



TU Clausthal

Clausthal University of Technology

A Simpler Semantics for Abilities under Uncertainty

Wojciech Jamroga¹, Thomas Ågotnes², and Wiebe
van der Hoek³

IfI Technical Report Series

IfI-08-01



Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany

Editor of the series: Jürgen Dix

Technical editor: Wojciech Jamroga

Contact: wjamroga@in.tu-clausthal.de

URL: <http://www.in.tu-clausthal.de/forschung/technical-reports/>

ISSN: 1860-8477

The IfI Review Board

Prof. Dr. Jürgen Dix (Theoretical Computer Science/Computational Intelligence)

Prof. Dr. Klaus Ecker (Applied Computer Science)

Prof. Dr. Barbara Hammer (Theoretical Foundations of Computer Science)

Prof. Dr. Kai Hormann (Computer Graphics)

Prof. Dr. Gerhard R. Joubert (Practical Computer Science)

apl. Prof. Dr. Günter Kemnitz (Hardware and Robotics)

Prof. Dr. Ingbert Kupka (Theoretical Computer Science)

Prof. Dr. Wilfried Lex (Mathematical Foundations of Computer Science)

Prof. Dr. Jörg Müller (Business Information Technology)

Prof. Dr. Niels Pinkwart (Business Information Technology)

Prof. Dr. Andreas Rausch (Software Systems Engineering)

apl. Prof. Dr. Matthias Reuter (Modeling and Simulation)

Prof. Dr. Harald Richter (Technical Computer Science)

Prof. Dr. Gabriel Zachmann (Computer Graphics)

A Simpler Semantics for Abilities under Uncertainty

Wojciech Jamroga¹, Thomas Ågotnes², and Wiebe van der Hoek³

¹ Department of Informatics, Clausthal University of Technology, Germany

² Department of Computer Engineering, Bergen University College, Norway

³ Department of Computer Science, University of Liverpool, UK

wjamroga@in.tu-clausthal.de, tag@hib.no, wiebe@csc.liv.ac.uk

Abstract

Modal logics of strategic ability form one of the fields where logic and game theory can successfully meet. However, defining a good semantics of abilities under imperfect information proved to be a difficult task. In our previous work, we have proposed a logic which – we believe – captures such abilities in an elegant and general way. However, the semantics of the logic is fairly non-standard, as it defines the truth of formulae for sets of states (rather than single states). In this paper, we present a standard, state-based semantics for a relevant subset of the logic. We also discuss explicit operators for complement, union, intersection, and transitive closure of epistemic relations. The resulting language is studied on a variant of the coordinated attack problem.

1 Introduction

Modal logics of strategic ability [1, 2, 19, 20] form one of the fields where logic and game theory can successfully meet. The logics have clear possible worlds semantics, are axiomatizable, and have some interesting computational properties. Moreover, they are underpinned by a clear and intuitively appealing conceptual machinery for modeling and reasoning about systems that involve multiple autonomous agents. Alternating-time Temporal Logic (ATL) [1, 2], is probably the most important logic of strategic ability that has emerged in recent years. However, ATL considers only agents that possess perfect information about the current state of the world, and such agents seldom exist in reality. A combination of ATL and epistemic logic, called *Alternating-time Temporal Epistemic Logic* (ATEL), was introduced to enable reasoning about agents acting under imperfect information [26]. Still, it has been pointed out in several places [11, 14, 15, 8] that the meaning of ATEL formulae can be counterintuitive. Most importantly, an agent’s ability

to achieve property φ should imply that the agent has enough control and knowledge to *identify* and *execute* a strategy that enforces φ .

In [13], we have proposed Constructive Strategic Logic (CSL) which – we believe – allows to capture knowledge and abilities of agents with imperfect information in an elegant and general way. However, the semantics of CSL is fairly non-standard. In this paper, we present a standard, state-based semantics for CSL^{nf} , a subset of CSL formulae in *constructive normal form* which is no less expressive than the full CSL. Although the more rigid structure prevents e.g. some typical axioms from being formulae of CSL^{nf} , they can be rewritten to equivalent CSL^{nf} counterparts.

Having defined the new semantics, we observe that a sequence of epistemic operators denotes a relation which is the sequential composition of the corresponding epistemic relations. We suggest that other ways of combining individual views of agents in a group are also meaningful. To this end, we introduce explicit operators that yield complement, union, and intersection of relations (analogous to *role constructors* used in description logics [3]). Moreover, we propose to use the transitive closure operator of dynamic logic [21, 9, 24]. We show that the typical collective knowledge operators can be naturally defined by these constructors from individual knowledge – but there are more possible combinations that can be used. We study the resulting language on a variant of the coordinated attack problem. Finally, we point out that the extension does not increase the model checking complexity.

2 What Agents Can Achieve

2.1 ATL: Ability in Perfect Information Games

ATL [1, 2] can be understood as a generalization of the branching time temporal logic CTL [4, 5], in which path quantifiers are replaced with so called *cooperation modalities*. The formula $\langle\langle A \rangle\rangle \varphi$, where A is a coalition of agents, expresses that A have a collective strategy to enforce φ . ATL formulae include temporal operators: “ \bigcirc ” (“in the next state”), \Box (“always from now on”) and \mathcal{U} (“until”). Operator \Diamond (“now or sometime in the future”) can be defined as $\Diamond \varphi \equiv \top \mathcal{U} \varphi$. Similarly to CTL, every occurrence of a temporal operator is immediately preceded by exactly one cooperation modality.¹ The broader language of ATL^* , in which no such restriction is imposed, is not discussed in this paper.

Let A be a set of agents. Formally, the recursive definition of ATL formulae is:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \bigcirc \varphi \mid \langle\langle A \rangle\rangle \Box \varphi \mid \langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi$$

¹ The logic to which such a syntactic restriction applies is sometimes called “*vanilla*” ATL (resp. “*vanilla*” CTL etc.).

Example ATL properties are: $\langle\langle jamesbond \rangle\rangle \Diamond \text{win}$ (James Bond has an infallible plan to eventually win), and $\langle\langle jamesbond, bondsgirl \rangle\rangle \text{fun } \mathcal{U} \text{shot} - \text{at}$ (Bond and his current girlfriend have a collective way of having fun until someone shoots at them).

A number of semantics have been defined for ATL, most of them equivalent [6, 7]. In this paper, we use a variant of *concurrent game structures* as models. A *concurrent game structure* (CGS) is a tuple $M = \langle \text{Agt}, St, \Pi, \pi, Act, d, o \rangle$ which includes a nonempty finite set of all agents $\text{Agt} = \{1, \dots, k\}$, a nonempty set of states St , a set of atomic propositions Π , a valuation of propositions $\pi : \Pi \rightarrow 2^{St}$, and a set of (atomic) actions Act . Function $d : \text{Agt} \times St \rightarrow (2^{Act} \setminus \emptyset)$ defines nonempty sets of actions available to agents at each state, and o is a (deterministic) transition function that assigns the outcome state $q' = o(q, \alpha_1, \dots, \alpha_k)$ to state q and a tuple of actions $\langle \alpha_1, \dots, \alpha_k \rangle$, $\alpha_i \in d(i, q)$, that can be executed by Agt in q . A (memoryless) *strategy* s_a of agent a is a conditional plan that specifies what a is going to do for every possible situation: $s_a : St \rightarrow Act$ such that $s_a(q) \in d(a, q)$. A *collective strategy* S_A for a group of agents A is a tuple of strategies, one per agent from A .

A *path* Λ in model M is an infinite sequence of states that can be effected by subsequent transitions, and refers to a possible course of action (or a possible computation) that may occur in the system; by $\Lambda[i]$, we denote the i th position on path Λ . Function $out(q, S_A)$ returns the set of all paths that may result from agents A executing strategy S_A from state q onward:

$$out(q, S_A) = \{ \lambda = q_0 q_1 q_2 \dots \mid q_0 = q \text{ and for every } i = 1, 2, \dots \text{ there exists a tuple of agents' decisions } \langle \alpha_1, \dots, \alpha_k \rangle \text{ such that } \alpha_a = S_A(a)(q_{i-1}) \text{ for each } a \in A, \text{ and } \alpha_a \in d(a, q_{i-1}) \text{ for each } a \notin A, \text{ and } o(q_{i-1}, \alpha_1, \dots, \alpha_k) = q_i \}.$$

The semantics of ATL formulae can be given via the following clauses:

$$\begin{aligned} M, q \models p & \text{ iff } p \in \pi(q) & (\text{for } p \in \Pi); \\ M, q \models \neg \varphi & \text{ iff } M, q \not\models \varphi; \\ M, q \models \varphi \wedge \psi & \text{ iff } M, q \models \varphi \text{ and } M, q \models \psi; \\ M, q \models \langle\langle A \rangle\rangle \bigcirc \varphi & \text{ iff there is a collective strategy } S_A \text{ such that, for every } \\ & \Lambda \in out(q, S_A), \text{ we have } M, \Lambda[1] \models \varphi; \\ M, q \models \langle\langle A \rangle\rangle \Box \varphi & \text{ iff there exists } S_A \text{ such that, for every } \Lambda \in out(q, S_A), \text{ we} \\ & \text{have } M, \Lambda[i] \models \varphi \text{ for every } i \geq 0; \\ M, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi & \text{ iff there exists } S_A \text{ such that for every } \Lambda \in out(q, S_A) \text{ there} \\ & \text{is an } i \geq 0, \text{ for which } M, \Lambda[i] \models \psi, \text{ and } M, \Lambda[j] \models \varphi \text{ for every } 0 \leq j < i. \end{aligned}$$

2.2 ATL with Epistemic Logic

Real-life agents seldom possess complete information about the current state of the world. On the other hand, imperfect information and knowledge are

handled in epistemic logic in a natural way. A combination of ATL and epistemic logic, called *Alternating-time Temporal Epistemic Logic* (ATEL) was introduced to enable reasoning about agents acting under imperfect information.

ATEL [26] enriches the picture with an epistemic component, adding to ATL operators for representing agents' knowledge: $K_a\varphi$ reads as "agent a knows that φ ". Additional operators $E_A\varphi$, $C_A\varphi$, and $D_A\varphi$, where A is a set of agents, refer to *mutual knowledge* ("everybody knows"), *common knowledge*, and *distributed knowledge* among the agents from A . Models for ATEL extend concurrent game structures with epistemic accessibility relations $\sim_1, \dots, \sim_k \subseteq Q \times Q$ (one per agent) for modeling agents' uncertainty.² We call such models *concurrent epistemic game structures* (CEGS).³ Agent a 's epistemic relation is meant to encode a 's inability to distinguish between the (global) system states: $q \sim_a q'$ means that, while the system is in state q , agent a cannot determine whether it is in q or q' . Then, the semantics of K_a is defined as:

$$M, q \models K_a\varphi \text{ iff } M, q' \models \varphi \text{ for every } q' \text{ such that } q \sim_a q'.$$

Relations \sim_A^E , \sim_A^C and \sim_A^D , used to model group epistemics, are derived from the individual relations of agents from A . First, \sim_A^E is the union of relations \sim_a , $a \in A$. Next, \sim_A^C is defined as the transitive closure of \sim_A^E . Finally, \sim_A^D is the intersection of all the \sim_a , $a \in A$. The semantics of group knowledge can be defined as below (for $\mathcal{K} = C, E, D$):

$$M, q \models \mathcal{K}_A\varphi \text{ iff } M, q' \models \varphi \text{ for every } q' \text{ such that } q \sim_A^{\mathcal{K}} q'.$$

Note that $K_a \equiv C_{\{a\}} \equiv E_{\{a\}} \equiv D_{\{a\}}$, so individual knowledge operators K_a are in fact redundant.

It has been pointed out in several places that such a straightforward combination of ATL and epistemic logic can be counterintuitive [11, 14, 15], as the following example shows.

Example 1 *There is a broken lightbulb and a switch in the room. Agent i has two available actions: turn the switch or replace the bulb. Before replacing it, he needs to make sure that the electricity is off, otherwise he can get a shock. However, he does not know in which position the switch is "on", and in which it is "off".*

Intuitively, there is no strategy that guarantees that the agent replaces the bulb safely. However, the ATL formula $\langle\langle i \rangle\rangle \text{safe} \mathcal{U} (\text{safe} \wedge \text{bulbOK})$ holds for both possible states of the switch. Suppose first that the switch is off: then, the strategy "replace the bulb" clearly achieves the goal. Likewise, there is a successful strategy for the switch being on, namely "turn the switch off and then replace the bulb". Even the ATEL formula $K_i \langle\langle i \rangle\rangle \text{safe} \mathcal{U} (\text{safe} \wedge \text{bulbOK})$ holds, because i has a strategy to enforce $\text{safe} \mathcal{U} (\text{safe} \wedge \text{bulbOK})$ for each state that he considers possible (switch on/switch off) – and the semantics of ATEL does not require these two strategies to be identical.

² The relations are assumed to be equivalences.

³ Additionally, we will assume that CEGS are *uniform*, i.e., agents have the same choices in indistinguishable states ($q \sim_a q'$ implies $d_a(q) = d_a(q')$).

Most importantly, one would expect that an agent’s ability to achieve property φ should imply that the agent has enough control and knowledge to *identify* and *execute* a strategy that enforces φ (cf. also [23]). The problem is closely related to the distinction between knowledge *de re* and knowledge *de dicto* [22, 17, 18, 28]. One can naturally distinguish at least four different levels of strategic ability (cf. [14]):

1. Agent a has a strategy “*de re*” to enforce φ , i.e., he has an executable winning strategy and knows the strategy (he “knows how to play”);
2. Agent a has a strategy “*de dicto*” to enforce φ (i.e., he knows only that *some* executable winning strategy is available);
3. Agent a has an executable strategy to enforce φ (but not necessarily even knows about it);
4. Agent a may *happen* to behave in such a way that φ is enforced. However, the behavior can have no executable specification (i.e., there might be no uniform strategy that describes it).

Unfortunately, ATEL enables to express only ability of type (4). A number of logics were proposed to capture abilities on various levels [11, 14, 23, 15, 27, 10], yet none of them seemed the ultimate definitive solution.

2.3 Constructive Strategic Logic

In [13], we have proposed Constructive Strategic Logic (CSL) which, as we believe, allows to address the interplay between knowledge and abilities under imperfect information in a way that is both elegant and general. By “general”, we mean that it allows to characterize as many meaningful levels of strategic ability as possible (and at least as many as ATOL [14]). In particular, it should enable the distinction between various readings of knowing a strategy “*de re*” and “*de dicto*” for individual as well as collective players. By “elegant”, we mean that it allows us to express various levels of ability by *composition* of epistemic operators with strategic operators, instead of assigning a specialized modality to every conceivable combination.

To achieve this, we built our proposal around new epistemic operators for what we call “practical” or “constructive” knowledge. The idea was inspired by the tradition of *constructivism* which argues that one must find (or “construct”) a mathematical object to prove that it exists [25]. Agents A *constructively know* that $\langle\langle B \rangle\rangle \varphi$ if they can present a strategy for B that guarantees achieving φ . In our semantics formulae are interpreted over *sets of states* rather than single states. This reflects the intuition that the “constructive” ability to enforce φ means that the agents in question have a single strategy that brings about φ from *all* subjectively possible initial situations. We

write $M, Q \models \langle\langle A \rangle\rangle \varphi$ to express the fact that A must have a strategy which is successful from all states in set Q . The constructive knowledge operators $\mathbb{K}_a, \mathbb{E}_A, \mathbb{C}_A, \mathbb{D}_A$ yield sets of states for which a single evidence (i.e., a successful strategy) should be presented.

The language of Constructive Strategic Logic can be defined as below:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \bigcirc \varphi \mid \langle\langle A \rangle\rangle \Box \varphi \mid \langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi \mid \mathbb{C}_A \varphi \mid \mathbb{E}_A \varphi \mid \mathbb{D}_A \varphi.$$

Models are concurrent epistemic game structures again, and we restrict strategies to so called *uniform* ones. Strategy s_a is uniform iff $q \sim_a q'$ implies $s_a(q) = s_a(q')$; a collective strategy is uniform iff it consists of only uniform individual strategies. The notion of a formula φ being satisfied by a (non-empty) set of states Q in a model M , written $M, Q \models \varphi$, can be defined as follows. Let $\text{img}(q, \mathcal{R})$ be the image of state q with respect to binary relation \mathcal{R} , i.e., the set of all states q' such that $q\mathcal{R}q'$. Moreover, we use $\text{out}(Q, S_A)$ as a shorthand for $\bigcup_{q \in Q} \text{out}(q, S_A)$, and $\text{img}(Q, \mathcal{R})$ as a shorthand for $\bigcup_{q \in Q} \text{img}(q, \mathcal{R})$. The new semantics is given through the following clauses.

$$\begin{aligned} M, Q \models p & \quad \text{iff } p \in \pi(q) \text{ for every } q \in Q; \\ M, Q \models \neg\varphi & \quad \text{iff } M, Q \not\models \varphi; \\ M, Q \models \varphi \wedge \psi & \quad \text{iff } M, Q \models \varphi \text{ and } M, Q \models \psi; \\ M, Q \models \langle\langle A \rangle\rangle \bigcirc \varphi & \quad \text{iff there exists memoryless uniform } S_A \text{ such that, for} \\ & \quad \text{every } \Lambda \in \text{out}(Q, S_A), \text{ we have that } M, \{\Lambda[1]\} \models \varphi; \\ M, Q \models \langle\langle A \rangle\rangle \Box \varphi & \quad \text{iff there exists } S_A \text{ such that, for every } \Lambda \in \text{out}(Q, S_A) \text{ and} \\ & \quad i \geq 0, \text{ we have } M, \{\Lambda[i]\} \models \varphi; \\ M, Q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi & \quad \text{iff there exists } S_A \text{ such that, for every } \Lambda \in \text{out}(Q, S_A), \\ & \quad \text{there is an } i \geq 0 \text{ for which } M, \{\Lambda[i]\} \models \psi \text{ and } M, \{\Lambda[j]\} \models \varphi \text{ for every} \\ & \quad 0 \leq j < i. \\ M, Q \models \hat{\mathcal{K}}_A \varphi & \quad \text{iff } M, \text{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi \text{ (where } \hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D} \text{ and } \mathcal{K} = C, E, D, \\ & \quad \text{respectively).} \end{aligned}$$

We also write $M, q \models \varphi$ as a shorthand for $M, \{q\} \models \varphi$. Additionally, we define: $\Diamond \varphi \equiv \top \mathcal{U} \varphi$, $\mathbb{K}_a \varphi \equiv \mathbb{C}_{\{a\}} \varphi$, and we use the usual definitions of Boolean connectives $\vee, \rightarrow, \leftrightarrow$. Standard knowledge can be defined as special case of constructive knowledge: $\mathcal{K}_A \varphi \equiv \hat{\mathcal{K}}_A \text{Now}(\varphi)$ where $\text{Now}(\varphi) \equiv \langle\langle \emptyset \rangle\rangle \varphi \mathcal{U} \varphi$ expresses that φ holds at the current state of the system (or, more, precisely, in every possible current state from Q). Finally, $\mathcal{K}_a \varphi \equiv C_{\{a\}} \varphi$.

The following proposition shows that CSL is more expressive than the most important previous proposals: F-ATEL [15], ATL_{ir} [23] and ATOL [14].

Proposition 1 ([13]) *There is a succinct translation from ATL_{ir} , ATOL and F-ATEL to CSL that preserves satisfaction and validity of formulae. Moreover, CSL is strictly more expressive than ATL_{ir} , ATOL and F-ATEL.*

The main advantage of CSL is that it allows the expression of the fact that an agent or coalition knows *how* to achieve its goal, as distinct from knowing that the goal can be *somehow* achieved. The former is captured by the formula saying that there is a strategy which is successful in all states the agent/coalition considers possible, while the latter is captured by the formula which states that in all possible states there is a successful strategy.

Proposition 2 ([13]) *CSL allows to capture abilities of types (1)-(3). More precisely:*

- $M, q \models \mathbb{K}_a \langle\langle a \rangle\rangle \varphi$ iff agent a has a strategy “de re” to enforce φ from q ;
- $M, q \models K_a \langle\langle a \rangle\rangle \varphi$ iff a has a strategy “de dicto” to enforce φ from q ;
- $M, q \models \langle\langle a \rangle\rangle \varphi$ iff a has a strategy to enforce φ from q .

Capturing different ability levels of coalitions is analogous, with various “epistemic modes” of collective recognizing the right strategy.

Thus, CSL provides a natural way of distinguishing between knowing how to play (ability “de re”) and weaker forms of strategic abilities. Note also that constructive knowledge operators capture the notion of ability “de re”, while standard epistemic operators refer to having a strategy “de dicto”.

Consider again the system from Example 1. We have that $K_i \langle\langle i \rangle\rangle \text{safe} \mathcal{U} (\text{safe} \wedge \text{bulbOK})$, but $\neg \mathbb{K}_i \langle\langle i \rangle\rangle \text{safe} \mathcal{U} (\text{safe} \wedge \text{bulbOK})$ – the latter because no single strategy is successful in *both* states (electricity on/off) the agent considers possible. Thus, CSL allows to express that the agent knows that there is a way of replacing the bulb, but he does not effectively know *how to do it*. Some subtler strategic properties can be studied on the variant of the Coordinated Attack problem, presented in the next section.

3 Example: Onion Soup Robbery

A virtual safe contains the recipe for the best onion soup in the world. The safe can only be opened by a k -digit binary code, where each digit c_i is sent from a prescribed location i ($1 \leq i \leq k$). To open the safe and download the recipe it is enough that at least $n \leq k$ correct digits are sent at the same moment. However, if a *wrong* value is sent from one of the locations, or if an insufficient number (i.e., between 1 and $n-1$) of digits is submitted, then the safe locks up and activates an alarm.

k agents are connected at the right locations; each of them can send 0, send 1, or do nothing (*nop*). Moreover, individual agents have only partial information about the code. To make the example more concrete, we assume that agent i (connected to location i) knows the values of $c_{i-1} \text{ XOR } c_i$ and $c_i \text{ XOR } c_{i+1}$ (we take $c_0 = c_{k+1} = 0$). This implies that only agents 1 and k know the values of “their” digits. Still, every agent knows whether his neighbors’ digits are the same as his.

Definition 1 ($Attack_k^n$)

- $\mathbb{A}gt = \{1, \dots, k\}$;
- $St = Q \cup S$, where states in $Q = \{q = \langle c_1, \dots, c_k \rangle \mid q \in \{0, 1\}^k\}$ identify possible codes for the (closed) safe, and states in $S = \{open, alarm\}$ represent the situations when the safe has been opened, or when the alarm has been activated;
- $\Pi = \{open\}$; $\pi(open) = \{open\}$; $Act = \{0, 1, nop\}$;
- $d(i, q) = \{0, 1, nop\}$ for $q \in Q$, and $d(i, s) = \{nop\}$ for $s \in S$;
- For all $x \in St$: $o(x, nop, \dots, nop) = x$. For $q \in Q$, and at least one $\alpha_i \neq nop$: $o(q, \alpha_1, \dots, \alpha_k) = open$ if $\alpha_j = c_j$ for at least n agents j and $\alpha_i \notin \{c_i, nop\}$ for no i ; else, $o(q, \alpha_1, \dots, \alpha_k) = alarm$.
- $q \sim_i q'$ iff $q[i-1] \text{ XOR } q[i] = q'[i-1] \text{ XOR } q'[i]$ and $q[i] \text{ XOR } q[i+1] = q'[i] \text{ XOR } q'[i+1]$.

For $Attack_k^n$, $k \geq 3$, the following CSL properties hold in every state $q \in Q$:

- $\langle\langle \mathbb{A}gt \rangle\rangle \Diamond open \wedge \neg \mathbb{E}_{\mathbb{A}gt} \langle\langle \mathbb{A}gt \rangle\rangle \Diamond open$: there is an executable strategy for the agents, which guarantees a win, but not all of them can identify it (in fact, none of them can in this case);
- $\mathbb{D}_{\mathbb{A}gt} \langle\langle \mathbb{A}gt \rangle\rangle \Diamond open$: if the agents share information they can recognize who should send what;
- $\mathbb{D}_{\{1, \dots, n-1\}} \langle\langle \mathbb{A}gt \rangle\rangle \Diamond open$: it is enough that the first $n-1$ agents devise the strategy. Note that the same holds for the last $n-1$ agents, i.e., the subteam $\{k-n+2, \dots, k\}$;
- Still, $\neg \mathbb{D}_{\{1, \dots, n-1\}} \langle\langle 1, \dots, n-1 \rangle\rangle \Diamond open$: all agents are necessary to execute the strategy.

We observe that constructive knowledge operators allow to approximate the amount of communication that is needed to establish a winning strategy in scenarios where explicit modeling of communication is impossible or too expensive. For instance, formula $\mathbb{D}_{\mathbb{A}gt} \langle\langle \mathbb{A}gt \rangle\rangle \Diamond open$ says that if the agents in $\mathbb{A}gt$ share their information they will be able to determine a strategy that opens the safe. Of course, the model does not include a possibility of such “sharing”, at least not explicitly. That is, there is no transition that leads to a state in which the epistemic relations of agents have been combined via intersection. Still, $D_A \varphi$ indicates that there is *epistemic potential* for agents in A to realize/infer φ ; what might be missing is means of exploiting the potential (e.g., communication). In the same way, $\mathbb{D}_A \langle\langle A \rangle\rangle \bigcirc \varphi$ says that the epistemic potential for A to determine the right strategy for $\bigcirc \varphi$ is there, too. So, it might be profitable to design efficient communication mechanisms to make the most of it.

4 Normal Form CSL

In this section, we first recall (and slightly refine) the concept of *constructive normal form* (CSNF) introduced in [13]. Then, we define formally the syntax and semantics of “normal form CSL”. The semantics is state-based in the sense that a single state appears on the left hand side of the satisfaction relation \models .

4.1 Constructive Normal Form

Definition 2 A CSL formula is in constructive normal form (CSNF) if every subformula starting with a \hat{K}_A operator is of the form $\hat{K}_{A_1} \dots \hat{K}_{A_n} \psi$ where ψ starts with a cooperation modality.

The following proposition is a straightforward corollary of [13, Theorem 60].

Proposition 3 Every CSL formula is strongly equivalent to a formula in constructive normal form. That is, for every formula φ of CSL, there is a normal form formula φ' such that $M, Q \models \varphi$ iff $M, Q \models \varphi'$. The transformation to normal form can be done in linear time and yields a formula which can be only linearly longer than φ .

We use CSL^{nf} to denote the sublanguage of CSL consisting only of formulae in CSNF. It turns out that the importance of the normal form goes beyond technicalities. In Section 4.3, we show that formulae of CSL^{nf} can be given standard state-based semantics, similar to those of ATL and CTL.

4.2 Rephrasing the Syntax

We recall from Section 4.1 that, by using only formulae in constructive normal form, we do not lose any expressivity. However, CSL^{nf} imposes some constraints on the usage of constructive knowledge operators, so that not every property can be written verbatim in CSL^{nf} —although for every property, an equivalent CSL^{nf} formula can be found (cf. Example 2 below).

Before defining the new semantics, we rephrase the definition of CSL^{nf} to a more convenient form:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varepsilon\langle A \rangle \bigcirc \varphi \mid \varepsilon\langle A \rangle \Box \varphi \mid \varepsilon\langle A \rangle \varphi \mathcal{U} \varphi,$$

where $p \in \Pi$ is an atomic proposition, $A \subseteq \text{Agt}$ a group of agents, and $\varepsilon = \hat{K}_{A_1}^1 \dots \hat{K}_{A_n}^n$ stands for a (possibly empty) sequence of constructive knowledge operators $\hat{K} = \mathbb{C}, \mathbb{E}, \mathbb{D}$, indexed by (possibly different) coalitions of agents. Additional operators are defined as in Section 2.3.

Example 2 Consider the CSL formula $\mathbb{C}_A(\text{safe} \rightarrow \langle\langle A \rangle\rangle \Box \text{safe})$. Clearly, it is not a formula of CSL^{nf} . Still, it has an equivalent normal form formula $\neg \mathbb{C}_A \text{Now}(\text{safe}) \vee \mathbb{C}_A \langle\langle A \rangle\rangle \Box \text{safe}$ (that is, these two formulae have the same truth value for every model and set of states in the model).

Note also that all the formulae mentioned in Section 3 belong to “normal form CSL”. In fact, for most properties that can be written in CSL, they are most naturally expressed in their strategic normal form.

4.3 Semantics

In Definition 5, we present the semantic clauses for CSL^{nf} . The first three clauses are completely standard for modal logic. The other three clauses define the semantics of cooperation modalities coupled with constructive knowledge operators (in pretty much the same way as temporal operators are coupled with path quantifiers in CTL). A sequence of operators $\hat{K} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ yields a composition of the corresponding epistemic relations, which is used to determine the set of states that seem subjectively possible in state q . First, we define the correspondence formally, and then we present the clauses.

Definition 3 Let $\mathcal{R}_1, \mathcal{R}_2 \subseteq St \times St$ be binary relations on states. The composition of \mathcal{R}_1 and \mathcal{R}_2 is defined as

$$\mathcal{R}_1 \circ \mathcal{R}_2 = \{\langle q, q'' \rangle \mid \text{there is } q' \text{ such that } q\mathcal{R}_1 q' \text{ and } q'\mathcal{R}_2 q''\}.$$

As the composition is additive, its definition naturally extends to an arbitrary number of relations ($\mathcal{R}_1 \circ \dots \circ \mathcal{R}_n$). We assume that a composition of an empty sequence of relations is just the identity relation.

Definition 4 Let $\varepsilon = \hat{K}_{A_1}^1 \dots \hat{K}_{A_n}^n$ be a sequence of constructive knowledge operators. The relation corresponding to ε is defined as $\text{rel}(\varepsilon) = \sim_{A_1}^{\mathcal{K}^1} \circ \dots \circ \sim_{A_n}^{\mathcal{K}^n}$.

In accordance with definition 3, we define the relation corresponding to the empty string as the identity relation on states: $\text{rel}(\emptyset) = id_{St}$.

Definition 5 The semantics of CSL^{nf} is defined through the clauses below.

- $M, q \models p$ iff $p \in \pi(q)$ (for $p \in \Pi$);
- $M, q \models \neg \varphi$ iff $M, q \not\models \varphi$;
- $M, q \models \varphi \wedge \psi$ iff $M, q \models \varphi$ and $M, q \models \psi$;
- $M, q \models \varepsilon \langle\langle A \rangle\rangle \bigcirc \varphi$ iff there is a memoryless uniform strategy S_A such that, for every $\Lambda \in \text{out}(\text{img}(q, \text{rel}(\varepsilon)), S_A)$, we have $M, \Lambda[1] \models \varphi$;
- $M, q \models \varepsilon \langle\langle A \rangle\rangle \Box \varphi$ iff there is S_A such that, for every $\Lambda \in \text{out}(\text{img}(q, \text{rel}(\varepsilon)), S_A)$, we have $M, \Lambda[i] \models \varphi$ for every $i \geq 0$;

$M, q \models \varepsilon \langle A \rangle \varphi \mathcal{U} \psi$ iff there is S_A such that for every $\Lambda \in \text{out}(\text{img}(q, \text{rel}(\varepsilon)), S_A)$ there is an $i \geq 0$, for which $M, \Lambda[i] \models \psi$, and $M, \Lambda[j] \models \varphi$ for every $0 \leq j < i$.

The above semantics is equivalent to the semantics of CSL presented in Section 2.3.

Proposition 4 Let φ be a formula of CSL^{uf} . Then, $M, q \models_{\text{CSL}^{\text{uf}}} \varphi$ iff $M, q \models_{\text{CSL}} \varphi$.

Proof. (Induction on the structure of φ)

- Case $\varphi \equiv p$: $M, q \models_{\text{CSL}^{\text{uf}}} p$ iff $p \in \pi(q)$ iff $M, q \models_{\text{CSL}} p$.
- Case $\varphi \equiv \neg\psi$: $M, q \models_{\text{CSL}^{\text{uf}}} \neg\psi$ iff $M, q \not\models_{\text{CSL}^{\text{uf}}} \psi$ iff (by induction) $M, q \not\models_{\text{CSL}} \psi$ iff $M, q \models_{\text{CSL}} \neg\psi$.
- Cases $\varphi \equiv \psi_1 \wedge \psi_2, \mathcal{K}_A \varphi$: analogous.
- Case $\varphi \equiv \hat{\mathcal{K}}_{A_1}^1 \dots \hat{\mathcal{K}}_{A_n}^n \langle A \rangle \bigcirc \psi$:
 $M, q \models_{\text{CSL}^{\text{uf}}} \hat{\mathcal{K}}_{A_1}^1 \dots \hat{\mathcal{K}}_{A_n}^n \langle A \rangle \bigcirc \psi$ iff there is S_A such that for every $\Lambda \in \text{out}(\text{img}(q, \sim_{A_1}^{\mathcal{K}_1} \circ \dots \circ \sim_{A_n}^{\mathcal{K}_n}), S_A)$ we have $M, \Lambda[1] \models_{\text{CSL}^{\text{uf}}} \psi$ iff (by induction) there is S_A such that for every $\Lambda \in \text{out}(\text{img}(q, \sim_{A_1}^{\mathcal{K}_1} \circ \dots \circ \sim_{A_n}^{\mathcal{K}_n}), S_A)$ we have $M, \Lambda[1] \models_{\text{CSL}} \psi$ iff (by the semantics of CSL) $M, \text{img}(q, \sim_{A_1}^{\mathcal{K}_1} \circ \dots \circ \sim_{A_n}^{\mathcal{K}_n}) \models_{\text{CSL}} \langle A \rangle \bigcirc \psi$ iff $M, q \models_{\text{CSL}} \hat{\mathcal{K}}_{A_1}^1 \dots \hat{\mathcal{K}}_{A_n}^n \langle A \rangle \bigcirc \psi$.
- Cases $\varphi \equiv \hat{\mathcal{K}}_{A_1}^1 \dots \hat{\mathcal{K}}_{A_n}^n \langle A \rangle \square \psi, \hat{\mathcal{K}}_{A_1}^1 \dots \hat{\mathcal{K}}_{A_n}^n \langle A \rangle \psi_1 \mathcal{U} \psi_2$: analogous.

■

5 Flexible Aggregation of Uncertainty Sets

Constructive knowledge operators are used in CSL (and CSL^{uf}) to “aggregate” a set of states for which the coalition should find a successful strategy. In the relational view of the previous section, such an operator corresponds to a relation on states, and the set is taken as the image of the current state with respect to the relation. An operator can appear alone or in a sequence; in the latter case, the corresponding relation is the composition of particular relations. The only way to combine individual epistemic relations in a different way (i.e., not through relational composition) is to employ the predefined operators of collective knowledge $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$.

In this section, we propose to use more basic operators for combining individual views of agents. Besides composition, explicit operators that yield union, intersection, complement and transitive closure are added. We show that the resulting language of epistemic expressions allows to express strictly

more epistemic properties than before. Moreover, the properties can be expressed by combining only *individual* knowledge operators.

The idea is very similar to relational operations on programs (or actions) that can be found in dynamic logic [21, 9] (cf. also Segerberg's work on logic of iteration [24]). There, the nondeterministic choice operator (\sqcup) is interpreted as the union of underlying relations, and iteration ($*$) facilitates the reflexive and transitive closure.⁴ Analogous operators (for complement, union, and intersection of relations) are used as *role constructors* in description logics [3].

5.1 Epistemic Expressions and Their Interpretation

So far, the only epistemic expressions that we used were sequences of knowledge operators (individual and/or collective). Now we allow the expressions to have more sophisticated structure. It turns out that, with the enhanced epistemic expressions, we do not need collective knowledge operators any more.

Formally, the language of CSL^{ef} is now redefined as follows. The production rule for formulae φ is the same as before; the only thing that changes is the set of epistemic expressions ε . Moreover, we only need to define an interpretation of the new knowledge terms in order to extend the semantics. The semantic clauses for formulae φ stay the same as in section 4.3.

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varepsilon\langle\langle A \rangle\rangle \bigcirc \varphi \mid \varepsilon\langle\langle A \rangle\rangle \Box \varphi \mid \varepsilon\langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi, \\ \varepsilon &::= \emptyset \mid \mathbb{K}_a \mid \bar{\varepsilon} \mid \varepsilon \sqcap \varepsilon \mid \varepsilon \varepsilon \mid \varepsilon^*.\end{aligned}$$

Additionally, we define $\varepsilon_1 \sqcup \varepsilon_2 = \overline{\bar{\varepsilon}_1 \sqcap \bar{\varepsilon}_2}$. The relation *rel* behind epistemic expressions is now extended as follows:

- $\text{rel}(\emptyset) = \text{id}_{S_t}$,
- $\text{rel}(\mathbb{K}_a) = \sim_a$,
- $\text{rel}(\bar{\varepsilon}) = \overline{\text{rel}(\varepsilon)}$,
- $\text{rel}(\varepsilon_1 \sqcap \varepsilon_2) = \text{rel}(\varepsilon_1) \cap \text{rel}(\varepsilon_2)$,
- $\text{rel}(\varepsilon_1 \varepsilon_2) = \text{rel}(\varepsilon_1) \circ \text{rel}(\varepsilon_2)$,
- $\text{rel}(\varepsilon^*)$ is the reflexive and transitive closure of $\text{rel}(\varepsilon)$.

It is easy to see that $\text{rel}(\varepsilon_1 \sqcup \varepsilon_2) = \text{rel}(\varepsilon_1) \cup \text{rel}(\varepsilon_2)$. Note that $\varepsilon_1 \sqcap \varepsilon_2$ refers to the knowledge obtained by *joining* knowledge behind ε_1 and ε_2 . Thus, on the semantic side, it is modeled by reducing the uncertainty (or *ignorance*) represented by $\text{rel}(\varepsilon_1), \text{rel}(\varepsilon_2)$ to their intersection. Analogously, $\varepsilon_1 \sqcup \varepsilon_2$ refers

⁴ There is also the sequential composition of programs ; that corresponds to ordinary relational composition.

to a situation of *reduced knowledge*, and thus extended ignorance ($rel(\varepsilon_1) \cup rel(\varepsilon_2)$).

Proposition 5 states that collective knowledge operators are not necessary any more.

Proposition 5 *Let tr be a translation of CSL^{ef} formulae that replaces:*

- every occurrence of $\mathbb{D}_{\{a_1, \dots, a_n\}}$ with $\mathbb{K}_{a_1} \sqcap \dots \sqcap \mathbb{K}_{a_n}$,
- every occurrence of $\mathbb{E}_{\{a_1, \dots, a_n\}}$ with $\mathbb{K}_{a_1} \sqcup \dots \sqcup \mathbb{K}_{a_n}$, and
- every occurrence of $\mathbb{C}_{\{a_1, \dots, a_n\}}$ with $(\mathbb{K}_{a_1} \sqcup \dots \sqcup \mathbb{K}_{a_n})^*$.

Then, $M, q \models \varphi$ iff $M, q \models tr(\varphi)$.

Proof. Straightforward. ■

Example 3 Consider $Attack_k^n$ again. We can rewrite the properties from Section 3 so that only individual knowledge operators are included: $\neg(\mathbb{K}_1 \sqcup \dots \sqcup \mathbb{K}_k) \langle \langle \mathbb{A}gt \rangle \rangle \Diamond open$, $(\mathbb{K}_1 \sqcap \dots \sqcap \mathbb{K}_k) \langle \langle \mathbb{A}gt \rangle \rangle \Diamond open$ etc. Naturally, all these properties hold in every state $q \in Q$.

Another property that can be expressed with extended epistemic expressions is: $\neg(\mathbb{D}_{\{1, \dots, n-1\}} \sqcup \mathbb{D}_{\{k-n+2, \dots, k\}}) \langle \langle \mathbb{A}gt \rangle \rangle \Diamond open \equiv \neg((\mathbb{K}_1 \sqcap \dots \sqcap \mathbb{K}_{n-1}) \sqcup (\mathbb{K}_{k-n+2} \sqcap \dots \sqcap \mathbb{K}_k)) \langle \langle \mathbb{A}gt \rangle \rangle \Diamond open$, which says that the subteams $\{1, \dots, n-1\}$, $\{k-n+2, \dots, k\}$ do not have the ability to independently share knowledge and identify the same winning strategy (despite the fact that each subteam can identify a successful strategy on its own, cf. Section 3). However, if the subteams overlap in at least $n-2$ agents then they are able to do so; the common strategy prescribes sending the right bits by the overlapping agents and their immediate neighbors, and refraining from action by the rest. Thus, for $x - y \geq n-3$, we have that $(\mathbb{D}_{\{1, \dots, x\}} \sqcup \mathbb{D}_{\{y, \dots, k\}}) \langle \langle \mathbb{A}gt \rangle \rangle \Diamond open$.

5.2 Epistemic Expressions for Standard Knowledge

As standard knowledge can be seen as a special case of constructive knowledge, one can think of combining standard knowledge in a similar way:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varepsilon \langle \langle A \rangle \rangle \bigcirc \varphi \mid \varepsilon \langle \langle A \rangle \rangle \Box \varphi \mid \varepsilon \langle \langle A \rangle \rangle \varphi \mathcal{U} \varphi \mid \varepsilon' \varphi, \\ \varepsilon' &::= \emptyset \mid K_a \mid \overline{\varepsilon'} \mid \varepsilon' \sqcap \varepsilon' \mid \varepsilon' \varepsilon' \mid (\varepsilon')^*. \end{aligned}$$

Definition 6 Let $constr(\varepsilon')$ be the “constructive” epistemic expression ε obtained by replacing every K_i in ε' by \mathbb{K}_i . The semantics of $\varepsilon' \varphi$ can be given as follows:

$$M, q \models \varepsilon' \varphi \quad \text{iff} \quad M, q \models \varphi \text{ for every } q' \in \text{img}(q, rel(constr(\varepsilon'))).$$

Alternatively, we can define a separate denotation rel' of expressions ε' to the same effect.

Again, it turns out that standard knowledge is a special case of constructive knowledge:

Proposition 6 $M, q \models \varepsilon' \varphi$ iff $M, q \models \text{constr}(\varepsilon') \text{Now}(\varphi)$.

An example non-trivial epistemic property that can be expressed in such language is “only knowing”/“all I know” from [16]. To recall, agent i “only knows” φ in M, q iff $q \sim_i q'$ for all states q' satisfying φ .

Proposition 7 Agent i “only knows” φ in M, q iff $M, q \models \bar{K}_i \neg \varphi$.

By this, and the fact that every formula of CSL can be equivalently rewritten to CSL with extended epistemic expressions, we get the following.

Corollary 8 CSL with extended epistemic terms has strictly more expressive power than “pure” CSL.

5.3 Model Checking

The *model checking* problem asks whether a given formula φ holds in a given model M and state q . Below, we present a straightforward adaptation of the ATL_{ir} model checking from [23] that allows to model-check formulae of CSL^{ef} with extended epistemic expressions. The algorithm invokes a general CTL model checker $\text{mctl}(\varphi, M)$ that returns the set of states in M which satisfy φ .

Function $\text{mcheck}(\varphi, M, q)$.

Case $\varphi \equiv p$: return(*true*) if $p \in \pi(q)$, else return(*false*);

Case $\varphi \equiv \neg \psi$: return(*true*) if $\text{mcheck}(\psi, M, q) = \text{false}$, else return(*false*);

Case $\varphi \equiv \psi_1 \wedge \psi_2$: return(*true*) if $\text{mcheck}(\psi_1, M, q) = \text{true}$ and $\text{mcheck}(\psi_2, M, q) = \text{true}$, else return(*false*);

Case $\varphi \equiv \varepsilon \langle\langle A \rangle\rangle \bigcirc \psi$: Run $\text{mcheck}(\psi, M, q)$ for every $q \in St$, and label the states in which the answer was *true* with an additional proposition *yes* (not used elsewhere). Then, guess the strategy of A , and “trim” model M by removing all the transitions inconsistent with the strategy (yielding a sparser model M'). Finally, return(*true*) if $\text{rel}(\varepsilon) \subseteq \text{mctl}(A \bigcirc \text{yes}, M')$, else return(*false*).

Note: subformula ψ is checked in the original model M , and not in M' !

Cases $\varphi \equiv \langle\langle A \rangle\rangle \Box \psi$ and $\varphi \equiv \langle\langle A \rangle\rangle \psi_1 \mathcal{U} \psi_2$: analogous.

We note that: (1) general model checking of CTL is linear with respect to the number of transitions in the model and the length of the formula; (2) complement, union, composition, and transitive closure of binary relations can be computed in time linear with respect to the number of the relations and the maximal number of tuples per relation. Thus, the above algorithm runs in deterministic polynomial time, with calls to an oracle of an NP problem.

Proposition 9 *The model checking problem for CSL^{nf} (with extended epistemic expressions) is in Δ_2^P , i.e., in the class of problems solvable in deterministic polynomial time with calls to an NP oracle.*

For the lower bound, we note that CSL^{nf} subsumes ATL_{ir} , since ATL_{ir} 's central operator $\langle\langle a_1, \dots, a_n \rangle\rangle_{ir}$ can be translated into CSL^{nf} as $(\mathbb{K}_{a_1} \sqcup \dots \sqcup \mathbb{K}_{a_n}) \langle\langle a_1, \dots, a_n \rangle\rangle$. Moreover, model checking ATL_{ir} is Δ_2^P -complete [23, 12]. Thus, we get the following.

Proposition 10 *The model checking problem for CSL^{nf} (with extended epistemic expressions) is Δ_2^P -complete in the number of transitions and epistemic links in the model, and the length of the formula.*

6 Final Remarks

We have shown how the syntax and the semantics of Constructive Strategic Logic can be redefined so that we obtain a more standard modal logic without compromising expressivity or computational complexity. It does not mean, however, that the original CSL is now obsolete. On the contrary, we believe that only the richer semantics of full CSL provides the appropriate conceptual framework for studying knowledge and ability under imperfect information. “Normal form CSL” may be technically simpler, and easier to use in theoretical analysis, but only by showing its relationship to full CSL do we validate its conceptual merits.

References

- [1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109. IEEE Computer Society Press, 1997.
- [2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

References

- [3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [4] E. Clarke and E. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proceedings of Logics of Programs Workshop*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71, 1981.
- [5] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.
- [6] V. Goranko. Coalition games and alternating temporal logics. In J. van Benthem, editor, *Proceedings of TARK VIII*, pages 259–272. Morgan Kaufmann, 2001.
- [7] V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280, 2004.
- [8] T. Ågotnes. A note on syntactic characterization of incomplete information in atel. In *Proceedings of the Workshop on Knowledge and Games, Liverpool*, pages 34–42, 2004.
- [9] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
- [10] A. Herzig and N. Troquard. Knowing how to play: Uniform choices in logics of agency. In *Proceedings of AAMAS’06*, pages 209–216, 2006.
- [11] W. Jamroga. Some remarks on alternating temporal epistemic logic. In B. Dunin-Keplicz and R. Verbrugge, editors, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 133–140, 2003.
- [12] W. Jamroga and J. Dix. Model checking ATL_{ir} is indeed Δ_2^P -complete. In *Proceedings of EUMAS’06*, 2006.
- [13] 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.
- [14] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2–3):185–219, 2004.
- [15] G. Jonker. Feasible strategies in Alternating-time Temporal Epistemic Logic. Master thesis, University of Utrecht, 2003.
- [16] H. Levesque. All I know: a study in auto-epistemic logic. *Artificial Intelligence*, 42(3):263–309, 1990.

- [17] R. Moore. A formal theory of knowledge and action. In J. Hobbs and R. Moore, editors, *Formal Theories of the Commonsense World*. Ablex Publishing Corp., 1985.
- [18] L. Morgenstern. Knowledge and the frame problem. *International Journal of Expert Systems*, 3(4), 1991.
- [19] M. Pauly. Game logic for game theorists. Technical Report INS-R0017, CWI, 2000.
- [20] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.
- [21] V. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings on the 17th IEEE Symposium on Foundations of Computer Science*, pages 109–121, 1976.
- [22] W. Quine. Quantifiers and propositional attitudes. *Journal of Philosophy*, 53:177–187, 1956.
- [23] P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2), 2004.
- [24] K. Segerberg. A completeness theorem in the modal logic of programs. *Notices of the American Mathematical Society*, 24, 1977.
- [25] A. S. Troelstra. History of constructivism in the twentieth century. Technical Report ML-91-05, University of Amsterdam, 1991.
- [26] W. van der Hoek and M. Wooldridge. Cooperation, knowledge and time: Alternating-time Temporal Epistemic Logic and its applications. *Studia Logica*, 75(1):125–157, 2003.
- [27] S. van Otterloo and G. Jonker. On Epistemic Temporal Strategic Logic. *Electronic Notes in Theoretical Computer Science*, XX:35–45, 2004. Proceedings of LCMAS’04.
- [28] M. Wooldridge. *Reasoning about Rational Agents*. MIT Press : Cambridge, Mass, 2000.