructive theory of intentions. *Synthese*, 155:237–263, 2007.
4. A. Clark and D. Chalmers. The extended mind. *Analysis*, 58: 1:7–19, 1998.
5. D. Kirsh. The intelligent use of space. *Artif. Intell.*, 73(1-2):31–68, 1995.
6. B. A. Nardi. *Context and Consciousness: Activity Theory and Human-Computer Interaction.* MIT Press, 1996.
7. D. Norman. Cognitive artifacts. In *Designing interaction: Psychology at the human–computer interface.* Cambridge University Press, New York, 1991.
8. A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17 (3), 2008.
9. A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini. Coordination Artifacts: Environment-based Coordination for Intelligent Agents. In *Proceedings of AAMAS'04*, volume 1, pages 286–293, New York, USA, 2004.
10. A. Ricci, M. Piunti, L. D. Acay, R. Bordini, J. Hubner, and M. Dastani. Integrating Artifact-Based Environments with Heterogeneous Agent-Programming Platforms. In *Proceedings of AAMAS'08*, 2008.
11. A. Ricci, M. Viroli, and A. Omicini. The A&A programming model & technology for developing agent environments in MAS. In *5th Workshop "Programming Multi-Agent Systems"(PROMAS07)*, volume 4908 of *LNAI*, pages 91–109. Springer, 2007.
12. M. Sierhuis and W. J. Clancey. Modeling and simulating work practice: A human-centered method for work systems design. *IEEE Intelligent Systems*, 17(5), 2002.
13. D. Weyns, A. Omicini, and J. Odell. Environment as a First-class Abstraction in MAS. In *Autonomous Agents and Multi-Agent Systems* [14], pages 5–30.
14. D. Weyns and H. V. D. Parunak. Special issue on environments for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):1–116, Feb. 2007.

# Dynamic logic on normal form games

R. Ramanujam and Sunil Simon

The Institute of Mathematical Sciences<br>
C.I.T. Campus, Chennai 600 113, India.<br>
E-mail: {jam,sunils}@imsc.res.in

**Abstract.** We consider a dynamic logic of game composition, where atomic games are in normal form. We suggest that it is useful to consider a modality indexed by game - play pairs. We show how we can reason not only about notions like strategy comparison, dominated strategies and equilibrium in such a framework, but also strategic response, whereby the choice of a player depends on plays observed in the past. This makes for a significant difference in the presence of unbounded iteration. We present a complete axiomatization of the logic and prove its decidability.

## 1 Overview

The central innovation introduced by game theory is its strategic dimension. A player's environment is not neutral, and she expects that other players will try to outguess her plans. Reasoning about such expectations and strategizing one's own response accordingly constitutes the main logical challenge of game theory.

Games are defined by sets of rules that specify what moves are available to each player, and every player plans her strategy according to her own preferences over the possible outcomes. In an extensive form game, the moves of players are explicitly presented and therefore strategies are not abstract atomic entities, but have a certain structure associated with them. The choice of which strategy to employ depends not only on the game structure but also on her expectation of what strategy other players choose. Thus at any game position, the past as well as the possible futures and players' expectations determine strategies.

In contrast, strategies are presented in an abstract way in a normal form game and the reasoning in such a game is driven by outcome specifications. Thus normal form games can be viewed as extensive form games abstracted into a tree of depth one, where edges are labelled by a tuple of strategies, one for each player, and thus strategies are atomic. Therefore there is no past and future that strategies refer to, and we only speak of notions like rational response, dominant strategies, equilibrium and so on. However, when we consider repeated games, or games composed of smaller games, the notion of strategic response of a player to other players' moves becomes relevant, pretty much in the same way as it is used in extensive form games. History information, as well as epistemic attitudes of players become relevant.

There have been several logical studies from this viewpoint. Notable among these is the work on alternating temporal logic (ATL) [AHK02] which considers selective quantification over paths that are possible outcomes of games in which players and an environment alternate moves. An ATL model is a concurrent game structure which consists of a single game arena whose edges correspond to concurrent moves of the players. Moves in the arena can therefore be thought of as a strategy profile of an appropriate normal form game. Thus each game position of the arena is associated with a single normal form game. The formulas of ATL make assertions about the tree unfolding of this arena. The emphasis is on the existence of a strategy for a coalition of players to force an outcome. Since the game tree encodes the past information, the logic itself can be extended with past modalities as well as knowledge modalities in order to reason about the history information and epistemic conditions used in strategizing by players ([JvdH04],[vdHW02]). Extensions of ATL where strategies are allowed to be named and referred to in the formulas of the logic are proposed in ([vdHJW05],[WvdHW07]). ([Ago06], [Bor07]) extends ATL with the ability to specify actions of players in the formulas.

The running thread in this line of work is the notion of strategies as being atomic, lacking structure. In principle, since concurrent actions (strategy profiles) can be named, their labels can be used to refer to strategies, and hence we can speak of $b$ being a response to $a$ in the past, to achieve $\alpha$. However, the tree models carry only temporal information, and strategies lack syntactic structure, and this is reflected in reasoning. Thus ATL-based logics can be seen as analogous to temporal logics (for games), as opposed to dynamic and process logics.

When games are themselves structured, strategic response reflects such structure as well. For games of bounded length, an action labelled modal logic reflects game and strategy structure well, but when we consider unbounded play as arising from unbounded repetition of games, the situation is different. This is the spirit in which game logic [Par85] was proposed and underlying framework of coalition logic [Pau01]. The strategies used by a player in such a composite game would depend on not just the outcome specification but also what strategy was used, especially by opponents, in the past. The history information can then be analysed by taking into account the underlying structure of the composite game.

We suggest that in reasoning about structured games, it is useful for the strategies of players to also reflect the structure. Thus rather than reasoning about the strategies in the composed game, one should look at strategies in the atomic game and compose such atomic game strategy pairs.

Suppose that we have a 2-player 2-stage game $g_1$ followed by $g_2$. Consider player 1 strategizing at the end of $g_1$, when $g_2$ is about to start; her planning depends not only how $g_2$ is structured, but also how her opponent had played in $g_1$, and the outcomes that resulted in $g_1$ for both of them. Thus her strategizing in the composite game $g_1; g_2$ is best described as follows: consider $g_1$ in extensive form as a tree, and the subtree obtained by the set of plays $\eta_1$; when $g_2$ starts from any of the leaf nodes of this subtree, consider the play $\eta_2$. We encode this as $(g_1, \eta_1); (g_2, \eta_2)$, and see $(g_2, \eta_2)$ as a response to $(g_1, \eta_1)$. Thus the "programs" of this logic are game - play pairs of this kind.

For extensive form games, this was done in [RS08], where we look at strategic reasoning done in extensive form games by making explicit use of the structure of the game tree. It defines a propositional dynamic logic where programs are regular expressions over game strategy pairs. This gives the ability to reason about the strategic response of players based on what happened in the past. A complete axiomatization of the dynamic logic is presented and the decidability of the logic is also shown. This paper proves similar results for composition of normal form games.

We consider composition of game play pairs in normal form games, corresponding to the fact that the reasoning performed in single stage is mostly outcome based. If we restrict the reasoning to bounded repetition of games or to multistage games where the number of stages are bounded, then we do not need to look at composition of game play pairs. It is the presence of unbounded iteration of games which makes it necessary to introduce a dynamic structure on game play pairs. We therefore study a dynamic logic where programs consist of regular expressions over game play pairs in normal form games. While the main technical result is a complete axiomatization for the logic, the central objective of the paper is to highlight the logical differences between composition of normal form games and that of extensive form games, in terms of the reasoning involved.

We wish to emphasize that what we study here is really a dynamic logic of tree composition. When we consider only bounded games, the logic is subsumed by the ATL frameworks, but the class of games with unbounded iteration studied in Section 5 is our main object of study. However, rather than presenting it all at one go, we discuss strategic response for bounded games before considering repetition.

In the case of extensive form games, the idea of taking into account the structure available within strategies and making assertions about a specific strategy leading to a specified outcome is developed in [vB01,vB02], where van Benthem uses dynamic logic to describe games as well as strategies. [Gho08] presents a complete axiomatisation of a logic describing both games and strategies in a dynamic logic framework where assertions are made about atomic strategies. The techniques developed in [Gho08] can be easily transferred to normal form games. Our point of departure from this line of work is in talking about the strategic response of players in the logical framework.

## 2 Preliminaries

### Normal form games

Let $N = \{1, 2\}$ be the set of players, $\Sigma_i$ for $i \in \{1, 2\}$ be a finite set of action symbols which represent moves of players and $\Sigma = \Sigma_1 \times \Sigma_2$. For each player $i$, let $R_i$ be the finite set of rewards, $\preceq^i \subseteq R_i \times R_i$ be a preference ordering on $R_i$ and let $R = R_1 \times R_2$.

Normal form (or strategic form) games are one shot games where the strategies of players corresponds to choosing an action from the action set. A strategy profile is simply a pair of actions, one for each player. A play of the game corresponds to each player choosing an action simultaneously without knowledge of the action picked by the other player. Thus a strategy profile constitutes a play in the game. Each play is associated with a pair of rewards for the players, the outcome of the play.

Suppose $|\Sigma_1| = m$ and $|\Sigma_2| = k$, then a strategic form game can be represented as an $m \times k$ matrix $A$ where the actions of player 1 constitute the rows of the matrix and that of player 2 the columns. The matrix entries specify the outcome of the play for each player, i.e. elements from $R$. An example game is given in Fig. 1. Here $\Sigma_1 = \{b, c\}$ and $\Sigma_2 = \{x, y\}$. The action profile $(b, x)$ where player 1 chooses to play $b$ and player 2 chooses $x$, results in the reward $r_1^1$ for player 1 and $r_2^1$ for player 2.

|  | $x$ | $y$ |
|---|---|---|
| $b$ | $(r_1^1, r_2^1)$ | $(r_1^2, r_2^2)$ |
| $c$ | $(r_1^3, r_2^3)$ | $(r_1^4, r_2^4)$ |

**Fig. 1.** Matrix game

Let $\bar{\imath} = 2$ when $i = 1$ and $\bar{\imath} = 1$ when $i = 2$. Unless specified, we will use the convention that $i = 1$ and $\bar{\imath} = 2$. We use $b, c$ to denote the actions of player $i$, $x, y$ to denote the actions of player $\bar{\imath}$ and $a$ to denote the strategy profile.

### Strategy comparison and equilibrium

For player $i$, the ordering $\preceq^i$ on the rewards $R_i$ induces an ordering on the strategy profiles as: $(b, x) \preceq^i (c, y)$ iff $A(b, x)[i] \preceq^i A(c, y)[i]$. Having defined the preference ordering on strategy profiles, the various game theoretic notions of interest include:

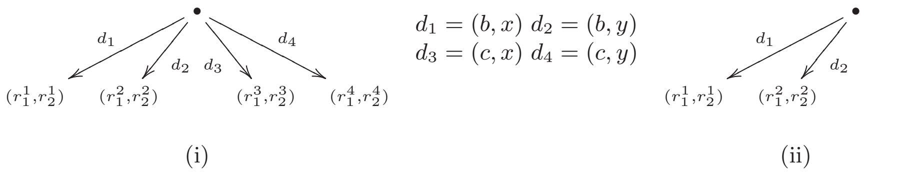
- Weak domination: We say a strategy $b$ of player $i$ weakly dominates a strategy $c$ if for all $x \in \Sigma_{\bar{\imath}}$, $(c, x) \preceq^i (b, x)$.
- Best response: Given strategies $b$ and $x$ of player $i$ and $\bar{\imath}$ respectively, we say that $b$ is the best response for $x$ iff for all $c \in \Sigma_i$, $(c, x) \preceq^i (b, x)$. A similar definition can be given for the best response of player $\bar{\imath}$.
- Equilibrium: A strategy profiles $(b, x)$ constitutes a Nash equilibrium iff $b$ is the best response for $x$ and $x$ is the best response for $b$.

## 3 Reasoning in strategic form games

As opposed to extensive form games where the game structure is explicit, normal form games are specified by the set of abstract strategies and the outcomes. In this scenario a player cannot strategize based on the past moves of the opponent. Strategizing would rather be based on his expectation of what strategies the opponent will choose, along with the outcomes which can be ensured by the player. In this section we look at how to logically reason about such abstract strategies with respect to the outcomes.

For a logical analysis, it is convenient to view the normal form game as a tree of depth one, where the edges are labelled by pairs of actions, one for each player. Formally $g = (S, \longrightarrow, s_0, \lambda)$ where $S$ is the set of states, $s_0$ is the root of the tree. The transition function $\longrightarrow: s_0 \times \Sigma \rightarrow S$ is a partial function also called the move function. The reward function $\lambda : S \rightarrow (R_1 \times R_2)$. For a node $s \in S$, let $\vec{s} = \{a \in \Sigma \mid \exists s' \in S \text{ where } s \xrightarrow{a} s'\}$ and $\Sigma^g = \{a \in \Sigma \mid \exists s, s' \in S \text{ where } s \xrightarrow{a} s'\}$. Thus for a game $g$, the set $\Sigma^g$ constitutes all the strategy profiles of $g$.

The game tree corresponding to the strategic form game in Fig. 1 is shown in Fig. 2(i). A play is simply an edge in the tree, this corresponds to both the players picking an action. A strategy for player $i$ is the subtree of $g$ where for player $i$ a unique action is chosen and for player $\bar{\imath}$ all the actions are taken into account. A strategy for player 1 in the game given in Fig. 2(ii) where he picks action "$b$", is shown in Fig. 2(ii). For the rest of the paper, we will use the tree representation for strategic games. For a pair $a = (b, x)$ and $j \in \{1, 2\}$, we denote by $a[j]$ the $j^{th}$ component of $a$.



(i) (ii)

**Fig. 2.** Normal form game

### The logic

**Syntax:** Let $P$ be a countable set of propositions and $g$ be a strategic form game (in the tree representation). The syntax of the logic is given by:

$$\Phi := p \in P \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \langle g, \eta \rangle^{\forall} \alpha$$

where $\eta \subseteq \Sigma^g$.

For the moment we let the semantic tree structure $g$ be part of the syntax of the formulas. In section 4 we show how game trees can be specified in the logic in a syntactic manner. For a game $g$ (specified as a tree of depth one), $\eta \subseteq \Sigma^g$ represents a set of plays in $g$. The intuitive meaning of the construct $\langle g, \eta \rangle^{\forall} \alpha$ is to say that the formula $\alpha$ holds at all nodes which results from the plays of $g$ specified by $\eta$. Note that strategies of a particular player can be easily represented in $\eta$. This can be done by fixing a single action for the player and considering all plays in $g$ where this action is fixed.

**Semantics:** The model $M = (g, V)$ where $g = (S, \longrightarrow, s_0, \lambda)$ is a normal form game and $V : S \to 2^P$ is a valuation function. To be able to perform strategic reasoning in the logic, we need to refer to rewards of the players in the formula. This is taken care of by using special propositions to code them up, in the spirit of the approach taken in [Bon02]. The preference ordering is then simply inherited by the implication available in the logic. Formally, let $R_1 = \{r_1^1, \ldots, r_1^l\}$ be the set of rewards for player 1. Without loss of generality we assume that $r_1^1 \preceq^1 r_1^2 \preceq^1 \cdots \preceq^1 r_1^l$. Let $\Theta_1 = \{\theta_1^1, \ldots, \theta_1^l\}$ be a set of special propositions used to encode the rewards in the logic, i.e. $\theta_1^j$ corresponds to the reward $r_1^j$. Likewise for player 2, corresponding to the set $R_2$, we have a set of propositions $\Theta_2$. The valuation function satisfies the condition:

– For all states $s$, for all $i \in \{1, 2\}$, $\{\theta_i^1, \ldots, \theta_i^j\} \subseteq V(s)$ iff $\lambda(s)[i] = r_i^j$.

The truth of a formula $\alpha \in \Phi$ in the model $M$ at a position $s$ (denoted $M, s \models \alpha$) is defined as follows:

– $M, s \models p$ iff $p \in V(s)$.
– $M, s \models \neg\alpha$ iff $M, s \not\models \alpha$.
– $M, s \models \alpha_1 \vee \alpha_2$ iff $M, s \models \alpha_1$ or $M, s \models \alpha_2$.
– $M, s \models \langle g, \eta \rangle^{\forall} \alpha$ iff $s$ is not a leaf node and $\forall s' \in \mathit{tail}(g, \eta)$, $M, s' \models \alpha$.

where for game $g$ and $\eta \in 2^{\Sigma}$, $\mathit{tail}(g, \eta) = \{s' \mid s_0 \xrightarrow{a} s' \text{ and } a \in \eta\}$.

When $\langle g, \eta \rangle^{\forall} \alpha$ is asserted at the root node $s_0$ of the game tree $g$, we get the following interpretation: $\langle g, \eta \rangle^{\forall} \alpha$ holds iff $\alpha$ holds at all leaf nodes resulting from plays specified by $\eta$. Since we are working with a single tree of depth one, interpreting $\langle g, \eta \rangle^{\forall} \alpha$ at the leaf nodes does not make sense. The dual modality $[g, \eta]^{\exists} \alpha$, would say that there exists a play of $g$ specified in $\eta$ such that $\alpha$ holds at the leaf node of the play.

**Strategy comparison in the logic:** We show that the various strategizing notions discussed in the earlier section can be expressed in the logic. For a game $g$, let $\Sigma^g = \{a_1, \ldots, a_k\}$ be the strategy profiles occurring in $g$. For $i \in \{1, 2\}$, let $\Sigma_i^g = \{a_1[i], \ldots, a_k[i]\}$ and for $b \in \Sigma_i^g$, let $\Sigma_g(b) = \{a \in \Sigma^g \mid a[i] = b \text{ and } a[\bar{\imath}] \in \Sigma_{\bar{\imath}}^g\}$. $\Sigma_g(b)$ thus consists of all the strategy profiles where player $i$'s strategy is fixed to $b$. Consider the formula:

$$\mathit{ensures}^i(g, \gamma) \equiv \bigvee_{b \in \Sigma_i^g} \langle g, \Sigma_g(b) \rangle^{\forall} \gamma.$$

*ensures*$^i(g, \gamma)$ says that given that the opponent chooses an action from the set $\Sigma^g_{\bar{\imath}}$, there is a strategy for player $i$ to achieve $\gamma$ no matter what choice player $\bar{\imath}$ makes. In the case of $\gamma \in R_i$, this corresponds to the rewards that player $i$ can ensure. If player $i$ expects that $\bar{\imath}$ will choose only actions from the set $\Sigma' \subseteq \Sigma^g_{\bar{\imath}}$, then the restriction of *ensures*$^i(g, \gamma)$ to $\Sigma'$ specifies what player $i$ can ensure in terms of his expectation. A player during the phase of strategizing might take into consideration what he can ensure given his expectation about the strategies of the opponent. The related concept of weakly dominating strategies can be defined as follows:

$$DOM^i(b, b') \equiv \bigwedge_{x \in \Sigma^g_{\bar{\imath}}} \bigwedge_{\theta_i \in \Theta_i} \left( \langle g, (b', x) \rangle^{\forall} \theta_i \supset \langle g, (b, x) \rangle^{\forall} \theta_i \right).$$

This says that whatever reward that can be ensured using the strategy $b'$ can also be ensured with the strategy $b$. In other words, this says that for player $i$, the strategy $b$ weakly dominates $b'$.

Given a strategy $x$ of player $\bar{\imath}$ we can express the fact that the strategy $b$ is better than $b'$ for player $i$ using the formula

$$Better^i_x(b, b') \equiv \bigwedge_{\theta_i \in \Theta_i} (\langle g, (b', x) \rangle^{\forall} \theta_i \supset \langle g, (b, x) \rangle^{\forall} \theta_i)$$

We can express $b$ is the best response of player $i$ for $x$ as $BR^i_x(b) \equiv \bigwedge_{b' \in \Sigma^g_i} Better^i_x(b, b')$.

Having defined best response, the fact that a strategy profile $(b, x)$ constitutes an equilibrium can be expressed as: $EQ(b, x) \equiv BR^i_x(b) \wedge BR^{\bar{\imath}}_b(x)$.

| | $d^2$ | $c^2$ |
|---|---|---|
| $d^1$ | $(P^1, P^2)$ | $(T^1, S^2)$ |
| $c^1$ | $(S^1, T^2)$ | $(R^1, R^2)$ |

**Fig. 3.** Prisoner's Dilemma

**Example:** Consider the prisoner's dilemma game given in Fig. 3. Let the actions $c$ and $d$ correspond to cooperate and defect respectively. The preference ordering over the rewards for $i \in \{1, 2\}$ is given by $S^i \preceq^i P^i \preceq^i R^i \preceq^i T^i$. Let the propositions representing the rewards be $\{\theta^S_i, \theta^P_i, \theta^R_i, \theta^T_i\}$. Consider the formulas:

- $\alpha_1 \equiv \langle g, (c^1, d^2) \rangle^{\forall} \theta^S_1 \supset \langle g, (d^1, d^2) \rangle^{\forall} \theta^S_1$.
- $\alpha_2 \equiv \langle g, (c^1, c^2) \rangle^{\forall} \theta^R_1 \supset \langle g, (d^1, c^2) \rangle^{\forall} \theta^R_1$.

The formula $\alpha_1$ holds since we have $S^1 \preceq^1 P^1$ and $\alpha_2$ holds since $R^1 \preceq^1 T^1$. The formula $\alpha_1 \wedge \alpha_2$ states that irrespective of the move made by player 2, it is better for player 1 to choose $d^1$. In other words, "defect" is a dominant strategy for player 1 in this game.

$\alpha_1$ says that the strategy $d^1$ is better than $c^1$ for player 1 against the strategy $d^2$ of player 2. Since there are only two strategies available for player 1, we get that $d^1$ is the best response for $d^2$. A similar reasoning with respect to player 2 shows that $d^2$ is the best response for $d^1$. From which we get that the strategy profile $(d^1, d^2)$ constitutes an equilibrium profile.

It can be seen, that quite a lot of reasoning that is done in the case of normal form games can be captured by considering game play pairs. The game play pairs in effect, provides us the power of reasoning about restrictions of the full game tree and the ability to compare various such restrictions in terms of the outcomes they guarantee. A player can thus make use of notions like dominant strategy, guaranteed outcome, best response and so on to come up with an appropriate plan of action for the game. The important strategizing notion which is missing in this approach is that of strategic response of a player to the opponent's action. To capture this aspect we need to move over to a model where instead of working with a fixed normal form game, we have a finite set of games and where composition of these games can be performed.

## 4 Strategic response

For the sake of clarity, in subsequent sections, we concentrate on the structure of the game $g$ with respect to the moves of the players and disregard the rewards associated in the game structure. In section 7, after the logic is presented in full generality, we mention the changes required to take care of the rewards present in the game.

Since formulas of the logic refer to the normal form game trees, we first present a syntax for representing such trees.

**Syntax for strategic form games:** Let *Nodes* denote a finite set of nodes, the strategic form game tree is specified using the syntax:

$$\mathrm{G} := \Sigma_{a_m \in J}(x, a_m, y_m).$$

where $x, y_m \in \mathit{Nodes}$, $J = J_1 \times J_2$ for $J_2 \subseteq \Sigma_1$ and $J_2 \subseteq \Sigma_2$.

The game tree $T_g$ generated by the game $g \in \mathrm{G}$ is defined as follows. Let $g = (x, a_1, y_1) + \ldots + (x, a_k, y_k)$, $T_g = (S_g, \Longrightarrow_g, s_{g,0})$ where

- $S_g = \{s_x, s_{y_1}, \ldots, s_{y_k}\}$ and $s_{g,0} = s_x$.
- For $1 \leq j \leq k$ we have $s_x \overset{a_j}{\Longrightarrow}_g s_{y_j}$.

**Syntax:** In addition to the set of propositions, let $\mathcal{G} \subseteq \mathrm{G}$ be a finite set of games. The syntax of the logic is very similar to what was presented earlier:

$$\Phi := p \in P \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \langle g, \eta \rangle^{\forall} \alpha$$

where $g \in \mathcal{G}$ and $\eta \subseteq \Sigma^g$.

**Models:** Formulas of the logic express properties about normal form game trees and plays in the game. Since the modality $\langle g, \eta \rangle^{\forall}$ can be nested, we are in effect talking about finite trees which are generated by composing individual game trees. However there can be an infinite set of finite game trees. One way of giving a finite presentation is to think of the tree being obtained by unfolding of a Kripke structure. As we will see later, the logic cannot distinguish between these two. A model $M = (W, \longrightarrow, V)$ where $W$ is the set of states (or game positions), the relation $\longrightarrow \subseteq W \times \Sigma \times W$ and $V : W \rightarrow 2^P$ is a valuation function.

**Semantics:** The truth of a formula $\alpha \in \Phi$ in a model $M$ and a state $u$ is defined as in the earlier case. The only difference is in the interpretation of $\langle g, \eta \rangle^{\forall} \alpha$ which is given as:

- $M, u \models \langle g, \eta \rangle^{\forall} \alpha$ iff *enabled*$(g, u)$ and for all $w \in$ *tail*$(T_u \upharpoonright g, \eta)$, $M, w \models \alpha$.

Intuitively *enabled*$(g, u)$ says that the $g$ structure can be embedded at state $u$ of the model, with respect to compatibility with the action labels. *tail*$(T_u \upharpoonright g, \eta)$ is the set of nodes of the resulting embedded tree when restricted to plays in $\eta$. $M, w \models \langle g, \eta \rangle^{\forall} \alpha$ says that firstly $g$ can be embedded at $u$ and if $X$ is the set of all states resulting from the plays specified in $\eta$, then the formula $\alpha$ holds in all $w \in X$. The dual $[g, \eta]^{\exists} \alpha$ says: if $g$ can be embedded at the state $u$ then there exists a state $w$ resulting from the plays specified in $\eta$ such that $\alpha$ holds at $w$. Formally the tree embedding and the restriction operation is defined below.

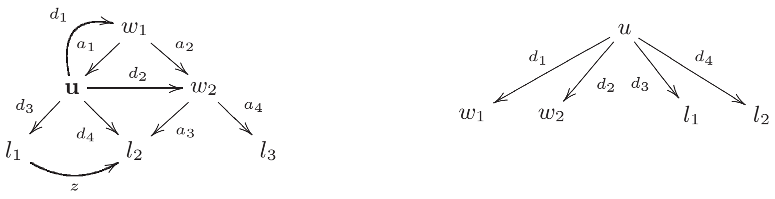
**Restriction on trees:** For $w \in W$, let $T_w$ denote the tree unfolding of $M$ starting at $w$. Given a state $w$ and $g \in \mathcal{G}$, let $T_w = (S_M^w, \Longrightarrow_M, s_w)$ and $T_g = (S_g, \Longrightarrow_g, s_{g,0})$. The restriction of $T_w$ with respect to the game $g$ (denoted $T_w \upharpoonright g$) is the subtree of $T_w$ which is generated by the structure specified by $T_g$. The restriction is defined as follows: $T_w \upharpoonright g = (S, \Longrightarrow, s_0, f)$ where $f : S \rightarrow S_g$. Initially $S = \{s_w\}$, $s_0 = s_w$ and $f(s_w) = s_{g,0}$.

Let $\{a_1, \ldots, a_k\}$ be the outgoing edges of $s_{g,0}$, i.e. for all $j : 1 \leq j \leq k$, $s_{g,0} \overset{a_j}{\Longrightarrow}_g t_j$. For each $a_j$, let $\{s_j^1, \ldots, s_j^m\}$ be the nodes in $S_M^w$ such that $s_w \overset{a_j}{\Longrightarrow}_M s_j^l$ for all $l : 1 \leq l \leq m$. Add nodes $s_j^1, \ldots, s_j^m$ to $S$ and the edges $s_0 \overset{a_j}{\Longrightarrow} s_j^l$ for all $l : 1 \leq l \leq m$. Also set $f(s_j^l) = t_j$.

We say that a game $g$ is enabled at $w$ (denoted *enabled*$(g, w)$) if the tree $T_w \upharpoonright g = (S, \Longrightarrow, s_0, f)$ has the following property:

- $\overrightarrow{s_0} = f(\overrightarrow{s_0})$.

As an illustration of the restriction operation, consider the game $g$ shown in Fig. 2(i) (disregarding the payoff labels). Let the Kripke structure $M$ be as given in Fig. 4(i). For the node $u$ of the Kripke structure, the restriction $T_u \upharpoonright g$ is shown in Fig. 4(ii)



(i)

(ii)

**Fig. 4.** Restriction

**Example:** Strategic response of players can easily be expressed in the logic. For instance, in the prisoner's dilemma game, the "tit-for-tat" strategy for player 1 would be to copy the action of player 2 in the earlier stage. This can be represented as: $\langle g,(c^1,d^2)\rangle^{\forall}\langle g,(d^1,d^2)\rangle^{\forall}\alpha \wedge \langle g,(c^1,c^2)\rangle^{\forall}\langle g,(c^1,c^2)\rangle^{\forall}\alpha$.

The above formalism enables reasoning of bounded levels of strategic response by players. The next step would be to look at unbounded iteration or composition of games. This cannot be achieved by the nesting of modalities and therefore the dynamic structure needs to be brought in at the level of the game play pairs.

## 5 Unbounded game composition

The syntax of the game play pair is enriched as follows:

$$\Gamma := (g,\eta) \mid \xi_1;\xi_2 \mid \xi_1 \cup \xi_2 \mid \xi^* \mid \beta?$$

where $g \in \mathrm{G}$, $\eta \subseteq \Sigma^g$ and $\beta \in \Phi$.

Here we allow $g$ to be any normal form game tree in G (syntax for trees given in section 4). The atomic game play pair $(g,\eta)$ would have the same interpretation as before. $\xi_1 \cup \xi_2$ would mean playing $\xi_1$ or $\xi_2$. Sequencing in our setting does not mean the usual relational composition of games. Rather, it is the composition of game play pairs of the form $(g_1,\eta_1);(g_2,\eta_2)$. A pair $(g,\sigma)$ gives rise to a tree and therefore composition over these trees need to be performed. $\xi^*$ is the iteration of the ';' operator and $\beta?$ tests whether the formula $\beta$ holds at the current state.

The syntax of the formulas of the logic is given by:

$$\Phi := p \in P \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \langle\xi\rangle^{\forall}\alpha$$

where $\xi \in \Gamma$.

Models for the logic are Kripke structures as in the earlier case and the semantics remains the same except for the construct $\langle\xi\rangle^{\forall}\alpha$ which is interpreted as:

- $M,u \models \langle\xi\rangle^{\forall}\alpha$ iff $\exists(u,X) \in R_\xi$ such that $\forall w \in X$ we have $M,w \models \alpha$.

For $\xi \in \Gamma$, we have $R_\xi \subseteq W \times 2^W$. The definition of $R$ in the atomic case is same as the interpretation of game play pair used earlier. i.e.:

- $R_{(g,\sigma)} = \{(u, X) \mid \mathit{enabled}(g, u) \text{ and } X = \mathit{tail}(T_u \upharpoonleft g, \eta)\}$.

The semantics for composite game strategy pairs is given as follows:

- $R_{\xi_1;\xi_2} = \{(u, X) \mid \exists Y = \{v_1, \ldots, v_k\}$ such that $(u, Y) \in R_{\xi_1}$ and $\forall v_j \in Y$ there exists $X_j \subseteq X$ such that $(v_j, X_j) \in R_{\xi_2}$ and $\bigcup_{j=1,\ldots,k} X_j = X\}$.
- $R_{\xi_1 \cup \xi_2} = R_{\xi_1} \cup R_{\xi_2}$.
- $R_{\xi^*} = \bigcup_{n \geq 0} (R_\xi)^n$.
- $R_{\beta?} = \{(u, \{u\}) \mid M, u \models \beta\}$.

The formulas of the logic can not only make assertions about strategies of players but also about the game structure itself. Thus states of the Kripke structure can be viewed as being associated with a set of atomic normal form games. The restriction operation identifies the specific game under consideration, which in turn is determined by the assertions made by formulas of the logic. Consider the following formula:

- $\langle (g, \eta_2); (g', \eta_1) \rangle^{\forall} \mathit{win}_1$ where $\eta_2$ is a strategy for player 2 in game $g$ and $\eta_1$ a strategy of player 1 in $g'$.

This says that assuming in game $g$, player 2 plays according to strategy $\eta_2$ then in $g'$, player 1 can follow $\eta_1$ and ensure $\mathit{win}_1$. Note that this is not same as saying player 1 can ensure $\mathit{win}_1$ in the composed game $g = g; g'$. The fact that player 2 employed strategy $\eta_2$ in game $g$ is used in strategizing by player 1. However, this specification involves only bounded level of strategic response and can thus be expressed in an ATL like framework extended with the appropriate action modalities and past operators. Consider a construct of the form:

- $((g_1, \eta_1); ((g_2, \eta_2) \cup (g_3, \eta_3)))^*; \mathit{win}_2?; (g, \eta)$
  where $\eta_1, \eta_2$ and $\eta_3$ are player 2 strategies in games $g_1, g_2$ and $g_3$ respectively and $\eta$ is a player 1 strategy in game $g$.

This says that if player 2 can ensure $\mathit{win}_2$ by iterating the structure $g_1$ followed by $g_2$ or $g_3$ and employing strategies $\eta_1$ followed by $\eta_2$ or $\eta_3$ then player 1 plays according to $\eta$ in game $g$. Here not only does player 1 assert that player 2 can ensure $\mathit{win}_2$ but also makes assertions about the specific game structure that is enabled and the atomic strategies that player 2 employs. Iteration performed here does not correspond to the assertion that a property holds through out the history. To express such properties, one needs to shift from the ATL setting to a dynamic logic framework.

## 6 Axiom system

We now present an axiomatization of the valid formulas of the logic. We will find the following notations and abbreviations useful.

For $a \in \Sigma$, let $g_a$ denote the normal form game with a unique strategy profile $a$, we define $\langle a \rangle \alpha$ as:

- $\langle a\rangle\alpha \equiv \langle g_a, \{a\}\rangle^{\forall}\top \wedge [g_a, \{a\}]^{\exists}\alpha$.

From the semantics it is easy to see that for $a \in \Sigma$, this gives the usual semantics for $\langle a\rangle\alpha$, i.e. $\langle a\rangle\alpha$ holds at a state $u$ iff there is a state $w$ such that $u \xrightarrow{a} w$ and $\alpha$ holds at $w$.

For a game $g = (x, a_1, y_1) + \ldots + (x, a_k, y_k)$, the formula $g^{\checkmark}$ denotes that the game structure $g$ is enabled. This is defined as:

- $g^{\checkmark} \equiv \bigwedge_{j=1,\ldots,k} \langle a_j\rangle\top$.

**The axiom schemes**

(A1) Propositional axioms:
 (a) All the substitutional instances of tautologies of PC.
(A2) Axiom for single edge games:
 (a) $\langle a\rangle(\alpha_1 \vee \alpha_2) \equiv \langle a\rangle\alpha_1 \vee \langle a\rangle\alpha_2$.
(A3) Dynamic logic axioms:
 (a) $\langle \xi_1 \cup \xi_2\rangle^{\forall}\alpha \equiv \langle \xi_1\rangle^{\forall}\alpha \vee \langle \xi_2\rangle^{\forall}\alpha$.
 (b) $\langle \xi_1; \xi_2\rangle^{\forall}\alpha \equiv \langle \xi_1\rangle^{\forall}\langle \xi_2\rangle^{\forall}\alpha$.
 (c) $\langle \xi^*\rangle^{\forall}\alpha \equiv \alpha \vee \langle \xi\rangle^{\forall}\langle \xi^*\rangle^{\forall}\alpha$.
 (d) $\langle \beta?\rangle^{\forall}\alpha \equiv \beta \supset \alpha$.

For $g = (x, a_1, y_1) + \ldots + (x, a_n, y_n)$ and $\eta \subseteq \Sigma^g$,

(A4) $\langle g, \eta\rangle^{\forall}\alpha \equiv g^{\checkmark} \wedge (\bigwedge_{a\in\eta}[a]\alpha)$.

**Inference rules**

$$(MP)\ \frac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (NG)\ \frac{\alpha}{[a]\alpha}$$

$$(IND)\ \frac{\langle \xi\rangle^{\forall}\alpha \supset \alpha}{\langle \xi^*\rangle^{\forall}\alpha \supset \alpha}$$

Since the relation $R$ is synthesised over tree structures, the interpretation of sequential composition is quite different from the standard one. Consider the usual relation composition semantics for $R_{\xi_1;\xi_2}$, i.e. $R_{\xi_1;\xi_2} = \{(u, X) | \exists Y$ such that $(u, Y) \in R_{\xi_1}$ and for all $v \in Y$, $(v, X) \in R_{\xi_2}\}$. It is easy to see that under this interpretation the formula $\langle \xi_1\rangle^{\forall}\langle \xi_2\rangle^{\forall}\alpha \supset \langle \xi_1; \xi_2\rangle^{\forall}\alpha$ is not valid.

## 7 Completeness

Here we present an overview of the completeness proof for the logic. Details and the full proof can be found in [RS08].

To show completeness, we prove that every consistent formula is satisfiable. Let $\alpha_0$ be a consistent formula, and $CL(\alpha_0)$ denote the subformula closure of $\alpha$. Let $\mathcal{AT}(\alpha_0)$ be the set of all maximal consistent subsets of $CL(\alpha_0)$, referred to as atoms. We use $u, w$ to range over the set of atoms. Each $u \in \mathcal{AT}$ is a finite set of formulas, we denote the conjunction of all formulas in $u$ by $\widehat{u}$. For a nonempty subset $X \subseteq \mathcal{AT}$, we denote by $\widetilde{X}$ the disjunction of all $\widehat{u}, u \in X$. Define a transition relation on $\mathcal{AT}(\alpha_0)$ as follows: $u \xrightarrow{a} w$ iff $\widehat{u} \wedge \langle a \rangle \widehat{w}$ is consistent. The valuation $V$ is defined as $V(w) = \{p \in P \mid p \in w\}$. The model $M = (W, \longrightarrow, V)$ where $W = \mathcal{AT}(\alpha_0)$. Once the Kripke structure is defined, the semantics given earlier defines the relation $R_{(g,\eta)}$ on $W \times 2^W$ for $g \in \mathrm{G}$.

However for the completeness theorem, we need to also specify the relation between a pair $(u, X)$ being in $R_{(g,\eta)}$ and the consistency requirement on $u$ and $X$. This is done in the following lemma:

**Lemma 7.1.** *For all $g \in \mathrm{G}$, for all $i \in \{1, 2\}$ and for all $\eta \subseteq \Sigma^g$, for all $X \subseteq W$ and for all $u \in W$ the following holds:*

1. *if $(u, X) \in R_{(g,\eta)}$ then $\widehat{u} \wedge \langle g, \eta \rangle^{\forall} \widetilde{X}$ is consistent.*
2. *if $\widehat{u} \wedge \langle g, \eta \rangle^{\forall} \widetilde{X}$ is consistent then there exists $X' \subseteq X$ such that $(u, X') \in R_{(g,\eta)}$.*

Using techniques developed in propositional dynamic logic, the following two lemmas can be shown.

**Lemma 7.2.** *For all $\xi \in \Gamma$, for all $X \subseteq W$ and $u \in W$, if $\widehat{u} \wedge \langle \xi \rangle^{\forall} \widetilde{X}$ is consistent then there exists $X' \subseteq X$ such that $(u, X') \in R_{\xi}$.*

**Lemma 7.3.** *For all $\langle \xi \rangle^{\forall} \alpha \in CL(\alpha_0)$, for all $u \in W$, $\widehat{u} \wedge \langle \xi \rangle^{\forall} \alpha$ is consistent iff there exists $(u, X) \in R_{\xi}$ such that $\forall w \in X$, $\alpha \in w$.*

**Theorem 7.1.** *For all $\beta \in CL(\alpha_0)$, for all $u \in W$, $M, u \models \beta$ iff $\beta \in u$.*

The theorem follows from lemma 7.3 by a routine inductive argument.

**Decidability:** Since $|\Sigma|$ is constant, the size of $CL(\alpha_0)$ is linear in $|\alpha_0|$. Atoms are maximal consistent subsets of $CL(\alpha_0)$, hence $|\mathcal{AT}(\alpha_0)|$ is exponential in the size of $\alpha_0$. It follows from the completeness theorem that given a formula $\alpha_0$, if $\alpha_0$ is satisfiable then it has a model of exponential size. For all $\xi \in \Gamma$ occurring in $\alpha_0$, the relation $R_{\xi}$ can be computed in time exponential in the size of the model. Therefore we get that the logic is decidable in nondeterministic double exponential time.

**Adding rewards to the game structure:** The syntax of game trees presented in section 4 can be easily modified to include the payoff (reward) information for the game. Each node "$y_j$" needs to be replaced with a tuple of the form $r_j = (r_1^j, r_2^j)$ where $r_j \in R$. Models are Kripke structures $M = (W, \longrightarrow, V, \lambda)$ where $\lambda : W \to R$. For a game $g$ the generated tree $T_g = (S_g, \Longrightarrow_g, \lambda_g, s_{g,0})$. The tree restriction $T_w \restriction g$ (presented in section 4) is therefore a structure of the form $(S, \Longrightarrow, \lambda, s_0, f)$ where $\lambda : S \to R$. The condition for a game $g$ being enabled at a state $w$ needs to capture the rewards of the game as well and therefore needs to be modified as follows:

- $\forall s \in S \setminus \{s_0\}$, $\lambda(s) = \lambda_g(f(s))$.
- $\overrightarrow{s_0} = f(\overrightarrow{s_0})$.

As mentioned in section 3, let $\Theta_i$ be the finite set of special propositions coding up the rewards of players $R_i$ for $i \in \{1, 2\}$. For a game $g = (x, a_1, (r_1^1, r_2^1)) + \ldots + (x, a_k, (r_1^k, r_2^k))$, the enabling of $g$ can be represented in the logic as:

- $g^{\checkmark} \equiv \bigwedge_{j=1,\ldots,k}(\langle a_j \rangle \top \wedge [a_j](\theta_1^j \wedge \theta_2^j))$.

In the axiom scheme (section 6), the following two axioms are added along with the propositional axioms to capture the ordering of the rewards.

- $(\bigvee_{\theta_i \in \Theta_i} \theta_i)$ for $i \in \{1, 2\}$.
- $\bigwedge_{\theta_i^j \in \Theta_i}(\theta_i^j \supset \bigwedge_{k=1,\ldots,j} \theta_i^k)$ for $i \in \{1, 2\}$.

It is easy to check that with the above mentioned modification, the completeness theorem follows.

## 8 Discussion

By considering game play pairs, we are able to reason about restrictions of the game tree and thereby express game theoretic notions like a player's best response for an opponents strategy and equilibrium. In contrast, the approach taken in [RS08] is closer to the style of game logics: the reasoning is about what a player can ensure by following a certain strategy specification where all possible strategies of the opponent is taken into account. However, at the compositional level, the axiom system remains the same. This shows that the framework being considered is quite general, and is not dependent on the exact game representation. For a specific representation under consideration, once the axioms for the atomic case are presented appropriately, the theory lifts quite neatly.

This paper deals with games of perfect information, since at the end of each stage, all the players know the strategy profile along with the outcomes. It also operates within the framework of foundations for game theory in modal logics. In this sense, it does not try to offer new models for game theory but explicate the reasoning involved. It is worth noting that almost all the analysis performed in reasoning about games, including the related works mentioned earlier, are based on games of perfect information. Coming up with logical formalisms and extending the techniques to reason in games of imperfect information is a challenging task.

To come up with prescriptive mechanisms which provides advice to players on how to play, it is essential to be able to represent a player's expectations about the behaviour of the opponent. The expectations need not necessarily be represented in a probabilistic manner. Introducing expectations of players is particularly interesting in the framework of unbounded game composition as it allows players to learn from the past information, revise their expectations and accordingly make use of it to generate sophisticated plans. Enriching the framework to be able to represent expectations of players is left as future work.

**Acknowledgements**

We thank the anonymous referees for their helpful comments and suggestions.

## References

[Ago06] T. Agotnes. Action and knowledge in alternating time temporal logic. *Synthese*, 149(2):377–409, 2006.

[AHK02] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.

[Bon02] G. Bonanno. Modal logic and game theory: Two alternative approaches. *Risk Decision and Policy*, 7:309–324, December 2002.

[Bor07] S. Borgo. Coalitions in action logic. In *Proceedings IJCAI'07*, pages 1822–1827, 2007.

[Gho08] S. Ghosh. Strategies made explicit in dynamic game logic. In *Logic and the Foundations of Game and Decision Theory*, 2008.

[JvdH04] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[Par85] R. Parikh. The logic of games and its applications. *Annals of Discrete Mathematics*, 24:111–140, 1985.

[Pau01] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, October 2001.

[RS08] R. Ramanujam and S. Simon. Dynamic logic on games with structured strategies. In *The Principles of Knowledge Representation and Reasoning (to appear)*, 2008. `http://www.imsc.res.in/~sunils/papers/pdf/rs-kr08.pdf`.

[vB01] J. van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[vB02] J. van Benthem. Extensive games as process models. *Journal of Logic Language and Information*, 11:289–313, 2002.

[vdHJW05] W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 157–164, 2005.

[vdHW02] W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems*, pages 1167–1174, 2002.

[WvdHW07] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *Theoretical Aspects of Rationality and Knowledge*, pages 269–278, 2007.

# Information-Based Argumentation

Carles Sierra[1] and John Debenham[2]

[1] Institut d'Investigació en Intel·ligència Artificial – IIIA,
Spanish Scientific Research Council, CSIC
08193 Bellaterra, Catalonia, Spain sierra@iiia.csic.es
[2] University of Technology, Sydney, Australia debenham@it.uts.edu.au

**Abstract.** Information-based argumentation aims to model the partner's reasoning apparatus to the extent that an agent can work with it to achieve outcomes that are mutually satisfactory and lay the foundation for continued interaction and perhaps lasting business relationships. Information-based agents take observations at face value, qualify them with a belief probability and build models solely on the basis of messages received. Using augmentative dialogue that describes *what* is *good* or *bad* about proposals, these agents observe such statements and aim to model the way their partners react, and then to generate dialogue that works in harmony with their partner's reasoning.

## 1 Introduction

This paper is in the area labelled: *information-based agency* [1]. An information-based agent has an identity, values, needs, plans and strategies all of which are expressed using a fixed ontology in probabilistic logic for internal representation and in an illocutionary language [2] for communication. All of the forgoing is represented in the agent's deliberative machinery.

In line with our "Information Principle" [2], an information-based agent makes no *a priori* assumptions about the states of the world or the other agents in it — represented in a world model inferred from the messages that it receives. These agents build up their models by comparing expectation with observation — in this way we have constructed general models of trust, honour and reliability in a single framework [1].

[2] describes a rhetorical argumentation framework that supports argumentative negotiation. It does this by taking into account: the relative information gain of a new utterance and the relative semantic distance between an utterance and the dialogue history. Then [4] considered the effect that argumentative dialogues have on the on-going *relationship* between a pair of negotiating agents. Neither of these contributions addressed the relationship between argumentative utterances or strategies for argumentation. In this paper we adress these two issues.

The basis of our approach differs from [5] who builds on the notion of one argument "attacking" another. With the exception of a logical 'attack', whether one argument attacks another or not will depend on the receiving agent's private circumstances that are unlikely to be fully articulated. Thus, the notion of attack is of little use to information-based agents that build their models on the contents of utterances. This paper considers how to *counter* the effect of the partner agent's arguments, and aims to lead a negotiation towards some desired outcome by persuasive argumentation.

This paper is based in rhetorical argumentation [6]. For example, suppose I am shopping for a new car and have cited "suitability for a family" as a criterion. The salesman says "This LandMonster is great value," and I reply "My grandmother could not climb into that." Classical argumentation may attempt to refute the matriarch's lack of gymnastic prowess or the car's inaccessibility. Taking a less confrontational and more constructively persuasive view we might note that this statement impacts negatively on the "suitability for a family" criterion, and attempt to counter that impact possibly with "It's been voted No 1 for children." Although a smarter response may look for an argument that is semantically closer: "The car's height ensures a very comfortable ride over rough terrain that is popular with old people."

Information-based agents build their world models using an expectation/observation framework; this includes a model of the negotiation partner's behaviour. Agents form an *a priori* expectation of the significance of every event that occurs, and when the effect of that event is finally observed they revise their expectations. The model of behaviour includes measures of: trust, honour, reliability, intimacy, balance and disposition — *disposition* attempts to model what the partner means which may not be what they say. These measures are summarised using: temporal criteria, the structure of the ontology, and the illocutionary category of observed utterances.

Our argumentation agent has to perform two key functions: to understand incoming utterances and to generate responses. In Section 2 we describe the communication model and an argumentation language that admits Prolog-like statements. The approach is founded on a model of contract acceptance that is described in Section 3. Section 4 details a scenario that provides the context for the discussion. Sections 5 and 6 consider the scenario from each side of the bargaining table. Reactive and proactive argumentation strategies are given in Section 7, and Section 8 concludes.

## 2 Communication Model

This paper is written from the point of view of an agent $\alpha$ that is engaged in argumentative interaction with agent $\beta$. The history of all argumentative exchanges is the agents' *relationship*. We assume that their utterances, $u$, can be organised into distinct dialogues, $\Psi^t$. For simplicity we assume that at most one dialogue exists at any time. We assume that $\alpha$ and $\beta$ are negotiating with the mutual aim of signing a contract, where the contract will be an instantiation of the mutually-understood object $o(\Psi^t)$. We assume that this negotiation is taking place through the exchange of proposals accompanied by argumentative dialogue.

### 2.1 Ontology

In order to define a language to structure agent dialogues we need an ontology that includes a (minimum) repertoire of elements: a set of *concepts* organised in a is-a hierarchy (e.g. platypus is a mammal, Australian-dollar is a currency), and a set of relations over these concepts (e.g. price(beer,AUD) [7]:

An ontology is a tuple $\mathcal{O} = (V, R, \leq, \sigma)$ where:

1. $V$ is a finite set of concept symbols (including basic data types), i.e. a vocabulary;
2. $R$ is a finite set of relation symbols;
3. $\leq$ is a reflexive, transitive and anti-symmetric relation on $V$ (a partial order), and
4. $\sigma : R \rightarrow V^+$ is the function assigning to each relation symbol its arity

where $\leq$ is the traditional *is-a* hierarchy. To simplify computations in the computing of probability distributions we will assume that there is a number of disjoint *is-a* trees covering different ontological spaces (e.g. a tree for types of fabric, a tree for shapes of clothing, and so on). $R$ contains relations between the concepts in the hierarchy, this is needed to define deals as tuples of issues. Semantic distance plays a fundamental role in strategies for information-based agency, see [1] for details.

### 2.2 Language

The general argumentation language described here was first reported in [4]. The discussion is from the point of view of an information-based agent $\alpha$ in a multiagent system where $\alpha$ interacts with negotiating agents, $\beta_i$, and information providing agents, $\theta_j$: $\{\alpha, \beta_1, \ldots, \beta_o, \theta_1, \ldots, \theta_t\}$.

The shape of the language that $\alpha$ uses to represent the information received and the content of its dialogues depends on two fundamental actions: (i) passing information, and (ii) exchanging proposals and contracts. A contract $(a, b)$ between agents $\alpha$ and $\beta$ is a pair where $a$ and $b$ represent the actions that agents $\alpha$ and $\beta$ are responsible for respectively. *Contracts* signed by agents and *information* passed by agents, are similar to norms in the sense that they oblige agents to behave in a particular way, so as to satisfy the conditions of the contract, or to make the world consistent with the information passed. Contracts and Information can thus be thought of as normative statements that restrict an agent's behaviour.

$\alpha$'s communication language has two fundamental primitives: $\text{Commit}(\alpha, \beta, \varphi)$ to represent, in $\varphi$, the world that $\alpha$ aims at bringing about and that $\beta$ has the right to verify, complain about or claim compensation for any deviations from, and $\text{Done}(u)$ to represent the event that a certain action $u$[3] has taken place. In this way, norms, contracts, and information chunks will be represented as instances of $\text{Commit}(\cdot)$ where $\alpha$ and $\beta$ are individual agents. Language $\mathcal{L}$ is the set of utterances $u$ defined as:

$$\begin{aligned} u &::= \textit{illoc}(\alpha, \beta, \varphi, t) \mid u; u \mid \textbf{Let } \textit{context } \textbf{In } u \textbf{ End} \\ \varphi &::= \textit{term} \mid \text{Done}(u) \mid \text{Commit}(\alpha, \beta, \varphi) \mid \varphi \wedge \varphi \mid \\ &\qquad \varphi \vee \varphi \mid \neg\varphi \mid \forall x.\varphi_x \mid \exists x.\varphi_x \\ \textit{context} &::= \varphi \mid \textit{id} = \varphi \mid \textit{prolog\_clause} \mid \textit{context}; \textit{context} \end{aligned}$$

where $\varphi_x$ is a formula with free variable $x$, $illoc$ is any appropriate set of illocutionary particles, ';' means sequencing, and *context* represents either previous agreements, previous illocutions, the ontological working context, that is a projection of the ontological trees that represent the focus of the conversation, or code that aligns the ontological

[3] Without loss of generality we will assume that all actions are dialogical.

differences between the speakers needed to interpret an (illocutionary) action $u$. Representing an ontology as a set predicates in Prolog is simple. The set *term* contains instances of the ontology concepts and relations.[4]

For example, we can represent the following offer: "If you spend a total of more than €100 in my shop during October then I will give you a 10% discount on all goods in November", as:

Offer( $\alpha$, $\beta$, spent($\beta$, $\alpha$, October, x) $\wedge$ x $\geq$ €100 $\rightarrow$
    $\forall$ y. Done(Inform($\beta$, $\alpha$, pay($\beta$, $\alpha$, y), November)) $\rightarrow$ Commit($\alpha$, $\beta$, discount(y,10%)))

## 3 Contract Acceptance

No matter what interaction strategy an agent uses, and no matter whether the communication language is that of simple bargaining or rich argumentation, a negotiation agent will have to decide whether or not to sign each contract on the table. We will argue in Section 5 that the buyer will be uncertain of his preferences in our Scenario described in Section 4. If an agent's preferences are uncertain then it may not make sense to link the agent's criterion for contract acceptance to a strategy that aims to optimise its utility. Instead, we pose the more general question: "how certain am I that $\delta = (\phi, \varphi)$ is a good contract to sign?" — under realistic conditions this may be easy to estimate. $\mathbb{P}^t(\text{sign}(\alpha, \beta, \chi, \delta))$ estimates the certainty, expressed as a probability, that $\alpha$ should sign[5] proposal $\delta$ in satisfaction of her need $\chi$, where in $(\phi, \varphi)$ $\phi$ is $\alpha$'s commitment and $\varphi$ is $\beta$'s. $\alpha$ will accept $\delta$ if: $\mathbb{P}^t(\text{sign}(\alpha, \beta, \chi, \delta)) > c$, for some level of certainty $c$.

To estimate $\mathbb{P}^t(\text{sign}(\alpha, \beta, \chi, \delta))$, $\alpha$ will be concerned about what will occur if contract $\delta$ is signed. If agent $\alpha$ receives a commitment from $\beta$, $\alpha$ will be interested in any variation between $\beta$'s commitment, $\varphi$, and what is actually observed, as the enactment, $\varphi'$. We denote the relationship between commitment and enactment:

$$\mathbb{P}^t(\text{Observe}(\alpha, \varphi')|\text{Commit}(\beta, \alpha, \varphi))$$

simply as $\mathbb{P}^t(\varphi'|\varphi) \in \mathcal{M}^t$, and now $\alpha$ has to estimate her belief in the acceptability of each possible outcome $\delta' = (\phi', \varphi')$. Let $\mathbb{P}^t(\text{acc}(\alpha, \chi, \delta'))$ denote $\alpha$'s estimate of her belief that the outcome $\delta'$ will be acceptable in satisfaction of her need $\chi$, then we have:

$$\mathbb{P}^t(\text{sign}(\alpha, \beta, \chi, \delta)) = f(\mathbb{P}^t(\delta'|\delta), \mathbb{P}^t(\text{acc}(\alpha, \chi, \delta'))) \tag{1}$$

for some function $f$;[6] if $f$ is the arithmetic product then this expression is mathematical expectation. $f$ may be more sensitive; for example, it may be defined to ensure that no contract is signed if there is a significant probability for a catastrophic outcome.

There is no prescriptive way in which $\alpha$ should define $\mathbb{P}^t(\text{acc}(\alpha, \chi, \delta'))$, it is a matter for applied artificial intelligence to capture the essence of what matters in the

[4] We assume the convention that $V(v)$ means that $v$ is an instance of concept $V$ and $r(v_1, \ldots, v_n)$ implicitly determines that $v_i$ is an instance of the concept in the $i$-th position of the relation $r$.

[5] A richer formulation is $\mathbb{P}^t(\text{eval}(\alpha, \beta, \chi, \delta) = e_i)$ where eval$(\cdot)$ is a function whose range is some descriptive evaluation space containing terms such as "unattractive in the long term".

[6] $\beta$ influences the equation in the sense that different $\beta$s yield different $\mathbb{P}^t(\delta'|\delta)$.

application. In any real application the following three components at least will be required. $\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta'))$ represents $\alpha$'s belief that enactment $\delta'$ will satisfy her need $\chi$. $\mathbb{P}^t(\text{obj}(\delta'))$ represents $\alpha$'s belief that $\delta'$ is a fair deal against the open marketplace — it represents $\alpha$'s *objective* valuation. $\mathbb{P}^t(\text{sub}(\alpha, \chi, \delta'))$ represents $\alpha$'s belief that $\delta'$ is acceptable in her own terms taking account of her ability to meet her commitment $\phi$ [2] [1], and any way in which $\delta'$ has value to her personally[7] — it represents $\alpha$'s *subjective* valuation. That is:

$$\mathbb{P}^t(\text{acc}(\alpha, \chi, \delta')) = g(\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta')), \mathbb{P}^t(\text{obj}(\delta')), \mathbb{P}^t(\text{sub}(\alpha, \chi, \delta'))) \tag{2}$$

for some function $g$.

Suppose that an agent is able to estimate: $\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta'))$, $\mathbb{P}^t(\text{obj}(\delta'))$ and $\mathbb{P}^t(\text{sub}(\alpha, \chi, \delta'))$. The specification of the aggregating $g$ function will then be a strictly subjective decision. A highly cautious agent may choose to define:

$$\mathbb{P}^t(\text{acc}(\alpha, \chi, \delta')) = \begin{cases} 1 & \text{if: } \mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta')) > \eta_1 \\ & \wedge\, \mathbb{P}^t(\text{obj}(\delta')) > \eta_2 \;\wedge\; \mathbb{P}^t(\text{sub}(\alpha, \chi, \delta')) > \eta_3 \\ 0 & \text{otherwise.} \end{cases}$$

for some threshold constants $\eta_i$. Whereas an agent that was prepared to permit some propagation of confidence from one factor to compensate another could define:

$$\mathbb{P}^t(\text{acc}(\alpha, \chi, \delta')) = \mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta'))^{\eta_1} \times \mathbb{P}^t(\text{obj}(\delta'))^{\eta_2} \times \mathbb{P}^t(\text{sub}(\alpha, \chi, \delta'))^{\eta_3}$$

where the $\eta_i$ balance the influence of each factor.

The point of this is: if an agent aims to produce persuasive argumentative dialogue then in the absence of any specific information concerning the structure of $g$ the agent should ignore $g$ and concentrate on the three categories: $\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta'))$, $\mathbb{P}^t(\text{obj}(\delta'))$ and $\mathbb{P}^t(\text{sub}(\alpha, \chi, \delta'))$.

So how then will $\alpha$ specify: $\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta))$, $\mathbb{P}^t(\text{sub}(\alpha, \chi, \delta))$ and $\mathbb{P}^t(\text{obj}(\delta))$? Of these three factors only $\mathbb{P}^t(\text{obj}(\delta))$ has a clear meaning, but it may only be estimated if there is sufficient market data available. In the case of selling sardines this may well be so, but in the case of Google launching a take-over bid for Microsoft it will not[8]. Concerning $\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta))$ and $\mathbb{P}^t(\text{sub}(\alpha, \chi, \delta))$ we assume that an agent will somehow assess each of these as some combination of the confidence levels across a set of privately-known *criteria*. For example, if I am buying a camera then I may be prepared to define:

$$\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta)) = h(\mathbb{P}^t(\text{easy-to-use}(\alpha, \delta)), \mathbb{P}^t(\text{well-built}(\alpha, \delta))) \tag{3}$$

for some function $h$. Any attempt to model another agent's $h$ function will be as difficult as modelling $g$ above. *But*, it is perfectly reasonable to suggest that by observing my argumentative dialogue an agent could form a view as to which of these two criteria above was more important.

[7] For example, when buying a new digital camera, $\alpha$ may give a high subjective valuation to a camera that uses the same memory cards as her existing camera.

[8] In this example the subjective valuation will be highly complex.

This paper considers how an agent may observe the argumentative dialogue with the aim of modelling, within each of the three basic factors, the partner's criteria and the relative importance of those criteria. In repeated dealings between two agents, this model may be strengthened when the objects of the successive negotiations are semantically close but not necessarily identical.

## 4 The Scenario

Rhetorical argumentation is freed from the rigour of classical argumentation and descriptions of it can take the form of "this is how it works here" and "this is how it works there" without describing a formal basis. We attempt to improve on this level of vagary by using a general scenario and describing the behaviour of our agents within it.

In a general retail scenario there is a seller agent, $\alpha$, and a buyer, $\beta$. The items for sale are abstracted from: digital cameras, mobile phones, PDAs, smart video recorders, computer software, sewing machines and kitchen mixers. The features of an item are those that are typically listed on the last few pages of an instruction booklet. For example, a camera's features could include the various shutter speeds that it is capable of, the various aperture settings, the number of years of warranty, and so on — together the *features* describe the capabilities of the item. For the purpose of comparison with other items, $\beta$ will consider a particular item as a typed Boolean vector over the (possible) features of each item available, this vector shows which feature is present. The *state* of an item is then specified by identifying which of the item's features are 'on'. For example, the state of a camera could be: 'ready' with aperture set to 'f8' and shutter speed set to '1 500'th of a second'. In this scenario an *offer* is a pair (supply of a particular item, supply of some money) being $\alpha$'s and $\beta$'s commitments respectively.

$\beta$ may wish to know how well an item performs certain tasks. Software agents are not naturally endowed with the range of sensory and motor functions to enable such an evaluation. We imagine that the seller agent has an associated tame human who will demonstrate how the various items perform particular tasks on request, but performs no other function. We also imagine that the buyer agent has an associated tame human who can observe what is demonstrated, articulates an evaluation of it that is passed to its own agent, but performs no other function.

To simplify our set up we assume that the seller, $\alpha$, is $\beta$'s only source of information about what tasks each item can perform, and, as we describe below, what sequence of actions are necessary to make an item perform certain tasks[9]. That is, our multiagent system consists only of $\{\alpha, \beta\}$, and the buyer is denied access to product reviews, but *does* have access to market pricing data. This restriction simplifies the interactions and focusses the discussion on the argumentation.

For example, if the item is a camera the buyer may wish to observe how to set the camera's states so that it may be used for 'point-and-shoot' photography. If the item is a sewing machine she may wish to see how to make a button hole on a piece of cloth. If the item is graphics software she may wish to see how to draw a polygon with a two-pixel red line and to colour the polygon's interior blue. These tasks will be achieved by

[9] In other words, the sort of information that is normally available in the item's Instruction Booklet — we assume that $\alpha$ conveys this information accurately.

enacting a process that causes the item to pass though a sequence of states that will be explained to $\beta$ by $\alpha$. So far our model consists of: features, states, sequences and tasks.

We assume that the object of the negotiation is clear where the object is an uninstantiated statement of what both agents jointly understand as the intended outcome — e.g. I wish to exchange a quantity of eggs of certain quality for cash. We assume that each agent is negotiating with the aim of satisfying some goal or need that is private knowledge. In determining whether a negotiation outcome is acceptable in satisfaction of a need we assume that an agent will blend the factors in our acceptance model described in Section 3. We assume that for each factor an agent will articulate a set of *criteria* that together determine whether the factor is acceptable. The criteria may include private information such as deadlines.

More formally, there is a set of feature names, $\mathcal{F}$, a set of item names, $\mathcal{I}$, a feature mapping: $\text{feature} : \mathcal{I} \rightarrow \times^n(\mathbb{B} : \mathcal{F})$ where there are $n$ feature names, and $\mathbb{B}$ is a boolean variable that may be $\top$ or $\perp$. Each item name belongs to a unique concept — e.g.: "Nikon123 is-a camera". For any particular item name, $\nu$, $\text{feature}(\nu)$ will be a typed Boolean vector indicating which features that item $\nu$ possesses. Let $\mathcal{F}_\nu$ be the set of $n_\nu$ features that item $\nu$ possesses. At any particular time $t$, the state of an item is a mapping: $\text{state}^t : \mathcal{I} \rightarrow \times^{n_\nu}(\mathbb{B} : \mathcal{F}_\nu)$ where the value $\top$ denotes that the corresponding feature of that item is 'on'. A *sequence* is an ordered set of states, $(\boldsymbol{w}_i)$, where successive states differ in one feature only being on and off. A sequence is normally seen as performing a *task* that are linked by the mapping: to-do : $\mathcal{T} \rightarrow 2^{\mathcal{S}}$ where $\mathcal{T}$ is the set of tasks and $\mathcal{S}$ the set of all possible sequences — that is, there many be several sequences that perform a task. If a sequence is *performed* on an item then, with the assistance of a human, the agent rates how well it believes the sequence performs the associated task. The evaluation space, $\mathcal{E}$, could be $\{$good, OK, bad$\}$. A criterion is a predicate: criterion($\nu$), meaning that the item $\nu$ satisfies criterion 'criterion'. The set of criteria is $\mathcal{C}$. The argumentation requirements include (where $x \in V$, $c \in \mathcal{C}$, $v = \text{feature}(x)$, $y \in \mathcal{T}$, $z \in \mathcal{S}$, and $r \in \mathcal{R}$):

- "I need an $x$"
- "What sort of $x$ do you need?"
- "I need an $x$ that satisfies criterion $c$"
- "What features does $x$ have?"
- "$x$ has features $v$"
- "How do you make $x$ do $y$"
- "The sequence $z$ performed on $x$ does $y$"
- "Perform sequence $z$ on $x$"
- "If sequence $z$ is performed on $x$ then how would you rate that?"
- "I rate the sequence $z$ as performed on $x$ as $r$"

## 5 The Buyer Assesses A Contract

In this Section we consider how the buyer might use the general framework in Section 3 to assess a contract[10]. In general an agent will be concerned about the enactment of

[10] The seller will have little difficulty in deciding whether a contract is acceptable if he knows what the items cost.

any contract signed as described in Equation 1. In the scenario described in Section 4, enactment is not an issue, and so we focus on Equation 2. To simplify things we ignore the subjective valuation factor. Before addressing the remaining two factors we argue that the buyer will not necessarily be preference aware.

Consider a human agent with a need for a new camera who goes to a trusted camera shop. If the agent is preference aware he will be able to place the twenty to fifty cameras on offer in order of preference. If is reasonable to suggest that a normal, intelligent human agent could not achieve this with any certainty, nor could he with confidence represent his uncertainty in his preferences as a probability distribution over his preferences. This lack of awareness of preferences may be partially due to lack of information about each camera. But, what could "perfect information" realistically mean in this example? Even if the purchaser could borrow all the cameras for a day and had access to trusted, skilled users of each camera even then we suggest that our human agent would still be unable to specify a preference order with confidence. The underlying reason being the size and complexity of the issue space required to describe all of the features of every camera on offer, and the level of subjective judgement required to relate combinations of those features to meaningful criteria.

### 5.1 Assessing $\mathbb{P}^t(\text{satisfy}(\beta, \chi, \delta))$

First $\beta$ must give meaning to $\mathbb{P}^t(\text{satisfy}(\beta, \chi, \delta))$ by defining suitable criteria and the way that the belief should be aggregated across those criteria. Suppose one of $\beta$'s criteria is $\mathbb{P}^t(\text{ease-of-use}(\beta, \delta))$. The idea is that $\beta$ will ask $\alpha$ to demonstrate how certain tasks are performed, will observe the sequences that $\alpha$ performs, and will use those observations to revise this probability distribution until some clear verdict appears.

Suppose the information acquisition process is managed by a plan $\pi$. Let random variable $X$ represent $\mathbb{P}^t(\text{ease-of-use}(\beta, \delta) = e_i)$ where the $e_i$ are values from an evaluation space that could be $\mathcal{E} = \{\text{fantastic, acceptable, just OK, shocking}\}$. Then given a sequence $s$ that was supposed to achieve task $\tau$, suppose that $\beta$'s tame human rates $s$ as evidence for ease-of-use as $e \in \mathcal{E}$ with probability $z$. Suppose that $\beta$ attaches a weighting $\mathbb{R}^t(\pi, \tau, s)$ to $s$, $0 < \mathbb{R} < 1$, which is $\beta$'s estimate of the *significance* of the observation of sequence $s$ within plan $\pi$ as an indicator of the true value of $X$. For example, the on the basis of the observation alone $\beta$ might rate ease-of-use as $e = \text{acceptable}$ with probability $z = 0.8$, and separately give a weighting of $\mathbb{R}^t(\pi, \tau, s) = 0.9$ to the sequence $s$ as an indicator of ease-of-use. For an information-based agent each plan $\pi$ has associated *update functions*, $J_\pi(\cdot)$, such that $J_\pi^X(s)$ is a set of linear constraints on the posterior distribution for $X$. In this example, the posterior value of 'acceptable' would simply be constrained to 0.8.

Denote the prior distribution $\mathbb{P}^t(X)$ by $\boldsymbol{p}$, and let $\boldsymbol{p}_{(s)}$ be the distribution with minimum relative entropy[11] with respect to $\boldsymbol{p}$: $\boldsymbol{p}_{(s)} = \arg\min_{\boldsymbol{r}} \sum_j r_j \log \frac{r_j}{p_j}$ that satisfies

[11] Given a probability distribution $\boldsymbol{q}$, the *minimum relative entropy distribution* $\boldsymbol{p} = (p_1, \ldots, p_I)$ subject to a set of $J$ linear constraints $\boldsymbol{g} = \{g_j(\boldsymbol{p}) = \boldsymbol{a_j} \cdot \boldsymbol{p} - c_j = 0\}, j = 1, \ldots, J$ (that must include the constraint $\sum_i p_i - 1 = 0$) is: $\boldsymbol{p} = \arg\min_{\boldsymbol{r}} \sum_j r_j \log \frac{r_j}{q_j}$. This may be calculated by introducing Lagrange multipliers $\boldsymbol{\lambda}$: $L(\boldsymbol{p}, \boldsymbol{\lambda}) = \sum_j p_j \log \frac{p_j}{q_j} + \boldsymbol{\lambda} \cdot \boldsymbol{g}$. Minimising $L$,

the constraints $J_s^X(s)$. Then let $\boldsymbol{q}_{(s)}$ be the distribution:

$$\boldsymbol{q}_{(s)} = \mathbb{R}^t(\pi, \tau, s) \times \boldsymbol{p}_{(s)} + (1 - \mathbb{R}^t(\pi, \tau, s)) \times \boldsymbol{p} \tag{4}$$

and then let:

$$\mathbb{P}^t(X_{(s)}) = \begin{cases} \boldsymbol{q}_{(s)} & \text{if } \boldsymbol{q}_{(s)} \text{ is more interesting than } \boldsymbol{p} \\ \boldsymbol{p} & \text{otherwise} \end{cases} \tag{5}$$

A general measure of whether $\boldsymbol{q}_{(s)}$ is more interesting than $\boldsymbol{p}$ is: $\mathbb{K}(\boldsymbol{q}_{(s)} \| \mathbb{D}(X)) > \mathbb{K}(\boldsymbol{p} \| \mathbb{D}(X))$, where $\mathbb{K}(\boldsymbol{x} \| \boldsymbol{y}) = \sum_j x_j \log \frac{x_j}{y_j}$ is the Kullback-Leibler distance between two probability distributions $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\mathbb{D}(X)$ is the expected distribution in the absence of any observations — $\mathbb{D}(X)$ could be the maximum entropy distribution. Finally, $\mathbb{P}^{t+1}(X) = \mathbb{P}^t(X_{(s)})$. This procedure deals with integrity decay, and with two probabilities: first, the probability $z$ in the rating of the sequence $s$ that was intended to achieve $\tau$, and second $\beta$'s weighting $\mathbb{R}^t(\pi, \tau, s)$ of the significance of $\tau$ as an indicator of the true value of $X$. Equation 5 is intended to prevent weak information from decreasing the certainty of $\mathbb{P}^{t+1}(X)$. For example if the current distribution is $(0.1, 0.7, 0.1, 0.1)$, indicating an "acceptable" rating, then weak evidence $\mathbb{P}(X = \text{acceptable}) = 0.25$ is discarded.

Equation 4 simply adds in new evidence $\boldsymbol{p}_{(s)}$ to $\boldsymbol{p}$ weighted with $\mathbb{R}^t(\pi, \tau, s)$. This is fairly crude, but the observations are unlikely to be independent and the idea is that $\pi$ will specify a "fairly comprehensive" set of tasks aimed to determine $\mathbb{P}^t(X)$ to a level of certainty sufficient for Equation 2.

### 5.2 Assessing $\mathbb{P}^t(\text{obj}(\delta))$

$\mathbb{P}^t(\text{obj}(\delta))$ estimates the belief that $\delta$ is acceptable in the open-market that $\beta$ may observe in the scenario. Information-based agents model what they don't know with certainty as probability distributions. Suppose that $X$ is a discrete random variable whose true value is the open-market value of an item. First, $\beta$ should be able to bound $X$ to an interval $(x_{\min}, x_{\max})$ — if this is all the evidence that $\beta$ can muster then $X$ will be the flat distribution (with maximum entropy) in this interval, and $\mathbb{P}^t(\text{obj}((\text{item}, y)) = \sum_{x \geq y} \mathbb{P}(X = x)$. $\beta$ may observe evidence, perhaps as observed sale prices for similar items, that enables him to revise particular values in the distribution for $X$. A method [2] similar to that described in Section 5.1 is used to derive the posterior distribution — it is not detailed here. An interesting aspect of this approach is that it works equally well when the valuation space has more than one dimension.

## 6 The Seller Models the Buyer

In this Section we consider how the seller might model the buyer's contract acceptance logic in an argumentative context. As in Section 5 we focus on Equation 2 and for reasons of economy concentrate on the factor: $\mathbb{P}^t(\text{satisfy}(\alpha, \chi, \delta))$.

$\{\frac{\partial L}{\partial \lambda_j} = g_j(\boldsymbol{p}) = 0\}, j = 1, \ldots, J$ is the set of given constraints $\boldsymbol{g}$, and a solution to $\frac{\partial L}{\partial p_i} = 0, i = 1, \ldots, I$ leads eventually to $\boldsymbol{p}$. Entropy-based inference is a form of Bayesian inference that is convenient when the data is sparse [9] and encapsulates common-sense reasoning [10].

### 6.1 Modelling Contract Acceptance

Suppose that $\beta$ has found an item that he wants to buy, $\alpha$ will be interested in how much he is prepared to pay. In a similar way to Section 5.2, $\alpha$ can interpret $\beta$'s proposals as willingness to accept the offers proposed, and counter-offers as reluctance to accept the agent's prior offer — all of these interpretations being qualified with an epistemic belief probability. Entropy-based inference is then used to derive a complete probability distribution over the space of offers for a random variable that represents the partner's limit offers. This distribution is "the least biased estimate possible on the given information; i.e. it is maximally noncommittal with regard to missing information" [11]. If there are $n$-issues then the space of limit offers will be an $(n-1)$-dimensional surface through offer space. As described in [2], this method works well as long as the number of issues is not large and as long as the agent is aware of its partner's preferences along each dimension of the issue space.

### 6.2 Estimating $\beta$'s key criteria.

$\alpha$'s world model, $\mathcal{M}^t$, contains probability distributions that model the agent's belief in the world, including the state of $\beta$. In particular, for every criterion $c \in \mathcal{C}$ $\alpha$ associates a random variable $C$ with probability mass function $\mathbb{P}^t(C = e_i)$.

The distributions that relate object to criteria may be learned from prior experience. If $\mathbb{P}^t(C = e|O = o)$ is the prior distribution for criteria $C$ over an evaluation space given that the object is $o$, then given evidence from a completed negotiation with object $o$ we use the standard update procedure described in Section 5.1. For example, given evidence that $\alpha$ believes with probability $p$ that $C = e_i$ in a negotiation with object $o$ then $\mathbb{P}^{t+1}(C = e|O = o)$ is the result of applying the constraint $\mathbb{P}(C = e_i|O = o) = p$ with minimum relative entropy inference as described previously, where the result of the process is protected by Equation 5 to ensure that weak evidence does not override prior estimates.

In the absence of evidence of the form described above, the distributions, $\mathbb{P}^t(C = e|O = o)$, should gradually tend to ignorance. If a decay-limit distribution [2] is known they should tend to it otherwise they should tend to the maximum entropy distribution.

In a multiagent system, this approach can be strengthened in repeated negotiations by including the agent's identity, $\mathbb{P}^t(C = e|(O = o, Agent = \beta))$ and exploiting a similarity measure across the ontology. So if $\beta$ purchased a kitchen mixer apparently with the criterion "easy to carry" then that would increase the prior probability that $\beta$ will use the criterion "easy to carry" in negotiating for a sewing machine. Two methods for propagating estimates across the world model by exploiting the $\mathrm{Sim}(\cdot)$ measure are described in [2]. An extension of the $\mathrm{Sim}(\cdot)$ measure to sets of concepts is straightforward, we will note it as $\mathrm{Sim}{*}(\cdot)$.

### 6.3 Disposition: shaping the stance

Agent $\beta$'s *disposition* is the underlying rationale that he has for a dialogue. $\alpha$ will be concerned with the confidence in $\alpha$'s beliefs of $\beta$'s disposition as this will affect the certainty with which $\alpha$ believes she knows $\beta$'s key criteria. Gauging disposition in human discourse is not easy, but is certainly not impossible. We form expectations about what will be said next; when those expectations are challenged we may well believe that there is a shift in the rationale.

The bargaining literature consistently advises (see for example [12]) that an agent should change its *stance* (one dimension of stance being the 'nice guy' / 'tough guy' axis) to prevent other agents from decrypting their private information, and so we should expect some sort of "smoke screen" surrounding any dialogue between competitive agents. It would be convenient to think of disposition as the mirror-image of stance, but what matters is the agent's confidence in its model of the partner. The problem is to differentiate between a partner that is skilfully preventing us from decrypting their private information, and a partner that has either had a fundamental change of heart or has changed his mind in a way that will significantly influence the set of contracts that he will agree to. The first of these is normal behaviour, and the second means that the models of the partner may well be inaccurate.

If an agent believes that her partner's disposition has altered then the entropy of her model of the partner should be increased — particularly beliefs concerning the key criteria should be relaxed to prevent the dialogue attempting to enter a "blind alley", and to permit the search for common ground to proceed on broader basis. The mechanics for achieving this are simple: if an agent believes that his partner's disposition has shifted then his certainty of belief in the structure of the model of the partner is decreased.

$\alpha$'s model of $\beta$'s *disposition* is $D_C = \mathbb{P}^t(C = e|O = o)$ for *every* criterion in the ontology, where $o$ is the object of the negotiation. $\alpha$'s confidence in $\beta$'s disposition is the confidence he has in these distributions. Given a negotiation object $o$, confidence will be aggregated from $\mathbb{H}(C = e|O = o)$ for *every* criterion in the ontology. Then the idea is that if in the negotiation for a camera "for family use" $\alpha$ is asked to demonstrate how to photograph a drop of water falling from a tap then this would presumably cause a dramatic difference between $\mathbb{P}^t(C = e|(O = \textit{“family use”}))$ and $\mathbb{P}^t(C = e|(O = \textit{“family use”}, O' = \textit{“photograph water drops”}))$. This difference causes $\alpha$ to revise her belief in "family use", to revise the disposition towards distributions of higher entropy, and to approach the negotiation on a broader basis. A high-level diagram of $\alpha$'s model of $\beta$'s acceptance criteria that includes disposition is shown in Figure 1.

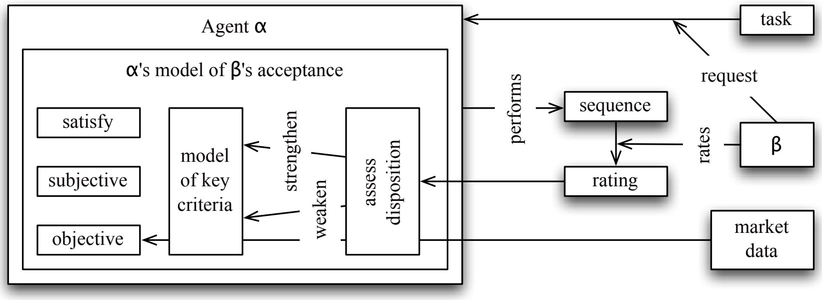
## 7 Strategies

In this section we describe the components of an argumentation strategy starting with tools for valuing information revelation that are used to model the fairness of a negotiation dialogue.

### 7.1 Information Revelation: computing counter proposals

Everything that an agent communicates gives away information. The simple offer "you may purchase this wine for €3" may be intrepretd in a utilitarian sense (e.g. the profit that you could make by purchasing it), and as information (in terms of the reduction of your entropy or uncertainty in your beliefs about my limit price for the item).

**Fig. 1.** The model of $\beta$'s acceptance criteria that lies at the heart of the argumentation strategy.



Information-based agents value information exchanged, and attempt to manage the associated costs and benefits.

*Illocutionary categories* and an *ontology* together form a framework in which the value of information exchanged can be categorised. The LOGIC framework for argumentative negotiation [4] is based on five illocutionary categories: Legitimacy of the arguments, Options i.e. deals that are acceptable, Goals i.e. motivation for the negotiation, Independence i.e: outside options, and Commitments that the agent has including its assets. In general, $\alpha$ has a set of illocutionary categories $\mathcal{Y}$ and a categorising function $\kappa : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{Y})$. The power set, $\mathcal{P}(\mathcal{Y})$, is required as some utterances belong to multiple categories. For example, in the LOGIC framework the utterance "I will not pay more for a bottle of Beaujolais than the price that John charges" is categorised as both Option (what I will accept) and Independence (what I will do if this negotiation fails).

Then two central concepts describe relationships and dialogues between a pair of agents. These are *intimacy* — degree of closeness, and *balance* — degree of fairness. In this general model, the *intimacy* of $\alpha$'s relationship with $\beta$, $A^t$, measures the amount that $\alpha$ knows about $\beta$'s private information and is represented as real numeric values over $\mathcal{G} = \mathcal{Y} \times V$.

Suppose $\alpha$ receives utterance $u$ from $\beta$ and that category $y \in \kappa(u)$. For any concept $x \in V$, define $\Delta(u, x) = \max_{x' \in concepts(u)} \text{Sim}(x', x)$. Denote the value of $A_i^t$ in position $(y, x)$ by $A^t_{(y,x)}$ then:

$$A^t_{(y,x)} = \rho \times A^{t-1}_{(y,x)} + (1 - \rho) \times \mathbb{I}(u) \times \Delta(u, x)$$

for any $x$, where $\rho$ is the discount rate, and $\mathbb{I}(u)$ is the *information*[12] in $u$. The *balance* of $\alpha$'s relationship with $\beta_i$, $B^t$, is the element by element numeric difference of $A^t$ and $\alpha$'s estimate of $\beta$'s intimacy on $\alpha$.

[12] Information is measured in the Shannon sense, if at time $t$, $\alpha$ receives an utterance $u$ that may alter this world model then the (Shannon) *information* in $u$ with respect to the distributions in $\mathcal{M}^t$ is: $\mathbb{I}(u) = \mathbb{H}(\mathcal{M}^t) - \mathbb{H}(\mathcal{M}^{t+1})$.

We are particularly interested in the concept of intimacy in so far as it estimates what $\alpha$ knows about $\beta$'s criteria, and about the certainty of $\alpha$'s estimates of the random variables $\{C_i\}$. We are interested in balance as a measure of the 'fairness' of the dialogue. If $\alpha$ shows $\beta$ how to take a perfect photograph of a duck then it is reasonable to expect some information at least in return.

Moreover, $\alpha$ acts proactively to satisfy her needs — that are organised in a hierarchy[13] of *needs*, $\varXi$, and a function $\omega : \varXi \rightarrow \mathcal{P}(W)$ where $W$ is the set of perceivable states, and $\omega(\chi)$ is the set of states that satisfy need $\chi \in \varXi$. Needs turn 'on' spontaneously, and in response to *triggers*. They turn 'off' because $\alpha$ believes they are satisfied. When a need fires, a plan is chosen to satisfy that need (we do not describe plans here). If $\alpha$ is to contemplate the future she will need some idea of her future needs — this is represented in her *needs model*: $\upsilon : T \rightarrow \times^{|\varXi|}[0,1]$ where $T$ is time, and: $\upsilon(t) = (\chi_1^t, \dots, \chi_{|\varXi|}^t)$ where $\chi_i^t = \mathbb{P}(\text{need } \chi_i \text{ fires at time } t)$.

Given the needs model, $\upsilon$, $\alpha$'s *relationship model* ($\text{Relate}(\cdot)$) determines the target *intimacy*, $A_i^{*t}$, and target *balance*, $B_i^{*t}$, for each agent $i$ in the known set of agents $Agents$. That is, $\{(A_i^{*t}, B_{*i}^t)\}_{i=1}^{|Agents|} = \text{Relate}(\upsilon, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$ where, $\boldsymbol{X}_i$ is the trust model, $\boldsymbol{Y}_i$ is the honour model and $\boldsymbol{Z}_i$ is the reliability model as described in [2]. As noted before, the values for intimacy and balance are not simple numbers but are structured sets of values over $\mathcal{Y} \times V$.

When a need fires $\alpha$ first selects an agent $\beta_i$ to negotiate with — the social model of trust, honour and reliability provide input to this decision, i.e. $\beta_i = \text{Select}(\chi, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$. We assume that in her social model, $\alpha$ has medium-term intentions for the state of the relationship that she desires with each of the available agents — these intentions are represented as the target intimacy, $A_i^{*t}$, and target balance, $B_i^{*t}$, for each agent $\beta_i$. These medium-term intentions are then distilled into short-term targets for the intimacy, $A_i^{**t}$, and balance, $B_i^{**t}$, to be achieved in the current dialogue $\varPsi^t$, i.e. $(A_i^{**t}, B_i^{**t}) = \text{Set}(\chi, A_i^{*t}, B_i^{*t})$. In particular, if the balance target, $B_i^{**t}$, is grossly exceeded by $\beta$ failing to co-operate then it becomes a trigger for $\alpha$ to terminate the negotiation.

### 7.2 Computing arguments

For an information-based agent, an incoming utterance is only of interest if it reduces the uncertainty (entropy) of the world model in some way. In information-based argumentation we are particularly interested in the effect that an argumentative utterance has in the world model including $\beta$'s disposition, and $\alpha$'s estimate of $\beta$'s assessment of current proposals in terms of its criteria.

Information-based argumentation attempts to counter the effect of the partner's arguments, in the simple negotiation protocol used here, an argumentative utterance, $u$, will either contain a justification of the proposal it accompanies, a rating and justification of one of $\alpha$ demonstration sequences, or a counter-justification of one of $\alpha$'s prior proposals or arguments. If $u$ requests $\alpha$ to perform a task then $u$ may modify $\beta$'s disposition i.e. the set of conditional estimates of the form: $\mathbb{P}^t(C = e|O = o))$. If $\beta$ rates and comments on the demonstration of a sequence then this affects $\alpha$'s estimate of $\beta$'s

[13] In the sense of the well-known Maslow hierarchy [14], where the satisfaction of needs that are lower in the hierarchy take precedence over the satisfaction of needs that are higher.

likelihood to accept a contract as described in Equation 1 (this is concerned with *how* $\beta$ will apply his criteria).

Suppose that $u$ rates and comments on the performance of a sequence then that sequence will have been demonstrated in response to a request to perform a task. Given a task, $\tau$, and a object, $s$, $\alpha$ may have estimates for $P^t(C = e|(O = o, \mathcal{T} = \tau))$ — if so then this suggests a link between the task and a set of one or more criteria $C_u$. The effect that $u$ has on $\beta$'s criteria (what ever they are) will be conveyed as the rating. In the spirit of the scenario, we assume that for every criterion and object pair $(C, o)$ $\alpha$ has a supply of positive argumentative statements $\mathcal{L}_{(C,o)}$. Suppose $\alpha$ wishes to counter the negatively rated $u$ with a positively rated $u'$. Let $\Psi_u$ be the set of all arguments exchanged between $\alpha$ and $\beta$ prior to $u$ in the dialogue. Let $M_u \subseteq \mathcal{L}_{(C,o)}$ for any $C \in C_\mu$. Let $N_u \subseteq M_u$ such that $\forall x \in N_u$ and $\forall u' \in \Psi_u$, $\text{Sim}{*}(\mathit{concepts}(x), \mathit{concepts}(u')) > \eta$ for some constant $\eta$. So $N_u$ is a set of arguments all of which (a) have a positive effect on at least one criterion associated with the negative $u$, and (b) are at 'some distance' (determined by $r$) from arguments already exchanged. Then:

$$u' = \begin{cases} \arg\min_{u' \in N_u} \text{Sim}{*}(\mathit{concepts}(u), \mathit{concepts}(u')) & \text{if } N_u \neq \emptyset \\ \arg\min_{u' \in M_u} \text{Sim}{*}(\mathit{concepts}(u), \mathit{concepts}(u')) & \text{otherwise.} \end{cases}$$

So using only 'fresh' arguments, $\alpha$ prefers to choose a counter argument to $u$ that is semantically close to $u$, and if that is not possible she chooses an argument that has some general positive effect on the criteria and may not have been used previously.

Suppose that $u$ proposes a contract. $\alpha$ will either decide to accept it or to make a counter offer. We do not describe the bargaining process here, see [2].

### 7.3 All together

If $\beta_i$ communicates $u$ then $\alpha$ responds with:

$$u' = \mathit{Argue}(u, \mathcal{M}^t, \Psi^t, A^{**t}, B^{**t}, C_u, N_u, M_u, D_u))$$

where:

- the *negotiation* mechanisms as explained in Section 7.1 sets parameters $A^{**t}, B^{**t}$) (see e.g. [4] for further details);
- the *argumentation* process determines the parameters $N_u, M_u$ needed to generate the accompanying arguments to the proposal, see Section 7.2;
- the *criteria* modeling process determines the set of criteria $C_u$ used by our opponent to assess the proposals, see Section 6.2; and,
- the *disposition* modeling sets the distributions $D_u$ used to interpret the stance of the opponent, see Section 6.3.

The personality of the agent will be determined by the particular $f$ chosen to select the answer to send. The study of concrete functions is subject of ongoing research as well as their application into a eProcurement setting.

## 8 Discussion

We have described an approach to argumentation that aims to:

- discover what the partner's key evaluative criteria are,
- model how the partner is evaluating his key criteria given some evidence,
- influence the partner's evaluation of his key criteria,
- influence the relative importance that the partner attaches to those criteria, and
- introduce new key criteria when it is strategic to do so.

The ideas described here are an attempt to develop an approach to argumentation that may be used in the interests of both parties. It aims to achieve this by unearthing the 'top layer' of the partner's reasoning apparatus and by attempting to work with it rather than against it. To this end, the utterances produced aim to influence the partner to believe what we believe to be in his best interests — although it may not be in fact. The utterances aim to convey what is so, and not to point out "where the partner is wrong". In the long term, this behaviour is intended to lead to the development of lasting relationships between agents that are underpinned both by the knowledge that their partners "treat them well" and that their partners act as they do "for the right reasons".

The ideas in this paper have been developed within a highly restrictive scenario that is deliberately asymmetric (being based on a buy / seller relationship). The structure of the analysis is far more general and applies to any scenario in which something has to be bought/made/designed that satisfies a need, and that can do various things. The agents who try to make it do things (use-cases if you like) subjectively rate what they see.

In previous work [4] we have advocated the gradual development of trust and intimacy[14] through successive argumentative exchanges as a way of building relationships between agents. The act of passing private information carries with it a sense of trust of the sender in the receiver, and having done so the sender will wish to observe that the receiver respects the information received. In this paper we have gone one step further by including a modest degree of *understanding* in the sense that an agent attempts to understand what her partner likes. This falls well short of a deep model of the partner's reasoning but we believe strikes a reasonable balance between being meaningful and being achievable. This augments the tools for building social relationships through argumentation by establishing:

- *trust* — my belief in the veracity of your *commitments*
- *intimacy* — my belief in the extent to which I know your private *information*
- *understanding* — my belief in the extent to which I know what you *like*

**Acknowledgements** This research has been supported by the Sabbatical Programme of the Catalan Government BE2007, the Australian Research Council Discovery Grant DP0557168, and by the Spanish Ministerio de Educación y Ciencia project ”Agreement Technologies” (CONSOLIDER CSD2007-0022, INGENIO 2010).

[14] The revelation of private information.

## References

1. Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI-07, Hyderabad, India (2007) 1513–1518
2. Sierra, C., Debenham, J.: Trust and honour in information-based agency. In Stone, P., Weiss, G., eds.: Proceedings Fifth International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2006, Hakodate, Japan, ACM Press, New York (2006) 1225 – 1232
3. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez, J.A., Sierra, C.: Environment engineering for multiagent systems. Journal on Engineering Applications of Artificial Intelligence **18** (2005)
4. Sierra, C., Debenham, J.: The LOGIC Negotiation Model. In: Proceedings Sixth International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2007, Honolulu, Hawai'i (2007) 1026–1033
5. Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence **77** (1995) 321–358
6. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, E.: Argumentation-based negotiation. Knowledge Engineering Review **18** (2003) 343–375
7. Kalfoglou, Y., Schorlemmer, M.: IF-Map: An ontology-mapping method based on information-flow theory. In Spaccapietra, S., March, S., Aberer, K., eds.: Journal on Data Semantics I. Volume 2800 of Lecture Notes in Computer Science. Springer-Verlag: Heidelberg, Germany (2003) 98–127
8. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering **15** (2003) 871 – 882
9. Cheeseman, P., Stutz, J.: On The Relationship between Bayesian and Maximum Entropy Inference. In: Bayesian Inference and Maximum Entropy Methods in Science and Engineering. American Institute of Physics, Melville, NY, USA (2004) 445 – 461
10. Paris, J.: Common sense and maximum entropy. Synthese **117** (1999) 75 – 93
11. Jaynes, E.: Information theory and statistical mechanics: Part I. Physical Review **106** (1957) 620 – 630
12. Lewicki, R.J., Saunders, D.M., Minton, J.W.: Essentials of Negotiation. McGraw Hill (2001)
13. Sierra, C., Debenham, J.: Information-based deliberation. In Padgham, L., Parkes, D., Müller, J., Parsons, S., eds.: Proceedings Seventh International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2008, Estoril, Portugal, ACM Press, New York (2008)
14. Maslow, A.H.: A theory of human motivation. Psychological Review **50** (1943) 370–396
15. Raiffa, H.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. Harvard U.P. (2002)

# Requirements towards automated mediation agents

Simeon Simoff[1,3],Carles Sierra[2,3] and Ramon López de Màntaras[2]

[1] School of Computing and Mathematics,
University of Western Sydney, NSW 1797, Australia s.simoff@uws.edu.au
[2] Institut d'Investigació en Intel·ligència Artificial – IIIA,
Spanish Scientific Research Council, CSIC
08193 Bellaterra, Catalonia, Spain {sierra, mantaras}@iiia.csic.es
[3] University of Technology, Sydney, Australia

**Abstract.** This paper presents a preliminary study of the relevant issues on the development of an automated mediation agent. The work is conducted within the 'curious negotiator' framework [1]. Mediation, considered from a knowledge perspective, is an information revelation process. The introduced formalism is used to demonstrate how via revealing the appropriate information and changing the understanding of the disputing parties mediation can succeed. Automating mediation needs to take into account that mediation is a knowledge intensive process, where the mediators utilise their past experiences and information from negotiating parties to change the positions of negotiating parties.

## 1 Introduction

Negotiation is the process whereby two (or more) individual agents with conflicting interests interact, aiming at reaching a mutually beneficial agreement on a set of issues. Engaging in such interactions is a daily activity —from a simple negotiation on the price of a product we buy at the market to the complicated negotiations in dispute resolutions on the international arena. Whatever is the level of negotiation, during such interactions, participants may need to make concessions in order to reach an agreement [2].

Negotiation is goal-directed in the sense that individual agents involved in a negotiation may —probably will— have agendas of their own. But the agendas of the negotiating agents may be incompatible —there may be no solution that satisfies them all. Further the existence of a solution is unlikely to be known when the negotiation commences [3]. So it may not be useful to consider negotiation as a search problem because the solution space may be empty whilst the negotiating agents may believe that it is not so. If the negotiation is a multi-issue negotiation for which the issue set is open (i.e. it can change at any stage in the negotiation) then the agendas of the individual negotiating agents must necessarily be at a higher level than the issues because the issues are unknown, and may even be issues that 'had never occurred' to one of the agents. So for multi-issue negotiation the agendas of the agents cannot in general be a high level goal such as 'to maximise profit on the deal' as the deal space is unknown. Environmental conflict resolution is a typical example where conflicts involve many different types of parties, issues and resources [4].

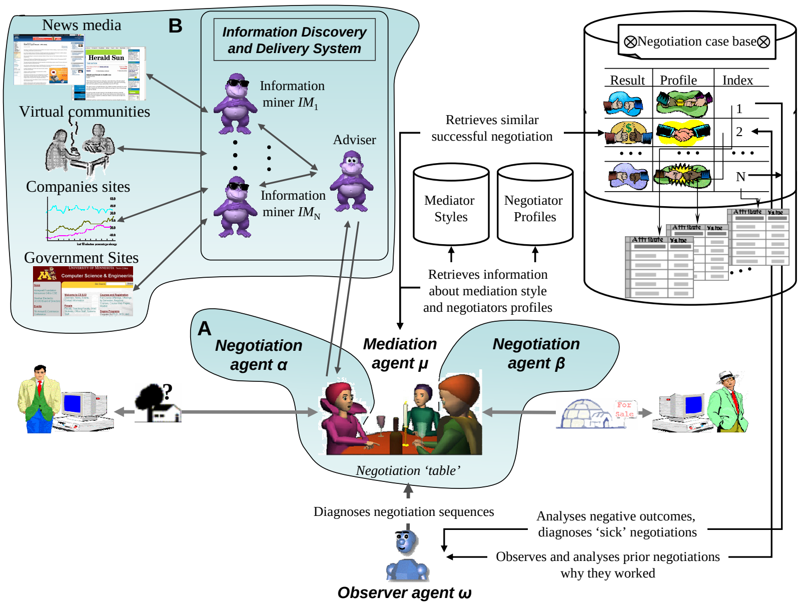


**Fig. 1.** The design of the 'curious negotiator' and the progress of the research

As a result negotiations may reach a deadlock, taking prohibitively long time without reaching tangible outcomes, or be terminated. This is when in real life the intervention of a mediator can influence the process, facilitating it towards a mutual agreement.

The design of the 'curious negotiator' automated negotiation system, outlined initially in [1], is an attempt to address these issues. Figure 1 shows an updated version of the overall design proposed in [1] and the progress of the work.

The 'curious negotiator' is founded on the intuition "it's what you know that matters" and investigates the use of information and information theory, including entropy-based (random worlds) inference, as a foundation for automated negotiation between agents with bounded rationality. The design presented in [1] aimed at exploiting the interplay between contextual information [5] and the development of offers in negotiation conducted in an electronic environment. This contextual information is derived from what happens at the bargaining table and away from it. The work on the negotiation agent (shaded area A in Figure 1) focused on identifying mechanisms and knowledge structures for utilisation information in the negotiation process. Negotiation agent $\alpha$ negotiates with agent $\beta$ by sending illocutions which represent offers and counter offers. The illocutions are represented in a communication language $\mathbb{C}$. An example of such language, where the kernel set of negotiation illocutions is extended with illocutions that enable persuasive negotiation and argumentation, is presented in [6] and [7]. Negotiation agent $\alpha$ also uses an internal language $\mathcal{L}$ for it's reasoning.

Negotiation agent $\alpha$ negotiates from a stance that assumes nothing about her opponent's motivations and applies maximum entropy logic to that which it has observed. The basic feasibility of this approach was reported in [8] where an agent for multi-issue bilateral bargaining signs contracts if it is sufficiently confident that they are acceptable. This work is orthogonal to the utility-based approach, and treats negotiation as an information discovery and revelation process. The output from the work covering the shaded area A in Figure 1 is known collectively as information-based agency [7].

The information that the agent utilises may come from at least two sources:

- from the 'negotiation table', e.g. from the counter offers that the opponent provides (this is incorporated in the work presented in [8]) (in general, all utterances agents make during a negotiation give away (valuable) information);
- from external sources, e.g. other deals, news, companies white papers, blogs of virtual communities, and other electronically accessible sources, all of which constitute part of the context in which negotiation happens.

The automation of the discovery, representation and delivery of the necessary information and knowledge to the agents has been the focus of the work on the information discovery and delivery system in the 'curious negotiator' (shaded area B in Figure 1). Elements of the approach and different technical aspects of the embedded information mining system have been presented in several works. One of its components has been presented in more detail in [9]. It includes an effective automated technique for extracting relevant articles from news web sites and their semi-structured representation, which then can be processed further by the information discovery and delivery system. In [10] it has been demonstrated how extracted unstructured or semi-structured news can be utilised to refine exchange rate predictions and provide the information to the negotiating agent. The choice of the application has been influenced by the literature indicators that daily economy news and political events influence the exchange rate daily movement [11, 12]. The mechanism includes news extraction algorithms, a quantitative process model based on the extracted news information, which is exemplified by an exchange rate prediction model, and a communication protocol between the data mining agents and the negotiation agents. The predictive information about the exchange rate can then be utilised as input to the agent's negotiation strategies. The system complements and services the information-based architecture in [8, 7]. The information request and the information delivery format is defined by the negotiation agent in the query. For example, if the topic of negotiation is buying a large number of digital cameras for an organisation, the shared ontology will include the product model of the camera, and some characteristics, like product reputation (which on their own can be a list of parameters), that are usually derived from additional sources (for example, from different opinions in a professional community of photographers or digital artists). Information requests can be formulated in a form that depends on the knowledge representation used by the agent, e.g. sets of possible values, value ranges, probability values. For example, if the negotiator is interested in high resolution cameras, and the brand is a negotiation parameter, the request for information to the information mining and delivery system can be formulated as a set consisting of camera models and corresponding requests for preference estimates. These preference estimates then can be computed from the information about these models, available in various sources, including various communities of professional experts, prosumers and consumers. Aciar et al present a recommender mechanism that utilises text mining to extract opinions about the products from consumer reviews available in electronic from and convert those opinions into a recommendation that then can be utilised by a negotiation agent [13].

The mechanisms for providing information and reasoning with such information, as well as the respective knowledge representation structures in the 'curious negotiator' framework have been developed. The mechanisms for dealing with negotiations that fail in reaching an agreement, or seemed to be leading to a failure, remain the undeveloped part of the 'curious negotiator'. It is indicated by the unshaded part in Figure 1, which includes the mediating agent $\mu$, the observer agent $\omega$ and their supporting knowledge representation structures.

The paper presents an extremely preliminary work on the principles of building an automated mediation agent within the 'curious negotiator' framework, consistent with the approach of the information based agency. It explores mediation as an information revelation process. It specifies the requirements towards the knowledge representation structures supporting mediation, including a case-based representation for storing the experience of past negotiations. Section 2 looks at mediation, as a knowledge-driven process and explores the changes that information revelation can make to the negotiation space and the outcomes of negotiation. It introduces the notion of 'mental model' of participants involved in the process and looks at mechanisms of how these models can be utilised in automated mediation. Section 3 considers some aspects in utilising past experiences and background knowledge in automated mediation. It looks also at the utilisation of information at the diagnosis stage.

## 2 Mediation as a knowledge driven process of information revelation.

Contemporary analysts in social and political sciences look at mediation as a process that enables conflict resolution. Mediators are often indispensable in the area of *dispute (or conflict) resolutions*, settling variety of disputes, spanning from conflicts between sovereign nations to conflicts between family members, friends, and colleagues. Successful mediation can make a dramatic difference to the outcome of a negotiation stalemate. For instance, on 14 January 1998 the President of United Nations Security Council issues statement demanding "that Iraq cooperate fully and immediately and without conditions with the Special Commission in accordance with the relevant resolutions."[4] As all UN weapons inspections in Iraq were frozen, during the following month all direct negotiations between the US and Iraq did not reach any agreement and the military conflict seemed unavoidable. The following event sequence illustrates the mediation process: (i) the US authorised the mediation effort; (ii) the UN secretary (the mediator) achieved a possible deal with Iraq; (iii) the UN secretary passed it back to the US; (iv) the US reviewed and accepted the deal. Several months later the conflict escalated, but this time no mediation was sought and military actions started. The mediation made a huge difference in the first dispute resolution.

[4] http://daccessdds.un.org/doc/UNDOC/GEN/N98/007/84/PDF/N9800784.pdf

### 2.1 Necessary and sufficient conditions for mediation

This example illustrates that mediation as a process involves *information revelation* and part of the mediator's strategy is guiding the process of information revelation. The following are the necessary (C1, C2) and sufficient (C3) conditions for a mediation to take place:

- *Condition C1*: Negotiating agents $\alpha$ and $\beta$ are willing to achieve a mutually beneficial agreement;
- *Condition C2*: Negotiating agents $\alpha$ and $\beta$ are seeking or will accept mediation (in the first case, the awareness about the conflict and the problem with the current state of the negotiation resides with the negotiating agents, in the second case either the mediator agent $\mu$ or, if present, the observer agent $\omega$ diagnoses the problem);
- *Condition C3*: A mediating agent $\mu$ is available (this condition is by default embedded in the 'curious negotiator' paradigm).

In the example with the 1998 Iraq crisis, in the second case condition C2 was not present. Conflicts may be a result of a contradiction of interests, as in the example with the 1998 Iraqi crisis, but can be also a result just of a different (but unknown to the disputing parties) perception of the disputed subject.

### 2.2 Mediation process within the 'curious negotiator' framework

Further we consider the following mediation process, illustrated in Figure 2, where agents $\alpha$ and $\beta$ are in a deadlock and direct exchange of offers between them has ceased. In a mediation session, $\alpha$ and $\beta$ interact with messages $m$ only with the mediating agent $\mu$.

$\mathcal{M}^t$ denotes a "mental model" at time $t$. We use the label "mental model" to denote the view (including related knowledge) of an agent about a dispute, about the views of the other parties on that dispute and the expected outcomes. This knowledge is internal to the agent. Each model is manifested to the other agents through the actions taken by the agent. The label "mental models" has been chosen to emphasise the key role of the mediator in the careful examination of the way negotiation parties have built their views on the disputed issues [14]. It is also in accordance with the view that negotiation can be conceptualised as a problem-solving enterprise in which mental models guide the behaviour of negotiating parties [15]. Further in the text we use the term mental model without quotation marks.

$\mathcal{M}^t_\alpha$ and $\mathcal{M}^t_\beta$ denote the mental models of agents $\alpha$ and $\beta$, respectively. $\mathcal{M}^t_\alpha$ is not known by $\beta$ and $\mathcal{M}^t_\beta$ is not known by $\alpha$. None of them is known by the mediating agent $\mu$. Each of these agents has its own approximations of the mental models of the other agents. $\mathcal{M}^t_{agent(party)}$ denotes the mental model that the *agent* has about another *party*. In particular, $\mathcal{M}^t_{\alpha(\beta)}$ is the mental model of $\alpha$ about $\beta$, i.e. about what $\beta$ wants out of the negotiation; respectively, $\mathcal{M}^t_{\beta(\alpha)}$ is the mental model of $\beta$ about $\alpha$, i.e. the position of $\alpha$ in the dispute. Further, $\mathcal{M}^t_{\mu(\alpha)}$ and $\mathcal{M}^t_{\mu(\beta)}$ are the mental models of the mediating agent $\mu$ about the positions of $\alpha$ and $\beta$ in the dispute, respectively. The actual formalism that expresses these models is beyond the scope of the paper.

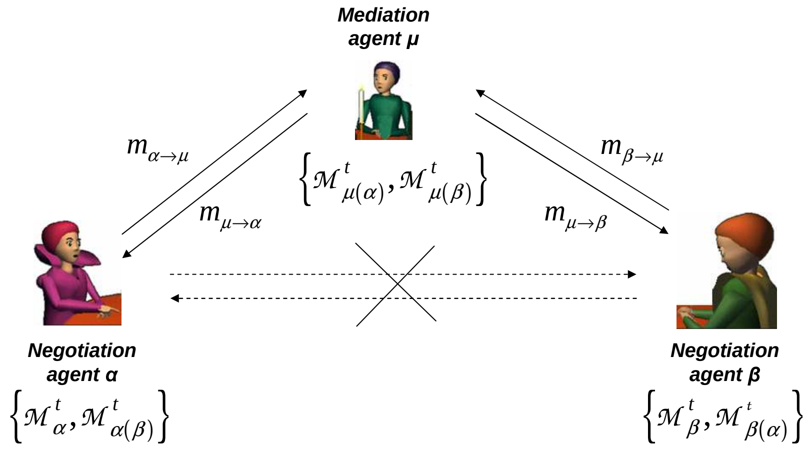

Mediation agent μ
$\left\{\mathcal{M}^t_{\mu(\alpha)}, \mathcal{M}^t_{\mu(\beta)}\right\}$
$m_{\alpha\rightarrow\mu}$
$m_{\mu\rightarrow\alpha}$
$m_{\beta\rightarrow\mu}$
$m_{\mu\rightarrow\beta}$
Negotiation agent α
$\left\{\mathcal{M}^t_{\alpha}, \mathcal{M}^t_{\alpha(\beta)}\right\}$
Negotiation agent β
$\left\{\mathcal{M}^t_{\beta}, \mathcal{M}^t_{\beta(\alpha)}\right\}$

**Fig. 2.** The mediation agent within the 'curious negotiator' framework

We use the above formalism to demonstrate some aspects of mediation that need to be taken into account when developing automated mediators. In the Orange Dispute [16], two sisters want the same orange. According to Kolodner [16] "MEDIATOR assumes they both want to eat it and solves the problem by having one sister cut the orange in two and the second chooses her half. When the second sister uses her peel for baking and throws away the pulp, MEDIATOR realises it made a mistake."[5]

Further we present the two mediation attempts in terms of the agreements reached and the information that can be passed to the mediator. Lets our agent $\alpha$ represent the first sister who wants to have the orange as a desert and agent $\beta$ represent the second sister who wants to have (only the peel of) the orange for cooking (the recipe requires the whole peel). If our mediation agent $\mu$ happens to be the case-based MEDIATOR, then the situation described in the Orange Dispute can be expressed through the mental models of the individual participants in Figure 3, making explicit the wrong assumption (the boxed expressions in Figure 3).

In these models $t_{break}$, and $t_{start}$ indicate the time when negotiation broke and when mediation started, respectively (in the case of the MEDIATOR it has been a one step act). The results of the agreements in terms of the outcomes - *Outcome (agent, issue, result)* are presented in Table 1, where result values are denoted as follows: "+", "+/-" and "-" for positive, acceptable, and negative, respectively for the corresponding agents in terms of the negotiated issue. In the original example [16], the result in the outcome for $\beta$ should be "+/-" as the second sister still used the peel from her half. Here we added the constraint of the recipe in order to get negotiation about the orange to a halt with an unacceptable "-" result and generate a request for mediation.

[5] MEDIATOR [17] is one of the early case-based mediators. The focus of the work was on the use of case-based reasoning for problem understanding, solution generation, and failure recovery. The failure recovery ability is demonstrated with the Orange Dispute in [16].

$$\alpha\ wants\ the\ orange\ as\ a\ dessert \in \mathcal{M}^{t}_{\alpha} \tag{1}$$

$$\beta\ wants\ the\ peel\ of\ the\ orange\ for\ cooking \in \mathcal{M}^{t}_{\beta} \tag{2}$$

$$\beta\ wants\ an\ orange \in \mathcal{M}^{t_{break}}_{\alpha(\beta)} \tag{3}$$

$$\alpha\ wants\ an\ orange \in \mathcal{M}^{t_{break}}_{\beta(\alpha)} \tag{4}$$

$$\boxed{\alpha\ wants\ the\ orange\ as\ a\ dessert} \in \mathcal{M}^{t_{start}}_{\mu(\alpha)} \tag{5}$$

$$\boxed{\beta\ wants\ the\ orange\ as\ a\ dessert} \in \mathcal{M}^{t_{start}}_{\mu(\beta)} \tag{6}$$

**Fig. 3.** The wrong initial assumption of the MEDIATOR [17] in terms of our mental models (Boxed expressions). This initial assumption (which didn't change as there were no mechanisms for that) caused the failure of that mediator.

The Orange Dispute can be considered an example of a dispute over resource scarcity. The resource in this case has a possible component-based separation (without change of the total amount of available resource) that allows to change the structure of the dispute through mediation, opening the space for a mutually beneficial solution. It exposes two aspects of mediation:

- The difference that a mediator can bring is in exploring the structure of the problem from a broader stance;
- An initial assumption by a mediator can lead to a failure of the mediation effort.

Consequently, we formulate the following postulates for the automated mediator:

- *Postulate P1*: An automated mediator $\mu$ should start interaction with extracting more information about the position of the parties on the negotiation;
- *Postulate P2*: An automated mediator should develop an independent "grand view" of the problem, which is more comprehensive than the individual views of $\alpha$ and $\beta$, respectively.;
- *Postulate P3*: An automated mediator $\mu$ should operate from the initial stance only of conditions C1 and C2.

Starting mediation without initial assumptions means that $\mu$ either does not have a model for each of the negotiating agents $\alpha$ and $\beta$, or accepts the models $\mathcal{M}^{t_{break}}_{\alpha(\beta)}$ and $\mathcal{M}^{t_{break}}_{\beta(\alpha)}$ these agents have about each other at the point of requesting mediation. In the case of the Orange Dispute, $\mu$ starts mediation with the exit models of $\alpha$ and $\beta$:

| **Agent** | **Agreement clauses** | **Outcome for $\alpha$** | **Outcome for $\beta$** |
|---|---|---|---|
| $\alpha$ | Cuts the orange into halves | Outcome($\alpha$, has orange, +/-) | Outcome($\beta$, has orange, -) |
| $\beta$ | chooses one half | Outcome($\alpha$, has orange, +/-) | Outcome($\beta$, has orange, -) |

**Table 1.** Outcomes of the Orange Dispute, based on mediation with initial assumption.

- $\mathcal{M}^{t_{start}}_{\mu(\alpha)} = \mathcal{M}^{t_{break}}_{\beta(\alpha)}$ , i.e. $\alpha \; wants \; an \; orange \in \mathcal{M}^{t_{start}}_{\mu(\alpha)}$, and
- $\mathcal{M}^{t_{start}}_{\mu(\beta)} = \mathcal{M}^{t_{break}}_{\alpha(\beta)}$ , i.e. $\beta \; wants \; an \; orange \in \mathcal{M}^{t_{start}}_{\mu(\beta)}$.

This information is not sufficient for mediation, e.g. the uncertainty in the mutual models of $\alpha$ and $\beta$, and the model $\mu$ has are the same. Research in conflict resolution in international relations demonstrates that if a mediator could credibly add information to the system of negotiators this alters the state of the system [18]. Consequently, $\mu$ takes steps in order to decrease this uncertainty. In addition, intuitively, it seems worth checking whether both parties have the same understanding of the issues in the dispute, i.e. technically, whether they operate with the same ontology or with compatible ontologies. In the Orange Dispute, $\mu$ obtains from each party what the orange is needed for. The Orange Dispute in terms of the mental models of the individual participants in the case of proposed mediation agent is presented in Figure 4. In these models $t_{break}$, $t_{start}$ and $t_{end}$ indicate the time when negotiation broke and when mediation started and ended, respectively. Note the difference of $\mathcal{M}^{t_{start}}_{\mu(\cdot)}$ for both $\alpha$ and $\beta$ in Figure 3 and Figure 4. The steps taken by the mediating agent are described in Figure 5 (we do not use a formal illocution based language, but the actions that the language should cater for are shown in italic).

$$\begin{align}
\alpha \; wants \; the \; orange \; as \; a \; dessert &\in \mathcal{M}^{t}_{\alpha} \tag{7}\\
\beta \; wants \; the \; peel \; of \; the \; orange \; for \; cooking &\in \mathcal{M}^{t}_{\beta} \tag{8}\\
\beta \; wants \; an \; orange &\in \mathcal{M}^{t_{break}}_{\alpha(\beta)} \tag{9}\\
\alpha \; wants \; an \; orange &\in \mathcal{M}^{t_{break}}_{\beta(\alpha)} \tag{10}\\
\boxed{\alpha \; wants \; an \; orange} &\in \mathcal{M}^{t_{start}}_{\mu(\alpha)} \tag{11}\\
\boxed{\beta \; wants \; an \; orange} &\in \mathcal{M}^{t_{start}}_{\mu(\beta)} \tag{12}\\
\boxed{\alpha \; wants \; the \; orange \; as \; a \; dessert} &\in \mathcal{M}^{t_{end}}_{\mu(\alpha)} \tag{13}\\
\boxed{\beta \; wants \; the \; peel \; of \; the \; orange \; for \; cooking} &\in \mathcal{M}^{t_{end}}_{\mu(\beta)} \tag{14}
\end{align}$$

**Fig. 4.** The respective mental models of $\alpha$, $\beta$ and $\mu$ in the mediation session of the Orange Dispute with our proposed agent.

The Orange Dispute illustrates also another important ability that an automated mediator should posses —the ability to refocus or restructure the dispute, based on the additional information about the models of each party. The outcomes of the restructured Orange Dispute are shown in Table 2.

The ability to restructure the problem is crucial for developing successful automated mediators. The Sinai Peninsula Dispute in the area of international relations shows similar properties to the Orange Dispute. The Sinai Peninsula is a piece of land of about 62,000 square km that separates Israel and Egypt. With its landscape Sinai has a *military value* for either side in terms of mechanised infantry transport or as a shelter for

1. $\mu$ : *ask* $\alpha$ to *send* its ontology of the negotiated item (orange).
2. $\mu$ : *ask* $\beta$ to *send* its ontology of the negotiated item (orange).
3. $\mu$ : *compare* ontologies received from$\alpha$ and $\beta$.
4. $\mu$ : *send* $\alpha$ and $\beta$ agreed ontology (orange as a fruit which has pulp and peel).
5. $\mu$ : *ask* $\alpha$ to *send* $\mu$ its preferences on the negotiated item in terms of agreed ontology.
6. $\mu$ : *ask* $\beta$ to *send* $\mu$ its preferences on the negotiated item in terms of agreed ontology.
7. $\mu$ : *advises* $\alpha$ and $\beta$ on $\mathcal{M}_{\alpha}^{t}$ and $\mathcal{M}_{\beta}^{t}$ based on their preferences
8. $\mu$ : *checks* the case base for past cases (resource disputes)
9. $\mu$ : *retrieves* resource disputes with divisible components
10. $\mu$ : *sends* $\alpha$ and $\beta$ action separate resource (peel the orange)
11. $\mu$ : *tells* $\alpha$ and $\beta$ to complete negotiation.
12. $\mu$ : mediation completed.

**Fig. 5.** Mediation as information revelation aiming at decreasing uncertainty within the negotiation system

guerrilla forces. The perceived importance of the territory is evidenced by the fact that Israelis and Egyptians fought in or over the Sinai Peninsula in 1948, 1956, 1967, 1968-1970, and 1973. Since 1967 Sinai had been occupied by Israel. Figure 6 shows a very simplified version of the models of the parties at the initial meeting in Jerusalem, when the negotiations started and halted and the change of the mediators models that lead to the outcomes. For the purpose of this paper we aim to emphasise the high level analogy with the Orange Dispute case (see Figure 4), i.e. the need for a mediator to reframe the problem. In fact, the need for restructuring the problem in order for a mediator to get a bigger picture has been recognised in PERSUADER [2], to resolve labor-management disputes. In recent works [19] the mediator is expected to have a complete knowledge of the solution space.

Following the initial interaction in Jerusalem, the US President Jimmy Carter initiated a *third-party mediation* effort that culminated in the Camp David accords. For the purposes of this paper we consider a simplified version of the second agreement of the Camp David accords on the future of the Sinai Peninsula. The items in the agreement are presented in Table 3, in a structure, similar to the presentation of the agreements in the Orange Dispute in Table 1 and Table 2. Without getting into the details of the mediation steps, from Table 3 it is evidenced that the initial mutually perceived models and about the need for territory and strategic military advantage have been transformed by the mediation into a Security/Sovereignty trade-off, with economic benefits.

The analogy with the Orange Dispute is in having the initial negotiation framed around a common resource Territory and a similar issue of having strategic military ad-

| **Agent** | **Agreement clauses** | **Outcome for $\alpha$** | **Outcome for $\beta$** |
|---|---|---|---|
| $\alpha$ | Peels the orange | Outcome($\alpha$, eat, +) | Outcome($\beta$, cook, +) |
| $\beta$ | Gets the whole peel | Outcome($\alpha$, eat, +) | Outcome($\beta$, cook, +) |

**Table 2.** Outcomes of the restructured Orange Dispute.

$$\begin{aligned}
\alpha\ \mathit{wants\ security, support\ for\ economy, recognition} &\in \mathcal{M}^{t}_{\alpha}\\
\beta\ \mathit{wants\ sovereignity\ (restored\ territory), support\ for\ economy, security} &\in \mathcal{M}^{t}_{\beta}\\
\beta\ \mathit{wants\ territory\ and\ strategic\ military\ advantage} &\in \mathcal{M}^{t_{break}}_{\alpha(\beta)}\\
\alpha\ \mathit{wants\ territory\ and\ strategic\ military\ advantage} &\in \mathcal{M}^{t_{break}}_{\beta(\alpha)}\\
\boxed{\alpha\ \mathit{wants\ territory\ and\ strategic\ military\ advantage}} &\in \mathcal{M}^{t_{start}}_{\mu(\alpha)}\\
\boxed{\beta\ \mathit{wants\ territory\ and\ strategic\ military\ advantage}} &\in \mathcal{M}^{t_{start}}_{\mu(\beta)}\\
\boxed{\alpha\ \mathit{wants\ security, support\ for\ economy, recognition}} &\in \mathcal{M}^{t_{end}}_{\mu(\alpha)}\\
\boxed{\beta\ \mathit{wants\ sovereignity\ (restored\ territory), support\ for\ economy, security}} &\in \mathcal{M}^{t_{end}}_{\mu(\beta)}
\end{aligned}$$

**Fig. 6.** The respective mental models of $\alpha$, $\beta$ and $\mu$ in the mediation session of the Sinai Dispute with our proposed agent

vantage as the main goals that can enable the security. Though both territorial and military components remain on the negotiation table, the mediator brought a background knowledge which changed the models of the parties: security and restoration may not necessarily be achieved with occupation of a territory or with expensive military presence.

The information injected by the mediator and proposed steps leads to decreasing the differences between perceived mental models $\mathcal{M}^{t}_{\alpha(\beta)}$ and $\mathcal{M}^{t}_{\beta(\alpha)}$, and the respective actual mental models $\mathcal{M}^{t}_{\beta}$ and $\mathcal{M}^{t}_{\alpha}$ of agents $\alpha$ and $\beta$, respectively, i.e. the intervention of the mediator decreases the uncertainty in the negotiation system.

### 2.3 Operating with information in mediation

As the mediator utilises information to decrease the uncertainty in the dispute, an automated mediation would require a measure of uncertainty $\mathbb{H}(\mathcal{M}^{t})$, allowing to quantify and compare the uncertainty coming from the incomplete knowledge of the mental models of the agents. In terms of the two party mediation in Figure 2, this decrease of uncertainty in mental models should be observable, i.e. $\mathbb{H}(\mathcal{M}^{t}_{\mu(\alpha)}) < \mathbb{H}(\mathcal{M}^{t}_{\beta(\alpha)})$ and $\mathbb{H}(\mathcal{M}^{t}_{\mu(\beta)}) < \mathbb{H}(\mathcal{M}^{t}_{\alpha(\beta)})$ . Within the framework of the information-based agency, which follows an information-theoretic approach, such measure should measure the information gain, as the mediator adds such gain. Viewing mediation as a dialogue system, e.g. in Figure 2, e.g. the mediator is engaged in a dialogue with each party, points also to the information-theoretic work in dialogue management strategies in conversational case-based reasoning [20]. In terms of an automated mediation system, the mediator should have the mechanism to determine the most informative question to ask at each stage of the interaction to each of the negotiating agents.

| **Agent** | **Agreement clauses** | **Outcome for $\alpha$** | **Outcome for $\beta$** |
|---|---|---|---|
| $\alpha$ | withdraw its armed forces from the Sinai | Outcome($\alpha$, Military, -) | Outcome($\beta$, Territory, +)<br>Outcome($\beta$, Sovereignty, +) |
| $\alpha$ | Evacuate its 4500 civilians | Outcome($\alpha$, Territory, -) | Outcome($\beta$, Territory, +)<br>Outcome($\beta$, Sovereignty, +) |
| $\alpha$ | Restory Sinai to Egypt | Outcome($\alpha$, Territory, -) | Outcome($\beta$, Territory, +)<br>Outcome($\beta$, Sovereignty, +) |
| $\alpha$ | Limit its forces within 3km from Egyptian Border | Outcome($\alpha$, Military, -)<br>Outcome($\alpha$, Security, +) | Outcome($\beta$, Security, +) |
| $\alpha$ | Lost the Abu-Rudeis oil fields in Western Sinai | Outcome($\alpha$, Economy, -) | Outcome($\beta$, Economy, +) |
| $\beta$ | Normal diplomatic relations with Israel | Outcome($\alpha$, Recognition, +) | Outcome($\beta$, Security, +) |
| $\beta$ | Freedom of passage through Suez Canal | Outcome($\alpha$, Economy, +)<br>Outcome($\alpha$, Security, +) | Outcome($\beta$, Security, +) |
| $\beta$ | Freedom of passage through nearby waters | Outcome($\alpha$, Economy, +)<br>Outcome($\alpha$, Security, +) | Outcome($\beta$, Economy, +)<br>Outcome($\beta$, Security, +) |
| $\beta$ | Restricted Egyptian forces in Sinai | Outcome($\alpha$, Security, +) | Outcome($\beta$, Military, -)<br>Outcome($\beta$, Security, +) |

**Table 3.** The Sinai Peninsula Dispute. $\alpha$ denotes Israel; $\beta$ denotes Egypt.

## 3 Utilising past experiences and background knowledge in automated mediation

The American Bar Association defines mediation as a process by which those who have a dispute, misunderstanding or conflict come together and, with *the assistance of a trained neutral mediator*, resolve the issues and problems in a way that meets the needs and interests of both parties.[6] This definition emphasises the key role of the past experience of the mediator and its unbiased nature. Further, we consider these two aspects, starting with mediator bias.

### 3.1 Unbiased mediator

The *bias of a mediator* is defined as the presence of a preference towards one of the

- outcomes in the negotiation; or,
- sides involved in the negotiation.

Not having preference towards any of the outcomes of a negotiation means also to keep open all options. For instance, the peace-loving broker's bias towards peaceful solutions makes his or her claims less believable compared to a broker who is indifferent to war or peace [18]. Such bias as a result can decrease the effectiveness of the mediation effort. Protecting automated mediation from introduction of a bias is not seen as a problem.

[6] http://www.abanet.org/cpr/clientpro/medpreface.html

### 3.2 Utilising past experiences

Experience is, perhaps, the distinct feature between successful and less successful mediators. Case-based reasoning (CBR) is an approach to problem solving that emphasizes the role of prior experience during future problem solving (i.e., new problems are solved by reusing and if necessary adapting the solutions to similar problems that were solved in the past) (see [21] for a recent review of the state-of-the-art in the CBR field). From a machine learning point of view, updating the case base is a lazy learning approach (i.e. learning without generalisation). Some aspects of using the past experience by the tandem Mediation and Observation agents have been discussed in [1]. In terms of required case representation, a starting point is the knowledge representation structure for representing negotiation cases, proposed in [22]. This structure needs to be updated for dealing with ontologies. For the mediation, the case based will be linked to the corresponding knowledge base of the mediation strategies used. The case structure now includes a negotiation case as its problem section and the collection of mediation steps, information used and other knowledge, as the solution part of the case.

Important from a computational perspective is the diagnosis stage of the mediation process [23]. The diagnostic function consists of monitoring the progress of negotiation or related interactions intended to settle or resolve disputed issues (Druckman and coauthors [23] refer to [24]). Monitoring provides a long view of unfolding developments, including trends in escalating and de-escalating dynamics. Within the framework of 'curious negotiator' we consider this stage as a pre-mediation stage, which is executed by the observer agent $\omega$. To some extent it resembles similarity with OLAP[7] —the pre-data mining steps in business intelligence, where summary statistics at different levels are generated and later provide guidance to the data mining strategies. Similar to OLAP, monitoring should be able to provide snapshots of the negotiation process at any moment of time at different levels of granularity. The mediator $\mu$ should be able to estimate the difference between $\mathcal{M}^t_{\alpha(\beta)}$ and $\mathcal{M}^t_{\beta(\alpha)}$from the respective actual mental models $\mathcal{M}^t_\beta$ and $\mathcal{M}^t_\alpha$ in order to define the intervention time of mediating interventions (if we follow a proactive approach and intervene before negotiation fails).

## 4 Conclusions

Though there has been some interest in automated mediation [17, 2, **?**,19] during the years, the field requires a significant effort in research and development. This paper has presented an early work on the principles of building an automated mediation agent. The mediation agent is part of the 'curious negotiator' framework, hence can utilise some of the machinery developed for it, in particular:

- the *information-based agency* [23, 7], which offers mechanisms that allow the interplay between argumentation and information;
- the *information-mining system*, [9, 10], which offer means for automated discovery and (to some extent) delivery of that information to negotiating agents;

[7] Online analytical processing.

- the *electronic/virtual institutions environment* [25, 26], which offers means not only for performing negotiation, but also for collecting the necessary data about the negotiation sessions in order to use it in mediation.

Mediation is an information revelation process. The Orange and Sinai disputes demonstrate how through the revealing of the appropriate information and changing the understanding of the disputes mediation can succeed. Computationally, the approach requires the specification of the introduced mental models of the agents and the definition of a measure of the difference between what is labelled as mental models. The aim of the mediation is to decrease the difference between what is a perceived model and the actual model. One possible way is by identifying alternative formulations of the dispute, demonstrated with the Orange dispute and Sinai dispute (the simplified version of the second agreement).

Case-based reasoning offers a potential mechanism for the mediator for handling past experiences, though the structure of the case will be complex (in comparison to the usually assumed attribute-value structure), extending the already complex structure for representing negotiation cases [22]. Overall, from a knowledge perspective, automating mediation needs to take in account that mediation is:

- a knowledge intensive process, where the mediators utilise their past experiences;
- a process that utilises information from negotiating parties and uses information for changing the positions these parties have on the negotiation table.

As it may deal with confidential information, mediation requires trust in the mediator from the parties involved, as much of the information about their position negotiating parties would not reveal to the other side. Though this has been beyond the scope of the paper, we are aware of this issue.

In conclusion, we would like to stress that the importance of mediation has been recognised world-wide. It's interesting to note that nowadays mediation skills are taught to students at various levels and schools spanning from elementary schools to university schools, including the Harvard Law School. Hence, the development of an automated mediation system is on the top priority of the research agendas.

**Acknowledgements** This research has been supported by the Sabbatical Progrmme of the Spanish Ministerio de Educación y Ciencia Grant SAB2006-0001, the Australian Research Council Discovery Grant DP0557168, the Generalitat de Catalunya Grant 2005/SGR/00093 and by the Spanish Ministerio de Educacion y Ciencia Consolider Grant CSD2007-0022.

## References

1. Simoff, S.J., Debenham, J.: Curious negotiator. In Klusch, M., Ossowski, S., Shehory, O., eds.: Proceedings of the Int. Conference on Cooperative Information Agents, CIA-2002, Springer, Heidelberg (2002)
2. Sycara, K.P.: Problem restructuring in negotiation. Problem restructuring in negotiation **37**(10) (1991) 1248–1268

3. Lewicki, R.J., Saunders, D.M., Minton, J.W.: Essentials of Negotiation. McGraw Hill (2001)
4. Franklin Dukes, E.: What we know about environmental conflict resolution: An analysis based on research. Conflict Resolution Quarterly **22**(1-2) (2004) 191–220
5. Gomes, A., Jehiel, P.: Dynamic process of social and economic interactions: on the persistence of inefficiencies. Centre for Economic Policy Research, CEPR, London (2001)
6. Ramchurn, S.D., Sierra, C., Godo, L., Jennings, N.R.: Negotiating using rewards. Artificial Intelligence **171** (2007) 805–837
7. Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI-07, Hyderabad, India (2007) 1513–1518
8. Debenham, J.: Bargaining with information. In Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M., eds.: Proceedings Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2004, ACM Press, New York (2004) 664–671
9. Zhang, D., Simoff, S.: Informing the curious negotiator: Automatic news extraction from the internet. In Simoff, S., Williams, G., eds.: Proceedings 3rd Australasian Data Mining Conference, 6 - 7th December, Cairns, Australia (2004) 55–72
10. Zhang, D., Simoff, S., Debenham, J.: Exchange rate modelling for e-negotiators using text mining techniques. In: E-Service Intelligence - Methodologies, Technologies and Applications. Springer, Heidelberg (2007) 191–211
11. Ehrmann, M., Fratzscher, M.: Exchange rates and fundamentals: new evidence from real-time data. Journal of International Money and Finance **24** (2005) 317–341
12. Prast, H.M., de Vor, M.P.H.: Investor reactions to news: a cognitive dissonance analysis of the euro-dollar exchange rate. European Journal of Political Economy **21** (2005) 115 – 141
13. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed recommender: Basing recommendations on consumer product reviews. IEEE Intelligent Systems **May/June** (May/June 2007) 39–47
14. Gentner, D., Stevens, A.L., eds.: Mental Models. Erlbaum, Hillsdale, NJ (1983)
15. Van Boven, L., Thompson, L.: A look into the mind of the negotiator: Mental models in negotiation. Group Processes & Intergroup Relations **6**(4) (2003) 387–404
16. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann Publishers, Inc., San Mateo, CA (1993)
17. Kolodner, J.L., Simpson, R.L.: The mediator: Analysis of an early case-based problem solver. Cognitive Science **13**(4) (1989) 507–549
18. Smith, A., Stam, A.: Mediation and peacekeeping in a random walk model of civil and interstate war. International Studies Review **5**(4) (2003) 115–135
19. Chalamish, M., Kraus, S.: Automed - an automated mediator for bilateral negotiations under time constraints. In: Proceedings of the International Conference on Autonomous Agents and Multi Agent Systems, AAMAS07, Honolulu, Hawaii, USA, IFAAMAS (2007)
20. Branting, K., Lester, J., Mott, B.: Dialogue management for conversational case-based reasoning. In: Proceedings of ECCBR04. (2004) 77–90
21. De Mantaras, R.L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review **20**(3) (2005) 215–240
22. Matos, N., Sierra, C.: Evolutionary computing and negotiating agents. In: Agent Megiated Electronic Commerce. Springer, Heidelberg (1999) 126–150
23. Druckman, D., Druckman, J.N., Arai, T.: e-mediation: Evaluating the impacts of an electronic mediator on negotiating behavior. Group Decision and Negotiation **13** (2004) 481–511
24. Zartman, I.W., Berman, M.R.: The Practical Negotiator. Yale University Press, New Haven, CT (1982)

25. Esteva, M.: Electronic Institutions: From specification to development. Phd thesis, Technical University of Catalonia, Barcelona (2003)
26. Bogdanovych, A.: Virtual Institutions. Phd thesis, Faculty of Information Technology, University of Technology, Sydney, Sydney (2007)