Verifying Existence of Resource-Bounded Coalition Uniform Strategies

Natasha Alechina
University of Nottingham
nza@cs.nott.ac.uk

Mehdi Dastani
University of Utrecht
m.m.dastani@uu.nl

Brian Logan
University of Nottingham
bsl@cs.nott.ac.uk

Abstract

We consider the problem of whether a coalition of agents has a knowledge-based strategy to ensure some outcome under a resource bound. We extend previous work on verification of multi-agent systems where actions of agents produce and consume resources, by adding epistemic pre- and postconditions to actions. This allows us to model scenarios where agents perform both actions which change the world, and actions which change their knowledge about the world, such as observation and communication. To avoid logical omniscience and obtain a compact model of the system, our model of agents' knowledge is syntactic. We define a class of coalition-uniform strategies with respect to any (decidable) notion of coalition knowledge. We show that the model-checking problem for the resulting logic is decidable for any notion of coalition-uniform strategies in these classes.

1 Introduction

We propose a new logical formalism, RB±ATSEL, for modelling and verifying multi-agent systems where agents execute both ontic actions (actions that change the world) and epistemic actions (actions that change their knowledge). This is a common situation in many multi-agent systems where agents have to explore and change their environment; for example, knowledge-based planning, diagnosis, etc. As an example, we focus on multi-agent systems where some agents *monitor* the behaviour of other agents to detect *norm violations* [Álvarez-Napagao *et al.*, 2011]. We would like to be able to automatically verify properties of such systems using model-checking; for example, to check whether monitoring agents have a strategy to detect all norm violations.

There has been considerable work on Alternating Time Temporal Logic (ATL) extended with epistemic operators and on the model-checking problem for the resulting logics, e.g., [van der Hoek and Wooldridge, 2002; Lomuscio *et al.*, 2009]. The motivation of this paper is closer to the work on Dynamic Epistemic Logic (DEL) e.g., [Baltag *et al.*, 1998; van Ditmarsch and Kooi, 2008], and epistemic planning, e.g., [Andersen *et al.*, 2012], where we can reason about how epistemic actions change the agents' epistemic states, which is impossible in epistemic ATL.

Our approach differs from previous work in two main respects: the adoption of syntactic knowledge, and considering costs of both ontic and epistemic actions. We interpret epistemic modalities *syntactically* rather than using an indistinguishability relation. This allows us to use simpler models, and to model different (non-omniscient) reasoning procedures for different agents. We also consider the costs of both ontic and epistemic actions, such as observation and communication. Clearly ontic actions (e.g., moving from one location to another) have costs (e.g., energy). However, observations often have non-trivial costs (e.g., an agent may need to use costly equipment, or pay some authority for verified information [Jamroga and Tabatabaei, 2013; Naumov and Tao, 2015]). Exchanging messages also has costs, for example, energy, or money. This is particularly relevant for norm monitoring scenarios: a successful monitoring strategy may exist, but could be prohibitively expensive and not practically feasible. For this reason, we chose as the basis for our formalism the logic RB±ATL, where actions produce and consume resources [Alechina *et al.*, 2014]. (If observations have a cost, we need to model resource production if monitoring is to be performed indefinitely.) Using RB±ATL allows us to check whether a strategy *that requires less than a given amount of resources* exists. However Alechina et al. [2014] consider only the resource consumption of ontic actions.

The notion of strategies we consider are *perfect recall* strategies, where the choice of the next action by an agent depends on all previously encountered states. Perfect recall strategies make more sense than *memoryless* strategies in our setting, as actions both produce and consume resources. (Intuitively, this is because an agent may need to 'loop' several times making some resource in order to execute an action that consumes the resource, e.g., recharging a battery for several timesteps.) In addition, strategies should also be *uniform*; that is, if an agent has the same knowledge at each point in two histories, then it should chose the same action in both of them. However, model-checking epistemic ATL with uniform perfect recall strategies and more than one agent is undecidable [Dima and Tiplea, 2011]. This result does not change for syntactic epistemics. We therefore propose a notion of *coalition-uniform* strategies for which the model-checking problem is decidable. A strategy is coalition-uniform for a coalition A if for any two histories indistinguishable for A (wrt some notion of indistinguishability), it chooses the same action. We call the resulting logic Resource-Bounded Alternating Time Syntactic Epistemic Logic (RB±ATSEL). The main contribution of this paper is a decidable model-checking procedure for RB±ATSEL with coalition-uniform strategies (wrt any decidable notion of indistinguishability).

2 Syntax and Semantics of RB±ATSEL

We adopt the approach to epistemic logic that interprets agents' knowledge syntactically, as a (finite) set of formulas, as in, e.g., [Konolige, 1986]. An agent knows that ϕ if, and only if, ϕ is in its knowledge base or is derivable from it by some simple terminating procedure (e.g., closure under modus ponens). This approach is very close to the notion of algorithmic knowledge of Fagin et al. [1995]. In what follows, to decide whether the agent knows ϕ, we simply check whether a formula ϕ is in agent i's state s_i, but this can be trivially replaced with a check for $alg(s_i, \phi) = true$, where alg is a terminating procedure that takes a set of formulas s_i and a formula ϕ and checks whether ϕ follows from s_i.

Syntactic knowledge provides a convenient and compact way of modelling *knowledge change*, compared to, for example, DEL. In DEL, the update mechanism involves combining models to produce new models, and requires considerably more space to represent and more computation to reason about. In the syntactic approach, we can simply specify post-conditions of actions which add and remove formulas from the agent's state. In DEL, we need to essentially associate an automaton with each action that can transform an epistemic model into a new epistemic model. Finally, it is worth noting that many epistemic planners use what are essentially syntactic knowledge bases (and as a result solve a decidable planning problem), e.g., [Petrick and Bacchus, 2004]. This contrasts with the undecidability of DEL-based epistemic planning [Aucher and Bolander, 2013].

The language of RB±ATSEL is built from the following components: $Agt = \{a_1, \ldots, a_n\}$ a set of n agents; $Res = \{res_1, \ldots, res_r\}$ a set of r resources; and Π a set of propositions. $B = Agt \times Res \rightarrow \mathbb{N}_\infty$ is a set of resource bounds, where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. (Note that the definition of bound and related definitions differ from those in [Alechina *et al.*, 2014] as we assume resources can't be transferred between agents.)

Formulas of the language $\mathcal{L}$ of RB±ATSEL are defined by the following syntax $\varphi, \psi ::=$

$$p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\langle A^b \rangle\rangle \bigcirc \varphi \mid \langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi \mid \langle\langle A^b \rangle\rangle \Box \varphi \mid K_a \varphi$$

where $p \in \Pi$ is a proposition, $A \subseteq Agt$, $b \in B$ is a resource bound and $a \in Agt$.

The meaning of RB±ATSEL formulas is as follows: $\langle\langle A^b \rangle\rangle \bigcirc \varphi$ means that a coalition A has a strategy executable within resource bound b to ensure that the next state satisfies φ; $\langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$ means that A has a strategy executable within resource bound b to ensure ψ while maintaining the truth of φ; $\langle\langle A^b \rangle\rangle \Box \varphi$ means that A has a strategy executable within resource bound b to ensure that φ is always true; and $K_a \varphi$ means that formula φ is in agent a's knowledge base.

Definition 1. *A model of RB±ATSEL is a structure* $M = (\Phi, Agt, Res, S, \Pi, Act, d, c, \delta)$ *where:*

- Φ *is a finite set of formulas of* $\mathcal{L}$.
- S *is a set of tuples* $(s_1, \ldots, s_n, s_e)$ *where* $s_e \subseteq \Pi$ *and for each* $a \in Agt$, $s_a \subseteq \Phi$.
- Agt *is a non-empty set of* n *agents,* Res *is a non-empty set of* r *resources.*
- Π *is a finite set of propositional variables;* $p \in \Pi$ *is true in* $s \in S$ *iff* $p \in s_e$.
- Act *is a non-empty set of actions which includes* $idle$, *and* $d : S \times Agt \rightarrow \wp(Act) \setminus \{\emptyset\}$ *is a function which assigns to each* $s \in S$ *a non-empty set of actions available to each agent* $a \in Agt$. *We assume that for every* $s \in S$ *and* $a \in Agt$, $idle \in d(s, a)$. *We denote joint actions by all agents in* Agt *available at* s *by* $D(s) = d(s, a_1) \times \cdots \times d(s, a_n)$.
- *for every* $s, s' \in S, a \in Agt$, $d(s, a) = d(s', a)$ *if* $s_a = s'_a$.
- $c : Act \times Res \rightarrow \mathbb{Z}$ *is the function which models consumption and production of resources by actions (a positive integer means consumption, a negative one production). Let* $cons_{res}(\alpha) = max(0, c(\alpha, res))$ *and* $prod_{res}(\alpha) = -min(0, c(\alpha, res))$. *We stipulate that* $c(idle, res) = 0$ *for all* $res \in Res$.
- $\delta : S \times Act^n \rightarrow S$ *is a partial function which for every* $s \in S$ *and joint action* $\sigma \in D(s)$ *returns the state resulting from executing* σ *in* s.

We denote by $D_A(s)$ the set of all joint actions by agents in coalition A at s. Let σ be a joint action by agents in A. The set of outcomes of this joint action in s is the set of states reached when A executes σ: $out(s, \sigma) = \{s' \in S \mid \exists \sigma' \in D(s) : \sigma = \sigma'_A \wedge s' = \delta(s, \sigma')\}$ (where s'_A is the restriction of σ' to A). A *strategy for a coalition* $A \subseteq Agt$ is a mapping $F_A : S^+ \rightarrow Act^{|A|}$ (from finite non-empty sequences of states to joint actions by A) such that, for every $\lambda s \in S^+$, $F_A(\lambda s) \in D_A(s)$. A computation $\lambda \in S^\omega$ is consistent with a strategy F_A iff, for all $i \geq 0$, $\lambda[i+1] \in out(\lambda[i], F_A(\lambda[0, i]))$. Overloading notation, we denote the set of all computations λ consistent with F_A that start from s by $out(s, F_A)$. Given a bound $b \in B$, a computation $\lambda \in out(s, F_A)$ is b-consistent with F_A iff, for every $i \geq 0$, for every $a \in A$,

$$\sum_{j=0}^{j=i-1} tot(F_a(\lambda[0, j])) + b_a \geq cons(F_a(\lambda[0, i]))$$

where $F_a(\lambda[0, j])$ is a's action as part of the joint action returned by F_A for the sequence of states $\lambda[0, j]$; $tot(\sigma) = prod(\sigma) - cons(\sigma)$ is the (vector) difference between the vector $prod(\sigma) = (prod_1(\sigma), \ldots, prod_r(\sigma))$ of resource amounts action σ produces and the vector of resource amounts $cons(\sigma)$ it consumes; b_a is $a's$ resource bound in b. This condition requires that the amount of resources a accumulated on the path so far, plus the original bound, is greater than or equal to the cost of executing the next action by a in the strategy. F_A is a b-strategy if all $\lambda \in out(s, F_A)$ are b-consistent.

In the presence of imperfect information, it makes sense to consider only *uniform* strategies rather than arbitrary ones.

A strategy is uniform if after epistemically indistinguishable histories, agents select the same actions. Two states s and t are epistemically indistinguishable by agent a, denoted by $s \sim_a t$, if a has the same local state (knows the same formulas) in s and t: $s \sim_a t$ iff $s_a = t_a$. For a coalition A, indistinguishability $s \sim_A s'$ means that A as a whole has the same knowledge in the two states. Various notions of coalitional knowledge can be used to define $\sim_A$. For example, $s \sim_A t$ iff $\bigcup_{a \in A} s_a = \bigcup_{a \in A} t_a$ (the distributed knowledge of A in s and t is the same). Another possible definition of $s \sim_A t$ is $\forall a \in A (s_a = t_a)$. $\sim_A$ can be lifted to histories in the obvious way: $s_1, \ldots, s_k \sim_A t_1, \ldots, t_k$ iff for all $j \in [1, k]$, $s_j \sim_A t_j$.

Definition 2. *A strategy F_A for A is coalition-uniform with respect to $\sim_A$ if for all $\bar{s} \sim_A \bar{t}$, $F_A(\bar{s}) = F_A(\bar{t})$.*

Note that any notion of action choice based on coalition knowledge presupposes that agents in the coalition share knowledge for the purpose of action selection. In other words, there is a 'silent step' before action selection when agents in the coalition can communicate with each other instantaneously and without any cost. The only explicit and potentially resource consuming communication actions which may be necessary for a successful strategy are actions communicating with agents outside of the coalition.

The truth definition for RB±ATSEL with coalition-uniform strategies (parameterised by $\sim_A$) is as follows:

- $M, s \models p$ iff $p \in s_e$
- boolean connectives have standard truth definitions
- $M, s \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff $\exists$ coalition-uniform b-strategy F_A such that for all $\lambda \in out(s, F_A)$: $M, \lambda[1] \models \phi$
- $M, s \models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ iff $\exists$ coalition-uniform b-strategy F_A such that for all $\lambda \in out(s, F_A)$, $\exists i \geq 0$: $M, \lambda[i] \models \psi$ and $M, \lambda[j] \models \phi$ for all $j \in \{0, \ldots, i-1\}$
- $M, s \models \langle\langle A^b \rangle\rangle \Box \phi$ iff $\exists$ coalition-uniform b-strategy F_A such that for all $\lambda \in out(s, F_A)$ and $i \geq 0$: $M, \lambda[i] \models \phi$.
- $M, s \models K_a \phi$ iff $\phi \in s_a$

Note that we do not impose any conditions on the syntactic knowledge (not consistency, not veracity etc.). Of course, in a particular modelling scenario such conditions may be imposed. The general results for decidability of model-checking stated below hold for such special cases too. They also hold for *strong coalition uniformity* where the truth definition for coalition modalities requires the existence of a coalition-uniform strategy from every indistinguishable state. For example, for $M, s \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ strong coalition uniformity requires that $\forall s' \sim_A s$, $\exists$ coalition-uniform b-strategy F_A such that for all $\lambda \in out(s', F_A)$: $M, \lambda[1] \models \phi$.

3 Model-Checking RB±ATSEL

In this section, we prove the following general result:

Theorem 1. *The model-checking problem for RB±ATSEL with coalition-uniform strategies, with respect to any decidable notion of $\sim_A$, is decidable.*

To prove decidability we give an algorithm which, given a structure $M = (\Phi, Agt, Res, S, \Pi, Act, d, c, \delta)$ and a formula ϕ_0, returns the set of states $[\phi_0]_M$ satisfying ϕ_0: $[\phi_0]_M = \{s \mid M, s \models \phi_0\}$. The theorem follows from Lemmas 1 and 2 which establish termination and correctness of the algorithm respectively.

Algorithm 1 Labelling ϕ_0

function RB±ATSEL-LABEL(M, ϕ_0)
  for $\phi' \in Sub(\phi_0)$ **do**
    case $\phi' = p,\ \neg\phi,\ \phi \vee \psi$ standard, see [Alur *et al.*, 2002]
    case $\phi' = K_a \phi$
      $[\phi']_M \leftarrow \{ s \mid s \in S \wedge \phi \in s_a \}$
    case $\phi' = \langle\langle A^b \rangle\rangle \bigcirc \phi$
      $[\phi']_M \leftarrow Pre(A, [\phi]_M, b)$
    case $\phi' = \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$
      $[\phi']_M \leftarrow \{ s \mid s \in S \wedge$ UNTIL$([node_0(s, b)], \{\ \}, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi) \}$
    case $\phi' = \langle\langle A^b \rangle\rangle \Box \phi$
      $[\phi']_M \leftarrow \{ s \mid s \in S \wedge$ BOX$([node_0(s, b)], \{\ \}, \langle\langle A^b \rangle\rangle \Box \phi) \}$
  return $[\phi_0]_M$

The algorithm is shown in Algorithm 1. Given ϕ_0, we produce a set of subformulas $Sub(\phi_0)$ of ϕ_0 in the usual way (but excluding subformulas in the scope of a knowledge modality), ordered in increasing order of complexity. We then proceed by cases. For all formulas in $Sub(\phi)$ apart from $K_a \phi$, $\langle\langle A^b \rangle\rangle \bigcirc \phi$, $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ (where b may contain ∞) we essentially run the standard ATL model-checking algorithm [Alur *et al.*, 2002]. Labelling states with $\langle\langle A^b \rangle\rangle \bigcirc \phi$ makes use of a function $Pre(A, \rho, b)$ which, given a coalition A, a set $\rho \subseteq S$ and a bound b, returns a set of states s in which A has a joint action σ_A with $cons(\sigma_A) \leq b$ such that $out(s, \sigma_A) \subseteq \rho$. Labelling states with $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ is more complex, and in the interests of readability we provide separate functions: UNTIL for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ formulas is shown in Algorithm 2, and BOX for $\langle\langle A^b \rangle\rangle \Box \phi$ formulas is shown in Algorithm 3.

Both algorithms proceed by depth-first and-or search on M. Information about the state of the search is recorded in a search tree of nodes. A *node* is a structure which consists of a state of M (including the epistemic states of the agents), the resources available to the agents A in that state (if any), and a finite path of nodes leading to this node from the root node. Edges in the tree correspond to joint actions by agents in A and are labelled with the action taken. Note that the resources available to the agents in a state s on a path constrain the edges from the corresponding node to be those actions σ_A where $cons(\sigma_A)$ is less than or equal to the available resources. For each node n in the tree, we have a function $s(n)$ which returns its state, $p(n)$ which returns the nodes on the path to n, and $a(n)$ which returns the joint action taken by A to reach $s(n)$ (i.e., the label of the edge to n from its predecessor). The function $e_{i,k}(n)$ returns the resource availability on the i-th resource in $s(n)$ for agent $k \in A$ as a result of following $p(n)$. The function $node_0(s, b)$ returns the root node, i.e., a node n_0 such that $s(n_0) = s$, $p(n_0) = [\ ]$, $a(n_0) =$ *no-op*, and $e_{i,k}(n_0) = b_{i,k}$ for all resources i and agents $k \in A$. The function $node(n, \sigma, s')$ returns a node n' where $s(n') = s'$,

$p(n') = [p(n) \cdot n]$, $a(n') = \sigma$, and for all resources i and agents $k \in A$, $e_{i,k}(n') = e_{i,k}(n) + prod_i(\sigma_k) - cons_i(\sigma_k)$. In addition, we assume functions $hd(u)$, $tl(u)$ which return the head and tail of a list u, and $u \circ v$ which concatenates the lists u and v. (We abuse notation slightly, and treat sets as lists, e.g., use $hd(u)$ where u is a set, to return an arbitrary element of u, and use $\circ$ between a set and a list.)

Algorithm 2 Labelling $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$

function UNTIL($B, C, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$)
  if $B = [\,]$ **then**
    return *true*
  $n \leftarrow hd(B)$
  if $\exists n' \in p(n) : s(n') = s(n) \wedge$
      $(\forall i, k : e_{i,k}(n') \geq e_{i,k}(n))$ **then**
    return *false*
  for $i, k \in \{i \in Res, k \in A \mid \exists n' \in p(n) :$
      $s(n') = s(n) \wedge e_{i,k}(n') < e_{i,k}(n) \wedge$
      $(\forall j, m : e_{j,m}(n') \leq e_{j,m}(n))\}$ **do**
    $e_{i,k}(n) \leftarrow \infty$
  if $s(n) \in [\psi]_M$ **then**
    return UNTIL($tl(B), C \cup \{n\}, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$)
  if $s(n) \notin [\phi]_M$ **then**
    return *false*
  if $\exists n' \in C : p(n) \cdot n \sim_A p(n')[1, |p(n) \cdot n|]$ **then**
    $\sigma \leftarrow a(p(n')[|p(n) \cdot n| + 1]$
    if $\sigma \in D_A(s(n)) \wedge cons(\sigma) \leq e(n)$ **then**
      $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
      return UNTIL($P \circ tl(B), C, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$)
  else
    $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid cons(\sigma) \leq e(n)\}$
    for $\sigma \in ActA$ **do**
      $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
      if UNTIL($P \circ tl(B), C, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$) **then**
        return *true*
  return *false*

UNTIL (Algorithm 2) takes a stack (list) of 'open' nodes B, a set of 'closed' nodes C, and a formula $\phi' = \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi \in Sub(\phi_0)$ as input. If there are no more open nodes to consider, UNTIL returns true, indicating that a strategy exists to enforce $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$. Otherwise we check whether the state $s(n)$ has been encountered before on $p(n)$, i.e., $p(n)$ ends in a loop. If the loop is unproductive (i.e., resource availability has not increased since the previous occurrence of $s(n)$ on the path $p(n)$), then the loop is not necessary for a successful strategy, and search on this branch is terminated. If, on the other hand, the loop strictly increases the availability of at least one resource i for some agent k and does not decrease the availability of other resources, then $e_{i,k}(n)$ is replaced with ∞ as a shorthand denoting that any finite amount of i can be produced by repeating the loop sufficiently many times. We then check if the second argument ψ of ϕ' is true in $s(n)$. If so, search terminates on the current branch, and continues on a different branch by expanding the next open node in B and adding the current node n to the set of closed nodes. Note that we only add 'successful' branches to the closed set rather than all visited nodes, as search proceeds depth-first. Coalition uniformity is ensured if action choices are consistent with those taken in $\sim_A$ states on all successful paths explored to date ($n_1, \ldots, n_k \sim_A n'_1, \ldots, n'_k$ iff $s(n_1), \ldots, s(n_k) \sim_A s(n'_1), \ldots, s(n'_k)$). If the current branch is not closed (i.e., the second argument ψ of ϕ' is not true in $s(n)$, but ϕ is true in $s(n)$), search continues on this branch. First we check if the current path (including the current node) is epistemically indistinguishable from a (prefix of) a path to a closed node ρ. If so, for a coalition-uniform strategy, the same action, σ, should be selected in the current state as in the corresponding state in ρ. (We use $p(n')[i]$ to denote the i-th node in the path $p(n')$, and $p(n')[1, j]$ to denote the prefix of $p(n')$ up to the j-th node.) If the cost of the action σ is less than the resource availability in the current state, we generate a new node for each possible outcome state of the action, and call UNTIL recursively to continue the search, pushing the nodes corresponding to the successor states onto the stack of open paths. If the cost of the required action is greater than the current resource availability, search terminates on the current branch with false. If no action is *required* at the current state for coalition uniformity, then for each action that is possible in the current state given the current resource availability, we attempt to find a strategy for each of the outcome states of that action. If a strategy cannot be found for any action possible in $s(n)$, UNTIL returns false.

Algorithm 3 Labelling $\langle\langle A^b \rangle\rangle \Box \phi$

function BOX($B, C, \langle\langle A^b \rangle\rangle \Box \phi$)
  if $B = [\,]$ **then**
    return *true*
  $n \leftarrow hd(B)$
  if $s(n) \notin [\phi]_M$ **then**
    return *false*
  if $\exists n' \in p(n) : s(n') = s(n) \wedge$
      $(\forall j, k : e_{j,k}(n') \geq e_{j,k}(n)) \wedge$
      $(\exists j, k : e_{j,k}(n') > e_{j,k}(n))$ **then**
    return *false*
  if $\exists n' \in p(n) : s(n') = s(n) \wedge$
      $(\forall j, k : e_{j,k}(n') \leq e_{j,k}(n))$ **then**
    return BOX($tl(B), C \cup \{n\}, \langle\langle A^b \rangle\rangle \Box \phi$)
  if $\exists n' \in C : p(n) \cdot n \sim_A p(n')[1, |p(n) \cdot n|]$ **then**
    $\sigma \leftarrow a(p(n')[|p(n) \cdot n| + 1]$
    if $\sigma \in D_A(s(n)) \wedge cons(\sigma) \leq e(n)$ **then**
      $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
      return BOX($P \circ tl(B), C, \langle\langle A^b \rangle\rangle \Box \phi$)
  else
    $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid cons(\sigma) \leq e(n)\}$
    for $\sigma \in ActA$ **do**
      $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
      if BOX($P \circ tl(B), C, \langle\langle A^b \rangle\rangle \Box \phi$) **then**
        return *true*
  return *false*

BOX (Algorithm 3) takes a stack (list) of 'open' nodes B, a set of 'closed' nodes C, and a formula $\phi' = \langle\langle A^b \rangle\rangle \Box \phi \in Sub(\phi_0)$ as input. If there are no more open nodes to consider, BOX returns true. It then checks if ϕ is false in the state $s(n)$. If so, it returns false immediately, terminating search of the current branch of the search tree. Otherwise we check whether the state $s(n)$ has been encountered before on $p(n)$, i.e., $p(n)$ ends in a loop. For BOX the loop check is slightly different. If the loop decreases the amount of at least one resource for one agent without increasing the availability of any other resource, it cannot form part of a successful strategy, and the search terminates returning false. If a non-decreasing loop is found, then it is possible to maintain the invariant formula ϕ forever without expending any resources, and the search terminates on the current branch and continues on a different branch by expanding the next open node in B and adding the current node n to the set of closed nodes. The remaining cases are similar to UNTIL. If the current branch is not closed, search continues on the branch, first checking whether an action is required for the strategy to be coalition-uniform, and, if not, for each action that is possible in the current state given the current resource availability.

Lemma 1 (Termination)**.** *Algorithm 1 terminates.*

Proof. All the cases in Algorithm 1 apart from the calls to Algorithms 2 and 3 clearly terminate. It therefore suffices to show that the calls to Algorithms 2 and 3 terminate.

In order to prove termination, we first show (Claim 1) that on each path explored by Algorithm 2 or Algorithm 3 there is no infinite loop where nodes with the same state and incomparable $e(n)$ occur. This implies that the tree explored for each element of B is of finite depth, since the number of states is finite and repeated states will necessarily occur in the search, and if the resource availability vectors are comparable the search will terminate for that node. Algorithm 2 returns false for a non-increasing loop on line 6, and resets resource availability to ∞ on line 8 for an increasing loop; if the same resource-increasing loop is encountered again with all resources set to ∞ or unchanged, the algorithm will return false on line 6. Algorithm 3 terminates returning false on line 8 if the loop is decreasing, and calls itself on the next member of B on line 10 if the loop is non-decreasing. Second, we show that (Claim 2) there cannot be infinitely many recursive calls generated by calls on line 10 of Algorithm 2 and of Algorithm 3. We do this by showing that the list B containing paths that must be checked with respect to a currently successful strategy, will eventually become empty. Together, these two claims provide the proof of the lemma, because they guarantee that after a finite number of recursive calls both algorithms terminate.

Claim 1: Algorithm 2 and Algorithm 3 cannot generate a path where nodes with the same state and incomparable $e(n)$ occur infinitely often.

This part of the termination proof is similar to that in [Alechina *et al.*, 2014] which in turn is similar to the proof of Lemma f in [Reisig, 1985, p.70], and proceeds by induction on the number of resource/agent pairs m. For $m = 1$, since $e(n)$ is always positive, the claim is immediate. Assume the claim holds for m and let us show it for $m + 1$. In other words, the first m positions in $e(n)$ will eventually become comparable. Then the $m + 1$ position will become comparable since there are only finitely many positive integers which are smaller than a given $e_{m+1}(n)$.

Claim 2. There can be only finitely many calls generated by the coalition uniformity check (line 13 of Algorithm 2 and line 11 of Algorithm 3).

Here we need to show that the open list B is used to explore a finite tree, hence B will eventually become empty. The depth of this tree is bounded by the depth of the longest possible path. Since the relation $\sim_A$ only holds between the paths of the same length, Claim 1 is sufficient to limit the depth of the tree. The finite branching factor of the tree follows from the fact that the set $out(s, \sigma_A)$ is always finite. □

Lemma 2 (Correctness)**.** *Given a model M, a state s in M and a formula ϕ, Algorithm 1 labels s with ϕ iff $M, s \models \phi$.*

Proof. The proof for all the cases in Algorithm 1 apart from the calls to Algorithms 2 and 3 is straightforward.

Let us look at the case for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$. We need to show that a call to UNTIL$([node_0(s, b)], \{\ \}, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns true if, and only if, $M, s \models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and similarly for $\langle\langle A^b \rangle\rangle \Box \phi$. By inductive hypothesis, the algorithm only explores paths where ϕ holds (line 11) until ψ is encountered (line 9), so the pure temporal semantics of $\mathcal{U}$ is respected. Note that it is enough to find a finite strategy which is guaranteed to achieve a state where ψ is true. After that, the agents can select the $idle$ action in all subsequent histories, which both ensures coalition uniformity and does not require any resources. The proof as regards resource bounds (and whether it is safe to reset a bound to ∞ when a productive loop is encountered, and explore a productive loop only once) is similar to the one for RB±ATL [Alechina *et al.*, 2014]. However, in addition we need to show that the algorithms return true if and only if there is a satisfying *coalition-uniform* strategy. Assume that the algorithm returns true. We need to show that the strategy found is coalition-uniform. This is ensured by the check on line 13. A current successful strategy is kept in the closed set C, and for all coalition-indistinguishable paths we check whether the same strategy returns true, and only then return true, otherwise we backtrack and try another strategy. For the other direction, assume that there is a coalition-uniform strategy for A to enforce $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$. An inspection of Algorithm 2 shows that if such a strategy exists, then there exists a sequence of recursive calls by the algorithm (corresponding to the choice of actions given by the strategy) which results in the algorithm returning true.

The case of $\langle\langle A^b \rangle\rangle \Box \phi$ is similar. We ensure that Algorithm 3 returns a coalition-uniform strategy by an identical check on line 11. □

4 Verifying Norm Monitoring Strategies

In this section, we show how RB±ATSEL can be used to reason about knowledge-based resource bounded strategies in a simple norm monitoring scenario. In the scenario, agents monitor and enforce a norm that visitors to a museum are prohibited from getting too close to the artwork on display: if a visitor approaches the artwork, s/he is warned; if s/he approaches again after being warned, s/he is required to leave the museum. For simplicity, we assume the museum has a single exhibition room, there are two monitoring agents 1 and 2, and one visitor 3. At each timestep, the visitor can perform an $idle$ action or approach the artwork, app. Agents 1 and 2 can perform an $idle$ action, an observation action, obs, issue a warning $warn$, escort the visitor out of the museum rem, or recharge their battery gen. The agents require a single resource, energy. The gen action produces energy; all other actions apart from $idle$ consume energy.

We use propositions a to denote that the visitor has approached the artwork, c_i ($i \in \{1,2\}$) to denote that agent i has just charged their battery, w to denote that the visitor has been warned, and r to denote that the visitor has been removed from the museum. The global system state is represented by $s = (s_1, s_2, s_3, s_e)$, where s_i ($i = \{1,2,3\}$) is the local state of i, and s_e is the state of the environment. The set of formulas Φ which constitute possible contents of agents' states includes information on whether the agents have (just) charged, whether the visitor has approached the artwork, been warned, or removed from the museum.

The museum scenario can be modelled by the structure $M = (\Phi, Agt, Res, S, \Pi, Act, d, c, \delta)$, where $\Phi = \{a, c_1, c_2, r, w\}$, $Agt = \{1,2,3\}$, $Res = \{energy\}$, $S = 2^{\{a,c_1,w\}} \times 2^{\{a,c_2,w\}} \times 2^{\{r,w\}} \times 2^{\Pi}$, $\Pi = \{a, c_1, c_2, w, r\}$, $Act = \{idle, app, obs, warn, rem, gen\}$.

d is defined for all $s \in S$ as follows:

1. $idle \in d(s,i)$ for all $i \in \{1,2,3\}$
2. $app \in d(s,3)$ iff $r \notin s_3$
3. $obs \in d(s,i)$ for all $i \in \{1,2\}$
4. $gen \in d(s,i)$ for all $i \in \{1,2\}$
5. $warn \in d(s,i)$ for all $i \in \{1,2\}$ iff $a \in s_i$ (a warning is only issued if a monitor knows the visitor has approached the artwork)
6. $rem \in d(s,i)$ for all $i \in \{1,2\}$ iff $a, w \in s_i$ (the visitor is only removed if s/he approaches the artwork and a warning has been issued)

$c(idle, energy) = 0$, $c(gen, energy) = -2$, $c(\alpha, energy) = 1$ for $\alpha \in Act \setminus \{idle, gen\}$

δ is defined based on the following post conditions of actions (action preconditions are given by d):

1. $idle$ performed by agent $i \in \{1,2\}$ removes c_i from s_i and s_e; $idle$ performed by agent 3 removes a from s_e
2. app performed by agent 3 adds a to the state of the environment
3. obs performed by agent $i \in \{1,2\}$ removes c_i from s_i and s_e, and, if performed in a state where a is true (false), adds (removes) a to (from) i's local state
4. $warn$ performed by agent $i \in \{1,2\}$ removes c_i from s_i and s_e, and adds w to s_1, s_2, s_3 and s_e
5. rem performed by agent $i \in \{1,2\}$ removes c_i from s_i and s_e, and adds r to s_3, s_e
6. gen performed by agent $i \in \{1,2\}$ adds c_i to s_i and s_e

The following property states that if the visitor approaches the artwork, then this will be known by one of the monitoring agents in the next state:

$$\langle\!\langle \{1,2\}^{1,0} \rangle\!\rangle \Box (a \rightarrow \langle\!\langle \{1,2\}^{0,0} \rangle\!\rangle \bigcirc (K_1 a \vee K_2 a)).$$

This formula is true for a notion of coalition uniformity based on distributed knowledge of the coalition. The strategy is as follows: agents take turns charging and observing; agent 1 chooses obs in the state where both agents's states don't contain c_i, hence it needs 1 unit of energy to start with. The following properties state that the monitoring agents are able to warn the visitor in two steps after the visitor's approach, and that after being warned, the visitor will be removed directly after another approach. They are true under the same notion of coalition uniformity.

$\phi_w = \langle\!\langle \{1,2\}^{1,0} \rangle\!\rangle \Box (a \rightarrow \langle\!\langle \{1,2\}^{0,0} \rangle\!\rangle \bigcirc (\langle\!\langle \{1,2\}^{0,0} \rangle\!\rangle \bigcirc w))$

$\langle\!\langle \{1,2\}^{1,0} \rangle\!\rangle \Box (w \rightarrow \phi_r)$, where $\phi_r = \phi_w[w/r]$.

5 Related Work

The motivation of work on epistemic logics where acquiring information requires resources [Jamroga and Tabatabaei, 2013; Naumov and Tao, 2015] is very similar to ours, however the technical approach is very different. In [Jamroga and Tabatabaei, 2013], a set of states an agent considers possible is updated by observations (which eliminate some states), and observations have resource costs. The logic introduced in the paper can express statements such as 'i can potentially achieve knowledge of whether ϕ is true under resource bound b'. In [Naumov and Tao, 2015], edges in an epistemic indistinguishability relation have weights corresponding to the costs of removing them (obtaining information which would distinguish the states). This allows the authors to define weighted knowledge operators which represent the costs of coming to know whether some proposition is true.

Other related work falls broadly into three categories: work on model-checking resource logics (without epistemics), work on model-checking epistemic ATL (under standard semantics for epistemics and without knowledge change), and work on model-checking DEL and epistemic planning. There exist several formalisms that extend Alternating Time Temporal Logic (ATL), [Alur *et al.*, 2002] with reasoning about resources available to agents and production and consumption of resources by actions. When the production of resources is allowed, the model-checking problem for many (but not all) of these logics is undecidable (for a survey, see [Alechina *et al.*, 2015]). Epistemic ATL has been studied extensively, see e.g., [van der Hoek and Wooldridge, 2002; Ågotnes, 2006; Lomuscio *et al.*, 2009; Guelev *et al.*, 2011; Dima and Tiplea, 2011]. Its model-checking problem with perfect recall and uniform strategies was shown to be undecidable in the case of more than one agent in [Dima and Tiplea, 2011]. In [Guelev *et al.*, 2011], it was shown that if uniform strategies are defined in terms of distributed knowledge of the coalition, the model-checking problem becomes decidable. The technique used to prove this is very different from the one used in this paper. Various notions of coalition uniformity were studied in [van Ditmarsch and Knight, 2014], and justified for a setting where agents in a coalition share their information; the model-checking problem for the resulting logic was not considered. There is a large body of work on DEL. The model-checking problem for full DEL was shown to be undecidable in [Aucher and Bolander, 2013] and decidable for a fragment of DEL in [Aucher and Schwarzentruber, 2013]. DEL-based epistemic planning is also undecidable in general, but is tractable for some special cases [Yu *et al.*, 2013; Bolander *et al.*, 2015].

Acknowledgement. This work was supported by the Engineering and Physical Sciences Research Council [grant EP/K033905/1].

References

[Ågotnes, 2006] T. Ågotnes. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):375–407, 2006.

[Alechina *et al.*, 2014] N. Alechina, B. Logan, H. N. Nguyen, and F. Raimondi. Decidable model-checking for a resource logic with production of resources. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, pages 9–14, 2014.

[Alechina *et al.*, 2015] N. Alechina, N. Bulling, B. Logan, and H. N. Nguyen. On the boundary of (un)decidability: Decidable model-checking for a fragment of resource agent logic. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015.

[Alur *et al.*, 2002] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[Álvarez-Napagao *et al.*, 2011] S. Álvarez-Napagao, H. Aldewereld, J. Vázquez-Salceda, and F. Dignum. Normative monitoring: Semantics and implementation. In *Coordination, Organizations, Institutions, and Norms in Agent Systems COIN 2010*, volume 6541 of *LNCS*, pages 321–336. Springer, 2011.

[Andersen *et al.*, 2012] M. B. Andersen, T. Bolander, and M. H. Jensen. Conditional epistemic planning. In *13th European Conference on Logics in Artificial Intelligence, JELIA 2012*, volume 7519 of *LNCS*, pages 94–106. Springer, 2012.

[Aucher and Bolander, 2013] G. Aucher and T. Bolander. Undecidability in epistemic planning. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.

[Aucher and Schwarzentruber, 2013] G. Aucher and F. Schwarzentruber. On the complexity of dynamic epistemic logic. In *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2013)*, 2013.

[Baltag *et al.*, 1998] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements and common knowledge and private suspicions. In *TARK'98 Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56, 1998.

[Bolander *et al.*, 2015] T. Bolander, M. H. Jensen, and F. Schwarzentruber. Complexity results in epistemic planning. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 2791–2797, 2015.

[Bulling and Farwer, 2010] N. Bulling and B. Farwer. On the (un-)decidability of model checking resource-bounded agents. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 567–572, 2010.

[Dima and Tiplea, 2011] C. Dima and F. L. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.

[Fagin *et al.*, 1995] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT, 1995.

[Guelev *et al.*, 2011] D. P. Guelev, C. Dima, and C. Enea. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics*, 21(1):93–131, 2011.

[Jamroga and Tabatabaei, 2013] W. Jamroga and M. Tabatabaei. Accumulative knowledge under bounded resources. In *Computational Logic in Multi-Agent Systems - 14th International Workshop, CLIMA XIV*, volume 8143 of *LNCS*, pages 206–222. Springer, 2013.

[Konolige, 1986] K. Konolige. *A Deduction Model of Belief*. Morgan Kaufmann Publishers, 1986.

[Lomuscio *et al.*, 2009] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. In *Proceedings of the 21st International Conference on Computer Aided Verification (CAV 2009)*, volume 5643 of *LNCS*, pages 682–688. Springer, 2009.

[Naumov and Tao, 2015] P. Naumov and J. Tao. Budget-constrained knowledge in multiagent systems. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pages 219–226, 2015.

[Petrick and Bacchus, 2004] R. P. A. Petrick and F. Bacchus. Extending the knowledge-based approach to planning with incomplete information and sensing. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 613–622, 2004.

[Reisig, 1985] W. Reisig. *Petri Nets: An Introduction*, volume 4. Springer, 1985.

[van der Hoek and Wooldridge, 2002] W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*, pages 1167–1174. ACM, 2002.

[van Ditmarsch and Knight, 2014] H. van Ditmarsch and S. Knight. Partial information and uniform strategies. In *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV*, volume 8624 of *LNCS*, pages 183–198. Springer, 2014.

[van Ditmarsch and Kooi, 2008] H. van Ditmarsch and B. Kooi. Semantic results for ontic and epistemic change. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, pages 87–117, 2008.

[Yu *et al.*, 2013] Q. Yu, X. Wen, and Y. Liu. Multi-agent epistemic explanatory diagnosis via reasoning about actions. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.