Multi-Valued Verification of Strategic Ability

Wojciech Jamroga
Institute of Computer Science,
Polish Academy of Sciences
ul. Jana Kazimierza 5
01-248 Warsaw, Poland
wojciech.jamroga@ipipan.waw.pl

Beata Konikowska
Institute of Computer Science,
Polish Academy of Sciences
ul. Jana Kazimierza 5
01-248 Warsaw, Poland
beata.konikowska@ipipan.waw.pl

Wojciech Penczek
Institute of Computer Science,
Polish Academy of Sciences
ul. Jana Kazimierza 5
01-248 Warsaw, Poland
wojciech.penczek@ipipan.waw.pl

ABSTRACT

Some multi-agent scenarios call for the possibility of evaluating specifications in a richer domain of truth values. Examples include runtime monitoring of a temporal property over a growing prefix of an infinite path, inconsistency analysis in distributed databases, and verification methods that use incomplete anytime algorithms, such as bounded model checking. In this paper, we present multi-valued ATL* (mv-ATL∗), an expressive logic to specify strategic abilities in multi-agent systems. We show that our general method for model-independent translation from multi-valued to two-valued model checking cannot be directly extended to mv-ATL∗. We also propose two ways of overcoming the problem. Firstly, we identify constraints on mv-ATL∗ formulas for which the model-independent translation can be suitably adapted. Secondly, we present a model-dependent reduction that can be applied to all formulas of mv-ATL∗. We show that, in all cases, the complexity of verification increases only polynomially when new truth values are added to the evaluation domain. We also consider several examples that show possible applications of mv-ATL∗ and motivate its use for model checking MAS.

Categories and Subject Descriptors
F.3.1 [Specifying, Verifying, and Reasoning about Programs]: Specification techniques; D.2.4 [Software/Program Verification]: Model checking; I.2.4 [Knowledge Representation Formalisms and Methods]: Modal logic, Temporal logic

General Terms
Verification, Theory

Keywords
model checking, multi-valued verification, alternating-time logic

1. INTRODUCTION

The alternating-time temporal logic ATL* and its less expressive variant ATL [3] are probably the most popular logics that allow for reasoning about agents’ abilities in strategic encounters. ATL∗ combines features of temporal logic and basic game theory, encapsulated in the main language construct of the logic, ⟨⟨A⟩⟩γ, which can be read as “the group of agents A has a strategy to enforce γ”. Property γ can include operators X (“next”), G (“always”), F (“eventually”) and/or U (“until”). Much research on ATL∗ has focused on the way it can be used for verification of multi-agent systems, including theoretical studies on the complexity of model checking, as well as practical verification algorithms.

Typically, model checking is a yes/no problem. However, it is sometimes convenient to consider the output of verification in a richer domain of values. This can be due to two reasons. First, our model of the system can be only partially conclusive with respect to properties of the system. A good example is runtime monitoring of temporal properties, where a temporal formula (interpreted with an infinite time horizon in mind) is checked on a finite but constantly growing sequence of events, observed so far. Consider for instance specification Fp. If p has already occurred then the formula is clearly true whatever happens next. What if it has not occurred? Then, the formula may still turn out true (because p may occur in subsequent steps), but it can also turn out false; effectively, the truth value is unknown in our current model. Likewise, formula Gp can be only proved false in the course of monitoring, or the model is inconclusive regarding its value. Indeed, well known approaches to runtime monitoring use multi-valued interpretation of temporal formulas in finite runs [7, 8].

Secondly, even if the model is complete and faithful, the verification procedure can be only partially conclusive. For instance, in bounded model checking [10, 47], a full transition system is given but the formula is checked on runs of length at most n. Now, again, Fp is clearly true if we find p to occur on every path in up to n steps. Otherwise, the output is inconclusive (because p might or might not occur in subsequent steps). The case of Gp is analogous. A more sophisticated motivating example is presented below.

EXAMPLE I (ANYTIME STATE ABSTRACTION). Let M be a transition system that models the behavior of k agents operating in a common environment (e.g., autonomous vehicles at a busy intersection). The following ATL formula specifies a sensible requirement on the system: ⟨⟨m⟩⟩Gsafea. Suppose further that the state of each agent is determined by n binary variables. It is easy to see that the system has O(2kn) global states, which makes straightforward model checking, and even explicit generation of the whole model problematic for large values of k.

State abstraction is a technique that consists in “clustering” global states into abstract states according to an equivalence relation R. For example, qRq′ if q, q′ coincide on the variables of agent a. Such states will end up in the same abstract state [q]R. However, it can happen that the value of a proposition (e.g., safea) varies across [q]R. Then, the most reasonable value for safea in the ab-
strict state is “conflicting” or “inconsistent”. Moreover, if we use
an anytime algorithm, it may be stopped before it computes the
truth of safeₐ in some of the new states. Then, the value of the
proposition can be set to “uncomputed” or “undecided”. Conse-
quently, we will have to manipulate at least four truth values when
model checking the formula ⟨⟨a⟩⟩Gsₐₑₐ in the abstract model.

In this paper, we study model checking for a multi-valued variant
of ATL∗ over arbitrary lattices of logical values. We call the new
logic mv-ATL∗₂. Our approach extends previous work on temporal
model checking: similarly to [38, 39], we do not propose dedicated
algorithms for mv-ATL∗₂. Instead, we look for general efficient
translations from the multi-valued case to the 2-valued case. We (i)
prove that no model-independent translation exists for the whole
language of mv-ATL∗₂, (ii) identify a broad subclass of formulas
for which such a translation can be obtained, and (iii) propose a
recursive model-dependent translation that works for all instances
of the problem. We also show that all the results easily extend to
verification of strategies under imperfect information.

Related work. Multi-valued interpretation of modal formulas has
been used in multiple approaches to verification. The main idea
was proposed by Fitting [22, 23] already 25 years ago. In the
2000s, a number of works adapted it to verification of distributed
and multi-agent systems. A variant of CTL∗ for models over finite
quasi-booleans was proposed in [38], together with a general
translation scheme that reduced multi-valued model checking
of CTL∗ specifications to the standard 2-valued case. This was later
extended to multi-valued modal μ-calculus [26, 12, 51, 46], and to
multi-valued modal μ-calculus with knowledge [39]. Our paper
follows this line of work, and extends the techniques to strategic
operators of ATL∗.¹ We also enrich the language with the “compar-
ison” operators ≲ and ≳, which provide: (i) the notions of material
implication and biconditional, useful in specifying general proper-
ties of multi-valued models; (ii) a way of model checking “thresh-
old properties” analogous to probabilistic temporal logics behind
PRISM [40]. As it turns out, the new operators require non-trivial
treatment, significantly different from [38, 26, 12, 51, 39].

Model checking methods for the special case of 3-valued tem-
poral logics were discussed in [25, 28, 30]. Related approaches
include runtime verification, which often uses 3-valued [7] or 4-
valued interpretation [8, 26] of temporal formulas. Moreover, 4-
valued semantics has been used to evaluate database queries [44,
45]. 3-valued semantics of strategic abilities was considered in [6]
for alternating μ-calculus and, recently, in [43] for ATL. In both
cases the main aim was to verify abstractions of multi-agent sys-
tems. Note that, while the agenda of our paper comes close to that
of [43], our semantics differs from [43] even in the 3-valued case.

A quite different but related strand of research concerns real-
valued logics over probabilistic models for temporal [20, 41, 32]
and strategic specifications [33]. We also mention the research on
probabilistic model checking of temporal and strategic logics [29,
5, 40, 14, 18] that evaluates specifications in the 2-valued domain
but recognizes different degrees of success and the need to aggre-
gate them over available strategies and possible paths.

2. PRELIMINARIES

We begin by presenting the basics of alternating-time temporal
logic and quasi-booleans of truth values.

¹Note that, despite having “games” in the title, [51] was not con-
cerned with strategic specifications. “Games” were used there only
in the technical sense, to define the semantics of μ-calculus. How-
ever, [4] deals with game applications of [51].

Figure 1: Autonomous vehicles at the intersection: model M₁

2.1 What Agents Can Achieve

Alternating-time temporal logic [3] generalizes branching-time
temporal logic CTL* by replacing path quantifiers E, A with strate-
gic modalities ⟨⟨A⟩⟩. Informally, ⟨⟨A⟩⟩γ says that a group of agents
A has a collective strategy to enforce temporal property γ. Simi-
larly to CTL∗ and CTL, we consider two syntactic variants of the
alternating-time logic, namely ATL∗ and ATL.

Let A be a finite set of agents, and Prop a countable set of
atomic propositions. The language of ATL∗ is defined as follows:

ϕ ::= p | ¬ϕ | ϕ ∧ ϕ | ⟨⟨A⟩⟩γ, γ ::= ⊤ | ¬γ | γ ∧ γ | X γ | γ U γ.

where A ⊆ A and p ∈ Prop. Derived boolean connectives
and constants (∨, ∨, ⊥) are defined as usual. “Sometimes”, “weak
until”, and “always from now on” are defined as Fγ ≡ T U γ,
γ₁ W γ₂ ≡ ¬((¬γ₁) U (¬γ₁ ∧ ¬γ₂)), and Gγ ≡ γ W ⊥. Also, we
can use ⟨⟨A⟩⟩γ ≡ ¬⟨⟨A⟩⟩γ to express that, for each strategy of A,
property γ fails on some paths.²

ATL (without “star”) is the syntactic variant in which strategic
and temporal operators are combined into compound modalities:

ϕ ::= p | ¬ϕ | ϕ ∧ ϕ | ⟨⟨A⟩⟩X ϕ | ⟨⟨A⟩⟩ϕ U ⟨⟨A⟩⟩ϕ | ⟨⟨A⟩⟩ϕ W ϕ.

Semantics. The semantics of ATL∗ is defined over a variant of
synchronous multi-agent transition systems.

Definition 1 (CGS). A concurrent game structure (CGS) is a
tuple M = ⟨A, St, Act, d, t, Prop, V⟩ which includes nonempty
finite sets of: agents A = {1, ..., k}, states St, actions Act,
atomic propositions Prop, and a propositional valuation V : St → 2Prop. The function d : A × St → 2Act defines availability of
actions. The (deterministic) transition function t assigns a successor
state q′ = t(q, α₁, ..., αₖ) to each state q ∈ St and any tuple of
actions α₁ ∈ d(i, q) that can be executed by A in q.

A pointed CGS is a pair ⟨M, q₀⟩ consisting of a concurrent game
structure M and an initial state q₀ in M.

Example 2 (Driving agents). Consider an intersection with k
autonomous vehicles around it. Each vehicle is modeled

²In some papers, [⟨⟨A⟩⟩γ ≡ ¬⟨⟨A⟩⟩γ] is used as the dual strategic
operator. We use ⟨⟨A⟩⟩γ instead for technical convenience, to avoid
negations of path subformulas when transforming to NNF.
as a separate agent, whose local state is characterized by either the proposition \texttt{out}, (when the vehicle is outside the intersection) or \texttt{in}, (when the vehicle is inside it). The available actions are: \texttt{in} ("drive in" or "stay in", depending on the current state) and \texttt{out} ("drive out" or "stay out"). When both agents are in and decide to leave at the same time, a collision occurs (collision), and the guilty party – or parties – have to pay a penalty (penalty).

Figure 1 presents a pointed CGS modeling the scenario for \(k = 2\). The combinations of actions that are not displayed in the graph do not change the state of the system. We assume that the responsibility for the accident is determined by a complex constraint satisfaction problem, involving variables like the trajectory and speed of the vehicles, their technical condition etc. To simplify the graph, we only include the possibility of agent 1 being guilty.

A \textit{path} \(\lambda = q_0q_1q_2\ldots\) in a CGS is an infinite sequence of states such that there is a transition between each \(q_i, q_{i+1}\). \(\lambda[i, \infty]\) denotes the \(i\)th position on \(\lambda\) and \(\lambda[i, \infty]\) the suffix of \(\lambda\) starting with \(i\).

A \textit{perfect recall strategy} (or IR-strategy) for agent \(a\) is a function \(s_{a} : S T^{+} \rightarrow \text{Act} \) such that \(s_{a}(q_0q_1\ldots q_n) \in d(a, q_n)\). A \textit{collective strategy} for agents \(A = \{a_1, \ldots, a_r\}\) is a tuple of individual strategies \(s_{a} = (s_{a_1}, \ldots, s_{a_r})\). The set of such strategies is denoted by \(\Sigma_{A}\). The "outcome" function \(\text{out}(q, s_{A})\) returns the set of all paths that can occur when agents \(A\) execute strategy \(s_{A}\) from state \(q\) onward. The semantics of ATL* is defined as follows:

\[
M, q \models p \text{ iff } q \in V(p), \text{ for } p \in \text{Prop};
\]

\[
M, q \models \neg \varphi \text{ iff } M, q \not\models \varphi;
\]

\[
M, q \models \varphi_1 \land \varphi_2 \text{ iff } M, q \models \varphi_1 \text{ and } M, q \models \varphi_2;
\]

\[
M, q \models \langle A \rangle \gamma \text{ iff there is a strategy } s_{A} \in \Sigma_{A}\text{ such that, for each path } \lambda \in \text{out}(q, s_{A}), \text{ we have } M, \lambda \models \gamma.
\]

\[
M, \lambda \models \varphi \text{ iff } M, \lambda[0] \models \varphi;
\]

\[
M, \lambda \models \neg \varphi \text{ iff } M, \lambda \not\models \varphi;
\]

\[
M, \lambda \models \gamma_1 \land \gamma_2 \text{ iff } M, \lambda \models \gamma_1 \text{ and } M, \lambda \models \gamma_2;
\]

\[
M, \lambda \models \chi \gamma \text{ iff } M, \lambda[1, \infty] \models \gamma; \text{ and }
\]

\[
M, \lambda \models \gamma_1 \cup \gamma_2 \text{ iff there is an } i \in \mathbb{N}_0 \text{ such that } M, \lambda[i, \infty] \models \gamma_2 \text{ and } M, \lambda[j, \infty] \models \gamma_1 \text{ for all } 0 \leq j < i.
\]

**Example 3. (Driving agents, CTD.)** For model \(M_1\), we have \(M_1, q_{oo} \not\models \langle 1 \rangle \text{Prop} - \text{penalty} : \text{agent 1 can avoid the penalty forever (the obvious strategy is to never enter the crossroads). On the other hand, the agent cannot make sure it will get the penalty even if it wants to: } M_1, q_{oo} \not\models \langle 1 \rangle \text{Fpenalty} + \text{Prop}. This can only be ensured if the agents cooperate: } M_1, q_{oo} \models \langle 1 \rangle \text{FProp} \land \langle 2 \rangle \text{Fpenalty}. Moreover, \(M_1, q_{oo} \models \langle 1 \rangle \text{Fin}1 \land \langle 2 \rangle \text{Fin}2\); each agent is able to enter the intersection. Still, it cannot successfully drive through the crossroads on its own (e.g., \(M_1, q_{oo} \not\models \langle 1 \rangle \text{F} (\text{Fin}1 \lor \text{Fin}2))\).

**Negation Normal Form.** Formulas of ATL* can be equivalently transformed so that negation is only applied to atomic propositions. For this, the set of primitive operators must be extended as follows:

\[
\varphi ::= p \mid \neg p \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \langle A \rangle \gamma \mid \langle \overline{A} \rangle \gamma, \gamma ::= \varphi \mid \gamma \land \gamma \mid \gamma \lor \gamma \mid X \gamma \mid U \gamma \mid \gamma W \gamma.
\]

The Negation Normal Form for ATL is constructed analogously.

### 2.2 Quasi-Boolean Lattices

We will use quasi-boolean and De Morgan algebras [11] to interpret formulas of our multi-valued logics.
It is well known [19] that every element of a finite distributive lattice can be uniquely decomposed into the join of all join-irreducible elements in its downward closure, i.e.

\[ x = \bigcup (\mathcal{I}(L) \cap \downarrow x) \]  

(1)

**Example 5.** The join-irreducible elements of the DM algebras in Figure 2 are marked with black dots. In the lattice 2 + 4, the element S, which is not join-irreducible, can be decomposed into the join of the join-irreducible elements DK and DC.

The characterization (1) is used to define translations from multi-valued to standard model checking using the following theorem.

**Theorem 1** ([39]). Let L be a finite DM algebra, and let \( l \in \mathcal{I}(L) \). Then the function \( f_l : L \rightarrow \{ \top, \bot \} \) defined by \( f_l(\uparrow l) = \top \), \( f_l(\downarrow l) = \bot \) preserves arbitrary bounds.

**Remark 1.** The above does not hold for lattices which are not distributive. To see this, consider the element \( 1 \) of M5, which is join-irreducible. However, \( f_1 \) does not preserve join, as \( f_1(1_2 \cup 1_3) = f_1(\top) = \top \) whereas \( f_1(1_2) \cup f_1(1_3) = \bot \cup \bot = \bot \).

### 3. MULTI-VALUED STRATEGIC LOGIC

In this section we extend the syntax and semantics of ATL* to allow for multi-valued reasoning.

#### 3.1 Multi-Valued ATL*

**Domain of interpretation.** We propose a variant of ATL* where formulas are interpreted in a quasi-boolean algebra \( L = (L, \leq, \sim) \), see Section 2.2 for details.

**Definition 6.** (Interpreted QBA algebras). Let \( C \) be a countable set of symbols. An interpreted QBA algebra over \( C \) (IQB algebra, for short) is a pair \( L^+ = (L, \sigma) \), where \( L = (L, \leq, \sim) \) is a QBA algebra and \( \sigma : C \rightarrow L \) is an interpretation of the symbols in \( C \) as truth values in \( L \).

**Syntax.** Logical operators can often be naturally interpreted as either maximizers or minimizers of the truth values. For example, disjunction (\( \varphi \lor \psi \)) can be understood as a maximizer (“the most that we can make of either \( \varphi \) or \( \psi \)”), and conjunction as a minimizer. This extends to existential quantification (maximizing) and universal quantification (minimizing) over paths, strategies, moments in time, etc. Since multi-valued negation is problematic, we will use the syntactic variant of ATL* in negation normal form. Moreover, valuation will be given explicitly for all literals and not only atoms.

To increase the expressive power of the language, we allow for the use of symbols in \( C \). Finally, we add the operator \( \preceq \) representing the lattice order, useful for comparing truth values of formulas. The resulting logic is called mv-ATL* and has the following syntax:

\[ \varphi := c | p | \neg p | \varphi \land \psi | \varphi \lor \psi | \langle A \rangle \gamma | \langle A \rangle \gamma | \varphi \preceq \psi. \]

where \( p \in Prop \), \( A \subseteq C \), and \( \gamma \) is a path formula. In what follows, by an order formula we shall mean any formula of the form \( \varphi_1 \preceq \varphi_2 \). Additionally, we define \( \varphi_1 \equiv \varphi_2 \iff (\varphi_1 \preceq \varphi_2) \land (\varphi_2 \preceq \varphi_1) \).

**Semantics.** The semantics is provided over concurrent game structures with multi-valued interpretation of literals.

**Definition 7.** (Multi-Valued CGS). Let \( Lit = \{ p, \neg p \mid p \in Prop \} \), and let \( L^+ = (L, \sigma) \) be an IQB algebra with \( L = (L, \leq, \sim) \). A multi-valued concurrent game structure (mv-CGS) over \( L \) is a tuple \( M = (A, St, Act, d, t, Prop, V, L^+) \), where \( A, St, Act, d, t, Prop \) are as before, and \( V : Lit \times St \rightarrow L^+ \) assigns all literals with truth values from the logical domain \( L \). Note that, for greater generality, we do not require that \( V(\neg p, q) = \sim V(p, q) \).

Let \( M = (A, St, Act, d, t, Prop, V, L^+) \) be an mv-CGS, with \( L^+ = (L, \sigma) \). The valuation function \( V \) is defined as below. We sometimes use \( \bigcup \gamma \{ Y \} \) as a shorthand for \( \bigcup \gamma \{ Y \mid X \} \), and similarly for the supremum. For any \( q \in St \) and any path \( \lambda \) in \( M \):

\[ [c]_{M,\lambda} = [c] \quad \text{for} \quad c \in C; \]
\[ [p]_{M,\lambda} = V(p, q) \quad \text{and} \quad \neg[p]_{M,\lambda} = V(\neg p, q) \quad \text{for} \quad p \in Prop; \]
\[ [\varphi_1 \land \varphi_2]_{M,\lambda} = \bigwedge \{ [\varphi_1]_{M,\lambda}, [\varphi_2]_{M,\lambda} \}; \]
\[ [\varphi_1 \lor \varphi_2]_{M,\lambda} = \bigvee \{ [\varphi_1]_{M,\lambda}, [\varphi_2]_{M,\lambda} \}; \]
\[ \lnot \gamma_{M,\lambda} = \bigwedge \{ [\gamma]_{M,\lambda}, [\gamma]_{M,\lambda} \}; \]
\[ \gamma_{M,\lambda} = \bigvee \{ [\gamma]_{M,\lambda}, [\gamma]_{M,\lambda} \}; \]
\[ [\langle A \rangle \gamma]_{M,\lambda} = \bigwedge \{ [\langle A \rangle \gamma]_{M,\lambda}, [\langle A \rangle \gamma]_{M,\lambda} \}; \]
\[ [\langle A \rangle \gamma]_{M,\lambda} = \bigvee \{ [\langle A \rangle \gamma]_{M,\lambda}, [\langle A \rangle \gamma]_{M,\lambda} \}; \]
\[ [\varphi_1 \lor \varphi_2]_{M,\lambda} = \top \quad \text{if} \quad [\varphi_1]_{M,\lambda} \leq [\varphi_2]_{M,\lambda} \quad \text{and} \quad \bot \quad \text{otherwise.} \]

**Example 6.** (State abstraction for driving agents). We take the CGS of Figure 1 and apply the method of state abstraction from [42] with the following mapping: \( \gamma(q_{oo}) = \gamma(q_{oo}) = \gamma(q_{oo}) = \gamma(q_{ii}) = \gamma(q_{ii}) = \gamma(q_{ii}) \). That is, situations where agent 1 is outside are represented by the abstract state \( q_{oo} \), those were the agent is inside but no collision has happened are mapped to \( q_{ii} \) etc. The resulting multi-valued CGS over lattice 2 + 4 is presented in Figure 4. Note that, e.g., \([m_2]_{M,\lambda} = i \), as the truth values of \([m_2]_{M,\lambda} \) in the original states \( q_{oo}, q_{oo} \) are in conflict. Moreover, let us assume that the model generating procedure had insufficient time to compute the values of \( \text{penalty}_1 \) and \( \text{penalty}_2 \) in state \( q_e \), which then \( \gamma(q_e) \). Thus, \([\text{penalty}_1]_{M,\lambda} = \gamma(q_{oo}), \gamma(q_{oo}) \). The valuations of those negated atoms that will appear in further examples are also given.

To explicitly show the connection between the truth values and their names in \( C \), for any IQB algebra \( L^+ = (L, \leq, \sim) \) and any truth value \( x \in \sigma(C) \), we will use the notation \( \langle x \rangle \) to denote any symbol \( c \in C \) such that \( \sigma(c) = x \). In other words, we will always have \( \sigma(\langle x \rangle) = x \) for all \( x \in \sigma(C) \).
 EXAMPLE 7 (State abstraction, cont.). We evaluate the formulas from Example 3 in our multi-valued model \(M_2\) from Figure 4. For some of them the truth values remain the same, e.g., 
\[
[(1)]G\text{-penalty}_{M_2, q_0} = \top, \text{ and } [(1)]\text{Fpenalty}_{M_2, q_0} = \bot, \text{ and } [(1)]\text{Fin}_{M_2, q_0} = \top.
\]
We also get 
\[
[(2)]\text{Fpenalty}_{M_2, q_0} = \bot.
\]
which suggests that the abstraction is too fine-grained wrt \(q_0\), and, obviously, for any ATL* formula \(\varphi, M', \iota \models \varphi\) iff \(M, \iota \models \varphi\).

We note that most approaches to multi-valued model checking of temporal specifications [38, 26, 39, 51] allow also for multi-valued transitions in the models, analogous to probabilistic transitions in Markov chains and Markov Decision Processes. That is, transitions can be assigned with “weights” drawn from the same algebra \(L\). Similarly, most 3-valued approaches to temporal abstraction and model checking implicitly assume 3-valued transitions by distinguishing between may and must transitions [24, 25, 28, 30]. However, the two approaches differ in how such transitions affect the semantics of formulas with universal quantification (such as “for all paths \(\gamma\)”). In the general multi-valued approach, the “weaker” the path is, the more it decreases the value of the formula. In the 3-valued approach, “weaker” paths have less influence on the overall value. We do not want to engage in this discussion here, and leave a proper treatment of multi-valued transitions for future work.

EXAMPLE 8 (Comparing truth values). The “comparison” operators provide several interesting specification patterns. For instance, they allow for specifications that are accepted when the “strength” of a property reaches a given threshold, similarly to the probabilistic approaches of [40, 18]. For example, the formula 
\[
[(1)]G\text{-collision} \leq [(1)]G\text{-penalty} \leq [(1)]G\text{-penalty}_2 \leq [(1)]G\text{-penalty}_o.
\]
expresses that formula \(\varphi\) holds in all reachable states of the model.

Levels of truth. We assume that \(\top\) is the single designated value, standing for full logical truth. In consequence, the truth and validity of formulas can be defined in a straightforward way as follows:

DEFINITION 8 (Levels of validity). Let \(M\) be mv-CGS, \(q\) a state in \(M\), and \(\varphi\) a formula of mv-ATL*. Then:

- \(\varphi\) is true in \(M, q\) (written \(M, q \models \varphi\)) iff \([\varphi]_{M,q} = \top\).
- \(\varphi\) is valid in \(M\) (\(M \models \varphi\)) iff \(\varphi\) is true in every state of \(M\).
- \(\varphi\) is valid (\(\models \varphi\)) iff \(\varphi\) is valid in every mv-CGS \(M\).
- Additionally, for a path formula \(\gamma\), we can say that \(\gamma\) holds on run \(\lambda\) in a mv-CGS \(M\) (written \(M, \lambda \models \gamma\)) iff \([\gamma]_{M,\lambda} = \top\).

3.2 Properties of mv-ATL*

We now show that mv-ATL* agrees with standard ATL* on 2-valued models, unlike the 3-valued version of ATL* from [43].

THEOREM 2. The logic mv-ATL* is a conservative extension of ATL*, i.e., every CGS \(M\) for ATL* can be identified with an mv-CGS \(M'\) for mv-ATL* over a two-valued lattice, where, for any ATL* formula \(\varphi\) and any state (path) \(\iota\) of \(M'\), \(\iota \models \varphi\) iff \(M, \iota \models \varphi\).

Proof. For any CGS \(M = (A, St, Act, d, t, AP, V)\) for ATL*, define mv-CGS \(M' = (A, St, Act, d, t, AP, V', L_2)\), where \(L_2 = (\{\bot, \top\}, \preceq, \sim, \sigma)\) with \(\bot \preceq \top\), \(\sim = \top\), and \(V'(p, q) = \top\) if \(q \in V(p), \bot\) otherwise. Then \(M'\) is an mv-CGS for mv-ATL*.

Note that the converse does not hold, i.e., a valid formula of ATL* need not be valid in mv-ATL*. An example of such a formula is \(p \lor \neg p\), which is valid in ATL* but not in mv-ATL*.

4. MODEL CHECKING mv-ATL*

Given a mv-CGS \(M\), a state \(q\), and a mv-ATL* formula \(\varphi\), the model checking problem consists in computing the value of \([\varphi]_{M,q}\).

4.1 Adapting Translations to mv-ATL*

A key result obtained in [38] was Theorem 1, providing a method for reducing a model for mv-CTL* based on a lattice \(L\) of logical values to a model based on a sublattice \(L'\) of \(L\). This ultimately allowed us to reduce model-checking mv-CTL* to model checking of (two-valued) CTL*.

An analogue of that theorem can be shown to hold for mv-ATL*, i.e., the sublanguage of mv-ATL* consisting of formulas without the \(\sim\) operator.

Theorem 3 (Bellman equations for mv-ATL*). The following formulas of mv-ATL* are valid:

\[

\begin{align*}
&\langle (A)\rangle \varphi_1 U \varphi_2 \equiv \varphi_2 \lor (\langle (A)\rangle \varphi_1) U \varphi_2, \\
&\langle (A)\rangle \varphi_1 W \varphi_2 \equiv \varphi_2 \lor (\langle (A)\rangle X \langle (A)\rangle \varphi_1) W \varphi_2.
\end{align*}
\]

Proof. The proof is analogous to [33, Proposition 3], and we omit it due to lack of space.
The proof follows easily from the key result given below:

**Lemma 1.** Let a state or path formula \( \varphi \) be such that
\[
[\varphi]_{M,i} = \bigcup_{i \in I} [\varphi_{j,i}]_{M,j,i}
\]
for any mv-CGS \( M \), any states and/or paths \( i,j \), of \( M \), any countable sets \( I, J \), and state (resp. path) formulas of mv-ATL \( \varphi_{j,i} \); or
\[
[\varphi]_{M,i} = \bigcup_{i \in I} [\varphi]_{M,i,j,i}
\]
for any mv-CGS \( M \), any states and/or paths \( i,j \), of \( M \), any countable sets \( I, J \), and state (resp. path) formulas of mv-ATL \( \varphi_{j,i} \); the other case follows by symmetry. As \( f \) preserves the bounds, by the assumption on \( \varphi \) we have \( f([\varphi]_{M,i}) = \bigcup_{i \in I} f([\varphi_{j,i}]_{M,j,i}) \).

Each \( \varphi_{j,i} \) satisfies (3), so \( f([\varphi]_{M,i}) = \bigcup_{i \in I} [\varphi_{j,i}]_{M,j,i} \), whence \( [\varphi]_{M,i} \), \( \varphi_{M,i} \), \( x \) iff \([\varphi]_{M,i} \in f^{-1}(x) \), and (3) holds for \( \varphi \). To prove Theorem 4, it suffices to check that the semantics of each formula of mv-ATL* can be given like in Lemma 1.

### 4.2 Translating Order Formulas

Unfortunately, Theorem 4 cannot be extended to mv-ATL* \( \ast \), i.e., to the language of mv-ATL* extended with order formulas of the form \( \varphi_1 \preceq \varphi_2 \), where \( \preceq \) represents the algebra order — because Equation (3) does not hold for such formulas.

To show a counterexample, we consider the QB algebra \( \mathcal{L}_o = (\mathbb{N}, \leq, \sim) \), where \( \mathbb{N} = \{0, k, 1, k+1, \ldots \} \), \( k \) is the usual total order on \( \mathbb{N} \), and \( \sim = (2k-1) - x \). Of course, in \( \mathcal{L}_o \) we have \( 0 \leq 2k-1 = 1 \) according to our lattice notation. Then the translation \( f : \mathcal{L}_o \to \mathcal{R} \) is given by \( f(x) = 2k-1 \) if \( x \geq k \), and \( f(x) = 0 \) if \( x < k \).

Now take arbitrary \( k_1, k_2 \) such that \( 0 < k_1 < k_2 < k \), and an mv-CGS \( M \) over \( \mathcal{L}_o \) for an arbitrary \( \sigma : C \to \mathcal{L}_o \) such that, for some state \( q \in S \) of \( M \) and atomic propositions \( p_1, p_2 \in PV \), we have \( V(p_1, q) = k_1 \) for \( i = 1, 2 \).

Next, let \( \varphi = p_2 \preceq p_1 \). Since \( [p_1]_{M,q} = k_1 \) for \( i = 1, 2 \) and \( k_2 > k_1 \), we have \( \neg[p_2]_{M,q} \leq [p_1]_{M,q} \), whence \([p_2]_{M,q} = 0 \).

However, for the model \( M_1 \) obtained from \( M \) with the translation \( f \) we get \( [p_1]_{M_1,q} = 0 \) for \( i = 1, 2 \) (as \( k_1 < k, f(k_1) = 0 \) for \( i = 1, 2 \)), where \([p_2]_{M_1,q} = 2k-1 \) which implies \([p_2]_{M,q} = 2k-1 \). Yet, as \( f^{-1}(2k-1) = \{k, k+1, \ldots \} \) we have \([p_2]_{M,q} = 0 \notin f^{-1}(2k-1) \), which contradicts Equation (3).

The above result can be generalized as follows:

**Lemma 2.** If \( \mathcal{L} = (L, \leq, \sim) \) of Theorem 4 contains a chain or anti-chain of cardinality \( n \), and \( \mathcal{L} = (L', \leq, \sim') \) is a subalgebra of \( \mathcal{L} \) of cardinality \( n', k \) of \( \mathcal{L} \), then there is no function \( f : L \to L' \) satisfying condition (3) if the language under consideration contains order formulas.

Now we give a necessary and sufficient condition for \( f : L \to L' \) as in Theorem 4, preserving bounds in \( \mathcal{L} \), to satisfy the translation condition (3) from Theorem 4 also for order formulas.

**Lemma 3.** Let the QB algebra \( \mathcal{L} = (L, \leq, \sim) \), its subalgebra \( \mathcal{L}' = (L', \leq', \sim') \), a mapping \( f : L \to L' \), and mv-CGSs \( M, M' \) satisfy the assumptions of Theorem 4. Then, (3) of that Theorem is satisfied for all order formulas iff the following conditions hold:

1. For any \( x_1, x_2 \in L \) \( x_1 < x_2 \Rightarrow f(x_1) < f(x_2) \)
2. For any \( x_1, x_2 \in L \) \( x_1 \sim x_2 \Rightarrow f(x_1) \sim f(x_2) \)

**Proof.** Note that an order formula is a state formula, and that for any such formula \( \psi \) we have \( [\psi]_{M,q} \in \{\bot, \top\} \) for any mv-CGS \( M \) and any \( q \in S \). Thus in order to prove (3) for such formulas, it suffices to show that, for any order formula \( \varphi \) and any state \( q \):

\[
[\varphi]_{M',q} = \top \iff [\varphi]_{M,q} \in f^{-1}(\top)
\]
Theorem 5. Let the QB algebra $L = (L, \leq, \sim)$, its subalgebra $L' = (L', \leq', \sim')$, $f : L \rightarrow L'$, and mv-CGS $M$, $M'$ satisfy the assumptions of Theorem 4. Then the condition (3) of that Theorem:

$$[\varphi]_{M', q} = x \quad \text{iff} \quad [\varphi]_{M, q} \in f^{-1}(x) \quad (7)$$

is satisfied for all formulas of $\text{mv-ATL}_x^*$ over $L$ iff conditions C1, C2 of Lemma 3 hold.

It can be seen that conditions C1, C2 imply that any translation $f$ meeting them must preserve the exact structure of the QB algebra $L$. An important consequence of that fact is:

Corollary 2. Given a QB algebra $L = (L, \leq, \sim)$ and its subalgebra $L' = (L', \leq', \sim')$, any function $f : L \rightarrow L'$ preserving the algebra bounds and satisfying the translation condition (3) for all order formulas must be one-to-one.

Proof. Suppose $f$ satisfies the above assumption, $x_1, x_2 \in L$ and $x_1 \neq x_2$. Then we have one of the following cases:

1. $x_1 < x_2$ or $x_2 < x_1$. Then $f(x_1) \neq f(x_2)$ by C1 of Theorem 5.
2. $x_1 \sim x_2$. Then $f(x_1) \sim f(x_2)$ by C2 of Theorem 5, which again implies $f(x_1) \neq f(x_2)$.

The meaning of Corollary 2 is that there is no way of reducing $n$-valued model checking to $k$-valued model checking for $k < n$, if we want to handle all order formulas. In particular:

Corollary 3. No reduction of $\text{mv-ATL}_x^*$ model checking to 2-valued model checking is possible.

4.3 Recursive Model Checking of $\text{mv-ATL}_x^*$

The impossibility result in Corollary 3 is due to the fact that order formulas can be used to encode the semantics in the language – including in particular its $n$-valued character. However, one usually wants to model-check one formula at a time. Then, Theorem 5 can be in many cases modified to provide the desired reduction:

Theorem 6. Let $L$, its subalgebra $L'$, $f : L \rightarrow L'$, and $M, M'$ be as in Theorem 4. Further, let $\varphi$ be a formula of $\text{mv-ATL}_x^*$ and $\text{Sub}(\varphi)$ be the set of all its subformulas. Then $\varphi$ satisfies the translation condition 7 of Theorem 5 whenever for any order formula $\psi \in \text{Sub}(\varphi)$ such that $\psi = \varphi_1 \leq \varphi_2$ and for $x_i = [\varphi_i]_{M, i}, i = 1, 2$, the following conditions hold:

(C1') $x_1 < x_2$ ⇒ $f(x_1) < f(x_2)$

(C2') $x_1 \sim x_2$ ⇒ $f(x_1) > f(x_2)$

Proof. To prove the thesis, we assume that C1', C2' are satisfied, and show by structural induction that the translation condition holds for any $\psi \in \text{Sub}(\varphi)$, i.e.,

$$[\psi]_{M', q} = x \quad \text{iff} \quad [\psi]_{M, q} \in f^{-1}(x) \quad (8)$$

For atomic or constant $\psi$, the thesis follows from Theorem 4. Suppose now (8) holds for all subformulas of $\varphi$ having rank $k$, and assume $\psi$ is of rank $k + 1$. If $\psi$ is obtained from subformulas of rank at most $k$ using any operator $Op$, then the satisfaction of (8) follows from the fundamental Lemma 3.

Thus it remains to consider the case of $\leq$. Assume $\psi = \psi_1 \leq \psi_2$, where (8) holds for $\psi_1, \psi_2$. Since $\psi$ is an order formula, according to what we have already noted in the proof of Lemma 2, proving (8) for $\psi$ reduces to showing

$$[\psi]_{M', q} = T \quad \text{iff} \quad [\psi]_{M, q} \in f^{-1}(T) \quad (9)$$

Note that since $\psi_1, \psi_2$ are in $\text{Sub}(\varphi)$, then C1', C2' hold for $x_i = [\varphi_i]_{M, i}, i = 1, 2$. By the inductive assumption, we also have

$$[\psi_i]_{M', q} = f([\psi_i]_{M, q}) \quad (i = 1, 2) \quad (10)$$

“$\Rightarrow$” We begin with the forward implication in (9). Assume that $[\psi_1]_{M', q} = T$. Then $[\psi_1]_{M, q} \leq [\psi_2]_{M, q}$. By (10), this implies $f([\psi_1]_{M, q}) \leq f([\psi_2]_{M, q})$. We show by contradiction that it implies

$$[\psi_1]_{M, q} \leq [\psi_2]_{M, q} \quad (11)$$

Suppose that (11) does not hold, then we have two possible cases:

Case 1: $[\psi_1]_{M, q} > [\psi_2]_{M, q}$. By condition C1' we have $f([\psi_1]_{M, q}) > f([\psi_2]_{M, q})$, whence from (10) we get $[\psi_1]_{M', q} > [\psi_2]_{M', q}$ and $[\psi_1]_{M', q} = T$, which is a contradiction.

Case 2: $[\psi_1]_{M, q} \sim [\psi_2]_{M, q}$. Then $f([\psi_1]_{M, q}) > f([\psi_2]_{M, q})$ by condition C2', which again leads to a contradiction by what we have already proved for Case 1.

Thus (11) above holds, whence $[\psi]_{M, q} = T \in f^{-1}(T)$, and the forward implication in (9) holds.

“$\Leftarrow$” The final step consists in proving the backward implication in (9). Assume that $[\psi]_{M, q} = f^{-1}(T)$. As $[\psi]_{M, q} \in (L, \leq, \sim)$ and $f(L) \neq f(L)$ by the preservation of bounds by $f$ and the non-triviality of $L, L'$, we get $[\psi]_{M, q} = T$, and consequently $[\psi]_{M, q} \leq [\psi_2]_{M, q}$. Since $f$ preserves bounds, this implies $f([\psi_1]_{M, q}) \leq f([\psi_2]_{M, q})$, whence from (10) we obtain $[\psi_1]_{M, q} \leq [\psi_2]_{M, q}$. This yields $[\psi]_{M', q} = T$, whence the backward implication in (9) holds, too.

Remark 2. Notice that Condition C2' is necessary, but C1' is not. Moreover, both are necessary if $\psi$ is symmetric w.r.t. to $\leq$, i.e., $(\varphi_2 \leq \varphi_1) \in \text{Sub}(\varphi)$ whenever $(\varphi_1 \leq \varphi_2) \in \text{Sub}(\varphi)$.

Assume that our mv-CGSs are defined over DM algebras, i.e., distributive QB algebras. We can show that the translation method based on join irreducible elements ($\text{JT}(L)$) [26, 12, 38, 37] can be applied to a formula $\varphi$ of $\text{mv-ATL}_x^*$ and an mv-CGS $M$, provided the assumptions of Theorem 6 are satisfied.

By (1), for each $x \in L$ we have $x = [\bigcup \text{JT}(L)] \cap \downarrow x$. Let $M'$ be the model obtained using the translation $f_1$. Therefore, according to Theorem 6: $[\varphi]_{M_1,q} = x$ iff $[\varphi]_{M_1} \in f_1^{-1}(x)$ whence $[\varphi]_{M_1,q} = T$ iff $[\varphi]_{M_1} \in \downarrow I$. Thus,

$$[\varphi]_{M_1,q} = \bigcup \{l \in \text{JT}(L) \mid [\varphi]_{M_1} = I \}. \quad (12)$$

In case the assumptions of Theorem 6 are not satisfied for $M$ and $\varphi$, we cannot translate model checking of $\text{mv-ATL}_x^*$ to model checking of $\text{ATL}$. Then, the simplest solution is to replace all the subformulas of $\varphi$ which are order formulas by fresh atomic propositions, and then apply our translation method to the resulting formula $\varphi'$ of $\text{mv-ATL}_x^*$ and the new model $M'$ which extends $M$ by the valuation of the new atomic propositions.

The algorithm is recursive. For each order subformula $\psi_1 \leq \psi_2$ of $\varphi$ such that $\psi_1, \psi_2 \in \text{mv-ATL}_x^*$, we define a fresh propositional variable $p_{\psi_i}$. Then, for each state $q$ of model $M'$ we have $V'(p_{\psi_i}) = \top$ iff $[\psi_i]_{M,q} \leq [\psi_2]_{M,q}$, and $V'(\neg p_{\psi_i}) = \bot$ iff $[\psi_i]_{M,q} \not\leq [\psi_2]_{M,q}$, where $[\psi_i]_{M,q}$ for $i \in \{1, 2\}$ is computed using our translation method. Next, when all order subformulas have been replaced with fresh propositions, we compute $[\varphi']_{M,q}$ using the translation method defined.

The main disadvantage of the above method compared to the direct translation method is that it requires computing the values of the new atomic propositions for all states of the model $M$. The method can be improved if we assume that a specific symbolic model checking method for two valued $\text{mv-ATL}_x^*$ is used. We
leave a study of this subject for future work. Nevertheless, the algorithm has two important consequences. First, it provides a general polynomial-time reduction from model checking mv-ATL* (resp. mv-ATL<\text{ir}>), to model checking standard 2-valued ATL* (resp. ATL). We state it formally as follows.

**Theorem 7.** Multi-valued verification of ATL* incurs only polynomial increase in the complexity compared to the 2-valued case. Specifically, model checking mv-ATL* is 2EXPTIME-complete, and model checking mv-ATL<\text{ir}> is P-complete in model size, formula length, and number of logical values.

Secondly, we note that correctness of the translation does not depend on the type of strategies being used in the semantics of mv-ATL*. As it is, the translation provides a model checking reduction to the IR variant of ATL* (perfect information + perfect recall). If we used memoryless strategies of type \( \text{sr}_a : \text{St} \rightarrow \text{Act} \) instead of perfect recall, the translation would yield reduction to the Ir variant of ATL* (perfect information + imperfect recall) [50]. Since the IR and IR semantics coincide in 2-valued ATL, we get the following.

**Theorem 8.** For mv-ATL<\text{ir}>, memory is irrelevant, i.e., its semantics can be equivalently given by memoryless strategies.

## 5. Imperfect Information

It can be argued that realistic multi-agent systems always include some degree of limited observability [50, 31, 1, 36, 2, 34, 49]. However, model checking of ATL with imperfect information is hard – more precisely, \( \Delta_2 \) to PSPACE-complete for agents playing memoryless strategies [50, 35, 13] and undecidable for agents with perfect recall [21]. Furthermore, the imperfect information semantics of ATL does not admit fixpoint equivalences [15], which makes incremental synthesis of strategies cumbersome if not impossible. Practical attempts at the problem have emerged only recently [48, 16, 27, 17], and do not go far beyond checking each of the exponentially many strategies. The experimental results show that this is feasible only for very small models.

To such hard problems, state abstraction can be successfully applied but it must be very coarse (clustering many concrete states together into a single abstract state) in order to reduce the model sufficiently. This can lead to a substantial reduction in the complexity, though possibly at the expense of introducing inconsistent or inconclusive labeling of propositions at some abstract states. In consequence, multivalued model checking can be extremely useful when reasoning about strategies under uncertainty. We formalize this by the following extension of mv-ATL<\text{ir}>.

**Multivalued ATL* with imperfect information.** Let us extend mv-CGS with indistinguishability relations \( \sim_1, \ldots, \sim_k \), one per agent in \( A \). Memoryless strategies with imperfect information (ir strategies, for short) are functions \( \text{sr}_a : \text{St} \rightarrow \text{Act} \) such that \( q \sim_a q' \) implies \( \text{sr}_a(q) = \text{sr}_a(q') \). Perfect recall strategies with imperfect information (shortly: Ir strategies) are functions \( \text{sr}_a : \text{St}^+ \rightarrow \text{Act} \) st. \( q_0 \sim_a q_{0,0}, \ldots, q_n \sim_a q_{0,n} \) implies \( \text{sr}_a(q_0 \ldots q_n) = \text{sr}_a(q_{0,0} \ldots q_{0,n}) \). Again, collective strategies for \( A \subseteq A \) are tuples of individual strategies for \( a \in A \). We denote them by \( \Sigma_A^\text{ir} \) and \( \Sigma_A^\text{pr} \), respectively.

The semantics of mv-ATL<\text{ir}>, parameterized by the type of strategies \( \mathcal{G} = \text{Ir}, \text{Ir}, \text{ir}, \text{ir}, \text{ir} \), can be defined by replacing the clause for \( \langle\langle A \rangle \rangle \gamma \) from Section 3.1 as follows:

\[
\langle\langle A \rangle \rangle \gamma_{M,q} = \bigcup_{a \in A} \bigcap_{\lambda \in \text{out}(q,a)} \{\gamma\}_{M,a,\lambda}.4
\]

This corresponds to the notion of objective ability, cf. [31, 15]. Subjective ability requires additional quantification over indistinguishable states; we omit the formal treatment for lack of space.

**Example 9.** (Abstraction for imperfect info). Take model \( M_3 \) from Example 2, and assume that agent 2 does not see the location of vehicle 1. This can be modeled by the following indistinguishability relation: \( q_{0,0} \sim_2 q_{0,1} \) and \( q_{0,1} \sim_2 q_{1,0} \). If we further apply the state abstraction \( \sigma \) from Example 6, we obtain model \( M_3 \) that adds \( q_{0,0} \sim_2 q_{1,0} \) to \( \text{mc-CGS} M_2 \). Now, e.g., \( \{\langle(2)\rangle \} \text{F(out}_2 \land \text{\neg penalty}_2\}\}_{M_3,q_{0,0}} = i \).

**Model checking techniques and formal results.** We emphasize again that the techniques proposed in Section 4 do not depend on the actual definition of strategy sets \( \Sigma_A \). In consequence, they carry over to the imperfect information case, and can be applied in exactly the same way to obtain model checking reductions from mv-ATL<\text{ir}> to the corresponding 2-valued cases. This demonstrates the power of the translation method that can be directly applied to a vast array of possible semantics for ATL. Combining the observation with the existing complexity results [13], we get the following as immediate consequences:

**Theorem 9.** Model checking mv-ATL<\text{ir}> is NP-complete, and model checking mv-ATL<\text{ir}*,> is PSPACE-complete in the size of the model and the formula, and the number of logical values.

**Theorem 10.** Model checking mv-ATL<\text{ir}> and mv-ATL<\text{ir}*,> is undecidable in general. For the fragment of mv-ATL<\text{ir}*,> with singleton coalitions only, model checking is EXPTIME-complete in the model size, formula length, and number of logical values.

## 6. Conclusions

In this paper, we study a variant of alternating-time temporal logic, called mv-ATL<\text{ir}>, where the truth values are drawn from an arbitrary De Morgan algebra. We also argue that multivalued model checking of mv-ATL<\text{ir}> specifications can be useful, especially for systems whose models cannot be fully analyzed due to their complexity and/or inaccessibility of the relevant information. Examples include verification of distributed databases and abstraction-based model checking of massive multi-agent systems, especially for specifications of strategies under imperfect information.

We prove that our multivalued semantics of ATL* provides a conservative extension of the classical 2-valued variant. Even more importantly, we propose efficient (i.e., polynomial-time) translations from multivalued model checking to the 2-valued case. The proposed techniques are elegant enough so that they can be directly applied to other semantic variants of strategic ability, for example ones that refer to imperfect information scenarios. Other results comprise analogues of Bellman equations and positionality of winning strategies for mv-ATL<\text{ir}> in perfect information scenarios.

Our translation method for mv-ATL<\text{ir}>, formulas allows us to benefit from the efficient 2-valued model checking algorithms. If a translation for a formula \( \varphi \) cannot be defined, our model checking method becomes less efficient, as it requires replacing all the order subformulas of \( \varphi \) with propositions and computing their values in a given mv-CGS.

In the future, we plan to have a closer look at state and action abstractions of complex models, including an appropriate semantic interpretation of multi-valued transitions.

**Acknowledgements.** Wojciech Jamroga acknowledges support of the 7th Framework Programme of the European Union under the Marie Curie IEF project ReVINK (PIEF-GA-2012-626398).

## References

[1] T. Ågotnes. A note on syntactic characterization of incomplete information in ATEL. In *Proceedings of Workshop*
on Knowledge and Games, pages 34–42, 2004.


[40] M. Kwiatkowska, G. Norman, and D. Parker. PRISM:


