Finding Uniform Strategies for Multi-Agent Systems

Jan Calta[1], Dmitry Shkatov[2], and Holger Schlingloff[3]

[1] Humboldt University, Berlin, Germany
calta@informatik.hu-berlin.de
[2] University of the Witwatersrand, Johannesburg, South Africa
dmitry@cs.wits.ac.za
[3] Humboldt University, Berlin, Germany
hs@informatik.hu-berlin.de

Abstract. We present an algorithm for finding uniform strategies in multi-agent systems with incomplete information. The algorithm finds all maximal uniform strategies for agents with incomplete information for enforcing a property expressible in the language of Alternating-time Temporal Logic ATL. The main application of the algorithm is automated program synthesis for systems that can be modeled as multi-agent systems with incomplete information (e.g., decentralized distributed systems).

1 Introduction

Over the last few years, the multi-agent systems paradigm has been deployed in safety-critical applications, such as sensor networks for detection of earthquakes (i.e., SOSEWIN network developed in the scope of the SAFER project [1]). For such applications, it is crucially important to verify or design software controlling the system using a formal procedure, which guarantees with certainty that the desired goal has been achieved. In this paper, we present an algorithm for designing software for such systems conforming to required properties.

As a formal model for multi-agent systems, we use Concurrent Epistemic Game Structures (CEGS) introduced in [10]; as a modeling language, we use the language of Alternating-time Temporal Logic ATL [2]. One of the most important aspects of CEGS's are *strategies* agents use to enforce properties expressible in a given modeling language. CEGS's allow us to model systems in which agents have incomplete information. In such systems, not all strategies are of interest, but only those in which every agent performs the same action in indistinguishable states; only such strategies can be viewed as formal models of algorithms. Such strategies are usually referred to as *uniform*. Given the choice of CEGS's as models for multi-agent systems, the problem of designing software conforming to a given property turns into the problem of finding uniform strategies enforcing a property. This property has to be expressible in a chosen modeling language. For the reasons that will become clearer later on, even though we use the syntax of ATL, we give those formulas the meaning they have in Constructive Strategic

Logic CSL from [5]. The result is the logic which we call ATL_u, which is thus our formalism of choice for verification and program synthesis in the context of multi-agent systems with incomplete information.

The main goal of the paper is to present an algorithm for finding uniform strategies in CEGS's with incomplete information. The paper is structured as follows. In section 2, we define the logic ATL_u as well as the concepts used in the rest of the paper. In section 3, we present the algorithm for finding uniform strategies and prove its correctness. In section 4, we present an example of running the algorithm on a simple multi-agent system. In section 5, we estimate the complexity of the algorithm. Finally, in conclusion, we summarize our results and point to directions for further research.

2 Preliminaries

We use concurrent epistemic game structures (CEGS), as defined in [5], as models for reasoning about agents with incomplete information.[4] A CEGS is a tuple

$$\mathfrak{M} = \langle Agt, St, \Pi, \pi, Act, d, o, \sim_1, ..., \sim_k \rangle,$$

where:

- $Agt = \{1, \ldots, k\}$ is a finite set of agents; a (possibly, empty) subset of Agt is called a *coalition*;
- St is a nonempty, finite set of states;
- Π is a set of atomic propositions;
- $\pi : \Pi \to \mathcal{P}(St)$ is a valuation function;
- Act is a nonempty, finite set of actions;
- $d : Agt \times St \to \mathcal{P}(Act)$ assigns to an agent and a state a nonempty subset of Act, which we think of as actions available to the agent at that state. For every $q \in St$, an *action vector* at q is a k-tuple $\langle \alpha_1, \ldots, \alpha_k \rangle$ such that $\alpha_a \in d(a, q)$, for every $1 \leq a \leq k$. The set of all action vectors at q is denoted by $D(q)$;
- o assigns to every $q \in St$ and every $v \in D(q)$ an outcome $o(q, v) \in St$;
- $\sim_1, \ldots, \sim_k \subseteq St \times St$ are indistinguishability relations for agents $1, \ldots, k$. We assume that $\sim_a$, for each $a \in Agt$, is an equivalence relation and, moreover, that $q \sim_a q'$ implies $d(a, q) = d(a, q')$ (i.e., an agent has the same choice of actions at indistinguishable states).

One can use CEGS's to synthesize software for distributed systems, or to verify that such systems satisfy certain properties. To do this in a formal way, we introduce logic ATL_u, whose syntax is defined as follows[5]:

[4] The notion of agent, as used in the literature, is quite an abstract one. For the purposes of this paper, however, the reader can think of agents as components of a distributed system.

[5] Note that the syntax of ATL_u is identical to Alternating-Time Temporal Logic (ATL) [2].

$$\varphi ::= \mathtt{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \mathsf{X}\, \varphi \mid \langle\langle A \rangle\rangle \mathsf{G}\, \varphi \mid \langle\langle A \rangle\rangle \varphi\, \mathsf{U}\, \varphi,$$

where $\mathtt{p} \in \Pi$ and $A \subseteq Agt$. The operator $\langle\langle\ \rangle\rangle$ is called *coalitional operator*, while the operators X, G and U are temporal operators *next, always* and *until*, respectively. We now introduce some notions necessary to define the semantics of ATL_u.

In what follows, given a tuple t, we denote by $t[i]$ the ith element of t. The symbol $\sharp$ denotes an unspecified action of an agent.

Definition 1. *Let $q \in St$ and let $A \subseteq \{1, .., k\}$ be a coalition of agents.*

- *An* A-move m_A *is a k-tuple such that $m_A[a] \in d(a, q)$, for every $a \in A$, and $m_A[a] = \sharp$ otherwise. For the reasons that will become clear later on, we also count as an A-move for an arbitrary coalition A the k-tuple $\sharp^k$. The set of all A-moves at state q is denoted by $D_A(q)$.*
- *An* assigned A-move *is a pair $\langle q, m_A \rangle$ where $q \in St$ and $m_A \in D_A(q)$.*

Definition 2. *Let $q \in St$ and let $m_A \in D_A(q)$ such that $m_A \neq \sharp^k$. The* outcome *of m_A at q, denoted by $out(q, m_A)$, is the set of all states q' such that there is an action vector $v \in D(q)$ with $v[a] = m_A[a]$, for all $a \in A$, such that $o(q, v) = q'$. The outcome of $\sharp^k$ at q is the set of states q' such that $o(q, v) = q'$ for some $v \in D(q)$.*

Definition 3. *A* strategy *S of a coalition A, denoted by S_A, is a nonempty set of assigned A-moves. A strategy S_A is* uniform *iff $q \sim_a q'$ implies $m_A[a] = m'_A[a]$ for every pair of assigned A-moves $\langle q, m_A \rangle$, $\langle q', m'_A \rangle \in S_A$ and every $a \in A$.*

As $\sim_a$, for every $a \in Agt$, is an equivalence relation, we have, in particular, $q \sim_a q$, and therefore, every uniform strategy of a coalition A is a function assigning A-moves to states. By contrast, a strategy that is not uniform can fail be a function, because it can assign more then one A-move to a state.

Definition 4. *Let M be a set of assigned A-moves. The* domain *of M, denoted by $dom(M)$, is the set $\{q \in St \mid \langle q, m_A \rangle \in M\}$.*

Given a sequence of states Λ, we denote by $|\Lambda|$ the number of states in Λ; if Λ is infinite, $|\Lambda| = \omega$. The ith state of Λ is denoted by $\Lambda[i]$.

Definition 5. *A* path *Λ is a (possibly, infinite) sequence of states $q_1, q_2, q_3 \ldots$ that can be effected by subsequent transitions; that is, for every $1 \leq i < |\Lambda|$, if $q_i \in \Lambda$, then there exists an action vector $v \in D(q_i)$ such that $q_{i+1} = o(q_i, v)$.*

We now define outcomes of uniform strategies. We use the notation S_A to refer to a strategy of coalition A.

Definition 6. *Let $q \in St$ and let S_A be a strategy such that $\langle q, m_A \rangle \in S_A$. The* outcome *of S_A at q, denoted by $out(q, S_A)$, is a set of paths $\{\Lambda \mid \Lambda[1] = q$ and for each $1 \leq i < |\Lambda|$ there is an assigned A-move $\langle \Lambda[i], m_A \rangle \in S_A$ such that $\Lambda[i+1] \in out(\Lambda[i], m_A)\}$. If $\Lambda \in out(q, S_A)$ is finite, then we require that either $\langle q_{|\Lambda|}, \sharp^k \rangle \in S_A$ or $q_{|\Lambda|} \notin dom(S_A)$.*

Intuitively, $\langle q_{|A|}, \sharp^k \rangle \in S_A$ means that it does not matter what the agents in A do at the last state of a path that is an outcome of S_A. This possibility of giving the agents in A a "free rein" at the end of carrying out a strategy is what motivated us to count $\sharp^k$ as an A-move, for every $A \subseteq Agt$.

Outcome $out(q, S_A)$ contains every path starting at q that may result from coalition A performing A-moves from S_A assigned to the states on the path. We use notation $out(Q, S_A)$ as a shorthand for $\bigcup_{q \in Q} out(q, S_A)$.

Example 1. Consider the CEGS, depicted in Fig. 1, with $Agt = \{1, 2\}$ and $Act = \{\alpha_1, \beta_1, \alpha_2, \beta_2\}$. Coalition A consists of agent 1 ($A = \{1\}$) and thus every A-move is a tuple with an action for agent 1 and the placeholder $\sharp$ for agent 2. Only one A-move is possible at state q, namely $\langle \alpha_1, \sharp \rangle$, so that $D_A(q) = \{\langle \alpha_1, \sharp \rangle\}$; analogously, $D_A(q') = \{\langle \alpha_1, \sharp \rangle, \langle \beta_1, \sharp \rangle\}$ and $D_A(q'') = \emptyset$. The outcome of $\langle \alpha_1, \sharp \rangle$ at q is $\{q', q''\}$. $S_A = \{\langle q, \langle \alpha_1, \sharp \rangle \rangle, \langle q', \langle \alpha_1, \sharp \rangle \rangle\}$ is a uniform strategy of A. The outcome of the S_A at q is $\{qq'', qq'q''\}$.

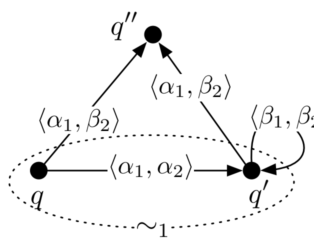

Fig. 1: An example of a CEGS.

Now we define the meaning of ATL$_u$-formulas over CEGS. The semantics we present is closely related to the semantics of Constructive Strategic Logic from [5], the only difference being that our language, unlike that of [5], does not contain epistemic operators.

Intuitively, given a CEGS $\mathfrak{M}$ and a set of states $Q \subseteq St$, we have $\mathfrak{M}, Q \models \langle\langle A \rangle\rangle \psi$ if there is a uniform strategy S_A such that ψ is satisfied by all paths in $out(Q, S_A)$. We evaluate ATL$_u$-formulas at sets of states, since when finding strategies, this will help us to find the maximal set at which the same strategy can be applied to achieve a certain outcome[6] (note that our motivation is quite different from that in [5]). Formally,

$\mathfrak{M}, Q \models \mathtt{p}$ iff $Q \subseteq \pi(\mathtt{p})$;
$\mathfrak{M}, Q \models \neg\varphi$ iff $\mathfrak{M}, Q \not\models \varphi$;
$\mathfrak{M}, Q \models \varphi_1 \wedge \varphi_2$ iff $\mathfrak{M}, Q \models \varphi_1$ and $\mathfrak{M}, Q \models \varphi_2$;
$\mathfrak{M}, Q \models \langle\langle A \rangle\rangle \psi$ iff there exists a uniform S_A such that $\mathfrak{M}, \Lambda \Vdash \psi$, for every $\Lambda \in out(Q, S_A)$;

[6] This does not increase the complexity of the strategy-finding algorithm.

$\mathfrak{M}, \Lambda \Vdash \mathsf{X}\,\varphi$ iff $\mathfrak{M}, \{\Lambda[2]\} \models \varphi$;
$\mathfrak{M}, \Lambda \Vdash \mathsf{G}\,\varphi$ iff Λ is infinite, and $\mathfrak{M}, \{\Lambda[i]\} \models \varphi$, for every $i \geq 1$;
$\mathfrak{M}, \Lambda \Vdash \varphi_1 \mathsf{U} \varphi_2$ iff $\mathfrak{M}, \{\Lambda[j]\} \models \varphi_2$, for some $j \geq 1$, and $\mathfrak{M}, \{\Lambda[i]\} \models \varphi_1$, for every $1 \leq i < j$.

Notice that the expressions $\mathsf{X}\,\varphi$, $\mathsf{G}\,\varphi$, and $\varphi_1 \,\mathsf{U}\, \varphi_2$ referred to in the last three clauses of the above definition are not ATL$_u$-formulas; they hold at paths rather that being satisfied by sets of states, as in the case of ATL$_u$-formulas.

For technical reasons, which will become clear later on, we want every ATL$_u$-formula—rather than only formulas beginning with a coalitional operator, also referred to as "strategic formulas"—to be defined in terms of strategies. To that end, we give an alternative semantics of ATL$_u$-formulas satisfying this property. We define the meaning of "non-strategic" formulas using the empty coalition $\emptyset$. As the only $\emptyset$-move is $m_\emptyset = \sharp^k$, every strategy of the empty coalition is uniform. Intuitively, a strategy $S_\emptyset$ can be thought of as the domain of $S_\emptyset$. We now redefine the meaning of non-strategic ATL$_u$-formulas in terms of strategies. Thus,

$\mathfrak{M}, Q \models \mathtt{p}$ iff there exists $S_\emptyset$ such that $Q = dom(S_\emptyset) \subseteq \pi(\mathtt{p})$;
$\mathfrak{M}, Q \models \neg\varphi$ iff there exists $S_\emptyset$ such that $Q = dom(S_\emptyset)$ and $\mathfrak{M}, dom(S_\emptyset) \not\models \varphi$;
$\mathfrak{M}, Q \models \varphi \wedge \psi$ iff there exists $S_\emptyset$ such that $Q = dom(S_\emptyset)$ and $\mathfrak{M}, dom(S_\emptyset) \models \varphi$ and $\mathfrak{M}, dom(S_\emptyset) \models \psi$;

In what follows, if CEGS $\mathfrak{M}$ is clear from the context, we write $Q \models \varphi$ instead of $\mathfrak{M}, Q \models \varphi$.

We now turn to the problem of finding uniform strategies enforcing a given ATL$_u$-formula in a given CEGS.

3 Finding Uniform Strategies in CEGSs

The main purpose of the present paper is to describe an algorithm for finding uniform strategies in CEGSs. The problem of finding such strategies can be viewed as a "constructive" model-checking problem for ATL$_u$; that is, given an ATL$_u$ formula $\langle\!\langle A \rangle\!\rangle\psi$ and a CEGS $\mathfrak{M}$, we want to find all uniform strategies of coalition A in $\mathfrak{M}$ that enforce ψ. For each such strategy S_A, we also get the set of states from which S_A can be effected and which, thus, satisfies the formula $\langle\!\langle A \rangle\!\rangle\psi$. Therefore, the problem that is solved by our algorithm is an extension of a model-checking problem for ATL$_u$.

Since uniform strategies can be viewed as programs, the extended algorithm presented in this section allows us to synthesize distributed programs achieving the outcomes that can be expressed using ATL$_u$ formulas.

Since the formula φ given as an input to our algorithm may contain a number of strategic subformulas, each requiring for its satisfaction the existence of (possibly, more than one) uniform strategy, we have to find all uniform strategies associated with each such subformula of φ, including φ itself. As with each strategic subformula of φ there might be associated several uniform strategies, some of which may contain others, we are only interested in finding maximal uniform strategies for each such subformula, i.e., the ones that can not be extended to a bigger set of states.

Definition 7. *Let S be a (not necessarily uniform) strategy of some coalition with the domain Q, and let φ be an ATL_u-formula.*

- *If φ is in the form $\langle\langle A\rangle\rangle\psi$ (i.e., φ is a strategic formula) then S is a strategy* for φ *if S is a strategy of coalition A, and $\Lambda \Vdash \psi$ for every $\Lambda \in out(Q, S)$;*
- *otherwise, S is a strategy* for φ *if S is a strategy of the coalition $\emptyset$, and $Q \models \varphi$.*

Definition 8. *Let S_A be a strategy of coalition A for an ATL_u-formula φ.*

- *S_A is a* maximal strategy for φ *if there is no other strategy S'_A for φ such that $S_A \subset S'_A$;*
- *S_A is a* maximal uniform strategy for φ *if S_A is uniform and there is no other uniform strategy S'_A for φ such that $S_A \subset S'_A$.*

We can now more precisely restate the problem our algorithm solves as follows: given a CEGS $\mathfrak{M}$ and an ATL$_u$-formula φ, find *all maximal uniform strategies S_A for every subformula of φ, including φ itself.*

The control structure of our algorithm (see appendix A) is based on the control structure of the model-checking algorithm for ATL from [2]. The major difference between the two algorithms is that our algorithm returns, for each subformula ψ of the input formula φ, the set of all maximal uniform strategies for φ, rather then just a set of states at which φ holds.

Note that, if ψ is a subformula of φ and there exist more than one uniform strategy for ψ, then the union of the domains of all those strategies is used to compute the maximal uniform strategies for φ. We denote by $[\varphi]$ the set of maximal uniform strategies for formula φ. We usually enumerate maximal uniform strategies for a formula using upper indices, as in S^1. The algorithm uses the following functions:

- *Subformulas*(φ) returns the list of subformulas of φ in the following order: if ψ is a subformula of τ then ψ precedes τ; if τ is in the form $\psi_1 \wedge \psi_2$ or $\langle\langle A\rangle\rangle\psi_1 \mathsf{U} \psi_2$ then ψ_1 precedes ψ_2.
- *Dom*(M) returns the domain of a given set M of assigned A-moves (recall Definition 4).
- $Pre(A, Q)$ returns, for a coalition A and a set of states Q, the pre-image of Q with respect to A, defined as the set $\{\langle p, m_A\rangle \mid m_A \in D_A(p)$ and $out(p, m_A) \subseteq Q\}$.
- *Uniform*(A, M), where M is the maximal (not necessarily uniform) strategy of coalition A for a given formula φ (recall Definition 8), returns all maximal uniform strategies of A for φ.

Note that there is exactly one maximal strategy M for every ATL$_u$-formula φ. If there were a strategy N for φ such that $N \not\subseteq M$ then $N \cup M$ would be a strategy for φ as well and $M \subset (N \cup M)$ would not be maximal. We now consider the issues involved in computing the set of maximal uniform strategies from such a strategy, which happens when we call the *Uniform* function.

Consider the maximal strategy M for a formula $\varphi = \langle\langle A \rangle\rangle \psi$ that is given as an input to *Uniform*. (An example for $\varphi = \langle\langle A \rangle\rangle \mathsf{F}\,\mathtt{p}$ is depicted in Fig. 2). Every maximal uniform strategy for φ must be a subset M' of M. To find every such subset, three issues must be resolved.

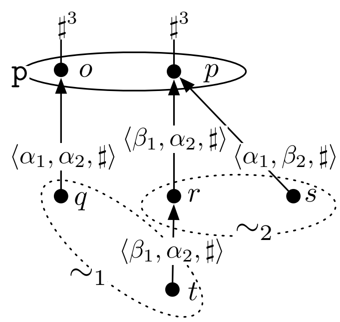

Fig. 2: The maximal strategy of coalition $A = \{1, 2\}$ for $\langle\langle A \rangle\rangle \mathsf{F}\,\mathtt{p}$. Agent 3, excluded from A, has only one action available at each state in $\{q, r, s, t\}$; thus, the arrows represent A-moves. The outcomes of $\sharp^3$ at o and p are omitted.

– *First*, two A-moves may assign different actions to an agent $a \in A$ in two states that are indistinguishable to a.

Definition 9. *Let $A \subseteq Agt$, and let $\langle q, m_A \rangle$ and $\langle q', m'_A \rangle$ be assigned A-moves. A-moves $\langle q, m_A \rangle$ and $\langle q', m'_A \rangle$ are* blocking *each other, symbolically $\langle q, m_A \rangle \leftrightsquigarrow \langle q', m'_A \rangle$, if $m_A[a] \neq m'_A[a]$ and $q \sim_a q'$, for some $a \in A$.*

In Fig. 2, $\langle q, \langle \alpha_1, \alpha_2, \sharp \rangle \rangle$ and $\langle t, \langle \beta_1, \alpha_2, \sharp \rangle \rangle$ are blocking each other; the same holds true for $\langle r, \langle \beta_1, \alpha_2, \sharp \rangle \rangle$ and $\langle s, \langle \alpha_1, \beta_2, \sharp \rangle \rangle$.

Remark 1. Note that a strategy S_A is uniform iff there is no pair of blocking assigned A-moves in S_A.

Remark 2. If there is a pair of blocking assigned A-moves in a strategy of A for φ then there may exist more then one maximal uniform strategy of A for φ.

In general, if M' is a uniform strategy, only one assigned A-move from a set of mutually blocking assigned A-moves may be included in M'.

– *Second*, assume that M is the maximal uniform strategy of A for φ and $\langle q, m_A \rangle \in M$. Consider strategy $M' = M \setminus \{\langle q^*, m^*_A \rangle \mid \langle q, m_A \rangle \leftrightsquigarrow \langle q^*, m^*_A \rangle\}$. Now, some $\langle q', m'_A \rangle \in M'$ may have become "disconnected" in M', i.e., for some state $q'' \in out(q', m'_A)$, all A-moves assigned to q'' have been thrown out of M'. Then, there may be an outcome of M' that is effected by $\langle q', m'_A \rangle$ but does not satisfy φ. We now define the notion of disconnectedness.

Definition 10. *Let M be a set of assigned A-moves and let $\langle q, m_A\rangle \in M$. We say that $\langle q, m_A\rangle$ is* disconnected in M *if there is $q' \in out(q, m_A)$ such that there is no A-move assigned to q' in M.*

As an example, in Fig. 2, assume that A-moves assigned to q and r are removed from M' because they block the A-moves assigned to t and s, respectively. Thus, $M' = \{\langle o, \sharp^3\rangle, \langle p, \sharp^3\rangle, \langle s, \langle \alpha_1, \beta_2, \sharp\rangle\rangle, \langle t, \langle \beta_1, \alpha_2, \sharp\rangle\rangle\}$. The outcome of the A-move assigned to t is r, but the only A-move assigned to r is in $M \setminus M'$. The assigned A-move $\langle t, \langle \beta_1, \alpha_2, \sharp\rangle\rangle$ is then disconnected in M'. Thus, there is a path $\Lambda = t, r$ in $out(dom(M'), M')$ that does not satisfy $\mathsf{F}\,\mathtt{p}$. Note that the A-moves assigned to o and p may be disconnected in M' as well (their outcome is not specified in the example). However, any path involving one of these assigned A-moves satisfies $\mathsf{F}\,\mathtt{p}$, because $\mathtt{p}$ holds both in o and in p. A-moves $\langle o, \sharp^3\rangle$ and $\langle p, \sharp^3\rangle$ immediately enforce $\langle\langle A\rangle\rangle \mathsf{F}\,\mathtt{p}$.

In any uniform strategy M' for $\varphi = \langle\langle A\rangle\rangle\psi$ returned by *Uniform*(A, M), every assigned A-move that is disconnected in M' must immediately enforce φ (in such a case the A-move is a singleton strategy for φ). Otherwise, there may be a path in $out(dom(M'), M')$ that does not satisfy φ.

–*Third*, assume that all disconnected assigned A-moves that do not immediately enforce φ are thrown out of M'. In our example $M' = \{\langle o, \sharp^3\rangle, \langle p, \sharp^3\rangle, \langle s, \langle \alpha_1, \beta_2, \sharp\rangle\rangle\}$. M' is now a uniform strategy for $\langle\langle A\rangle\rangle \mathsf{F}\,\mathtt{p}$, but it is not necessarily maximal. There may be another $N \subseteq M$ that is also a uniform strategy for $\langle\langle A\rangle\rangle \mathsf{F}\,\mathtt{p}$ and that contains all assigned A-moves from M' as well; $M' \subset N \subseteq M$. For example, in Fig. 2, $N = M' \cup \langle q, \langle \alpha_1, \alpha_2, \sharp\rangle\rangle$ is a superset of M' that is a uniform strategy for $\langle\langle A\rangle\rangle \mathsf{F}\,\mathtt{p}$ (even a maximal one).

We now summarize the above discussion in the following proposition.

Proposition 1. *Let $\varphi = \langle\langle A\rangle\rangle\psi$ be an ATL_u-formula and let M be the maximal strategy for φ. A set $M' \subseteq M$ is a maxitmal uniform strategy for φ iff*

1. *there is no pair of blocking assigned A-moves in M';*
2. *every disconnected assigned A-move in M' is a singleton strategy for φ;*
3. *there is no uniform strategy N for φ such that $M' \subset N \subseteq M$.*

The first condition is necessary and sufficient to ensure uniformity of a strategy, as noted in Remark 1. It is clear that the third condition amounts to maximality of M'. The second condition ensures that M' is a strategy *for* φ, as shown in the following lemma.

Lemma 1. *Let $\varphi = \langle\langle A\rangle\rangle\chi$ be an ATL_u-formula and let M be the maximal strategy for φ. A set $M' \subseteq M$ is a strategy for φ iff every assigned A-move disconnected in M' is a singleton strategy for φ.*

Proof. We have three cases to consider:

1. $\varphi = \langle\langle A\rangle\rangle \mathsf{X}\,\psi$:
   A set S of assigned A-moves is a strategy for $\langle\langle A\rangle\rangle \mathsf{X}\,\psi$ iff $\{\Lambda[2]\} \models \psi$ for every $\Lambda \in out(dom(S), S)$ iff $out(q, m_A) \models \psi$, for every $\langle q, m_A\rangle \in S$. Thus, every

$\{\langle q, m_A\rangle\} \subseteq S$, disconnected or not, is a strategy for $\langle\langle A\rangle\rangle \mathsf{X}\,\psi$. Now, since the union of strategies for φ is a strategy for φ, it follows from the above that S is a strategy for $\langle\langle A\rangle\rangle \mathsf{X}\,\psi$ iff every nonempty $S' \subseteq S$ is a strategy for $\langle\langle A\rangle\rangle \mathsf{X}\,\psi$. Therefore, M' is a strategy for $\langle\langle A\rangle\rangle \mathsf{X}\,\psi$, and we are, thus, done.

2. $\varphi = \langle\langle A\rangle\rangle \mathsf{G}\,\psi$:
   ($\Rightarrow$) M' is a strategy for $\langle\langle A\rangle\rangle \mathsf{G}\,\psi$ iff (a) every $\Lambda \in out(dom(M'), M')$ is infinite and (b) $\{\Lambda[i]\} \models \psi$ for every $i \geq 1$. It follows from (a) that no assigned A-move is disconnected in M', as the latter would imply the existence of a finite path in M'.
   ($\Leftarrow$) We argue by contraposition. Since M' is a subset of a strategy for $\langle\langle A\rangle\rangle \mathsf{G}\,\psi$, for every $\Lambda \in out(dom(M'), M')$ and every $1 \leq i \leq |\Lambda|$, we have $\{\Lambda[i]\} \models \psi$. Thus, if $M' \subseteq M$ is not a strategy for $\langle\langle A\rangle\rangle \mathsf{G}\,\psi$, this can only happen if some $\Lambda \in out(dom(M'), M')$ is finite. The last state of every such Λ must be in the outcome of some assigned A-move disconnected in M'. Since a single A-move can only be a (singleton) strategy for $\langle\langle A\rangle\rangle \mathsf{G}\,\psi$ if it "loops back", which is incompatible with being disconnected, none of these assigned A-moves can be a strategy for $\langle\langle A\rangle\rangle \mathsf{G}\,\psi$.
3. $\varphi = \langle\langle A\rangle\rangle \psi_1 \mathsf{U}\,\psi_2$:
   ($\Rightarrow$) M' is a strategy for $\langle\langle A\rangle\rangle \psi_1 \mathsf{U}\,\psi_2$ iff, for every $\Lambda \in out(dom(M'), M')$, there is $i \geq 1$ such that $\{\Lambda[i]\} \models \psi_2$ and $\{\Lambda[j]\} \models \psi_1$, for every $1 \leq j \leq i$. If $\langle q, m_A\rangle \in M'$ is an assigned A-move disconnected in M', then $\{q\} \models \psi_2$ must hold (otherwise there would be a path $\Lambda \in out(q, M')$ such that $\Lambda[i] \models \psi_2$ does not hold for any i). Thus, every assigned A-move disconnected in M' is a singleton strategy for $\langle\langle A\rangle\rangle \psi_1 \mathsf{U}\,\psi_2$.
   ($\Leftarrow$) We argue by contraposition. Since M is a strategy for $\langle\langle A\rangle\rangle \psi_1 \mathsf{U}\,\psi_2$, (a) there is no infinite $\Lambda \in out(dom(M), M)$ such that $\{\Lambda[i]\} \not\models \psi_2$ for all $i \geq 1$ and (b) there is no $q \in dom(M)$ such that $\{q\} \not\models \psi_1$ and $\{q\} \not\models \psi_2$. Since (b), if $M' \subseteq M$ is not a strategy for $\langle\langle A\rangle\rangle \psi_1 \mathsf{U}\,\psi_2$, this can only happen if, for some $\Lambda \in out(dom(M'), M')$, we have $\{\Lambda[i]\} \not\models \psi_2$, for every $i \geq 1$. Since (a), every such path must be finite. Thus, the last state q of every such path must be in the outcome of some assigned A-move $\langle q', m_A\rangle$ disconnected in M'. For this $q \in out(q', m_A)$, we have $\{q\} \not\models \psi_2$; thus, $\{\langle q', m_A\rangle\}$ cannot be a singleton strategy for $\langle\langle A\rangle\rangle \psi_1 \mathsf{U}\,\psi_2$.

□

The function *Uniform*(A, M), which ensures that the three conditions from Proposition 1 hold, is described in Alg. *Uniform*. It returns all maximal uniform strategies of A for φ, where φ is a formula enforced by M.

Uniform(A, M) works as follows. First, the problem of avoiding blocking pairs of assigned A-moves in a resulting maximal uniform strategy for φ is solved via reduction to the problem of listing all maximal cliques in the components of a graph derived from the blocking relation between assigned A-moves. (The Bron-Kerbosh algorithm [3], or its variant [7], can be used for listing of all maximal cliques.) Afterwards, we remove from every maximal uniform strategy $M' \subseteq M$ all disconnected assigned A-moves that are not singleton strategies of A for φ. Thus, only uniform strategies for φ remain. Finally, every strategy that is a subset of another strategy is removed. The remaining strategies are maximal uniform strategies for φ.

We now prove the correctness of the algorithm for finding uniform strategies:

Claim. Given an ATL$_u$-formula φ, the algorithm returns all maximal uniform strategies for every subformula of φ (including φ itself).

Proof. For the case that φ is a non-strategic formula (i.e., either φ is an atomic proposition, or $\varphi = \neg\psi$, or $\varphi = \psi_1 \wedge \psi_2$), the strategy $S_\emptyset$ returned by the algorithm assigns the $\emptyset$-move to every state that satisfies φ. Thus, the domain of $S_\emptyset$ is the set of states where φ holds, and $S_\emptyset$ is the maximal strategy for φ. As every strategy consisting of $\emptyset$-moves is uniform, $S_\emptyset$ is uniform, as desired.

For the case that $\varphi = \langle\langle A\rangle\rangle\psi$, we first show that the set of assigned A-moves passed to function *Uniform* is the maximal strategy for φ:

– $\varphi = \langle\langle A\rangle\rangle \mathsf{X}\rho$ – The arguments for function Pre are coalition A and the set Q of all states from which ρ can be enforced. Function Pre returns a strategy P containing all assigned A-moves that immediately lead to Q. Thus, every path from $out(dom(P), P)$ satisfies $\mathsf{X}\rho$, and P is a strategy for $\langle\langle A\rangle\rangle\mathsf{X}\rho$. No strategy of A for $\langle\langle A\rangle\rangle\mathsf{X}\rho$ may contain an A-move out of P and thus P is maximal.
– $\varphi = \langle\langle A\rangle\rangle\mathsf{G}\rho$ – The set T_1 passed to *Uniform* is the greatest fixed point of $F(X) = \{\langle q, m_A\rangle \in Pre(A, dom(X)) \mid \rho \textit{ can be enforced from } q\}$. Every path from $out(dom(T_1), T_1)$ satisfies $\mathsf{G}\rho$ and thus T_1 is a strategy for $\langle\langle A\rangle\rangle\mathsf{G}\rho$. Since every strategy for $\langle\langle A\rangle\rangle\mathsf{G}\rho$ must be a fixed point of F and T_1 is the greatest fixed point, T_1 is the maximal strategy for $\langle\langle A\rangle\rangle\mathsf{G}\rho$.
– $\varphi = \langle\langle A\rangle\rangle\rho_1 \mathsf{U} \rho_2$ – The set T_1 passed to *Uniform* is the least fixed point of $G(X) = \{\langle q, \sharp^k\rangle \mid \rho_2 \textit{ can be enforced from } q\} \cup \{\langle q, m_A\rangle \in Pre(A, dom(X)) \mid \rho_1 \textit{ can be enforced from } q\}$. Every path from $out(dom(T_1), T_1)$ satisfies $\rho_1 \mathsf{U} \rho_2$ and thus T_1 is a strategy for $\langle\langle A\rangle\rangle\rho_1 \mathsf{U} \rho_2$. No strategy for $\langle\langle A\rangle\rangle\rho_1 \mathsf{U} \rho_2$ may contain an assigned A-move that is not a member of the least fixed point of G and thus T_1 is the maximal strategy for $\langle\langle A\rangle\rangle\rho_1 \mathsf{U} \rho_2$.

Second, Proposition 1 imposes three necessary and sufficient conditions on a subset of the maximal strategy for a given $\varphi = \langle\langle A\rangle\rangle\psi$ to be a maximal uniform strategy for φ. Since the arguments passed to function *Uniform* are coalition A and the maximal strategy M for φ, we will show that *Uniform*(A, M) returns all subsets M' of M that satisfy the conditions from Proposition 1:

1. The first condition prohibits the presence in M' of pairs of blocking assigned A-moves. Consider a graph $G = \langle M, B\rangle$ where $(\langle q, m_A\rangle, \langle q', m'_A\rangle) \in B$ iff $\langle q, m_A\rangle$ and $\langle q', m'_A\rangle$ are blocking each other. A subset M' does not contain any pair of assigned A-moves blocking each other iff M' is an independent set in G, i.e., a clique (a complete subgraph) in the complement graph $\overline{G}$. Since we want only maximal uniform strategies, we select only maximal independent sets in G. Since no pair of assigned A-moves from two different disconnected components in G is blocking each other, the set of all maximal independent sets in G is $\{\bigcup_{i=1,\dots,n} I_i \mid I_i \textit{ is a maximal independent set in } C_i\}$

Function Uniform(A, M)

```
Input: a coalition A, the maximal strategy M of A for a formula φ
Output: all maximal uniform strategies of A for φ
begin
    build a graph G = ⟨M, B⟩ where (⟨q, m_A⟩, ⟨q′, m′_A⟩) ∈ B iff ⟨q, m_A⟩ and
    ⟨q′, m′_A⟩ are blocking;
    find all components C_1, .., C_m of G;
    foreach component C_i of G do
        /* find all maximal independent sets I_i^1, .., I_i^n in C_i          */
        build the complement graph C̄_i of C_i;
        𝕀_i := {all maximal cliques I_i^1, .., I_i^l of C̄_i} ; // Bron-Kerbosh algorithm
    // generate all combinations of cliques, one clique per component
    𝕊 := {∅};
    foreach component C_i of G do
        𝕊′ := {∅};
        foreach S_j ∈ 𝕊 do
            foreach clique I_i^k ∈ 𝕀_i do
                𝕊′ := 𝕊′ ∪ {I_i^k ∪ S_j};
        𝕊 := 𝕊′;
    if φ is not in the form ⟨⟨A⟩⟩Xψ then
        // remove all disconnected assigned A-moves
        foreach S_i ∈ 𝕊 do
            foreach ⟨q, m_A⟩ ∈ S_i do
                foreach q′ ∈ out(q, m_A) do
                    if there is no A-move assigned to q′ in S_i then
                        /* remove recursively all assigned A-moves
                           potentially leading to q′                      */
                        R := {⟨q, m_A⟩};
                        while R ≠ ∅ do
                            S_i := S_i \ R;
                            R′ := ∅;
                            foreach ⟨q, m_A⟩ ∈ R do
                                foreach ⟨q′, m′_A⟩ ∈ S_i do
                                    if q ∈ out(q′, m′_A) then R′ := R′ ∪ ⟨q′, m′_A⟩ ;
                            R := R′;
        // remove all non-maximal strategies
        foreach S_i ∈ 𝕊 do
            foreach S_j ∈ 𝕊 do
                if S_i ⊆ S_j then 𝕊 := 𝕊 \ S_i;
    return 𝕊;
```

where $C_1, \ldots, C_n$ are all disconnected components of G. Function *Uniform* finds all combinations of maximal independent sets from every disconnected component of G, one set per component, thus producing all uniform strategies in M.

2. The second condition excludes from M' every assigned A-move that is disconnected in M' and also is not a singleton strategy for φ. If $\varphi = \langle\langle A \rangle\rangle \mathsf{X} \rho$, the maximal strategy M consists only of the assigned A-moves disconnected in M and each of them is a singleton strategy for φ. Thus, every uniform strategy $M' \subseteq M$ is a uniform strategy for φ. If $\varphi = \langle\langle A \rangle\rangle \mathsf{G} \rho$ or $\varphi = \langle\langle A \rangle\rangle \rho_1 \mathsf{U} \rho_2$, function *Uniform* finds in every uniform strategy $M' \subseteq M$ all disconnected assigned A-moves and removes them recursively. No assigned A-move M' that is a singleton strategy for φ is disconnected. Thus, every uniform strategy M' in M after the removal of all disconnected assigned A-moves becomes a uniform strategy for φ.
3. Function *Uniform* drops every uniform strategy for φ that is a subset of another uniform strategy for φ. Thus, the results of function *Uniform* are maximal.

Since our algorithm passes to function *Uniform* the maximal strategy M for φ and function *Uniform* finds all subsets of M that satisfy the conditions from Proposition 1, we have that the algorithm returns for every ATL_u subformula φ all maximal uniform strategies for φ. □

4 Example

To demonstrate finding uniform strategies, we use the example of the SOSEWIN sensor network for detection of earthquakes [4]. This network consists of a fixed number of nodes. Every node is connected to two other nodes. A protocol for election of a leader is necessary to fulfill the task of the network. The aim of the protocol is to establish exactly one leader in the cluster. For the sake of simplicity, we fix the size of the cluster to three nodes and further limit the information available to each node by reducing the size of their neighborhood.

We represent the cluster of 3 nodes as a one-dimensional cellular automaton – a finite row of cells. Each cell may have one of two colors—black when the cell is designated as the leader and white otherwise—and is connected to the cell on its left and right side. The neighborhood of each cell consists of the cell on its left, that is, each cell knows its own color and the color of the cell on its left. The neighbor of the leftmost cell is the rightmost cell. In each step of the system each cell synchronously observes the colors in its neighborhood and decides to either keep it current color or swap it, according to an applicable rule. Given a cell and the colors of the cell's neighborhood the rule defines the next cell's color. Thus, a set of rules for all cells can be seen as a uniform strategy of the cellular automaton.

We want to find all maximal uniform strategies of the cellular automaton consisting of three cells for the following property: In one step a state of the cellular automaton is reached, where exactly one cell is black and the other two are white. Moreover, if the cellular automaton is at such state, the cells may not change their colors.

First, we specify a CEGS representing our system:

- Each cell in the cellular automaton is one agent. We denote the leftmost cell by c_1, the middle cell by c_2 and the rightmost cell by c_3 . For the sake of simplicity, we do not consider the environment to be an agent. Thus, the set of agents is $Agt = \{c_1, c_2, c_3\}$. Since all cells are involved in enforcing the property, they are all included in the coalition A.
- A state of the system is given by colors of all three cells. We denote the state where the cell c_1 is black and the cell c_2 and c_3 is white by ■□□. The set of all states is $St = \{$□□□, □□■, □■□, □■■, ■□□, ■□■, ■■□, ■■■$\}$.
- There is a single atomic proposition p that holds in those states where exactly one cell is black. That is, $\pi(\mathtt{p}) = \{$□□■, □■□, ■□□$\}$.
- Each cell can either keep its color (denoted by $-$) or swap it (denoted by s). Thus, the set of actions is $Act = \{-, s\}$.
- Each cell has both actions available at all states except those where proposition p holds. In states where p holds only action – is available to every cell (this constraint is a part of the property that the system should enforce). Thus, for every cell $c \in Agt$ we have $d(c, q) = \{-\}$ for every state $q \in \pi(\mathtt{p})$ and $d(c, q') = \{-, s\}$ for every state $q' \notin \pi(\mathtt{p})$.
- The transition function o can be derived from the description of the action. For example, $o($■□■$, \langle -, -, s\rangle) =$ ■□□.
- Since each cell knows only its own color and the color of the cell on its left, the two states that differ in the color of the cell on its right are indistinguishable for the cell.

We can now express the desired property formally: $\varphi = \langle\langle A\rangle\rangle \mathsf{X}\mathtt{p}$. The algorithm (see appendix A) calls function $Pre(A, \pi(\mathtt{p}))$ to find the maximal strategy M of A for φ. M consists of assigned A-moves represented by nodes in Fig. 3. The domain of M consists of all system states.

Function $Uniform(A, M)$ constructs the graph G for the blocking relation on M (Fig. 3). There is only one component in G and one maximal independent set in G is depicted in grey in the figure. In case of *next* operator every assigned A-move in the maximal strategy for φ is a singleton strategy for φ. Thus, every maximal independent set in G represents one maximal uniform strategy φ and no check for disconnected assigned A-moves that are not singleton strategies is necessary.

There is a number of maximal uniform strategies for φ, but none of them consists of A-moves assigned to every state of the system. That is, there is no uniform S_A for φ such that $dom(S_A) = St$. There are four uniform strategies for φ that contain A-moves assigned to every state from $\pi(\mathtt{p})$. From these four strategies only the one depicted on the figure consists of such assigned A-moves that to every cell the same action is prescribed for the same combination of its color and the color of its left neighbor. From this strategy one set of rules can be derived that can be performed by every cell, namely: swap the color if both

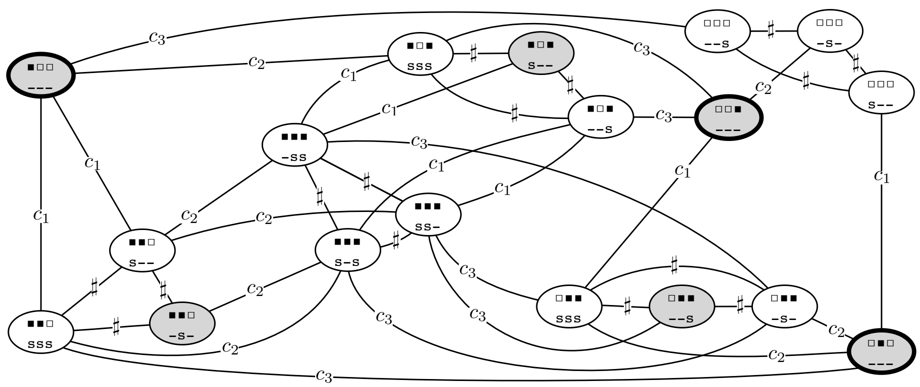

Fig. 3: The maximal strategy M for $\varphi = \langle\!\langle A \rangle\!\rangle \mathsf{X}\mathtt{p}$: A node represents an assigned A-move. An edge connects assigned A-moves that block each other and the label of the edge denotes the agent to that the connected states are indistinguishable. ♯ denotes all agents. Proposition $\mathtt{p}$ is true at the states from the bold nodes. One maximal uniform strategy for φ (a maximal independent set) consists of the grey nodes.

colors are black, keep the color otherwise. If every cell follows this set of rules, the system reaches a state from $\pi(\mathtt{p})$ in one step from all states except ■■■ and □□□ (while respecting the constraint on the behavior at states from $\pi(\mathtt{p})$).

5 The Complexity of Finding Uniform Strategies

In the following, $n = |St|$ denotes the number of states of a CEGS, $k = |Agt|$ denotes the number of all agents, $a = |Act|$ denotes the number of all actions and $m \leq n \cdot a^k$ denotes the number of transitions.

For each subformula φ of a given ATL$_u$-formula, the algorithm (see appendix A) first finds the maximal strategy M for φ. In the worst case, this involves computing the least or the greatest fixed point of *Pre*, which finds a pre-image for a given set of states and a coalition. When doing this, the membership of each assigned A-move $\langle p, m_A \rangle$ in M is considered at most once. The number of assigned A-moves is at most $m \leq n \cdot a^k$. To include $\langle p, m_A \rangle$ in M, every outcome of m_A at p must be checked, that is, at most a^k checks must be done. Thus, finding the maximal strategy M takes $O(n \cdot a^{2k})$ steps.

From the maximal strategy M for φ all maximal uniform strategies for φ must be extracted by function *Uniform*.

The first step of *Uniform(A, M)* is the construction of the graph $G = \langle M, B \rangle$ representing the blocking relation on the assigned A-moves from M. The vertices of G are the assigned A-moves from M and the edges of G are those pairs from M that are blocking each other. The size of M is at most the number of all transitions in the CEGS, i.e., $|M| \leq n \cdot a^k$. To decide whether a pair of vertices is connected by an edge, the actions from the assigned A-moves must be compared for each agent from A that cannot distinguish between the states to that the A-moves are assigned. This involves at most k comparisons for a pair of vertices. Since the presence of edges must be decided for every pair of vertices, the construction of G takes $O(k \cdot |M|^2) \leq O(k \cdot n^2 \cdot a^{2k})$ steps. The disconnected components of G can be identified during the construction for no extra price.

Next, each disconnected component C of graph G must be turned into its complement graph $\overline{C}$. That is, every pair of vertices in $\overline{C}$ must be included or excluded from the set of edges depending on their presence in the set of edges in C. In the worst case, G consists of only one component and the construction of $\overline{G}$ requires $O(|M|^2) \leq O(n^2 \cdot a^{2k})$ steps.

Next, all maximal cliques are found in every complement graph $\overline{C}$. The Bron-Kerbosh algorithm solving this task has the worst-time complexity $O(3^{v/3})$ for a graph with v vertices [9] and is optimal since there are at most $3^{v/3}$ maximal cliques in such graph [8]. If G consists of j disconnected components $C_1, \ldots, C_j$ with $c_1, \ldots, c_j$ vertices, respectively, then the task requires $O(\sum_{i=1}^{j} 3^{c_i/3}) \leq O(3^{n \cdot a^k/3})$ steps.

Next, all combinations of maximal cliques – one for each disconnected complement component – are generated to provide all maximal uniform strategies within M. Since there is at most $3^{c_i/3}$ maximal cliques in a complement component $\overline{C_i}$, we need up to $O(\prod_{i=1}^{j} 3^{c_i/3}) = O(3^{\sum_{i=1}^{j} c_i/3}) \leq O(3^{n \cdot a^k/3})$ steps to produce all maximal uniform strategies within M.

Next, in some cases all disconnected assigned A-moves must be removed from the strategies. To find out whether an assigned A-move is disconnected in a strategy, each its outcome must be checked. Each assigned A-move has at most $a^{k-|A|}$ states in the outcome and each uniform strategy has at most n assigned A-moves. Since there is up to $O(3^{n \cdot a^k/3})$ maximal uniform strategies, it takes up to $O(n \cdot a^{k-1} \cdot 3^{n \cdot a^k/3})$ steps to remove all disconnected assigned A-moves.

Lastly, every strategy that is a subset of another strategy must be removed so that only maximal uniform strategies for φ remain. At worst case, every pair of strategies must be compared (at most $3^{2 \cdot n \cdot a^k/3}$ comparisons) and each comparison may involve checks for each pair of assigned A-moves (n^2 checks). Thus, removing the non-maximal strategies may take at most $O(n^2 \cdot 3^{2 \cdot n \cdot a^k/3})$ steps.

For each subformula φ of a given ATL_u formula that contains a coalitional operator the maximal strategy M must be found (at most $O(n \cdot a^{2k})$ steps) and all maximal uniform strategies of the given coalition for φ must be extracted (at most $O(n^2 \cdot 3^{2 \cdot n \cdot a^k/3})$ steps). If we denote by l the number of the strategic subformulas in a given ATL_u formula then the worst-case complexity of finding uniform strategies is $O(l \cdot n^2 \cdot 3^{2 \cdot n \cdot a^k/3})$.

6 Conclusion

We presented an algorithm for finding uniform strategies for multi-agent systems with agents with incomplete information. Given a strategic formula $\varphi = \langle\langle A \rangle\rangle \psi$, we find all maximal uniform strategies that agents in A can use to enforce ψ, rather then simply the set of states satisfying φ. The formulas we consider have the syntax of ATL, but their semantics is taken from CSL ([5]). The algorithm has double-exponential worst-case complexity, since it uses. as a subroutine, a procedure for finding all cliques in a graph—problem that is known to be NP-complete—and the structures we are dealing with are exponential in the size of the input. Further research will be focused on reducing the complexity of the problem by using a model with an implicit representation of the incomplete information, e.g., modular interpreted systems [6].

Acknowledgements. This work is supported by the grant SUA 08/034 from the National Research Foundation of South Africa and the Deutsches Zentrum für Luft- und Raumfahrt, as well as by the grant from the Deutsche Forschungsgemeinschaft, Graduiertenkolleg METRIK (GRK 1324/1).

References

1. SAFER - Seismic eArly warning For EuRope. `http://www.saferproject.net/`, 2010.
2. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.
3. Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.
4. Joachim Fischer, Frank Kühnlenz, Klaus Ahrens, and Ingmar Eveslage. Model-based development of self-organizing earthquake early warning systems. *Proceedings of the 6th Vienna International Conference on Mathematical Modelling*, 2009.
5. Wojciech Jamroga and Thomas Ågotnes. Constructive knowledge: what agents can achieve under imperfect information. *Journal of Applied Non-classical Logics*, 17(4):423–475, 2007.
6. Wojciech Jamroga and Thomas Ågotnes. Modular interpreted systems. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8. ACM, 2007.
7. Ina Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250(1-2):1 – 30, 2001.
8. J. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, 1965.
9. Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28 – 42, 2006.
10. Wiebe van der Hoek and Michael Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

A Algorithm for finding uniform strategies

---

foreach φ' ***in*** *Subformulas*(φ) **do**

  $[\varphi'] := \emptyset$;

  case $\varphi' = \mathtt{p}$:

    $S^1 := \{ \langle q, \sharp^k \rangle \mid q \in \pi(\mathtt{p})\}$;

    $[\varphi'] := \{S^1\}$;

  case $\varphi' = \neg\psi$:

    $S^1 := \{ \langle q, \sharp^k \rangle \mid \nexists S \in [\psi] : q \in Dom(S)\}$;

    $[\varphi'] := \{S^1\}$;

  case $\varphi' = \psi_1 \wedge \psi_2$:

    $S^1 := \emptyset$;

    foreach $S^i \in [\psi_1]$ **do**

      foreach $S^j \in [\psi_2]$ **do**

        $S^1 := S^1 \cup \{\langle q, \sharp^k \rangle \mid q \in Dom(S^i) \cap Dom(S^j)\}$;

        $[\varphi'] := \{S^1\}$;

  case $\varphi' = \langle\!\langle A \rangle\!\rangle \mathsf{X}\, \psi$:

    $S := \bigcup_{S^i \in [\psi]} S^i$;

    $P := Pre(A, Dom(S))$;

    $i := 1$;

    foreach $S \in$ *Uniform*(A, P) **do**

      $S^i := S$;

      $[\varphi'] := [\varphi'] \cup \{S^i\}$;

      $i := i + 1$;

  case $\varphi' = \langle\!\langle A \rangle\!\rangle \mathsf{G}\, \psi$:

    $S := \bigcup_{S^i \in [\psi]} S^i$;

    $T_1 := \{\langle q, \sharp^k \rangle \mid q \in Dom(S)\};\ T_2 := \{\langle q, \sharp^k \rangle \mid q \in St\}$;

    while $Dom(T_2) \nsubseteq Dom(T_1)$ **do**

      $T_2 := T_1;\ T_1 := Pre(A, Dom(T_1))$;

      $T_1 := T_1 \setminus \{\langle q, m \rangle \in T_1 \mid q \notin Dom(S)\}$;

    $i := 1$;

    foreach $S \in$ *Uniform*(A, T_1) **do**

      $S^i := S$;

      $[\varphi'] := [\varphi'] \cup \{S^i\}$;

      $i := i + 1$;

  case $\varphi' = \langle\!\langle A \rangle\!\rangle\, \psi_1 \mathsf{U}\, \psi_2$:

    $S^1 := \bigcup_{S^i \in [\psi_1]} S^i$;

    $S^2 := \bigcup_{S^i \in [\psi_2]} S^i$;

    $T_1 := \emptyset;\ T_2 := \{\langle q, \sharp^k \rangle \mid q \in Dom(S^2)\}$;

    while $Dom(T_2) \nsubseteq Dom(T_1)$ **do**

      $T_1 := T_1 \cup T_2;\ T_2 := Pre(A, Dom(T_1))$;

      $T_2 := T_2 \setminus \{\langle q, m \rangle \in T_2 \mid q \notin Dom(S^1)\}$;

    $i := 1$;

    foreach $S \in$ *Uniform*(A, T_1) **do**

      $S^i := S$;

      $[\varphi'] := [\varphi'] \cup \{S^i\}$;

      $i := i + 1$;

  return $[\varphi']$;
