When observation and communication have costs

Natasha Alechina

Lorentz Center Workshop
To Be Announced! Synthesis of Epistemic Protocols

Plan of the talk

- motivation: why combine ontic and epistemic actions
- motivation: why consider costs of epistemic actions
- resource logics: logics to reason about actions with costs
- resource logics with (syntactic) knowledge: logics to reason about epistemic actions which have costs
- arguments in favour of syntactic representation of knowledge when reasoning about epistemic actions

joint work with Brian Logan, Nga Nguyen, Franco Raimondi, Nils Bulling, and Mehdi Dastani

Combining ontic and epistemic actions

- many different motivations (planning; diagnostics; knowledge-based plans/programs;...)
- this talk: monitoring for norm (social rule) violations
- for example, some agents want to check that an inexperienced robot bartender is doing its job properly
- no drink order is left unserved; no requests are served out of turn
- basically, need to model a combination of agents acting (coming in and ordering drinks, serving drinks) and other agents trying to check whether a particular temporal pattern (norm violation) has occurred

Costs of epistemic actions

- monitoring involves making observations and communicating with other agents (who may be in a position to make observations about another part of the pattern)
- observations often have non-trivial costs (have to drop what you are doing and go somewhere to have a look, or have to use costly equipment, or pay some authority for verified information)
- exchanging messages also has costs, for example, energy, or money

Original idea

- take a logic of action (for example, ATL)
- add costs to actions (done: later in the talk)
- add epistemic actions (DEL style?) with costs
- check how much a norm monitoring strategy would cost

Problem with the original idea

- epistemic planning with factual epistemic preconditions and post-conditions is undecidable [Bolander and Andersen 2011]
- not much hope that ATL model-checking with both ontic and epistemic (DEL style) actions is decidable

However...

- Ron's PKS planner has both types of actions, but produces plans in PSPACE; other perfectly feasible looking knowledge based planners exist (Jérôme's talk, Sheila's talk)
- in other words, the planning problem with epistemic actions (and also model-checking with mixed ontic and epistemic actions) may be made decidable and not any harder than the classical planning problem
- what makes the difference is the syntactic nature of the agent's knowledge base
- syntactic knowledge/belief in a nutshell: an agent knows/believes ϕ iff ϕ is in its knowledge base (or derivable from it by some simple terminating procedure, e.g. $K_i(p \underline{\vee} q)$, $K_i \neg p \Rightarrow K_i q$)

Syntactic knowledge vs interpreted systems

- states of the system are tuples $(s_e, s_1, \ldots, s_n)$ where s_e is the state of the environment and s_i is a local state of agent i
- in this, similar to interpreted systems
- however, each s_i is a finite set of formulas (= a knowledge base in planning)
- $M, s \models K_i\phi$ iff $\phi \in s_i$
- NOT iff all s' with $s'_i = s_i$ satisfy ϕ
- NOT iff all s' with the same 'ϕ-part of i's state' satisfy ϕ

Why syntactic knowledge

- no omniscience
- easy to implement
- may be computationally more efficient (more discussion later in the talk)

Shortcomings of syntactic knowledge

- update policy needs to be crafted by hand, just like pre- and post-conditions of ontic actions
- deductive closure conditions (if any) also need to be crafted by hand
- however this also makes the approach flexible
- for example, can represent different types of epistemic reasoners:
  - agent 1 believes everything that agent 2 believes: in agent 1's KB, $K_2\phi \Rightarrow \phi$
  - agents may do projection differently, for example if both 1 and 2 observe agent 3 in room1, then maybe 1 adds $K_2 in(room1, 3)$ and 2 adds only $K_3 in(room1, 3)$.

Resource Logics

Resource Logics

- variants of Alternating-Time Temporal Logic (ATL) where transitions have costs (or rewards) and the syntax can express resource requirements of a strategy, e.g.:

  agents A can enforce outcome φ *if they have at most* b_1 *units of resource* r_1 *and* b_2 *units of resource* r_2

- various flavours of resource logics exist: RBCL (IJCAI 2009), RB-ATL (AAMAS 2010), RB$\pm$ATL (ECAI 2014), RAL (Bulling & Farwer), PRB-ATL (Della Monica et al.), QATL* (Bulling & Goranko)

Model-checking resource Logic

- model-checking problem: given a structure, a state in the structure and a formula, does the state satisfy the formula?
- using model-checking, we can verify resource requirements of a multi-agent system (specify the system as a model, and write a formula expressing a system objective)

Model-checking resource Logics

- for most resource logics the model-checking problem is undecidable: in particular, various flavours of RAL, and QATL*
- here, I present a resource logic RB$\pm$ATL (ECAI 2014) with decidable model-checking problem

RB±ATL: syntax

- $Agt = \{a_1, \ldots, a_n\}$ a set of n agents
- $Res = \{res_1, \ldots, res_r\}$ a set of r resources,
- Π a set of propositions
- $B = \mathbb{N}^r_\infty$ a set of resource bounds, where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$

RB±ATL: syntax

Formulas of RB±ATL are defined by the following syntax

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

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

RB±ATL: meaning of formulas

- $\langle\langle A^b \rangle\rangle \bigcirc \psi$ means that a coalition *A* can ensure that the next state satisfies φ under resource bound *b*
- $\langle\langle A^b \rangle\rangle \psi_1 \mathcal{U} \psi_2$ means that *A* has a strategy to enforce ψ while maintaining the truth of φ, and the cost of this strategy is at most *b*
- $\langle\langle A^b \rangle\rangle \Box \psi$ means that *A* has a strategy to make sure that φ is always true, and the cost of this strategy is at most *b*

Resource-bounded concurrent game structure

A RB-CGS is a tuple $M = (Agt, Res, S, \Pi, \pi, Act, d, c, \delta)$ where:

- Agt is a non-empty set of n agents, Res is a non-empty set of r resources and S is a non-empty set of states;
- Π is a finite set of propositional variables and $\pi : \Pi \to \wp(S)$ is a truth assignment
- Act is a non-empty set of actions which includes *idle*, and $d : S \times Agt \to \wp(Act) \setminus \{\emptyset\}$ is a function which assigns to each $s \in S$ a non-empty set of actions available to each agent $a \in Agt$
- $c : S \times Agt \times Act \to \mathbb{Z}^r$ (the integer in position i indicates consumption or production of resource res_i by the action a)
- $\delta : (s, \sigma) \mapsto S$ for every $s \in S$ and joint action $\sigma \in D(s)$ gives the state resulting from executing σ in s.

Additional assumptions and notation

- for every $s \in S$ and $a \in Agt$, $idle \in d(s, a)$
- $c(s, a, idle) = \bar{0}$ for all $s \in S$ and $a \in Agt$ where $\bar{0} = 0^r$
- 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 a coalition A, $D_A(s)$ is the set of all joint actions by agents in A
- $out(s, \sigma) = \{s' \in S \mid \exists \sigma' \in D(s) : \sigma = \sigma'_A \wedge s' = \delta(s, \sigma')\}$
- $cost(s, \sigma) = \sum_{a \in A} c(s, a, \sigma_a)$
- if one agent consumes 10 units of resource and another agent produces 10 units of resource, the cost of their joint action is 0

Strategies and their costs

- a *strategy for a coalition* $A \subseteq Agt$ is a mapping $F_A : S^+ \rightarrow Act$ 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 $i \geq 0$, $\lambda[i+1] \in out(\lambda[i], F_A(\lambda[0,i]))$
- $out(s, F_A)$ the set of all consistent computations λ of F_A that start from s
- given a bound $b \in B$, a computation $\lambda \in out(s, F_A)$ is b-consistent with F_A iff, for every $i \geq 0$, $\sum_{j=0}^{i} cost(\lambda[j], F_A(\lambda[0,j])) \leq b$
- F_A is a b-strategy if all $\lambda \in out(s, F_A)$ are b-consistent

Truth definition

- $M, s \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff $\exists$ 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 $\exists$ 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 $\exists$ b-strategy F_A such that for all $\lambda \in out(s, F_A)$ and $i \geq 0$: $M, \lambda[i] \models \phi$

Infinite bound versions

Since the infinite resource bound version of RB-ATL modalities correspond to the standard ATL modalities, we write

- $\langle\langle A^{\bar{\infty}} \rangle\rangle \bigcirc \phi$ as $\langle\langle A \rangle\rangle \bigcirc \phi$
- $\langle\langle A^{\bar{\infty}} \rangle\rangle \phi \mathcal{U} \psi$ as $\langle\langle A \rangle\rangle \phi \mathcal{U} \psi$
- $\langle\langle A^{\bar{\infty}} \rangle\rangle \Box \phi$ as $\langle\langle A \rangle\rangle \Box \phi$

Model-checking RB±ATL

The model-checking problem for RB±ATL is the question whether, for a given RB-CGS structure M, a state s in M and an RB±ATL formula ϕ, $M, s \models \phi$.

Theorem (Alechina, Logan, Nguyen, Raimondi 2014):

The model-checking problem for RB±ATL is decidable

Complexity

- the model-checking problem for RB$\pm$ATL is EXPSPACE-hard
- no upper bound known
- may be an upper bound can be obtained from the (non-elementary) upper bound for vector addition systems (Leroux & Schmitz, LICS 2015)
- however, model-checking problem for RB$\pm$ATL with one resource type is in PSPACE

Adding syntactic knowledge

Adding syntactic knowledge

- extend the language to include K_i, $i \in Agt$ formulas
- fix a finite set of formulas Φ that can occur in an agent's state
- in $M = (Agt, Res, S, \Pi, \pi, Act, d, c, \delta)$, S is a set of tuples $(s_e, s_1, \ldots, s_n)$ where each $s_i \subseteq \Phi$
- d for every $s \in S, i \in Agt$ satisfies $d(s,i) = d(s',i)$ if $s_i = s'_i$

Model-checking with syntactic knowledge

Without requiring uniform strategies:

Theorem

The model-checking problem for syntactic epistemic RB$\pm$ATL is decidable

Coalition-uniform strategies

- consider the following notion of uniform strategies
- for $A \subseteq Agt$ and $s, s' \in S$, let $s \sim_A s'$ if for all $i \in A$, $s_i = s'_i$
- lift $\sim_A$ to finite sequences: $(s_1, \ldots, s_k) \sim_A (t_1, \ldots, t_k)$ iff for each $j \in [1, k]$, $s_j \sim_A t_j$
- a strategy F for A is coalition-uniform if for all $\bar{s} \sim_A \bar{t}$, $F(\bar{s}) = F(\bar{t})$
- truth definition quantifies over coalition-uniform strategies

Model-checking with syntactic knowledge

With coalition-uniform strategies:

Theorem

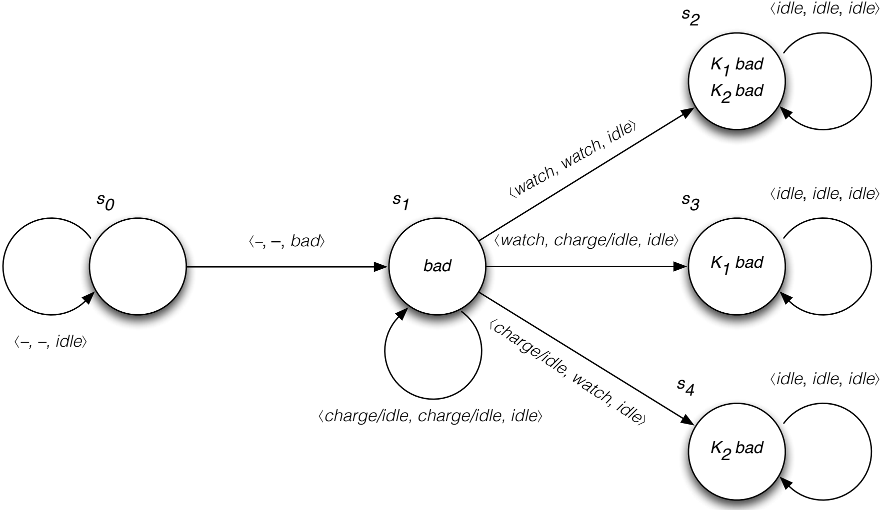
The model-checking problem for syntactic epistemic RB$\pm$ATL with coalition-uniform strategies is decidable

Verifying costs of norm compliance monitoring

Example

- consider a system of three agents 1, 2 and 3
- agents 1 and 2 are monitoring agents
- agent 3 is a normal agent which is being monitored
- agents 1 and 2 have ‘watch’ action which consumes one unit of energy
- if an agent executes ‘watch’ in a state where a violation occurred, in the next state this agent knows that a violation occured
- 1 and 2 also have ‘charge’ action which produces one unit of energy, and an idle action
- agent 3 has an action to violate the norm and an idle action

Example



s_0
⟨–, –, idle⟩
⟨–, –, bad⟩
s_1
bad
⟨charge/idle, charge/idle, idle⟩
⟨watch, watch, idle⟩
⟨watch, charge/idle, idle⟩
⟨charge/idle, watch, idle⟩
s_2
⟨idle, idle, idle⟩
K_1 bad
K_2 bad
s_3
⟨idle, idle, idle⟩
K_1 bad
s_4
⟨idle, idle, idle⟩
K_2 bad

Example

- agents 1 and 2 have a successful monitoring strategy which costs nothing:

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

- the simplest (uniform) strategy is for one agent to always charge and for another to always watch
- if the agent record in their state the last action they performed, a strategy where the agents alternate charging and watching is also possible.

Future work

- extend MCMAS implementation to proper synthesis (return strategies)
- extend syntactic epistemic setting to reasoning about temporal patterns