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$^*_{\preccurlyeq}$), 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$^*_{\preccurlyeq}$. We also propose two ways of overcoming the problem. Firstly, we identify constraints on mv-ATL$^*_{\preccurlyeq}$ 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$^*_{\preccurlyeq}$. 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$^*_{\preccurlyeq}$ 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, $\langle\langle A\rangle\rangle\gamma$, 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 $\mathsf{F}\mathsf{p}$. 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 $\mathsf{G}\mathsf{p}$ 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, $\mathsf{F}\mathsf{p}$ 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 $\mathsf{G}\mathsf{p}$ is analogous. A more sophisticated motivating example is presented below.

EXAMPLE 1 (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: $\langle\langle a\rangle\rangle\mathsf{G}\mathsf{safe}_\mathsf{a}$. Suppose further that the state of each agent is determined by m binary variables. It is easy to see that the system has $O(2^{k\cdot m})$ 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 $\mathcal{R}$. For example, $q\mathcal{R}q'$ iff q, q' coincide on the variables of agent a. Such states will end up in the same abstract state $[q]_\mathcal{R}$. However, it can happen that the value of a proposition (e.g., safe_a) varies across $[q]_\mathcal{R}$. Then, the most reasonable value for safe_a in the ab-*

[*] Also affiliated with: Institute of Informatics, University of Natural Sciences and Humanities, Siedlce, Poland.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016),*
J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.),
May 9–13, 2016, Singapore.
Copyright © 2016, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

stract state is “conflicting” or “inconsistent”. Moreover, if we use an anytime *algorithm, it may be stopped before it computes the truth of* $\mathsf{safe_a}$ *in some of the new states. Then, the value of the proposition can be set to “uncomputed” or “undecided”. Consequently, we will have to manipulate at least four truth values when model checking the formula* $\langle\!\langle a\rangle\!\rangle \mathsf{G}\mathsf{safe_a}$ *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$^*_{\preccurlyeq}$. Our approach extends previous work on temporal model checking: similarly to [38, 39], we do not propose dedicated algorithms for mv-ATL$^*_{\preccurlyeq}$. 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$^*_{\preccurlyeq}$, (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-boolean lattices 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 “comparison” operators $\preccurlyeq$ and $\cong$, which provide: (i) the notions of material implication and biconditional, useful in specifying general properties of multi-valued models; (ii) a way of model checking “threshold 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 temporal 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 systems. 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 aggregate them over available strategies and possible paths.

2. PRELIMINARIES

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

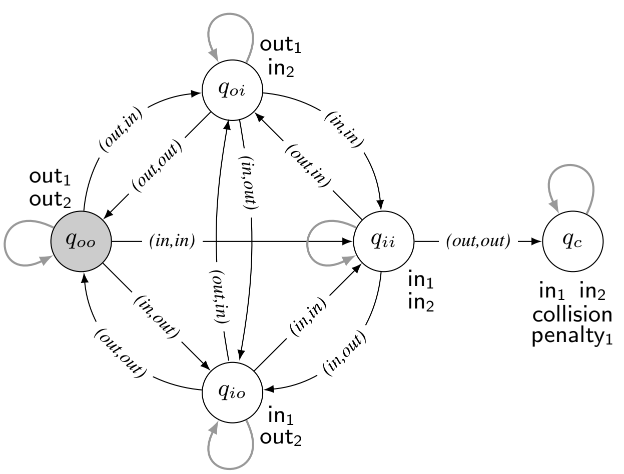


Figure 1: Autonomous vehicles at the intersection: model M_1

2.1 What Agents Can Achieve

Alternating-time temporal logic [3] generalizes branching-time temporal logic CTL* by replacing path quantifiers E, A with *strategic modalities* $\langle\!\langle A\rangle\!\rangle$. Informally, $\langle\!\langle A\rangle\!\rangle\gamma$ says that a group of agents A has a collective strategy to enforce temporal property γ. Similarly to CTL* and CTL, we consider two syntactic variants of the alternating-time logic, namely ATL* and ATL.

Let $\mathcal{A}$ be a finite set of agents, and $Prop$ a countable set of atomic propositions. The language of ATL* is defined as follows:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi\wedge\varphi \mid \langle\!\langle A\rangle\!\rangle\gamma,$$
$$\gamma ::= \varphi \mid \neg\gamma \mid \gamma\wedge\gamma \mid \mathsf{X}\,\gamma \mid \gamma\,\mathsf{U}\,\gamma.$$

where $A \subseteq \mathcal{A}$ and $\mathsf{p} \in Prop$. Derived boolean connectives and constants ($\vee, \top, \bot$) are defined as usual. “Sometime”, “weak until”, and “always from now on” are defined as $\mathsf{F}\gamma \equiv \top\,\mathsf{U}\,\gamma$, $\gamma_1\,\mathsf{W}\,\gamma_2 \equiv \neg((\neg\gamma_2)\,\mathsf{U}\,(\neg\gamma_1\wedge\neg\gamma_2))$, and $\mathsf{G}\gamma \equiv \gamma\,\mathsf{W}\,\bot$. Also, we can use $\overline{\langle\!\langle A\rangle\!\rangle}\,\gamma \equiv \neg\langle\!\langle A\rangle\!\rangle\gamma$ 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:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi\wedge\varphi \mid \langle\!\langle A\rangle\!\rangle\mathsf{X}\,\varphi \mid \langle\!\langle A\rangle\!\rangle\varphi\,\mathsf{U}\,\varphi \mid \langle\!\langle A\rangle\!\rangle\varphi\,\mathsf{W}\,\varphi.$$

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 $M = \langle\mathcal{A}, St, Act, d, t, Prop, V\rangle$ which includes nonempty finite sets of: agents $\mathcal{A} = \{1,\dots,k\}$, states St, actions Act, atomic propositions $Prop$, and a propositional valuation $V : St \to 2^{Prop}$. The function $d : \mathcal{A}\times St \to 2^{Act}$ defines availability of actions. The (deterministic) transition function t assigns a successor state $q' = t(q,\alpha_1,\dots,\alpha_k)$ to each state $q \in St$ and any tuple of actions $\alpha_i \in d(i,q)$ that can be executed by $\mathcal{A}$ in q.*

A pointed CGS *is a pair (M, q_0) consisting of a concurrent game structure M and an initial state q_0 in M.*

EXAMPLE 2 (DRIVING AGENTS). *Consider an intersection with k autonomous vehicles around it. Each vehicle is modeled*

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

[2] In some papers, $[\![A]\!]\gamma \equiv \neg\langle\!\langle A\rangle\!\rangle\neg\gamma$ is used as the dual strategic operator. We use $\overline{\langle\!\langle A\rangle\!\rangle}\,\gamma$ instead for technical convenience, to avoid negations of path subformulae when transforming to NNF.

as a separate agent, whose local state is characterized by either the proposition $\mathsf{out_i}$ *(when the vehicle is outside the intersection) or* $\mathsf{in_i}$ *(when the vehicle is inside it). The available actions are: in ("drive in" or "stay in", depending on the current state) and out ("drive out" or "stay out"). When both agents are in and decide to leave at the same time, a collision occurs (*$\mathsf{collision}$*), and the guilty party – or parties – have to pay a penalty (*$\mathsf{penalty_i}$*).*

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 *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}. $\lambda[i]$ denotes the ith position on λ and $\lambda[i, \infty]$ the suffix of λ starting with i.

A *perfect recall strategy* (or *IR*-strategy) for agent a is a function $s_a : St^+ \to Act$ such that $s_a(q_0q_1\ldots q_n) \in d(a, q_n)$. A *collective strategy* for agents $A = \{a_1, \ldots, a_r\}$ is a tuple of individual strategies $s_A = \langle s_{a_1}, \ldots, s_{a_r}\rangle$. The set of such strategies is denoted by Σ_A^{IR}. The "outcome" function $out(q, s_A)$ 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:

$M, q \models \mathsf{p}$ iff $q \in V(\mathsf{p})$, for $\mathsf{p} \in Prop$;

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;

$M, q \models \varphi_1 \wedge \varphi_2$ iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;

$M, q \models \langle\!\langle A\rangle\!\rangle\gamma$ iff there is a strategy $s_A \in \Sigma_A^{IR}$ such that, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda \models \gamma$.

$M, \lambda \models \varphi$ iff $M, \lambda[0] \models \varphi$;

$M, \lambda \models \neg\gamma$ iff $M, \lambda \not\models \gamma$;

$M, \lambda \models \gamma_1 \wedge \gamma_2$ iff $M, \lambda \models \gamma_1$ and $M, \lambda \models \gamma_2$;

$M, \lambda \models \mathsf{X}\,\gamma$ iff $M, \lambda[1, \infty] \models \gamma$; and

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

EXAMPLE 3 (DRIVING AGENTS, CTD.). *For model* M_1*, we have* $M_1, q_{oo} \models \langle\!\langle 1\rangle\!\rangle \mathsf{G}\neg\mathsf{penalty}_1$*: 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:* $M_1, q_{oo} \not\models \langle\!\langle 1\rangle\!\rangle \mathsf{F}\,\mathsf{penalty}_1$*. This can only be ensured if the agents cooperate:* $M_1, q_{oo} \models \langle\!\langle 1, 2\rangle\!\rangle \mathsf{F}\,\mathsf{penalty}_1$*. Moreover,* $M_1, q_{oo} \models \langle\!\langle 1\rangle\!\rangle \mathsf{F}\mathsf{in}_1 \wedge \langle\!\langle 2\rangle\!\rangle \mathsf{F}\mathsf{in}_2$*: each agent is able to enter the intersection. Still, it cannot successfully drive through the crossroads on its own (e.g.,* $M_1, q_{oo} \not\models \langle\!\langle 1\rangle\!\rangle \mathsf{F}(\mathsf{in}_1 \wedge \mathsf{F}\mathsf{out}_1)$*).*

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:

$$\varphi ::= \mathsf{p} \mid \neg\mathsf{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle\!\langle A\rangle\!\rangle\gamma \mid \overline{\langle\!\langle A\rangle\!\rangle}\,\gamma,$$
$$\gamma ::= \varphi \mid \gamma \wedge \gamma \mid \gamma \vee \gamma \mid \mathsf{X}\,\gamma \mid \gamma\,\mathsf{U}\,\gamma \mid \gamma\,\mathsf{W}\,\gamma.$$

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.

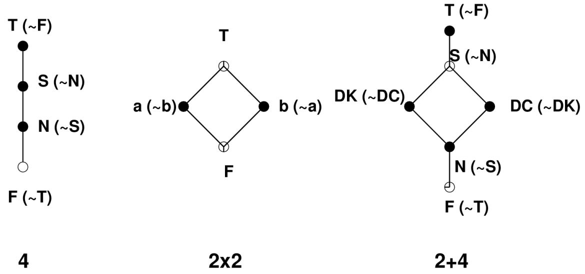


Figure 2: De Morgan algebras and join-irreducible elements

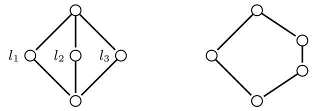


Figure 3: Non-distributive lattices M5 and N5

DEFINITION 2. *A* lattice *is a partial order* $\mathcal{L} = (L, \leq)$*, where every pair of elements* $x, y \in L$ *has the greatest lower bound (called* meet *and denoted by* $x \cap y$*) and the least upper bound (called* join *and denoted by* $x \cup y$*).*[3]

In what follows, we only consider finite lattices. We denote the least and the greatest elements of $\mathcal{L}$ by $\bot, \top$, respectively. Also, we write (i) $x_1 < x_2$ iff $x_1 \leq x_2$ and $x_1 \neq x_2$, and (ii) $x_1 \bowtie x_2$ iff neither $x_1 \leq x_2$ nor $x_2 \leq x_1$. Moreover, let:

- $\uparrow x = \{y \in L \mid x \leq y\}$ denote the upward closure of x, and
- $\downarrow x = \{y \in L \mid y \leq x\}$ denote the downward closure of x.

DEFINITION 3. $\mathcal{L} = (L, \leq, \sim)$ *is a* quasi-boolean (QB) algebra *if* $(L, \leq)$ *is a lattice and* $\sim$ *is a unary operator (called* complement*) such that, for any* $x, y \in L$*: (i)* $\sim(x \cap y) = \sim x \cup \sim y$*, (ii)* $\sim(x \cup y) = \sim x \cap \sim y$*, (iii)* $x \leq y$ *iff* $\sim y \leq \sim x$*, (iv)* $\sim\sim x = x$*.*

DEFINITION 4. *A lattice* $\mathcal{L} = (L, \leq)$ *is distributive if, for any* $x, y, z \in L$*, the following two conditions hold: (i)* $z \cup (x \cap y) = (z \cup x) \cap (z \cup y)$*, (ii)* $z \cap (x \cup y) = (z \cap x) \cup (z \cap y)$*. A distributive QB algebra is called a* De Morgan *algebra, shortly:* DM algebra*.*

EXAMPLE 4. *Figure 2 presents three DM algebras with applicability motivated by clear practical intuitions: the total order* $\mathbf{4}$ *for representing uncertainty, lattice* $\mathbf{2} \times \mathbf{2}$ *for representing disagreement, and lattice* $\mathbf{2} + \mathbf{4}$ *for representing both uncertainty and disagreement. In our view, lattice* $\mathbf{2} + \mathbf{4}$ *seems especially useful, as it provides truth values for both inconsistent and inconclusive evidence. The usual interpretation of its logical elements is:* $\bot$ *- must not (absolute falsity), N - should not, DC - do not care, DK - do not know, S - should,* $\top$ *- must (absolute truth). In this paper, we will use a somewhat different notation, writing* i *("inconsistent") instead of DC and* u *("undecided") instead of DK.*
Example non-distributive QB lattices are shown in Figure 3. Note that a lattice is distributive iff it contains neither M5 nor N5 [11].

DEFINITION 5. *Let* $\mathcal{L} = (L, \leq)$ *be a lattice. An element* $l \in L$ *is called* join-irreducible *iff* $l \neq \bot$ *and, for any* $x, y \in L$*, if* $l = x \cup y$*, then either* $l = x$ *or* $l = y$*.*
The set of all join-irreducible elements of $\mathcal{L}$ *is denoted by* $\mathcal{JI}(\mathcal{L})$*.*

[3] It follows from antisymmetry of $\leq$ that both the greatest lower bound and the least upper bound of x, y are uniquely determined.

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.

$$x = \bigcup(\mathcal{JI}(\mathcal{L}) \cap \downarrow x) \qquad (1)$$

EXAMPLE 5. *The join-irreducible elements of the DM algebras in Figure 2 are marked with black dots. In the lattice* $\mathbf{2}+\mathbf{4}$*, 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* $\mathcal{L}$ *be a finite DM algebra, and let* $l \in \mathcal{JI}(\mathcal{L})$*. Then the function* $f_l : L \longrightarrow \{\bot, \top\}$ *defined by* $f_l(\uparrow l) = \top$*,* $f_l(L \setminus \uparrow l) = \bot$ *preserves arbitrary bounds.*

REMARK 1. *The above does not hold for lattices which are not distributive. To see this, consider the element* l_1 *of* **M5***, which is join-irreducible. However,* f_{l_1} *does not preserve join, as* $f_{l_1}(l_2 \cup l_3) = f_{l_1}(\top) = \top$ *whereas* $f_{l_1}(l_2) \cup f_{l_1}(l_3) = \bot \cup \bot = \bot$*.*

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 $\mathcal{L} = (\mathrm{L}, \leq, \sim)$, see Section 2.2 for details.

DEFINITION 6 (INTERPRETED QB ALGEBRAS). *Let* $\mathcal{C}$ *be a countable set of symbols. An interpreted QB algebra over* $\mathcal{C}$ *(*IQB algebra*, for short) is a pair* $\mathcal{L}^+ = (L, \sigma)$*, where* $\mathcal{L} = (L, \leq, \sim)$ *is a QB algebra and* $\sigma : \mathcal{C} \to L$ *is an interpretation of the symbols in* $\mathcal{C}$ *as truth values in L.*

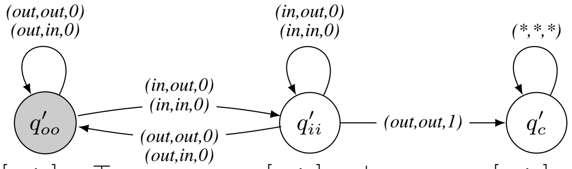
Syntax. Logical operators can often be naturally interpreted as either maximizers or minimizers of the truth values. For example, disjunction ($\varphi \vee \psi$) can be understood as a maximizer ("the most that we can make of either φ or ψ), and conjunction as a minimizer. This extends to existential 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 $\mathcal{C}$. Finally, we add the operator $\preccurlyeq$ representing the lattice order, useful for comparing truth values of formulas. The resulting logic is called mv-ATL$^*_{\preccurlyeq}$ and has the following syntax:

$$\varphi ::= c \mid \mathsf{p} \mid \neg\mathsf{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle \gamma \mid \overline{\langle\!\langle A \rangle\!\rangle}\, \gamma \mid \varphi \preccurlyeq \varphi,$$
$$\gamma ::= \varphi \mid \gamma \wedge \gamma \mid \gamma \vee \gamma \mid \mathsf{X}\, \gamma \mid \gamma\, \mathsf{U}\, \gamma \mid \gamma\, \mathsf{W}\, \gamma.$$

where $\mathsf{p} \in Prop$, $A \subseteq \mathcal{A}$, and $c \in \mathcal{C}$. In what follows, by an *order formula* we shall mean any formula of the form $\varphi_1 \preccurlyeq \varphi_2$. Additionally, we define $\varphi_1 \cong \varphi_2 \equiv (\varphi_1 \preccurlyeq \varphi_2) \wedge (\varphi_2 \preccurlyeq \varphi_1)$.

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

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



$[\mathsf{out}_1] = \top$	$[\mathsf{out}_1] = \bot$	$[\mathsf{out}_1] = \bot$
$[\mathsf{out}_2] = i$	$[\mathsf{out}_2] = i$	$[\mathsf{out}_2] = \bot$
$[\mathsf{in}_1] = \bot$	$[\mathsf{in}_1] = \top$	$[\mathsf{in}_1] = \top$
$[\mathsf{in}_2] = i$	$[\mathsf{in}_2] = i$	$[\mathsf{in}_2] = \top$
$[\mathsf{collision}] = \bot$	$[\mathsf{collision}] = \bot$	$[\mathsf{collision}] = \top$
$[\mathsf{penalty}_1] = \bot$	$[\mathsf{penalty}_1] = \bot$	$[\mathsf{penalty}_1] = u$
$[\mathsf{penalty}_2] = \bot$	$[\mathsf{penalty}_2] = \bot$	$[\mathsf{penalty}_2] = u$
$[\neg\mathsf{penalty}_1] = \top$	$[\neg\mathsf{penalty}_1] = \top$	$[\neg\mathsf{penalty}_1] = u$

Figure 4: Abstract model of the driving agents: M_2

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

$[c]_{M,q} = \sigma(c)$ for $c \in \mathcal{C}$;
$[\mathsf{p}]_{M,q} = V(\mathsf{p}, q)$ and $[\neg\mathsf{p}]_{M,q} = V(\neg\mathsf{p}, q)$ for $\mathsf{p} \in Prop$;
$[\varphi_1 \wedge \varphi_2]_{M,q} = [\varphi_1]_{M,q} \cap [\varphi_2]_{M,q}$;
$[\varphi_1 \vee \varphi_2]_{M,q} = [\varphi_1]_{M,q} \cup [\varphi_2]_{M,q}$;
$[\gamma_1 \wedge \gamma_2]_{M,\lambda}$, $[\gamma_1 \vee \gamma_2]_{M,\lambda}$: analogously;
$[\varphi]_{M,\lambda} = [\varphi]_{M,\lambda[0]}$; $\qquad [\mathsf{X}\, \gamma]_{M,\lambda} = [\gamma]_{M,\lambda[1..\infty]}$;
$[\gamma_1\, \mathsf{U}\, \gamma_2]_{M,\lambda} =$
$\qquad \bigcup_{i=0,1,\ldots} \{\bigcap\{[\gamma_2]_{M,\lambda[i..\infty]}, [\gamma_1]_{M,\lambda[j..\infty]} \mid 0 \leq j < i\}\}$;
$[\gamma_1\, \mathsf{W}\, \gamma_2]_{M,\lambda} = \bigcap_{i=0,1,\ldots}\{[\gamma_1]_{M,\lambda[i..\infty]}\} \cup$
$\qquad \bigcup_{i=0,1,\ldots} \{\bigcap\{[\gamma_2]_{M,\lambda[i..\infty]}, [\gamma_1]_{M,\lambda[j..\infty]} \mid 0 \leq j < i\}\}$;
$[\langle\!\langle A \rangle\!\rangle \gamma]_{M,q} = \bigcup_{s_A \in \Sigma_A} \bigcap_{\lambda \in out(q,s_A)}\{[\gamma]_{M,\lambda}\}$;
$[\overline{\langle\!\langle A \rangle\!\rangle}\, \gamma]_{M,q} = \bigcap_{s_A \in \Sigma_A} \bigcup_{\lambda \in out(q,s_A)}\{[\gamma]_{M,\lambda}\}$;
$[\varphi_1 \preccurlyeq \varphi_2]_{M,q} = \top$ if $[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}$ and $\bot$ otherwise.

EXAMPLE 6 (STATE ABSTRACTION FOR DRIVING AGENTS). *We take the CGS of Figure 1 and apply the metod of state abstraction from [42] with the following mapping:* $\sigma(q_{oo}) = \sigma(q_{oi}) = q'_{oo}$*,* $\sigma(q_{io}) = \sigma(q_{ii}) = q'_{ii}$*, and* $\sigma(q_c) = q'_c$*. That is, situations where agent 1 is outside are represented by the abstract state* q'_{oo}*, those where 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.,* $[\mathsf{in}_2]_{M_2,q'_{oo}} = i$*, as the the truth values of* in_2 *in the original states* q_{oo}, q_{oi} *are in conflict. Moreover, let us assume that the model generating procedure had insufficient time to compute the values of* $\mathsf{penalty}_1$ *and* $\mathsf{penalty}_2$ *in state* q_c *(and hence also* q'_c*). Thus,* $[\mathsf{penalty}_1]_{M_2,q'_c} = [\mathsf{penalty}_2]_{M_2,q'_c} = u$*.*

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 $\mathcal{C}$, for any IQB algebra $\mathcal{L}^+ = ((\mathrm{L}, \leq, \sim), \sigma)$ and any truth value $x \in \sigma(\mathcal{C})$, we will use the notation $\boxed{x}$ to denote any symbol $c \in \mathcal{C}$ such that $\sigma(c) = x$. In other words, we will always have $\sigma(\boxed{x}) = x$ for all $x \in \sigma(\mathcal{C})$.

EXAMPLE 7 (STATE ABSTRACTION, CTD.). *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.,* $[\langle\langle 1\rangle\rangle \mathsf{G}\neg \mathsf{penalty}_1]_{M_2,q'_{oo}} = \top$, *and* $[\langle\langle 1\rangle\rangle \mathsf{F}\,\mathsf{penalty}_1]_{M_2,q'_{oo}} = \bot$, *and* $[\langle\langle 1\rangle\rangle \mathsf{F}\,\mathsf{in}_1]_{M_2,q'_{oo}} = \top$. *We also get* $[\langle\langle 1,2\rangle\rangle \mathsf{F}\,\mathsf{penalty}_1]_{M_2,q'_{oo}} = u$, *which suggests that the abstraction is too fine-grained wrt* $\mathsf{penalty}_1$ *to verify the formula* $\langle\langle 1,2\rangle\rangle \mathsf{F}\,\mathsf{penalty}_1$ *efficiently. Finally, we get* $[\langle\langle 2\rangle\rangle \mathsf{F}\,\mathsf{in}_2]_{M_2,q'_{oo}} = i$, *which suggests that the abstraction is too coarse wrt* in_2, *i.e., it loses the information necessary to compute the value of* $\langle\langle 2\rangle\rangle \mathsf{F}\,\mathsf{in}_2$.

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 $\mathcal{L}$. 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 γ"). 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* $\boxed{u} \preccurlyeq \langle\langle 1\rangle\rangle \mathsf{G}\neg \mathsf{collision}$ *can be used to specify that the truth value of* $\langle\langle 1\rangle\rangle \mathsf{G}\neg \mathsf{collision}$ *is at least u (intuitively: there is no evidence that the formula is false). Moreover,* $\langle\langle 2\rangle\rangle \mathsf{G}\neg \mathsf{penalty}_2 \preccurlyeq \langle\langle 1\rangle\rangle \mathsf{G}\neg \mathsf{penalty}_1$ *says that assuming the ability of agent 1 to avoid penalty is more reasonable than in the case of agent 2. Formula* $\langle\langle atk\rangle\rangle \mathsf{F}(\mathsf{name}_{(007,\mathsf{Bond})} \cong \boxed{i})$ *says that the attacker has a strategy to effect a particular inconsistency in the database. Finally,* $\langle\langle \emptyset\rangle\rangle \mathsf{G}(\varphi \cong \boxed{\top})$ *expresses that formula φ 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 φ a state formula of mv-ATL$^*_{\preccurlyeq}$. Then:*

- *φ is* true in *M, q (written $M, q \models \varphi$) iff $[\varphi]_{M,q} = \top$.*
- *φ is* valid in *M ($M \models \varphi$) iff φ is true in every state of M.*
- *φ is* valid *($\models \varphi$) iff φ is valid in every mv-CGS M.*
- *Additionally, for a path formula γ, we can say that γ holds on run λ 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$^*_{\preccurlyeq}$ agrees with standard ATL* on 2-valued models, unlike the 3-valued version of ATL* from [43].

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

Proof. For any CGS $M = \langle \mathcal{A}, St, Act, d, t, AP, V\rangle$ for ATL*, define mv-CGS $M' = \langle \mathcal{A}, St, Act, d, t, AP, V', \mathcal{L}_2\rangle$, where $\mathcal{L}_2 = (\{\bot, \top\}, \leq, \sim, \sigma)$ with $\sim\bot = \top, \sim\top = \bot$, and $V'(p, q) = \top$ if $q \in V(p)$, $\bot$ otherwise. Then M' is an mv-CGS for mv-ATL$^*_{\preccurlyeq}$, and, obviously, for any ATL* formula φ, $M', \iota \models \varphi$ iff $M, \iota \models \varphi$.

The following is an immediate consequence:

COROLLARY 1. *Each valid formula of mv-ATL$^*_{\preccurlyeq}$ which does not contain the operator $\preceq$ is a valid formula of ATL*.*

Note that the converse does not hold, i.e., a valid formula of ATL* need not be valid in mv-ATL$^*_{\preccurlyeq}$. An example of such a formula is $p \vee \neg p$, which is valid in ATL* but not in mv-ATL$^*_{\preccurlyeq}$. Some important validities that do carry over from classical ATL are the fixpoint equivalences (which in case of multi-valued models can be seen as analogues of Bellman equations for MDPs [9]):

THEOREM 3 (BELLMAN EQUATIONS FOR MV-ATL$_{\preccurlyeq}$). *The following formulas of mv-ATL$_{\preccurlyeq}$ are valid:*

- $\langle\langle A\rangle\rangle \varphi_1 \,\mathsf{U}\, \varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \langle\langle A\rangle\rangle \mathsf{X}\, \langle\langle A\rangle\rangle \varphi_1 \,\mathsf{U}\, \varphi_2$;
- $\langle\langle A\rangle\rangle \varphi_1 \,\mathsf{W}\, \varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \langle\langle A\rangle\rangle \mathsf{X}\, \langle\langle A\rangle\rangle \varphi_1 \,\mathsf{W}\, \varphi_2$.

Proof. The proof is analogous to [33, Proposition 3], and we omit it due to lack of space.

4. MODEL CHECKING mv-ATL$^*_{\preccurlyeq}$

Given a mv-CGS M, a state q, and a mv-ATL$^*_{\preccurlyeq}$ formula φ, the model checking problem consists in computing the value of $[\varphi]_{M,q}$. This can be done in two ways: either by a dedicated algorithm (based e.g. on the fixpoint equations of Prop. 3), or by an efficient reduction to the 2-valued model checking. The latter option has many advantages. First and foremost, it allows us to benefit from the ongoing developments in 2-valued model checking, including symbolic model checking techniques, heuristics, model reduction techniques, etc. In this section, we show how model checking of mv-ATL$^*_{\preccurlyeq}$ can be reduced to the 2-valued variant of this problem.

4.1 Adapting Translations to mv-ATL$^*_{\preccurlyeq}$

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$^*_{\preccurlyeq}$ consisting of formulas without the $\preccurlyeq$ operator. That is, we have the following:

THEOREM 4. *Let $\mathcal{L} = (L, \leq, \sim)$ be an arbitrary finite QB algebra, $\mathcal{L}' = (L', \leq', \sim')$ a subalgebra of $\mathcal{L}$, and $f : L \to L'$ a mapping which preserves arbitrary bounds in $\mathcal{L}$, i.e.,*

$$f(\bigcap_{i\in I} x_i) = \bigcap_{i\in I} f(x_i), \qquad f(\bigcup_{i\in I} x_i) = \bigcup_{i\in I} f(x_i) \tag{2}$$

where I is an arbitrary set of indices. Further, let $\mathcal{L}^+ = (\mathcal{L}, \sigma)$ be an IQB algebra, $M = \langle \mathcal{A}, St, Act, d, t, Prop, \mathcal{V}, \mathcal{L}^+\rangle$ an mv-CGS, and $M' = \langle \mathcal{A}, St, Act, d, t, Prop, \mathcal{V}, \mathcal{L}'^+\rangle$ an mv-CGS obtained out of M by replacing each value $x \in L$ returned by σ or $\mathcal{V}$ in M with $f(x)$, i.e., $\sigma'(c) = f(\sigma(c))$ and $V'(p, q) = f(V(p, q))$, for $c \in \mathcal{C}, q \in St$, and $p \in Prop \cup \overline{Prop}$.

Then, for any state (path) formula φ of mv-ATL* over $\mathcal{L}$ and any state (resp. path) ι, we have

$$[\varphi]_{M',\iota} = x \textit{ iff } [\varphi]_{M,\iota} \in f^{-1}(x) \tag{3}$$

The proof follows easily from the key result given below:

LEMMA 1. *Let a state or path formula φ be such that*

$$[\varphi]_{M,\iota} = \bigcup_{i\in I} \bigcap_{j_i\in J_i} [\varphi_{j_i}]_{M,\iota_{j_i}} \quad or \quad [\varphi]_{M,\iota} = \bigcap_{i\in I} \bigcup_{j_i\in J_i} [\varphi_{j_i}]_{M,\iota_{j_i}}$$

for any mv-CGS M, any states and/or paths ι, ι_{j_i} of M, any countable sets I, J_i, and state (resp. path) formulas of mv-ATL* φ_{j_i} for $j_i \in J_i, i \in I$, such that φ_{j_i} satisfy (3). Then φ satisfies (3) too.

Proof. We consider the case $[\varphi]_{M,\iota} = \bigcup_{i\in I}\bigcap_{j_i\in J_i}[\varphi_{j_i}]_{M,\iota_{j_i}}$; the other case follows by symmetry. As f preserves the bounds, by the assumption on φ we have $f([\varphi]_{M,\iota}) = \bigcup_{i\in I}\bigcap_{j_i\in J_i} f([\varphi_{j_i}]_{M,\iota_{j_i}})$. Each φ_{j_i} satisfies (3), so $f([\varphi]_{M,\iota}) = \bigcup_{i\in I}\bigcap_{j_i\in J_i}[\varphi_{j_i}]_{M',\iota_{j_i}} = [\varphi]_{M',\iota}$, whence $[\varphi]_{M',\iota} = x$ iff $[\varphi]_{M,\iota} \in f^{-1}(x)$, and (3) holds for φ. 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$^*_{\preceq}$, 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 $\mathcal{L}_o = (\mathrm{L}_o, \leq, \sim)$, where $\mathrm{L}_o = \{0, ..., k-1, k, ..., 2k-1\}$, $\leq$ is the usual total order on L_o, and $\sim x = (2k-1) - x$. Of course, in $\mathcal{L}_o$ we have $0 = \bot, 2k-1 = \top$ according to our lattice notation. Then the translation $f : \mathrm{L}_o \to \{0, 2k-1\}$ given by $f(x) = 2k-1$ if $x \geq k$, and $f(x) = 0$ if $x < k$ preserves the bounds in $\mathcal{L}_o$.

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

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

However, for the model M_1 obtained from M with the translation f we get $[p_i]_{M_1,q} = 0$ for $i = 1, 2$ (as $k_i < k$, $f(k_i) = 0$ for $i = 1, 2$), where $[p_2]_{M_1,q} \leq [p_1]_{M_1,q}$, which implies $[\varphi]_{M_1,q} = 2k-1$. Yet, as $f^{-1}(2k-1) = \{k, k+1, ..., 2k-1\}$ we have $[\varphi]_{M,q} = 0 \notin f^{-1}(2k-1)$, 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 $n' < n$, 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:*

C1 $(\forall x_1, x_2 \in L)\, [x_1 < x_2 \;\Rightarrow\; f(x_1) < f(x_2)]$

C2 $(\forall x_1, x_2 \in L)\, [x_1 \bowtie x_2 \;\Rightarrow\; f(x_1) \bowtie f(x_2)]$

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

$$[\varphi]_{M',q} = \top \text{ iff } [\varphi]_{M,q} \in f^{-1}(\top) \tag{4}$$

"$\Rightarrow$": We start by proving the necessity of conditions C1,C2. Assume first f satisfies (4) for order formulas. We should prove that f satisfies Conditions C1, C2 for all such formulas. For what follows, denote $\xi = p_1 \preceq p_2, \psi = p_2 \preceq p_1$, where $p_1, p_2 \in PV$, $p_1 \neq p_2$.

C1: We argue by contradiction. Suppose $x_1, x_2 \in \mathrm{L}, x_1 < x_2$ and $f(x_1) \geq f(x_2)$. Then, as $x_1 < x_2$ implies $x_1 \leq x_2$ and f preserves bounds, we also have $f(x_1) \leq f(x_2)$, which yields $f(x_1) = f(x_2)$.

Now let M be an mv-CGS over $\mathcal{L}$ such that, for some state $q \in St$, we have $V(p_i, q) = x_i$ for $i = 1, 2$, and let M' be the image of M under f. Since $\psi = p_2 \preceq p_1$ and $[p_2]_{M,q} = x_2 > x_1 = [p_1]_{M,q}$, we have $[\psi]_{M,q} = \bot$. However, $[\psi]_{M',q} = \top$, because $[p_2]_{M',q} = f(x_2) = f(x_1) = [p_1]_{M',q}$. As $\bot \notin f^{-1}(\top)$, this contradicts (4).

C2: We again argue by contradiction. Suppose $x_1, x_2 \in L, x_1 \bowtie x_2$ and $\neg(f(x_1) \bowtie f(x_2))$. Without any loss of generality, we can assume that $f(x_1) \leq f(x_2)$. Let mv-CGSs M, M' and state q of M be like in the preceding item. Then, as $[p_1]_{M,q} = x_1 \bowtie x_2 = [p_2]_{M,q}$, we have in particular $[p_1]_{M,q} \not\leq [p_2]_{M,q}$. Since $\xi = p_1 \preceq p_2$, this implies $[\xi]_{M,q} = \bot$. In turn, $[\xi]_{M',q} = \top$, because $[p_1]_{M',q} = f(x_1) \leq f(x_2) = [p_2]_{M',q}$, which again contradicts (4).

"$\Leftarrow$": It remains to prove the sufficiency of conditions C1,C2. We assume that C1, C2 hold, and prove that (4) holds for formulas of the form $\varphi = \varphi_1 \preceq \varphi_2$. We start by proving this result for non-nested order formulas, i.e., we assume that φ_1, φ_2 do not contain $\preceq$. Then, by Theorem 4, (4) holds for φ_1, φ_2, which implies that

$$[\varphi_i]_{M',q} = f([\varphi_i]_{M,q}), \;\; i = 1, 2. \tag{5}$$

"$\Rightarrow$": We begin with the forward implication in (4). Assume that $[\varphi]_{M',q} = \top$. Then $[\varphi_1]_{M',q} \leq [\varphi_2]_{M',q}$. By (5), this implies $f([\varphi_1]_{M,q}) \leq f([\varphi_2]_{M,q}$. We show by contradiction that it implies

$$[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}. \tag{6}$$

Suppose that (6) does not hold, then we have two possible cases:

Case 1: $[\varphi_1]_{M,q} > [\varphi_2]_{M,q}$. Then by C1 we have $f([\varphi_1]_{M,q}) > f([\varphi_2]_{M,q})$, whence from (5) we get $[\varphi_1]_{M',q} > [\varphi_2]_{M_1,q}$ and $[\varphi]_{M',q} = \bot$ — which is a contradiction.

Case 2: $[\varphi_1]_{M,q} \bowtie [\varphi_2]_{M,q}$. Then $f([\varphi_1]_{M,q}) \bowtie f([\varphi_2]_{M,q})$ by C2, whence from (5) we get $\neg([\varphi_1]_{M',q} \leq [\varphi_2]_{M',q})$. Consequently, $[\varphi]_{M',q} = \bot$ — which is again a contradiction.

Thus (6) above holds, whence $[\varphi]_{M,q} = \top \in f_1(\top)$, and the forward implication in (4) holds.

"$\Leftarrow$": The final step is proving the backward implication in (4). Assume that $[\varphi]_{M,q} = f^{-1}(\top)$. As $[\varphi]_{M,q} \in \{\bot, \top\}$ and $f(\bot) \neq \top$ by the preservation of bounds by f and the non-triviality of L, L', we obtain $[\varphi]_{M,q} = \top$, whence $[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}$. Since f preserves bounds, this implies $f([\varphi_1]_{M,q}) \geq f([\varphi_2]_{M,q})$, whence from (5) we obtain $[\varphi_1]_{M_1,q} \leq [\varphi_2]_{M',q}$. This yields $[\varphi]_{M',q} = \top$, whence the backward implication in (4) holds, too.

Now assume (4) holds for order formulas with $\preceq$ nested at most k times, and assume φ is an order formula with $\preceq$ nested $k+1$ times. Then $\varphi = \varphi_1 \preceq \varphi_2$, where $\preceq$ is nested at most k times in φ_1, φ_2. Consequently, by the inductive assumption (6) holds for φ_1, φ_2, and repeating the proof given above for order formulas without nesting of $\preceq$ we can show that (4) holds for φ too.

This completes the proof of the sufficiency of C1, C2 for all order formulas, and the proof of Lemma 3. From Lemma 3 we can easily derive by induction the following general result:

THEOREM 5. *Let the QB algebra $\mathcal{L} = (L, \leq, \sim)$, its subalgebra $\mathcal{L}' = (L', \leq', \sim')$, $f : L \to L'$, and mv-CGS M, M' satisfy the assumptions of Theorem 4. Then the condition (3) of that Theorem:*

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

is satisfied for all formulas of mv-ATL$^*_{\preccurlyeq}$ over $\mathcal{L}$ 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 $\mathcal{L}$. An important consequence of that fact is:

COROLLARY 2. *Given a QB algebra $\mathcal{L} = (L, \leq, \sim)$ and its subalgebra $\mathcal{L}' = (L', \leq', \sim')$, any function $f : L \to L'$ 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. $x_1 \bowtie x_2$. Then $f(x_1) \bowtie f(x_2)$ 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 mv-ATL$^*_{\preccurlyeq}$ model checking to 2-valued model checking is possible.*

4.3 Recursive Model Checking of mv-ATL$^*_{\preccurlyeq}$

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

C1' $x_1 < x_2 \;\Rightarrow\; f(x_1) < f(x_2)$

C2' $x_1 \bowtie x_2 \;\Rightarrow\; f(x_1) > f(x_2)$

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

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

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

Thus it remains to consider the case of $\preceq$. Assume $\psi = \psi_1 \preceq \psi_2$, where (8) holds for ψ_1, ψ_2. Since ψ is an order formula, according to what we have already noted in the proof of Lemma 2, proving (8) for ψ reduces to showing

$$[\psi]_{M',q} = \top \quad \text{iff} \quad [\psi]_{M,q} \in f^{-1}(\top) \tag{9}$$

Note that since ψ_1, ψ_2 are in $Sub(\varphi)$, then C1', C2' hold for $x_i = [\varphi_i]_{M,q}, i = 1, 2$. By the inductive assumption, we also have

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

<u>**"⇒"**</u>: We begin with the forward implication in (9). Assume that $[\psi]_{M',q} = \top$. 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

$$[\psi_1]_{M,q} \leq [\psi_2]_{M,q} \tag{11}$$

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

Case 1: $[\psi_1]_{M,q} > [\psi_2]_{M,q}$. Then 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 $[\psi]_{M',q} = \bot$, which is a contradiction.

Case 2: $[\psi_1]_{M,q} \bowtie [\psi_2]_{M,q}$. 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 $[\psi]_{M,q} = \top \in f^{-1}(\top)$, and the forward implication in (9) holds.

<u>**"⇐"**</u>: The final step consists in proving the backward implication in (9). Assume that $[\psi]_{M,q} = f^{-1}(\top)$. As $[\psi]_{M,q} \in \{\bot, \top\}$ and $f(\bot) \neq \top$ by the preservation of bounds by f and the non-triviality of $\mathcal{L}, \mathcal{L}'$, we get $[\psi]_{M,q} = \top$, and consequently $[\psi_1]_{M,q} \leq [\psi_2]_{M,q}$. 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 $[\psi]_{M',q} = \top$, whence the backward implication in (9) holds, too.

REMARK 2. *Notice that Condition C2' is necessary, but C1' is not. Moreover, both are necessary iff φ is symmetric w.r.t. to $\preceq$, i.e., $(\varphi_2 \preceq \varphi_1) \in Sub(\varphi)$ whenever $(\varphi_1 \preceq \varphi_2) \in Sub(\varphi)$.*

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 ($\mathcal{JI}(\mathcal{L})$) [26, 12, 38, 37] can be applied to a formula φ of mv-ATL$^*_{\preccurlyeq}$ and an mv-CGS M, provided the assumptions of Theorem 6 are satisfied.

By (1), for each $x \in$ L we have $x = \bigsqcup(\mathcal{JI}(\mathcal{L}) \cap \downarrow x)$. Let M^l be the model obtained using the translation f_l. Therefore, according to Theorem 6: $[\varphi]_{M_1,\iota} = x$ iff $[\varphi]_{M,\iota} \in f_l^{-1}(x)$ whence $[\varphi]_{M_1,\iota} = \top$ iff $[\varphi]_{M,\iota} \in \uparrow l$. Thus,

$$[\varphi]_{M,\iota} = \bigsqcup\{l \in \mathcal{JI}(\mathcal{L}) \;\mid\; [\varphi]_{M_1,\iota} = \top\}. \tag{12}$$

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

The algorithm is recursive. For each order subformula $\psi = \psi_1 \leq \psi_2$ of φ such that $\psi_1, \psi_2 \in$ mv-ATL*, we define a fresh propositional variable p_ψ. Then, for each state q of model M' we have $V'(p_\psi, q) = \top$ iff $[\psi_1]_{M,q} \leq [\psi_2]_{M,q}$, and $V'(\neg p_\psi, q) = \bot$ iff $\neg([\psi_1]_{M,q} \leq [\psi_2]_{M,q})$, 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 mv-ATL$^*_{\preccurlyeq}$ 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$^*_{\preccurlyeq}$ (resp. mv-ATL$_{\preccurlyeq}$) 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$^*_{\preccurlyeq}$ is* $\mathbf{2EXPTIME}$*-complete, and model checking mv-ATL$_{\preccurlyeq}$ is* $\mathbf{P}$*-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$^*_{\preccurlyeq}$. 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 $s_a : St \rightarrow Act$ 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$_{\preccurlyeq}$, 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, $\mathbf{\Delta_2^P}$- to $\mathbf{PSPACE}$-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$^*_{\preccurlyeq}$.

Multivalued ATL* with imperfect information. Let us extend mv-CGS with indistinguishability relations $\sim_1, \dots, \sim_k$, one per agent in $\mathcal{A}$. Memoryless strategies with imperfect information (*ir* strategies, for short) are functions $s_a : St \rightarrow Act$ such that $q \sim_a q'$ implies $s_a(q) = s_a(q')$. Perfect recall strategies with imperfect information (shortly: *iR* strategies) are functions $s_a : St^+ \rightarrow Act$ st. $q_0 \sim_a q'_0, \dots, q_n \sim_a q'_n$ implies $s_a(q_0 \dots q_n) = s_a(q'_0 \dots q'_n)$. Again, collective strategies for $A \subseteq \mathcal{A}$ are tuples of individual strategies for $a \in A$. We denote them by Σ_A^{ir} and Σ_A^{iR}, respectively.

The semantics of mv-ATL$^*_{\mathfrak{S}\preccurlyeq}$, parameterized by the type of strategies $\mathfrak{S} = IR, Ir, ir, iR$, can be defined by replacing the clause for $\langle\langle A\rangle\rangle\gamma$ from Section 3.1 as follows:

$$[\langle\langle A\rangle\rangle\gamma]_{M,q} \;=\; \textstyle\bigcup_{s_A\in\Sigma_A^{\mathfrak{S}}} \bigcap_{\lambda\in out(q,s_A)}\{[\gamma]_{M,\lambda}\}.$$ [4]

EXAMPLE 9 (ABSTRACTION FOR IMPERFECT INFO). *Take model M_1 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: $q_{oo} \sim_2 q_{io}$ and $q_{oi} \sim_2 q_{ii}$. If we further apply the state abstraction σ from Example 6, we obtain model M_3 that adds $q'_{oo} \sim_2 q'_{ii}$ to mv-CGS M_2. Now, e.g., $[\langle\langle 2\rangle\rangle \mathsf{F}(\mathsf{out}_2 \wedge \neg\mathsf{penalty}_2)]_{M_3,q'_{oo}} = i$.*

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 Σ_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$^*_{\mathfrak{S}\preccurlyeq}$ 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$_{ir\preccurlyeq}$ is* $\mathbf{NP}$*-complete, and model checking mv-ATL$^*_{ir\preccurlyeq}$ is* $\mathbf{PSPACE}$*-complete in the size of the model and the formula, and the number of logical values.*

THEOREM 10. *Model checking mv-ATL$^*_{iR\preccurlyeq}$ and mv-ATL$_{iR\preccurlyeq}$ is undecidable in general. For the fragment of mv-ATL$_{iR\preccurlyeq}$ with singleton coalitions only, model checking is* $\mathbf{EXPTIME}$*-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$^*_{\preccurlyeq}$, where the truth values are drawn from an arbitrary De Morgan algebra. We also argue that multivalued model checking of mv-ATL$^*_{\preccurlyeq}$ 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$_{\preccurlyeq}$ in perfect information scenarios.

Our translation method for mv-ATL$^*_{\preccurlyeq}$ formulas allows us to benefit from the efficient 2-valued model checking algorithms. If a translation for a formula φ cannot be defined, our model checking method becomes less efficient, as it requires replacing all the order subformulas of φ 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 *Procedings of Workshop*

[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.

on Knowledge and Games, pages 34–42, 2004.

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

[3] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[4] A. M. B. Aminof, O. Kupferman. Improved model checking of hierarchical systems. *Information Computation*, 210:68–86, 2012.

[5] C. Baier, M. Kwiatkowska, and G. Norman. Computing probability bounds for linear time formulas over concurrent probabilistic systems. *Electronic Notes in Theoretical Computer Science*, 21(19), 1999.

[6] T. Ball and O. Kupferman. An abstraction-refinement framework for multi-agent systems. In *Proceedings of LICS*, pages 379–388. IEEE Computer Society, 2006.

[7] A. Bauer, M. Leucker, and C. Schallhart. Monitoring of real-time properties. In *Proceedings of FSTTCS*, volume 4337 of *Lecture Notes in Computer Science*, pages 260–272. Springer, 2006.

[8] A. Bauer, M. Leucker, and C. Schallhart. The good, the bad, and the ugly, but how ugly is ugly? In *Proceedings of RV*, volume 4839 of *Lecture Notes in Computer Science*, pages 126–138. Springer, 2007.

[9] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[10] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In *Proceedings of TACAS*, volume 1579 of *Lecture Notes in Computer Science*, pages 193–207. Springer, 1999.

[11] G. Birkhoff. *Lattice Theory (2nd edition)*. American Mathematical Society, 1948.

[12] G. Bruns and P. Godefroid. Model checking with multi-valued logics. In *ICALP*, pages 281–293, 2004.

[13] N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability: Complexity. In M. Dastani, K. Hindriks, and J.-J. Meyer, editors, *Specification and Verification of Multi-Agent Systems*, pages 125–159. Springer, 2010.

[14] N. Bulling and W. Jamroga. What agents can probably enforce. *Fundamenta Informaticae*, 93(1-3):81–96, 2009.

[15] N. Bulling and W. Jamroga. Comparing variants of strategic ability: How uncertainty and memory influence general properties of games. *Journal of Autonomous Agents and Multi-Agent Systems*, 28(3):474–518, 2014.

[16] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. Improving the model checking of strategies under partial observability and fairness constraints. In *Formal Methods and Software Engineering*, volume 8829 of *Lecture Notes in Computer Science*, pages 27–42. 2014.

[17] P. Cermak, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A model checker for the verification of strategy logic specifications. In *Proc. of CAV'14*, volume 8559 of *LNCS*, pages 525–532. Springer-Verlag, 2014.

[18] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *Proceedings of TACAS*, volume 7795 of *LNCS*, pages 185–191. Springer, 2013.

[19] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[20] L. de Alfaro, M. Faella, T. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. *Theoretical Computer Science*, 345:139–170, 2005.

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

[22] M. Fitting. Many-valued modal logics. *Fundamenta Informaticae*, 15(3-4):335–350, 1991.

[23] M. Fitting. Many-valued modal logics II. *Fundamenta Informaticae*, 17:55–73, 1992.

[24] P. Godefroid and R. Jagadeesan. Automatic abstraction using generalized model checking. In *Proceedings of CAV*, volume 2404 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 2002.

[25] P. Godefroid and R. Jagadeesan. On the expressiveness of 3-valued models. In *Proceedings of VMCAI'03*, volume 2575 of *LNCS*, pages 206–222. Springer-Verlag, 2003.

[26] A. Gurfinkel and M. Chechik. Multi-valued model checking via classical model checking. In *Proceedings of CONCUR*, volume 2761 of *LNCS*, pages 266–280. Springer-Verlag, 2003.

[27] X. Huang and R. van der Meyden. Symbolic model checking episte- mic strategy logic. In *Proceedings of AAAI*, pages 1426–1432, 2014.

[28] M. Huth, R. Jagadeesan, and D. A. Schmidt. A domain equation for refinement of partial systems. *Mathematical Structures in Computer Science*, 14:469–505, 2004.

[29] M. Huth and M. Z. Kwiatkowska. Quantitative analysis and model checking. In *Logic in Computer Science*, pages 111–122, 1997.

[30] M. Huth and S. Pradhan. Consistent partial model checking. In *Proc. of the Workshop on Domains VI*, volume 73 of *ENTCS*, pages 45–85. Elsevier, 2004.

[31] W. Jamroga. Some remarks on alternating temporal epistemic logic. In B. Dunin-Keplicz and R. Verbrugge, editors, *Proceedings of FAMAS*, pages 133–140, 2003.

[32] W. Jamroga. A temporal logic for Markov chains. In *Proceedings of AAMAS'08*, pages 697–704, 2008.

[33] W. Jamroga. A temporal logic for stochastic multi-agent systems. In *Proceedings of PRIMA'08*, volume 5357 of *LNCS*, pages 239–250, 2008.

[34] W. Jamroga and T. Ågotnes. Constructive knowledge: What agents can achieve under incomplete information. *Journal of Applied Non-Classical Logics*, 17(4):423–475, 2007.

[35] W. Jamroga and J. Dix. Model checking abilities of agents: A closer look. *Theory of Computing Systems*, 42(3):366–410, 2008.

[36] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2–3):185–219, 2004.

[37] B. Konikowska and W. Penczek. Reducing model checking from multi-valued CTL* to CTL*. In *Proceedings CONCUR*, volume 2421 of *LNCS*, pages 226–239. Springer-Verlag, 2002.

[38] B. Konikowska and W. Penczek. Model checking for multi-valued computation tree logics. In M. Fitting and E. Orłowska, editors, *Beyond Two: Theory and Applications of Multiple Valued Logic*, pages 193–210. Physica-Verlag, 2003.

[39] B. Konikowska and W. Penczek. Model checking of multivalued logic of knowledge and time. In *Proceedings of AAMAS*, pages 169–176. ACM, 2006.

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

probabilistic symbolic model checker. In *Proceedings of TOOLS*, volume 2324 of *Lecture Notes in Computer Science*, pages 200–204. Springer, 2002.

[41] A. Lluch-Lafuente and U. Montanari. Quantitative μ-calculus and CTL based on constraint semirings. *Electr. Notes Theor. Comput. Sci.*, 112:37–59, 2005.

[42] A. Lomuscio and J. Michaliszyn. An abstraction technique for the verification of multi-agent systems against ATL specifications. In *Proceedings of KR*. AAAI Press, 2014.

[43] A. Lomuscio and J. Michaliszyn. Verifying multi-agent systems by model checking three-valued abstractions. In *Proc. of the 2015 Inter. Conf. on Autonomous Agents and Multiagent Systems, (AAMAS 2015)*, pages 189–198, 2015.

[44] J. Maluszynski and A. Szalas. Logical foundations and complexity of 4QL, a query language with unrestricted negation. *Journal of Applied Non-Classical Logics*, 21(2):211–232, 2011.

[45] J. Maluszynski and A. Szalas. Partiality and inconsistency in agents' belief bases. In *Proceedings of KES-AMSTA*, volume 252 of *Frontiers in Artificial Intelligence and Applications*, pages 3–17. IOS Press, 2013.

[46] H. Pan, Y. Li, Y. Cao, and Z. Ma. Model checking Computation Tree Logic over finite lattices. *TCS*, to appear, 2015.

[47] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. In *Proceedings of AAMAS'03*, pages 209–216, New York, NY, USA, 2003. ACM Press.

[48] J. Pilecki, M. Bednarczyk, and W. Jamroga. Synthesis and verification of uniform strategies for multi-agent systems. In *Proceedings of the 15th Workshop on Computational Logic in Multi-Agent Systems CLIMA XV*, volume 8624 of *LNCS*, pages 166–182. Springer, 2014.

[49] H. Schnoor. Strategic planning for probabilistic games with incomplete information. In *Proceedings of AAMAS'10*, pages 1057–1064, 2010.

[50] P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electr. Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

[51] S. Shoham and O. Grumberg. Multi-valued model checking games. *J. Comput. Syst. Sci.*, 78(2):414–429, 2012.