This is the published version of a chapter published in *Handbook of Epistemic Logic*.

Citation for the original published chapter:


N.B. When citing this work, cite the original published chapter.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:diva-123448
Chapter 11

Knowledge and Ability

Thomas Ågotnes, Valentin Goranko, Wojciech Jamroga and Michael Wooldridge

Contents

11.1 Philosophical Treatments ......................... 544
11.2 Ability in Artificial Intelligence ................... 548
11.3 Logics for Abilities of Agents and Coalitions ..... 552
11.4 Abilities under Incomplete Information .......... 563
11.5 Reasoning about Strategies and Knowledge ..... 573
11.6 Concluding remarks ............................. 583
11.7 Notes .................................. 583
References ................................ 586

Abstract In this chapter we relate epistemic logics with logics for strategic ability developed and studied in computer science, artificial intelligence and multi-agent systems. We discuss approaches from philosophy and artificial intelligence to modelling the interaction of agents’ knowledge and abilities and then focus on concurrent game models and the alternating-time temporal logic ATL. We show how ATL enables reasoning about agents’ coalitional abilities to achieve qualitative objectives in concurrent game models, first assuming complete information and then under incomplete information and uncertainty about the structure of the game model. We then discuss epistemic extensions of ATL enabling explicit reasoning about the interaction of knowledge and strategic abilities on different epistemic levels, leading inter alia to the notion of constructive knowledge.
Our aim in this chapter is to survey logics that attempt to capture the interplay between knowledge and ability. The term “ability”, in the sense we mean it in this chapter, corresponds fairly closely to its everyday usage. That is, ability means the capability to do things, and to bring about states of affairs. There are several reasons why ability is worth studying in a formal setting:

- First, the concept of ability is surely worth studying from the perspective of philosophy, and in particular the philosophy of language. In this context, the point is to try to gain an understanding of what people mean when they make statements like “I can X”, and in particular to understand what such a claim implies about the mental state of the speaker.

- Second, and more recently, researchers in computer science and artificial intelligence are interested in the notion of what machines can achieve. For example, imagine an artificial intelligence to whom we can issue instructions in natural English. Then, if we give the machine an instruction “X” (where X might be “make me a cup of tea” or, famously, “bring me a beer”), then the question of what this instruction means in relation to the abilities of the machine becomes important. An artificial intelligence that is presented with an instruction to X surely needs to understand whether or not it actually can in fact X before it proceeds; this implies some model or theory of ability, and the ability of the artificial intelligence to reason with this model or theory.

The remainder of this chapter is structured as follows. We begin, in the following section, with a discussion on the way that philosophers have considered knowledge and ability. In section 11.2, we move on to discuss how the concept of ability has been considered and formalised in artificial intelligence; we focus in particular on the seminal and enormously influential work of Robert Moore on the relationship between knowledge and ability. We then go on to review more recent contributions to the logic of ability, arising from the computer aided verification community: specifically, Alternating-time Temporal Logic (ATL) and the issues that arise when attempting to integrate ATL with a theory of knowledge.

11.1 Philosophical Treatments of Knowledge and Ability

We will begin by reviewing the way that the concept of ability has been considered in philosophy. We will start with the work of the philosopher
Gilbert Ryle (1900–1976). Ryle was greatly interested in the concept of mind, and one of the questions to which he addressed himself was the distinction between knowing how and knowing that. Crudely, “knowing that” is the concept of knowledge with which this handbook is largely concerned: we think of “knowing that” as a kind of relation between agents and true propositions, and in this book we write $K_i \varphi$ to mean that agent $i$ knows that $\varphi$, where $\varphi$ is some proposition. The concept of “knowing how” seems related, but is clearly different: it is concerned with the knowledge of how to achieve things.

One question in particular that Ryle considered was whether know-how could be reduced to know-that: that is, whether know-how was in fact just a type of know-that. Ryle argued that such a reduction was not possible. His argument against such a reduction has since become quite celebrated (it is commonly known as “Ryle’s regress”, and has been applied in various other settings). Broadly, the argument goes as follows:

If know-how were a species of know-that, then, to engage in any action, one would have to contemplate a proposition. But, the contemplation of a proposition is itself an action, which presumably would itself have to be accompanied by a distinct contemplation of a proposition. If the thesis that knowledge-how is a species of knowledge-that required each manifestation of knowledge-how to be accompanied by a distinct action of contemplating a proposition, which was itself a manifestation of knowledge-how, then no knowledge-how could ever be manifested.

In other words, know-how does not reduce to know-that because if we had to know-that a proposition every time we exercised know-how, we would get an infinite regress of know-thats.

There are other arguments against a reduction of know-how to know-that. For example, consider the following sentence:

Michael knows how to ski. (*)

If know-how reduces to know-that, then there is some proposition, call it $\varphi^*$, such that knowing how to ski is equivalent to knowing $\varphi^*$. Presumably, telling somebody the proposition $\varphi^*$ would then be enough to convey the ability to ski, assuming of course the hearer of the message was able to process and understand it. We are trivialising the arguments at stake, of course, but nevertheless, anybody who has learned to ski will recognise that such a simplistic reduction of know-how to know-that is implausible.
The exact relationship of know-how to know-that remains the subject of some debate, with arguments on either side; see, e.g., Stanley and Williamson for a contemporary discussion.

Despite the philosophical objections of Ryle and others, researchers in artificial intelligence have largely adopted the view that know-how can be reduced to know-that. In the section that follows, we will see how researchers in artificial intelligence have attempted to formalise such reductions. Interestingly, with a sufficiently rich logical formalism, the two notions turn out distinct again, as we will argue in Section 11.5.3.

There have been several other early philosophical approaches to developing logics of agency and ability, including works of Georg Henrik von Wright (1916–2003), Stig Kanger (1924–88), Brian Chellas, Mark Brown and Nuel Belnap and Michael Perloff. In particular, Brown proposes a modal logic with non-normal modalities formalising the idea that the modality for ability has a more complex, existential-universal meaning (the agent has some action or choice, such that every outcome from executing that action (making that choice) achieves the aim), underlying all further approaches to formalizing agents’ ability that will be presented here. At about the same time Belnap and Perloff developed the basics of their theory of seeing to it that, usually abbreviated to “STIT”. The STIT family of logics were formulated in an attempt to be able to give a formal semantics to the concept of agency, where an agent is loosely understood to be an entity that makes deliberate purposeful actions. Although accounts differ on details, the general aim is to be able to make sense of statements such as “the agent i brings it about that φ”. We will denote that STIT operator by $\nabla_i \varphi$. The semantics of Belnap and Perloff’s operator (technically, known as the achievement stit operator), which makes the idea behind Brown’s ability operators explicit, was as follows. Formulae of the form $\nabla_i \varphi$ are interpreted in a branching-time model containing histories of the world, each of which contains various “moments”. Branching in the model occurs as a result of choices made by agents. Intuitively, in making a particular choice, an agent constrains the possible future histories of the world. Roughly speaking, we then say that $\nabla_i \varphi$ is true in a world/moment pair if there was some earlier moment in the history at which point the agent i made a choice such that:

- $\varphi$ is true in all histories consistent with that choice;
- at the point at which the choice was made, the status of $\varphi$ was not settled, i.e., there were histories arising from the choice moment in which $\varphi$ was false.

Expressed differently, $\nabla_i \varphi$ means that the agent i made a choice such that $\varphi$ was a necessary consequence of this choice, while $\varphi$ would not necessarily
have been true had the agent not made this choice.

Although we have not presented the full formal semantics for Belnap and Perloff’s operator, it should be obvious that the operator is semantically rather complex, at least compared to conventional tense and dynamic operators (the semantics constitute a “formidable definition” in the words of Belnap, Perloff, and Xu). The main source of difficulty is that an expression $\nabla_i \varphi$ seems to refer to both the past (the moment at which the relevant choice was made) and the present/future (the moments at which $\varphi$ is true). Several subsequent proposals have been made to refine and simplify the logic of STIT, with the aim of dealing with some of their counterintuitive properties, and contemporary formulations are much simpler and more intuitive. Here is a list of simple candidate axioms and deduction rules for a logic of agency, due to Troquard:

$$(M) \quad \nabla_i (\varphi \land \psi) \rightarrow (\nabla_i \varphi \land \nabla_i \psi)$$

$$(C) \quad (\nabla_i \varphi \land \nabla_i \psi) \rightarrow \nabla_i (\varphi \land \psi)$$

$$(N) \quad \nabla_i \top$$

$$(\text{No}) \quad \neg \nabla_i \top$$

$$(T) \quad \nabla_i \varphi \rightarrow \varphi$$

$$(\text{RE}) \quad \text{If } \varphi \leftrightarrow \psi \text{ then } \nabla_i \varphi \rightarrow \nabla_i \psi.$$
11.2 Ability in Artificial Intelligence

In this section, we survey attempts within the artificial intelligence (AI) community to develop logics of ability. Any survey of such work must surely begin with the 1969 paper *Some Philosophical Problems from the Standpoint of Artificial Intelligence* by John McCarthy and Pat Hayes. McCarthy and Hayes start from the following observation:

We want a computer program that decides what to do by inferring in a formal language [i.e., a logic] that a certain strategy will achieve a certain goal. This requires formalizing concepts of causality, ability, and knowledge.

Much of their article is concerned with speculating about the features that would be required in a logic to be used for decision-making of the type they describe. To illustrate their discussion, they introduced some ideas that have subsequently become more-or-less standard in the AI literature: in particular, the idea of what we now call the situation calculus (“sit calc”), a formalism for reasoning about actions in which world states (a.k.a. situations) are introduced as terms into the object language that the program uses to reason about the world. The situation calculus is one of the cornerstone formalisms of AI, and has been developed significantly since McCarthy and Hayes’ original work. Although they did not present a formalisation of ability, they did speculate on what such a formalism might look like. They suggested three possible interpretations of what it means for a computer program \( \pi \) to be able to achieve a state of affairs \( \varphi \):

1. There is a sub-program \( \sigma \) and room for it in memory which would achieve \( \varphi \) if it were in memory, and control were transferred to \( \pi \). No assertion is made that \( \pi \) knows \( \sigma \) or even knows that \( \sigma \) exists.
2. \( \sigma \) exists as above and that \( \sigma \) will achieve \( \varphi \) follows from information in memory according to a proof that \( \pi \) is capable of checking.
3. \( \pi \)’s standard problem-solving procedure will find \( \sigma \) if achieving \( \varphi \) is ever accepted as a subgoal.

The distinction between cases (1) and (2) is that, in the first, ability is seen from the standpoint of an omniscient external observer, who can see that there is some action or procedure such that, if \( \pi \) executed the action or followed the procedure, then the achievement of \( \varphi \) would result, while in the second, there is some potential awareness of this on the part of the program. The third conception implies *practical* ability: not only does the possibility
for the program to achieve $\varphi$ exist, but the program would actually be able
to compute and execute an appropriate strategy $\sigma$. Considerations such as
these are reflected in most subsequent logical treatments of ability.

After the seminal work of McCarthy and Hayes, probably the best-
known and most influential studies of ability in AI was Robert Moore’s
analysis of the relationship between knowledge and ability in a dynamic
variant of epistemic logic. Although the intuitions underpinning Moore’s
account of knowledge and action are intuitively appealing and deeply in-
sightful, the technical details of the actual formalism he used are rather
involved. Moore’s aim was to develop a formalism that could be used in
a classical first-order theorem proving system, of the type that was widely
studied in artificial intelligence research at the time. However, he realised
the value of using a modal language to express dynamic epistemic prop-
erties. His solution was to axiomatise a quantified modal logic using a
first-order logic meta-language, in effect, allowing statements in the quan-
tified modal language to be translated into the first-order language. The
technical details are along the lines of the “standard translation” of modal
logic into first-order logic. The resulting framework is rich and powerful;
but it is technically very involved, and not intuitive for people to read. For
these reasons, we here present a greatly simplified version of Moore’s for-
malism, which is not intended to be a fully-fledged logic, but a notation for
representing the key insights of the original work.

The basic components of the framework are a set of possible worlds
(essentially, system states), which we denote by $\mathbf{St}$, and set of actions, which
we denote by $\mathbf{Act}$. To keep things simple, we will assume there is just one
agent in the system. We write $q \models \varphi$ to mean that the proposition $\varphi$ is true
in state $q$ (we won’t worry about the full formal definition of the satisfaction
relation $\models$).

Epistemic properties of the system are captured in an epistemic accessibility
relation $\sim \subseteq \mathbf{St} \times \mathbf{St}$, with the usual meaning: $q \sim q’$ iff $q’$ and $q$ are
indistinguishable to the agent. A unary knowledge modality $K$ is defined
in the standard way:

$$ q \models K \varphi \text{ iff } \forall q’ : q \sim q’ \implies q’ \models \varphi. $$

As an aside, readers familiar with S5 treatments of knowledge might be
interested to hear that Moore only required epistemic accessibility relations
$\sim$ to be reflexive and transitive, giving an epistemic logic $\text{KT4} = \text{S4}$. In
this respect, his treatment of knowledge was slightly unorthodox, in that
he rejected the negative introspection axiom. However, the omission of
negative introspection plays no significant part in his formalisation of ability.

To capture the effects of actions, Moore used a ternary relation $R \subseteq
\mathbf{Act} \times \mathbf{St} \times \mathbf{St}$, where $R(a, q, q’)$ means that $q’$ results from performing $a$ in
when in state \( q \). Moore assumed actions were deterministic, so that only one state could result from the performance of an action in a given state.

Moore then introduced a modal operator \((\text{Res } a \varphi)\) to mean that after action \( a \) is performed, \( \varphi \) will be true – that is, \( \varphi \) is a “result” of performing action \( a \). The truth condition for this operator is as follows:

\[
q \models (\text{Res } a \varphi) \iff \exists q' \cdot R(a, q, q') \text{ and } q' \models \varphi
\]

Thus, Moore’s expression \((\text{Res } a \varphi)\) is similar to the dynamic logic expression \((a)\varphi\); and since Moore assumed actions are deterministic, it is in turn closely related to the dynamic logic expression \([a]\varphi\).

Quantification plays an important role in Moore’s logic of ability, and in particular, the theory relies on some important properties of quantifying in to modal knowledge contexts. To illustrate the issues, we assume that we have in our logic the technical machinery of first-order quantification. Now, consider the distinction between the following two quantified epistemic formulae, where \( \text{Murderer}(x) \) is intended to mean that the individual denoted by \( x \) is a murderer:

\[
K(\exists x : \text{Murderer}(x)) \tag{11.1}
\]

\[
\exists x : (K \text{Murderer}(x)) \tag{11.2}
\]

The first formula is said to be a \textit{de dicto} formula, while the second is a \textit{de re} formula. These two formulae do not express the same properties. The first formula asserts that, in every world consistent with the agent’s knowledge, the formula \( \exists x : \text{Murderer}(x) \) is true. Observe that, in one such world, the individual \( x \) could be Alice, while in another such world, the individual \( x \) could be Bob. In this case, the identity of the murderer is different in different epistemic alternatives for the agent. Thus, (11.1) asserts that the agent knows somebody is a Murderer, but does not imply that the agent knows the identity of the individual in question.

In contrast, (11.2) asserts something stronger. It says that there is some individual \( x \) such that in all epistemic alternatives for the agent, \( x \) is a Murderer. This time, the value of \( x \) is fixed across all epistemic alternatives for the agent, and so the agent knows the identity of the individual in question.

This distinction – between knowing that some \( x \) has a property, and knowing the identity of the \( x \) that has the property – is, as we will see, important in Moore’s theory of ability.

With these concepts in place, we can now turn to Moore’s formalisation of ability. He was concerned with developing a theory of ability that would capture the following two aspects of the interaction between knowledge and action:
1. As a result of performing an action, an agent can gain knowledge, and in particular, agents can perform “test” actions, in order to find things out.

2. In order to perform some actions, an agent needs knowledge: these are knowledge pre-conditions. For example, in order to open a safe, it is necessary to know the combination.

We will develop Moore’s model in two stages. The aim is ultimately to define a unary operator \((\text{Can } \varphi)\), which is intended to mean that the agent has the ability to achieve \(\varphi\). First, we will see a definition that captures some important intuitions, but which fails to capture some other of our intuitions about ability. Then, we will adapt our definition to rectify these problems. Here is our first attempt to define \(\text{Can}\):

\[
(\text{Can } \varphi) \leftrightarrow (\exists a : K(\text{Res } a \varphi))
\]

This definition says that the agent can \(\varphi\) if there exists some action \(a\) such that the agent knows that \(\varphi\) will result from the performance of \(a\). Notice that the variable \(a\) denoting the relevant action is quantified de re, and so this definition implies that the agent knows the identity of the action that results in the achievement of \(\varphi\). To see why this is important, suppose \(\text{Can}\) had instead been defined as follows:

\[
(\text{Can } \varphi) \leftrightarrow K(\exists a : (\text{Res } a \varphi))
\]

In this de dicto definition of ability, in every epistemic alternative there is some action that will result in \(\varphi\), but the identity of the action may be different in different epistemic alternatives: but then, since the agent is uncertain about which of its epistemic alternatives is the actual world, which of these actions should it perform? While the de dicto definition of ability tells us something, it does not seem strong enough to serve as a practical definition of ability.

So, according to our first definition, the agent can achieve \(\varphi\) if there exists some action \(a\), such that the agent knows the identity of \(a\), and that the result of performing \(a\) is \(\varphi\). Moore pointed out, however, that while this definition certainly embodies some intuition of value in understanding ability, it suffers from one major drawback: the agent is required to know the identity of the whole of the action required to achieve the desired state. In everyday usage, this seems much too strong a requirement. For example, we might say to a friend “I can be at your house at 8pm”, being entirely comfortable about the truth of this statement, without knowing in advance exactly what action will be used or required to achieve it. Moore therefore proposed an adapted version of the definition, as follows:
(**Can** $\varphi$) $\leftrightarrow$
\[
\exists a. K (\text{Res } a \varphi) \lor \\
\exists a. K (\text{Res } a (\text{Can } \varphi))
\]

Thus, an agent has the ability to achieve $\varphi$ if either:

1. it knows the identity of an action such that it knows that after this action is performed, $\varphi$ will hold; or

2. it knows the identity of an action such that it knows that after this action is performed, (Can $\varphi$) will hold.

Thus, the second case allows for the possibility of performing an action that will furnish the agent with the capability to achieve $\varphi$.

Moore’s formalism was enormously influential in the AI community. For example, Douglas Appelt used an enriched version of Moore’s formalism directly in a theorem-proving system for planning natural language utterances. Many other researchers used ideas from Moore’s work to formalise ability; of particular lasting significance has been the idea of an agent requiring de re knowledge of an action. In this chapter, we focus on a somewhat more complex notion of ability, that looks at strategies, i.e., conditional plans rather than simple actions. Still, the issue to what extent the agent (or group of agents) know the right strategy, and the difference between knowing a good strategy de re vs. de dicto is a central one. We demonstrate it in Section 11.5.

### 11.3 Logics for Abilities of Agents and Coalitions

In this section we present semantic models of game-like interaction in multi-agent systems, as well as two basic logics for reasoning about such models, namely **Coalition Logic** (CL) and **Alternating-time Temporal Logic** (ATL). We begin with an example leading to the concept of concurrent game model.

**Example 11.1 (Shared file updates)**

Two agents share a file in the cyberspace. The file can be updated only at designated moments (e.g., only on the hour). Agents can try to *update* the file (action U) or *do nothing* (action N). However, updating is disabled in some states. Both agents act simultaneously. Initially (at state $E$), both agents are enabled to apply U. If at any moment both agents try to update simultaneously, there is a conflict, and as a consequence the file is locked forever. If one agent tries to update while the other applies N, the file gets updated by the updating agent $i$ (resulting in state $U_i$). Also, as a very
11.3. LOGICS FOR ABILITIES OF AGENTS AND COALITIONS

simple form of fairness, we assume that no agent can update the file twice in a row. That is, if agent \( i \) applies again \( U \) before the other has applied \( U \), then the process goes to a state \( D_i \) where \( i \) is disabled from updating, and can only apply \( N \). At that state the other agent can enable \( i \) to apply further updates by applying \( N \), too, as the procedure then goes back to state \( U_i \), for \( i = 1, 2 \); alternatively, she can decide to apply \( U \) herself. If both agents apply \( N \) at any state but \( D_i \), then nothing happens. After, if ever, the other agent applies \( U \) at state \( U_i \) or \( D_i \), while agent \( i \) applies \( N \) the procedure goes to state \( P \) (“the file has been processed”) where both agents are disabled from making further updates. From that state, either nothing happens, if both agents apply \( N \), or agent 1 (and only agent 1) can reset the process to state \( E \), by applying action \( R \) (reset).

The example above describes a procedure formalisable as a transition system. At each state of that system every agent chooses one of its available actions, and the resulting tuple of actions is executed synchronously. The combination of actions determines the transition to a successor state, and so on, ad infinitum. Below we define formally such multi-agent transition systems.

11.3.1 Concurrent Game Models

Definition 11.1 (Concurrent game structures and models)
A concurrent game structure (CGS) is a tuple \( S = (Ag, St, Act, act, out) \) which consists of:

- a finite, non-empty set of players or agents\(^1\) \( Ag = \{1, \ldots, k\} \); the subsets of \( Ag \) are often called coalitions;
- a non-empty set of states (game positions) \( St \);
- a non-empty set of actions, or moves \( Act \);
- an action manager function \( act : Ag \times St \rightarrow \mathcal{P}(Act) \) assigning to every player \( a \) and a state \( q \) a non-empty set of actions available for execution by \( a \) at the state \( q \).

An action profile is a tuple of actions \( \alpha = \langle \alpha_1, \ldots, \alpha_k \rangle \in Act^k \). The action profile is executable at the state \( q \) if \( \alpha_i \in act(i, q) \) for every \( i \in Ag \). We denote by \( act(q) \) the subset of \( \prod_{a \in Ag} act(a, q) \) consisting of all action profiles executable at the state \( q \).

- a transition function \( out \) that assigns a unique outcome state \( out(q, \alpha) \) to every state \( q \) and every action profile \( \alpha \) which is executable at \( q \).

\(^1\)We use the terms ‘agent’ and ‘player’ as synonyms.
A concurrent game model (CGM) is a CGS endowed with a labeling \( L: \text{St} \rightarrow \mathcal{P}(\text{PROP}) \) of the states with sets of atomic propositions from a fixed set \( \text{PROP} \). As usual, that labeling describes which atomic propositions are true at a given state.

**Example 11.2 (Shared file updates as a concurrent game model)**
The procedure described in Example 11.1 is encoded in the concurrent game model on Figure 11.1. Each transition is labelled by a pair of actions (the action of agent 1 and the action of agent 2), and the atomic propositions are the same as the names of states where they hold.

**11.3.2 Plays and strategies**
A play in a CGS/CGM \( \mathcal{M} \) is just a path in \( \mathcal{M} \), that is, an infinite sequence of states that can result from subsequent transitions in \( \mathcal{M} \) obtained by joint moves of all players. The formal definitions follow below.

**Definition 11.2 (Plays and histories)**
We fix a concurrent game model \( \mathcal{S} = (\text{Ag}, \text{St}, \text{Act}, \text{act}, \text{out}, L) \). For a play \( \lambda \) and positions \( i, j \geq 0 \), we use \( \lambda[i] \), \( \lambda[j, i] \), and \( \lambda[j, \infty] \) to denote the \( i \)th state
of \( \lambda \), the finite segment \( q_j, q_{j+1}, \ldots, q_i \), and the tail subsequence \( q_j, q_{j+1}, \ldots \) of \( \lambda \), respectively. A play with \( \lambda[0] = q \) will be called an \( q \)-play.

The finite initial segments \( \lambda[0, i] \) of plays \( \lambda \) will be called histories in \( S \); a typical history will be denoted by \( h \) and its length by \( |h| \).

**Definition 11.3 (Strategies)**

A positional (aka. memoryless) strategy in \( S \) for a player \( a \in Ag \) is a function \( s_a : St \rightarrow Act \), such that \( s_a(q) \in act(a, q) \). A positional strategy in \( S \) for a coalition \( C \subseteq Ag \) is a tuple of positional strategies, one for each player in \( C \).

A perfect recall strategy in \( S \) for the player \( a \) is a function \( s_a : St^+ \rightarrow Act \) such that \( s_a(\ldots, q_i) \in act(a, q) \), where \( St^+ \) is the set of all histories in \( S \). A perfect recall strategy in \( S \) for the coalition \( C \) is a tuple of perfect recall strategies, one for each player in \( C \).

Clearly, every positional strategy can be seen as a perfect recall strategy.

**Definition 11.4 (Coalitional actions)**

Let \( q \in St \) and let \( C \subseteq Ag \). We denote the complement \( Ag \setminus C \) by \( \overline{C} \).

- A \( C \)-action at the state \( q \) is a tuple \( \alpha_C \) such that \( \alpha_C(i) \in act(i, q) \) for every \( i \in C \) and \( \alpha_C(j) = \tilde{z}_j \) for every \( j \notin C \), where \( \tilde{z}_j \) is a fixed symbol used as a placeholder for an arbitrary action of player \( j \).

We denote by \( act(C, q) \) the set of all \( C \)-actions at state \( q \).

Alternatively, \( C \)-actions at \( q \) can be defined as equivalence classes on the set of all action profiles at \( q \), where each equivalence class is determined by the choices of actions of the players in \( C \).

- An action profile \( \alpha \in act(q) \) extends a \( C \)-action \( \alpha_C \), denoted \( \alpha_C \sqsubseteq \alpha \), if \( \alpha(i) = \alpha_C(i) \) for every \( i \in C \).

- Given a \( C \)-action \( \alpha_C \in act(C, q) \) and a \( \overline{C} \)-action \( \alpha_{\overline{C}} \in act(\overline{C}, q) \), we denote by \( \alpha_C \oplus \alpha_{\overline{C}} \) the unique action profile \( \alpha \in act(q) \) such that both \( \alpha_C \sqsubseteq \alpha \) and \( \alpha_{\overline{C}} \sqsubseteq \alpha \).

**Definition 11.5 (Outcome sets)**

Let \( q \in St, C \subseteq Ag \), and \( \alpha_C \in act(C, q) \).

The outcome set of the \( C \)-action \( \alpha_C \) at \( q \) is the set of states

\[
\text{out\_set}(q, \alpha_C) := \{ \text{out}(q, \alpha) \mid \alpha \in act(q) \text{ and } \alpha_C \sqsubseteq \alpha \}
\]

The outcome set function \( \text{out\_set} \) can be naturally extended to act not just on joint actions but on all joint strategies applied at a given state.
(resp., history) in a given CGM. Then $\text{out\_set}(q, s_C)$ (resp., $\text{out\_set}(h, s_C)$) returns the set of all possible successor states that can result from applying a given positional (resp., perfect recall) joint strategy $s_C$ of the coalition $C$ at the state $q$ (resp., the history $h$).

Further, we extend the function $\text{out\_set}$ to $\text{out\_plays}$ that returns the set of all plays which can be realized when the players in $C$ follow the strategy $s_C$ from a given state $q$ in $S$ onward. Formally, it is defined as:

For memoryless strategies,

$$\text{out\_plays}(q, s_C) := \{\lambda \in \text{St}^\omega \mid \lambda[0] = q \text{ and } \lambda[j + 1] \in \text{out\_set}(\lambda[j], s_C) \text{ for each } j \in \mathbb{N}\}$$

Respectively, for perfect recall strategies,

$$\text{out\_plays}(q, s_C) := \{\lambda \in \text{St}^\omega \mid \lambda[0] = q \text{ and } \lambda[j + 1] \in \text{out\_set}(\lambda[0, j], s_C) \text{ for each } j \in \mathbb{N}\}$$

A fundamental question about a multi-player game is: What can a given player or coalition achieve in the game? So far the objectives of players and coalitions have not been formally specified. A typical objective would be to reach a state satisfying a given property expressed in terms of the atomic propositions, e.g. a winning state. Generally, an objective is a property of plays, e.g. one can talk about winning or losing plays for the given player or coalition. More precisely, if the current state of the game is $q$, we say that a coalition of players $C$ can bring about (or enforce) an objective $\varphi$ from that state if there is a joint strategy $s_C$ for $C$ such that every play from $\text{out\_plays}(q, s_C)$ satisfies the objective $\varphi$. We will now introduce suitable logics that can be used to formally specify strategic objectives of players and coalitions and to formally determine their ability to bring about such objectives.

11.3.3 Expressing Local Coalitional Powers: Coalition Logic

The power of a coalition to enforce the outcome to be in one or another designated outcome set naturally corresponds to a non-normal modal operator with monotone neighbourhood semantics. This observation was formalized by the Coalition Logic (CL) introduced by Marc Pauly as a multi-modal logic capturing coalitional abilities in strategic games. It extends classical propositional logic with a family of modal operators $\langle C \rangle$ indexed with the subsets (coalitions) of the set of agents $A$. Formulas of CL are defined recursively as follows:

$$\varphi := p \mid \neg \varphi \mid \varphi \lor \psi \mid \langle A \rangle \varphi.$$
11.3. LOGICS FOR ABILITIES OF AGENTS AND COALITIONS

where \( p \in \Pi \) is a proposition, and \( A \subseteq \text{Ag} \) is a set (coalition) of agents.

Intuitively, the formula \( \langle C \rangle \varphi \) says that the coalition \( C \) has, at the given game state, the power to guarantee an outcome satisfying \( \varphi \). Formally, one can interpret the operator \( \langle C \rangle \) in a state \( q \) of concurrent game model \( M \) as follows:

\[
M, q \models \langle C \rangle \varphi \text{ iff } \text{out\_set}(q, \alpha_C) \subseteq [\varphi]_M \text{ for some } \alpha_C \in \text{act}(C, q),
\]

where \([\varphi]_M := \{ q \in \text{St} | M, q \models \varphi \} \).

**Example 11.3 (Expressing properties in CL)**

Here are some properties expressed in CL. The reader might ponder each of them whether it should be satisfiable, valid, or neither.

1. “If Player 1 has an action to guarantee a winning successor state, then Player 2 cannot prevent reaching a winning successor state.”
   \[
   \langle 1 \rangle \text{win}_1 \rightarrow \neg \langle 2 \rangle \neg \text{win}_1
   \]
2. “Player \( a \) has an action to guarantee a successor state where he is rich, and has an action to guarantee a successor state where he is happy, but does not have an action to guarantee a successor state where he is both rich and happy.”
   \[
   \langle a \rangle \text{rich} \wedge \langle a \rangle \text{happy} \wedge \neg \langle a \rangle (\text{rich} \wedge \text{happy})
   \]
3. “None of the players 1 and 2 has an action ensuring an outcome state satisfying \( \text{goal} \), but they both have a collective action ensuring such an outcome state.”
   \[
   \neg \langle 1 \rangle \text{goal} \wedge \neg \langle 2 \rangle \text{goal} \wedge \langle 1,2 \rangle \text{goal}
   \]

Pauly has shown that the following is a sound and complete axiomatization of CL over the classical propositional logic:

1. \( \langle \text{Ag} \rangle \top \)
2. \( \neg \langle C \rangle \bot \) for any coalition \( C \subseteq \text{Ag} \)
3. \( \neg \langle \emptyset \rangle \varphi \rightarrow \langle \text{Ag} \rangle \neg \varphi \)
4. \( \langle C \rangle \varphi \land \langle D \rangle \psi \rightarrow \langle C \cup D \rangle (\varphi \land \psi) \) for any disjoint \( C, D \subseteq \text{Ag} \),

plus the inference rule Monotonicity:

\[
\frac{\varphi \rightarrow \psi}{\langle C \rangle \varphi \rightarrow \langle C \rangle \psi}
\]
11.3.4 Alternating-time temporal logics

Coalition logic is a very natural language to express local strategic abilities of players and coalitions, that is, their powers to guarantee desired properties in the successor states. However, it is not expressive enough for reasoning about long term strategic abilities. A suitable logic for that purpose is Alternating-time Temporal Logic $\text{ATL}^*$, widely used for specifying and reasoning about temporal properties of plays in multiagent systems that different players and coalitions can enforce by adopting suitable strategies.

Syntax of ATL$^*$ and ATL

The language of ATL$^*$ extends that of the Computation Tree Logic CTL$^*$ with strategic path quantifiers indexed by all subsets of a fixed finite non-empty set $\mathbb{A}_g$ of agents (or players).

**Definition 11.6 (ATL$^*$-formulae)**

The formulae of ATL$^*$ are defined for the fixed set of agents $\mathbb{A}_g$ and a fixed countably infinite set $\text{PROP}$ of atomic propositions, recursively as follows:

$$
\varphi := p \mid \neg \varphi \mid (\varphi \land \varphi) \mid X\varphi \mid G\varphi \mid (\varphi U \varphi) \mid \langle C \rangle \varphi,
$$

where $p$ ranges over $\text{PROP}$ and $C$ ranges over $\mathcal{P}(\mathbb{A}_g)$.

The intuitive interpretation of $\langle C \rangle \varphi$ is: "The coalition $C$ has a joint strategy that ensures that every path enabled by that strategy satisfies $\varphi$", and the temporal operators have the usual interpretation: $X$ stands for "next", $G$ for "always from now on", and $U$ for "until". The other boolean connectives and the propositional constants $\top$ ("truth") and $\bot$ ("falsum") are defined as usual. Just like in LTL and CTL$^*$, $F \varphi$ ("eventually") can be defined as $\top U \varphi$. We will also use dual strategic path quantifiers $\langle C \rangle$ defined as

$$
\langle C \rangle \varphi := \neg \langle C \rangle \neg \varphi,
$$

with the intended interpretation "the coalition $C$ cannot prevent an outcome satisfying $\varphi$".

As in CTL$^*$, the formulae of ATL$^*$ can be classified in two sorts: state formulae, that are evaluated at game states, and path formulae, that are evaluated on game plays. These are respectively defined by the following grammars, where $C \subseteq \mathbb{A}_g, p \in \text{PROP}$:

**State formulae:** $\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \langle C \rangle \gamma,$

**Path formulae:** $\gamma ::= \varphi \mid \neg \gamma \mid (\gamma \land \gamma) \mid X\gamma \mid G\gamma \mid (\gamma U \gamma).$
11.3. LOGICS FOR ABILITIES OF AGENTS AND COALITIONS

By analogy with CTL, we can now define the fragment ATL consisting of the ‘purely-state-formulae’ of ATL*, i.e., those ATL* formulae where every temporal operator is in the immediate scope of a strategic path quantifier. Formally:

**Definition 11.7**
The formulae of ATL are defined recursively as follows:

$$\varphi := p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \langle C \rangle X \varphi \mid \langle C \rangle G \varphi \mid \langle C \rangle (\varphi U \varphi),$$

where $p$ ranges over $PROP$ and $C$ ranges over $\mathcal{P}(Ag)$.

ATL operators can be intuitively interpreted as follows:

- $\langle C \rangle X \varphi$: ‘The coalition $C$ has a joint action that ensures $\varphi$ in the next moment (state)’;
- $\langle C \rangle G \varphi$: ‘The coalition $C$ has a joint strategy to maintain forever outcomes satisfying $\varphi$’;
- $\langle C \rangle (\psi U \varphi)$: ‘The coalition $C$ has a joint strategy to eventually reach an outcome satisfying $\varphi$, while meanwhile maintaining the truth of $\psi$’.

**Example 11.4 (Expressing properties in ATL*)**
Here are some properties expressed in ATL* (actually, all but the last one, are formulae in ATL).

- If the system has a strategy to eventually reach a safe state, then the environment cannot prevent it from reaching a safe state:
  $$\langle \text{system} \rangle F \text{safe} \rightarrow [[\text{env}]] F \text{safe}$$

- If the system has a strategy to stay in a safe state forever and has a strategy to eventually achieve its goal, then it has a strategy to stay in a safe state until it achieves its goal:
  $$\langle \text{system} \rangle G \text{safe} \land \langle \text{system} \rangle F \text{goal} \rightarrow \langle \text{system} \rangle (\text{safe U goal})$$

- The coalition $C$ has a joint action to ensure that the coalition $B$ cannot prevent $C$ from eventually winning:
  $$\langle C \rangle [B] F \text{win}_C.$$
CHAPTER 11. KNOWLEDGE AND ABILITY

Semantics of ATL

Definition 11.8 (Semantics of ATL on concurrent game models)
Let \( M = (Ag, St, Act, act, out, L) \) be a concurrent game model over a fixed set of atomic propositions \( PROP \). The truth of ATL\(^\ast\) -formulae is defined by mutual induction on state and path formulae as follows:

1. For state formulae, at a state \( q \in St \):
   
   - \( M, q \models p \iff p \in L(q) \), for all \( p \in PROP \);
   - \( M, q \models \neg \varphi \iff M, q \not\models \varphi \);
   - \( M, q \models \varphi \land \psi \iff M, q \models \varphi \) and \( M, q \models \psi \);
   - \( M, q \models \langle C \rangle \varphi \iff \) there exists a perfect recall \( C \)-strategy \( s_C \) such that \( M, \lambda \models \varphi \) holds for all \( \lambda \in \text{out}\_\text{plays}(q, s_C) \).

2. For path formulae, at a path \( \lambda \in St^\omega \):
   
   - \( M, \lambda \models \varphi \iff M, \lambda[0] \models \varphi \), for every state formula \( \varphi \);
   - \( M, \lambda \models \neg \gamma \iff M, \lambda \not\models \gamma \);
   - \( M, q \models \gamma_1 \land \gamma_2 \iff M, q \models \gamma_1 \) and \( M, q \models \gamma_2 \);
   - \( M, \lambda \models X\gamma \iff M, \lambda[1, \infty) \models \gamma \);
   - \( M, \lambda \models G\gamma \iff M, \lambda[i, \infty) \models \gamma \) holds for all positions \( i \geq 0 \);
   - \( M, q \models \gamma_1 \cup \gamma_2 \iff \) for some position \( i \geq 0 \) both \( M, \lambda[i, \infty) \models \gamma_2 \) and \( M, \lambda[j, \infty) \models \gamma_1 \) hold for all positions \( 0 \leq j < i \).

Because ATL only involves state formulae, the memory used in the strategies required to satisfy \( \langle C \rangle \)-formulae turns out redundant and these can be replaced with positional strategies. Formally, let us denote by \( \models^p \) the truth definition for ATL\(^\ast\) -formulae given above, where perfect recall strategies are replaced by positional strategies throughout. Then, the following holds:

For every concurrent game model \( M \), ATL-formula \( \varphi \) and a state \( q \in M \), we have that \( M, q \models \varphi \iff M, q \models^p \varphi \).

Thus, we have the following equivalent, but more explicit truth definition for ATL-formulae.

\(^2\) Note that the analogous claim does not hold for the broader language of ATL\(^\ast\). To see this, consider the last formula in Example 11.5, which holds according to the perfect recall semantics \( \models \) but not according to the positional semantics \( \models^p \).
Definition 11.9 (Truth of ATL-formulae)
Let $\mathcal{M} = (\text{Ag}, \text{St}, \text{Act}, \text{act}, \text{out}, L)$ be a concurrent game model over a fixed set of atomic propositions PROP. The truth of ATL-formulae at a state $q \in \text{St}$ is defined inductively as follows:

- $\mathcal{M}, q \models p$ iff $p \in L(q)$, for all $p \in \text{PROP}$;
- $\mathcal{M}, q \models \neg \varphi$ iff $\mathcal{M}, q \not\models \varphi$;
- $\mathcal{M}, q \models \varphi \land \psi$ iff $\mathcal{M}, q \models \varphi$ and $\mathcal{M}, q \models \psi$;
- $\mathcal{M}, q \models \langle C \rangle \varphi$ iff there exists a $C$-action $\alpha_C \in \text{act}(C, q)$ such that $\mathcal{M}, q' \models \varphi$ for all $q' \in \text{out}_{set}(q, \alpha_C)$;
- $\mathcal{M}, q \models \langle C \rangle \varphi \upharpoonright \psi$ iff there exists a positional $C$-strategy $s_C$ such that, for all $\lambda \in \text{out}_{plays}(q, s_C)$, $\mathcal{M}, \lambda[i] \models \varphi$ holds for all positions $i \geq 0$;
- $\mathcal{M}, q \models \langle C \rangle \varphi \downharpoonright \psi$ iff there exists a positional $C$-strategy $s_C$ such that, for all $\lambda \in \text{out}_{plays}(q, s_C)$, there exists a position $i \geq 0$ such that $\mathcal{M}, \lambda[i] \models \psi$ and $\mathcal{M}, \lambda[j] \models \varphi$ holds for all positions $0 \leq j < i$.

While ATL and CL were developed independently, it is easy to see that CL is equivalent to the next-time fragment of ATL. That is, we have $\mathcal{M}, q \models_{CL} \langle C \rangle \varphi$ iff $\mathcal{M}, q \models_{ATL} \langle C \rangle \varphi$, for every $\mathcal{M}, q, C$, and $\varphi$.

Example 11.5 (Properties of shared file updates)
Consider the shared file updates in Example 11.1. Denote the model by $\mathcal{M}_1$. With a slight abuse we use the same notation for states and for the atomic propositions (nominals) that identify them completely. We leave to the reader to verify the following.

- $\mathcal{M}_1, E \models \neg \langle 1 \rangle X D \land \langle 1 \rangle X \neg D \land \langle 1 \rangle X (E \lor U_2) \land \neg \langle 2 \rangle X (E \lor U_2)$
- $\mathcal{M}_1, E \models \neg \langle 1 \rangle F D \land \neg \langle 2 \rangle F D \land \langle 1 \rangle G \neg D \land \langle 2 \rangle G \neg D$
- $\mathcal{M}_1, U_1 \models \neg \langle 1 \rangle F P \land \langle 1, 2 \rangle G P$
- $\mathcal{M}_1, U_2 \models \langle 1, 2 \rangle (\neg D_2) \cup U_1 \land \neg \langle 1 \rangle (\neg D_2) \cup U_1$
- $\mathcal{M}_1, E \models \langle 1, 2 \rangle F (P \land X (E \land G \neg P))$. NB: the truth of this formula requires a strategy for 1 and 2 using memory.
CHAPTER 11. KNOWLEDGE AND ABILITY

Validity and Satisfiability in ATL

Definition 11.10
Let $\varphi$ be an ATL-formula and $\Gamma$ be a set of ATL-formulae.

- $\varphi$ is **satisfiable** if $M, q \models \varphi$ for some CGM $M$ and a state $q \in M$;
- $\varphi$ is **valid** if $M, q \models \varphi$ for every CGM $M$ and every state $q \in M$.

Satisfiability and validity of $\Gamma$ are defined similarly.

Pauly’s axioms for Coalition Logic, that we presented in Section 11.3.3, can be rephrased in ATL to provide validities of one-step ability:

\[
\begin{align*}
(T) & \quad \langle \text{Ag}\rangle X \top \\
(\bot) & \quad \lnot \langle C\rangle X \bot \\
(Ag) & \quad \lnot \langle \emptyset \rangle X \varphi \rightarrow \langle \text{Ag}\rangle X \lnot \varphi \\
(Sup) & \quad \langle C\rangle X \varphi \land \langle D\rangle X \psi \rightarrow \langle C \cup D\rangle X (\varphi \land \psi) \text{ for any disjoint } C, D \subseteq \text{Ag}.
\end{align*}
\]

Long-term abilities satisfy the following validities in ATL that define operators $\langle C\rangle X$, $\langle C\rangle G$, $\langle C\rangle U$ recursively as fixpoints of certain monotone operators:

\[
\begin{align*}
(FP_G) & \quad \langle C\rangle G \varphi \leftrightarrow \varphi \land \langle C\rangle X \langle C\rangle G \varphi \\
(GFP_G) & \quad \langle \emptyset \rangle G (\theta \rightarrow (\varphi \land \langle C\rangle X \theta)) \rightarrow \langle \emptyset \rangle G (\varphi \rightarrow \langle C\rangle G \varphi) \\
(FP_U) & \quad \langle C\rangle \psi U \varphi \leftrightarrow \varphi \lor (\psi \land \langle C\rangle X \langle C\rangle \psi U \varphi) \\
(LFP_U) & \quad \langle \emptyset \rangle G ((\varphi \lor (\psi \land \langle C\rangle X \theta)) \rightarrow \theta) \rightarrow \langle \emptyset \rangle G (\langle C\rangle \psi U \varphi \rightarrow \theta).
\end{align*}
\]

Axioms $(T)$–$(LFP_U)$ plus the inference rules of $\langle C\rangle X$-**Monotonicity** and $\langle \emptyset \rangle G$-**Necessitation**:

\[
\begin{align*}
\varphi \rightarrow \psi & \quad \frac{\langle C\rangle X \varphi \rightarrow \langle C\rangle X \psi}{\langle C\rangle X \varphi} \\
\varphi & \quad \frac{\langle \emptyset \rangle G \varphi}{\langle C\rangle X \varphi}
\end{align*}
\]

provide a sound and complete axiomatization for the validities of ATL. Furthermore, the satisfiability in ATL is decidable and EXPTIME-complete. The whole ATL$^*$ is 2EXPTIME-complete, but no explicit complete axiomatization for it is known yet.
11.4 Abilities under Incomplete Information

Usually agents have incomplete knowledge about the environment where they act, as well as about the current course of affairs, including the current states of the other agents, the actions they take, etc. That, of course, affects essentially their abilities to achieve individual or collective objectives. In this section, we combine concurrent game models and epistemic models in order to give semantics to a language expressing strategic ability under imperfect/incomplete information. We do not involve epistemic operators in the object language yet (this will be done in Section 11.5). Thus, knowledge is reflected only in the models, through agents’ epistemic relations (aka indistinguishability relations), but not explicitly referred in the language. As we will show, taking knowledge into account on purely semantic level is already sufficient to make a strong impact on the meaning of strategic operators, and the patterns of ability that can emerge.

In game theory, two different terms are traditionally used to indicate lack of information: “incomplete” and “imperfect” information. Usually, the former refers to uncertainties about the game structure and rules, while the latter refers to uncertainties about the history, current state, etc. of the specific play of the game. The models that we use allow for representing both types of uncertainty in a uniform way. We take the liberty to use the term “imperfect information” in the stricter game-theoretic sense, whereas we will use “incomplete information” more loosely, to indicate any possible relevant lack of information.

11.4.1 Incomplete Information and Uniform Strategies

The decision making and abilities of strategically reasoning players are strongly influenced by the knowledge they possess about the world, the other players, past actions, etc. So far we have considered structures of complete and (almost) perfect information in the sense that players were completely aware of the rules and structure of the game system and of the current state of the play, and the only information they lack is the choice of actions of the other players at the current state. However, in reality this is seldom the case: usually players have only partial information, both about the structure and the rules of the game, and about the precise history and the current state of a specific play of the game. In the following we are concerned with the following question: What can players achieve in a game if they are not completely informed about its structure and the current play?

We represent players’ incomplete information about the world by indistinguishability relations $\sim_a \subseteq S_t \times S_t$ on the state space, as discussed in Section 11.2. We write $q \sim_a q'$ to describe player a’s inability to discern
between states \( q \) and \( q' \). Both states appear identical from \( a \)'s perspective. The indistinguishability relations are traditionally assumed to be equivalence relations. The knowledge of a player is then determined as follows: a player knows that property \( \varphi \) holds in a state \( q \) if \( \varphi \) is the case in all states indistinguishable from \( q \) for that player.

How does the incomplete information modelled by an indistinguishability relation affect the strategic abilities of the player? In the case of ATL with complete information, abilities were derived from strategies defined on states or their sequences, i.e., memoryless or perfect recall strategies. For incomplete information, the picture is similar. However, the two notions of a strategy (as well as the definition of outcome) must take into account some constraints due to uncertainty of the players.

Formally, a concurrent game with incomplete information can be modelled by a concurrent epistemic game structure (CEGS) is a tuple

\[ S = (\mathbb{A}_g, \mathbb{S}, \{\sim_a \mid a \in \mathbb{A}_g\}, \mathbb{A}_{ct}, \mathbb{A}_{out}) \]

where \((\mathbb{A}_g, \mathbb{S}, \mathbb{A}_{ct}, \mathbb{A}_{out})\) is a CGS and \(\sim_a\) are indistinguishability relations over \(\mathbb{S}\), one per agent in \(\mathbb{A}_g\). A basic assumption in the case of incomplete information is that players have the same choices of actions in indistinguishable states, for otherwise, they would have a way to discern between these states. That is, we require that if \(q \sim_a q'\) then \(\mathbb{A}_{ct}(q) = \mathbb{A}_{ct}(q')\).

Just like CGM, a concurrent epistemic game model (CEGM) is defined by adding to a CEGS a labeling of game states with sets of atomic propositions.

The notion of strategy must be refined in the incomplete information setting in order to be realistic. In such setting an “executable” strategy has to assign the same choices to indistinguishable situations. Such strategies are called uniform.

**Definition 11.11 (Uniform strategies)**

A memoryless strategy \(s_a\) is uniform if the following condition is satisfied:

\[ \text{for all states } q, q' \in \mathbb{S}, \text{ if } q \sim_a q' \text{ then } s_a(q) = s_a(q'). \]

For perfect recall strategies we first lift the indistinguishability between states to indistinguishability between sequences of states. Two histories \(h = q_0q_1 \ldots q_n\) and \(h' = q'_0q'_1 \ldots q'_{n'}\) are indistinguishable for agent \(a\), denoted by \(h \approx_a h'\), if and only if \(n = n'\) and \(q_i \sim_a q'_i\) for \(i = 1, \ldots, n\). Now, a perfect recall strategy \(s_a\) is uniform if the following condition holds:

\[ \text{for all histories } h, h' \in \mathbb{S}^+, \text{ if } h \approx_a h' \text{ then } s_a(h) = s_a(h'). \]

Uniform joint strategies are defined as tuples of uniform individual strategies. Note that this definition presumes no communication between
11.4. ABILITIES UNDER INCOMPLETE INFORMATION

Figure 11.2: “Poor duck model” $\mathcal{M}_2$ with one player ($a$) and transitions labeled with $a$’s actions. Dotted lines depict the indistinguishability relations. Automatic transitions (i.e., such that there is only one possible transition from the starting state) are left unlabeled.

the agents that could reduce their uncertainties. We will come back to this issue again later.

Example 11.6 (Poor duck)
Consider model $\mathcal{M}_2$ from Figure 11.2, with the following story. A man wants to shoot down a yellow rubber duck in a shooting gallery. The man knows that the duck is in one of the two cells in front of him, but he does not know in which one. He can either decide to shoot to the left (action $\text{shoot}_L$), or to the right (action $\text{shoot}_R$), or reach out to the cells and look what is in (action $\text{look}$). Note that only one of the shooting actions can be successful – which one it depends on the starting state of the game.\footnote{For more seriously minded readers, we propose an alternative story: agent $a$ is a doctor who can apply two different treatments to a patient with symptoms that fit two different diseases. Additionally, the doctor can order a blood test to identify the disease precisely.}

Intuitively, the man does not have a strategy to ensure shooting down the duck in one step, at least not from his subjective point of view. On the other hand, he should be able to ensure it in multiple steps if he has a perfect recall of his observations. We will formalize the intuitions in the next subsection.
Example 11.7 (Shared file updates with incomplete information)

Consider again the shared file updating procedure in Example 11.1 and now suppose that Agent 2 cannot distinguish states $E$ and $U_1$ (e.g., because he cannot observe the action of Agent 1 at state $E$) and, likewise, Agent 1 cannot distinguish states $U_2$ and $D_2$. The resulting concurrent game model with incomplete information is given in Figure 11.3.

11.4.2 Reasoning about Abilities under Uncertainty

Agents’ incomplete information and use of memory can be incorporated into ATL$^*$ in different ways. One possible approach is not to change the logical language but to consider variations of the semantics, reflecting these by suitably varying the notion of strategy employed in the truth definition of the strategic operators. Combining perfect recall and memoryless strategies, on the one hand, with complete and incomplete information, on the other hand, gives rise to 4 natural variants of the semantics for ATL$^*$ (as originally discussed by Schobbens):

$\models_{IR}$: complete information and perfect recall strategies;
11.4. ABILITIES UNDER INCOMPLETE INFORMATION

|=_{IR}: complete information and memoryless strategies;
|=_{iR}: incomplete information and perfect recall strategies;
|=_{ir}: incomplete information and memoryless strategies.

The 4 semantic variants are obtained by different updating of the main semantic clause from Section 11.3. Let \( \mathcal{M} \) be a CEGM, and let \( \sim_{C} := \bigcup_{a \in C} \sim_{a} \) be the epistemic relation corresponding to the group knowledge of coalition \( C \) (i.e., what everybody in \( C \) knows). Moreover, let \([q]_{a} = \{q' \mid q \sim_{a} q'\}\) denote the information set of agent \( a \), and analogously for coalitions \( C \). The truth definitions for \( |=_{xy} \), where \( x \in \{I, i\} \), \( y \in \{R, r\} \), read as follows:

\[\mathcal{M}, q |=_{IR} \langle C \rangle \gamma \text{ iff there is a joint perfect recall strategy } s_{C} \text{ for } C \text{ such that } \mathcal{M}, \lambda |= \gamma \text{ for every play } \lambda \in \text{out}\_\text{plays}(q, s_{C})\]

\[\mathcal{M}, q |=_{iR} \langle C \rangle \gamma \text{ iff there is a joint memoryless strategy } s_{C} \text{ for } C \text{ such that } \mathcal{M}, \lambda |= \gamma \text{ for every play } \lambda \in \text{out}\_\text{plays}(q, s_{C})\]

\[\mathcal{M}, q |=_{IR} \langle C \rangle \gamma \text{ iff there is a uniform joint perfect recall strategy } s_{C} \text{ for } C \text{ such that } \mathcal{M}, \lambda |= \gamma \text{ for every play } \lambda \in \bigcup_{q' \in [q]_{C}} \text{out}\_\text{plays}(q', s_{C})\]

\[\mathcal{M}, q |=_{ir} \langle C \rangle \gamma \text{ iff there is a uniform joint memoryless strategy } s_{C} \text{ for } C \text{ such that } \mathcal{M}, \lambda |= \gamma \text{ for every play } \lambda \in \bigcup_{q' \in [q]_{C}} \text{out}\_\text{plays}(q', s_{C}).\]

It is easy to see that the standard semantic relation \( |= \) introduced in Section 11.3 corresponds to \( |=_{IR} \), and the positional semantics \( |=^{p} \) is the same as \( |=_{IR} \).

Another option is to extend the syntax of cooperation modalities with subscripts: \( \langle A \rangle_{xy} \), where \( x \) indicates the use of memory in the strategies (memoryless if \( x = r \) and perfect recall if \( x = R \)) and \( y \) indicates the information setting (complete if \( y = I \) and incomplete information if \( y = i \)).

**Example 11.8 (Shooting the duck in variants of ATL)**

Consider the “poor duck” model \( \mathcal{M}_{2} \) from Example 11.6. There is no good strategy for the man to shoot down the duck in one step regardless of the kind of memory that the man possesses – formally, \( \mathcal{M}_{2}, q_{0} |=_{iR} \neg \langle a \rangle X\text{shot} \) and \( \mathcal{M}_{2}, q_{0} |=_{ir} \neg \langle a \rangle X\text{shot}. \) However, he should be able to achieve it in multiple steps if he can remember and use his observations, i.e., \( \mathcal{M}_{2}, q_{0} |=_{IR} \langle a \rangle F\text{shot}. \)

On the other hand, suppose that it has been a long party, and the man is very tired, so he is only capable of using memoryless strategies at the moment. Does he have a memoryless strategy which he knows will achieve the goal? No. In each of these cases the man risks that he will fail
(at least from his subjective point of view). In consequence, $M_2, q_0 \models_{ir} \neg \langle a \rangle F\text{shot}$. Interestingly enough, the man can identify an opening strategy that will guarantee his knowing how to shoot the duck in the next moment: $M_2, q_0 \models_{ir} \langle a \rangle X\langle a \rangle F\text{shot}$. The opening strategy is to look; if the system proceeds to $q_4$ then the second strategy is to shoot to the left, otherwise the second strategy is to shoot to the right.

Example 11.9 (Properties of shared file under incomplete information)
Consider Example 11.7 about shared file updates with incomplete information. Denote the model on Figure 11.3 by $M_3$. Again, we use the same notation for states and for the atomic propositions identifying them. The following hold:

- $M_3, E \models_{ir} \langle 1, 2 \rangle XU_1$. Indeed, there is a joint uniform strategy for $\{1, 2\}$ that enforces $XU_1$ from all states $q$ such that $E \sim_{\{1, 2\}} q$, i.e., states $E, U_1$. The strategy is agent 1 playing $U$ at $E$ and $N$ at $U_1$; agent 2 playing $N$ at $E, U_1$ (and anything at the other states).

- On the other hand, $M_3, E \models_{ir} \neg \langle 1, 2 \rangle XU_2$. Although there is a joint uniform strategy that objectively enforces $XU_2$ (e.g., 1 playing $N$ and 2 playing $U$ at $E, U_1$, and anything at the other states), the strategy does not guarantee success in the subjective judgment of agent 2, because it does not achieve the goal from state $U_1$ (which player 2 thinks might be the case).

- $M_3, E \models_{ir} \langle 1, 2 \rangle ((\neg D_1 \wedge \neg D_2) U P)$. A joint uniform strategy that enforces $(\neg D_1 \wedge \neg D_2) U P$ prescribes agent 1 to play $N$ at $E$ and $U_1$, $U$ at $U_2$ and anything at all other states, while it requires agent 2 to play $U$ at states $E$ and $U_1$, $N$ at $U_2$ and anything at all other states. Note that the requirement for agent 1 to play $N$ at $U_1$ is irrelevant for the objective outcome of the strategy, but it is needed to assure the subjective judgment of agent 2 that the joint strategy will also work from state $U_1$.

- $M_3, U_2 \models_{ir} \langle 1, 2 \rangle (\neg D_2 U P)$. Again, the goal $(\neg D_2 U P)$ can be achieved objectively from state $U_2$ by a uniform strategy, but not subjectively, because for all that agent 1 knows at $U_2$ the formula may be already falsified if the current state were $D_2$.

- Still, $M_3, E \models_{ir} \langle 1, 2 \rangle (\neg D_2 U P)$. Indeed, the uncertainty of agent 2 between states $U_2$ and $D_2$ no longer matters, because being at state $E$ agent 1 knows for sure that the joint action $(N, U)$ would take the system to state $U_2$ and not to $D_2$ and therefore playing $(U, N)$ at that state must succeed. All he needs to do is follow blindly that strategy.
11.4. ABILITIES UNDER INCOMPLETE INFORMATION

when/if the system gets in his information set \( \{U_2, D_2\} \). Thus, the following strategy enforces \( \neg D_2 U P \) from \( E, U_1 \): agent 1 plays \( N \) at state \( E \) and \( U \) at \( U_2, D_2 \); agent 2 plays \( U \) at states \( E, U_1 \) and \( N \) at \( U_2 \).

- \( \mathcal{M}_3, E \models_{ir} \neg \langle 1, 2 \rangle F(D_1 \land F(U_1 \land FP)) \), because any successful joint memoryless strategy would have to prescribe joint actions \( (U, N) \) at \( E \) and \( U_1 \) and \( (N, N) \) at \( D_1 \) and therefore would never reach \( P \) thereafter.

- However, \( \mathcal{M}_3, \langle E \rangle \models_{ir} \langle 1, 2 \rangle F(D_1 \land F(U_1 \land FP)) \), where \( \langle E \rangle \) is the one state history starting at state \( E \). Indeed, a joint uniform memory-based strategy prescribing a play beginning with \( E, U_1, D_1, U_1, P, \ldots \) would succeed. Note that this strategy only requires only 1 extra memory cell per agent, needed to record the passing through state \( D_1 \). We leave the details to the reader.

Some remarks:

1. Ability under incomplete information can additionally be classified into objective and subjective. The former refers to existence of a strategy that is guaranteed to succeed from the perspective of an external observer with complete information about the model, while the latter requires that strategy to be guaranteed to succeed from the perspective of the player/coalition executing it. The definition above is based on subjective ability. As it requires that the executing players must be able to identify the right strategy on their own (within the limits of their incomplete information), it imposes stronger requirements on the strategy than the objective ability. Technically, this is because in the evaluation of a state formula \( \langle C \rangle \gamma \) in state \( q \), when judging the suitability of the selected uniform strategy for \( C \) in terms of the coalition’s subjective ability all epistemic alternatives of \( q \) wrt \( \sim_C \) are to be taken into account, whereas objectively it suffices to check that the strategy only succeeds from \( q \).

2. The distinction between objective and subjective ability is closely related to the first two interpretations of ability by McCarthy and Hayes, that were discussed in Section 11.2. Objective ability to enforce \( \varphi \) means that there exists a strategy (a “subprogram” \( \sigma \) in McCarthy and Hayes’ terminology) that, if executed, will guarantee \( \varphi \). Subjective ability means that the decision maker(s) (the “main program” \( \pi \) for McCarthy and Hayes) has enough information to verify that \( \sigma \) enforces \( \varphi \). We note that the third, strongest level of ability from
McCarthy and Hayes cannot be expressed in the logics presented in this chapter, as they embed no notion of problem-solving procedure \emph{inside} their semantics.

3. Alternative semantics, where common or distributed knowledge of $C$ are taken into account when judging the ability of the coalition to identify the right strategy, have also been considered. We will return to this topic in Section 11.5 where knowledge operators are added explicitly to the language.

4. Most modal logics of strategic ability agree that \emph{executable} strategies are exactly the ones that obey the uniformity constraints. For a joint strategy, this means that each member of the coalition must be able to carry out his part of the joint plan individually. An alternative approach has been presented by Guelev and colleagues, where the notion of uniform joint strategy for a coalition is redefined, based on a suitable \emph{group indistinguishability relation} for the coalition. The most interesting case is when the “distributed knowledge” relation is used. Conceptually, this amounts to assuming that members of the coalition have unlimited communicating capabilities, and freely share relevant information \emph{during the execution of the strategy}.

11.4.3 Impact of Knowledge on Strategic Ability

Clearly, knowledge and abilities are not independent: the more one knows, the more one can achieve by choosing a better suited strategy. We have seen that limited information influences the range of available strategies that agents and coalitions can choose. Respectively, it also affects the semantics of claims about an agent (or a coalition) being able to enforce a given outcome of the game. How big is the impact? As it turns out, adding agents’ knowledge (or, dually, uncertainty) to the semantics of ATL changes a lot. We will review the impact \emph{very} briefly here. The interested reader is referred to the bibliographic notes for details.

Valid sentences

Traditionally, a logic is identified with the set of sentences that are true in the logic; a semantics is just a possible way of defining the set, alternative to an axiomatic inference system. Thus, by comparing the sets of validities we compare the respective logics in the traditional sense. As it turns out, all semantic variants of ATL* discussed here are different on the level of general properties they induce. The pattern is as follows: properties of perfect information strategies refine properties of imperfect information strategies,
11.4. ABILITIES UNDER INCOMPLETE INFORMATION

Figure 11.4: Comparison of the sets of validities induced by various semantics of (a) ATL\textsuperscript{\textdagger}, and (b) ATL. Arrows depict strict subsumption of the sets of validities, e.g., “ATL\textsubscript{iR} \rightarrow ATL\textsuperscript{\textdagger}ATL\textsubscript{iR}” means that Validities(Atl\textsubscript{iR}) \subseteq Validities(Atl\textsuperscript{\textdagger}ATL\textsubscript{iR}). Dotted lines connect semantic variants with incomparable sets of validities. We do not include links that follow from transitivity of the subsumption relation. The subscript \textit{i} refers to “subjective” ability under imperfect information, that is, the standard “…” semantics presented in Section 11.4.2. Subscript \textit{i}o refers to the “objective” ability under imperfect information, where we only look at the paths starting from the actual initial state of the game (see the remark at the end of Section 11.4.2).

Thus, assuming imperfect information changes the set of properties that are universally true, i.e., ones that hold in every model and every state. Moreover, the change is essential in the sense that some fundamental validities of standard ATL (with perfect information) do not hold anymore under imperfect information. We give three examples of such formulae here:

\begin{align}
\langle A \rangle G \varphi \leftrightarrow \varphi \land \langle A \rangle X \langle A \rangle G \varphi \\
\langle A \rangle \varphi_1 U \varphi_2 \leftrightarrow \varphi_2 \lor \varphi_1 \land \langle A \rangle X \langle A \rangle \varphi_1 U \varphi_2 \\
\langle A_g \rangle F \varphi \leftrightarrow \neg \langle 0 \rangle G \neg \varphi
\end{align}

(11.3) (11.4) (11.5)

Formulae (11.3) and (11.4) provide fixpoint characterizations of strategic-temporal modalities. Formula (11.5) addresses the duality between necessary and obtainable outcomes in a game. All the three sentences are validities of ATL\textsubscript{iR} and ATL\textsubscript{iR}, but they are \textit{not} valid in ATL\textsubscript{iR} and ATL\textsubscript{iR}.\textsuperscript{4}

\textsuperscript{4} Interestingly, (11.5) becomes valid again with the “objective” interpretation of ability under imperfect information.
Ir \hspace{0.2cm} P-complete \hspace{0.2cm} P-complete \hspace{0.2cm} PSPACE-complete
IR \hspace{0.2cm} P-complete \hspace{0.2cm} P-complete \hspace{0.2cm} 2EXPTIME-complete
\nu \hspace{0.2cm} P-complete \hspace{0.2cm} \Delta^P_2\text{-complete} \hspace{0.2cm} PSPACE-complete
iR \hspace{0.2cm} P-complete \hspace{0.2cm} undecidable \hspace{0.2cm} undecidable

Figure 11.5: Overview of model checking complexity for variants of strategic logics. Each cell represents the logic over the set of formulae given in the column using the semantics given in the row. $\Delta^P_2 = NP^{NP}$ is the class of problems solvable by a deterministic Turing machine making adaptive calls to an oracle of a problem in $NP$.

**Expressivity**

Assuming imperfect information in the semantics of ATL-related logics changes also their relative expressivity. It is well known that, under the perfect information semantics: (i) ATL$^*$ is strictly less expressive than the alternating $\mu$-calculus, and (ii) ATL is strictly less expressive than alternation-free alternating $\mu$-calculus. Alternating $\mu$-calculus (AMC) is the logic obtained by adding the least fixpoint operator to Coalition Logic. Thus, characterization of long-term abilities under perfect information can be reduced to one-step abilities and fixpoint reasoning.

In contrast, when imperfect information semantics is used, the languages of ATL and alternation-free alternating $\mu$-calculus become incomparable with respect to their expressive power. In particular, the simple ATL formula $\langle A \rangle Fp$ cannot be equivalently translated into alternation-free AMC. Thus, long-term abilities cannot be equivalently expressed with one-step abilities and fixpoints anymore. This has an interesting consequence. Recall Moore’s notion of ability that we presented in Section 11.2. It is clearly based on fixpoint reasoning. Thus, under imperfect information, alternating $\mu$-calculus seems a better logic than ATL to formalize Moore’s concept of ability.

**Computational complexity**

Limited information of players in a game has also impact on the complexity of “solving” the game. In logical terms, this means that the complexity of model checking for a strategic logic is usually worse when imperfect information is involved. That is because the fixpoint characterizations of strategic-temporal modalities do not hold anymore under imperfect information, and hence model checking cannot be done with the standard fixpoint algorithm.
In essence, synthesizing strategies for imperfect information games cannot be done incrementally. Figure 11.5 summarizes how adding epistemic relations to the semantics of strategic operators increases the complexity of model checking for CL, ATL, and ATL*. Interestingly, it turns out that for the restricted language of ATL the type of information has a bigger impact on the complexity than the type of recall. For the broader language of ATL*, it is the other way around. The table indicates also some really bad news: for imperfect information and perfect recall, model checking becomes undecidable for any logic that can express long-term abilities.

Few results are known for the other decision problems. Satisfiability checking (and hence also validity checking) under perfect information is known to be PSPACE-complete for Coalition Logic, and 2EXPTIME-complete for ATL. To the best of our knowledge, there are no complexity results for satisfiability and validity checking of imperfect information variants of ATL. It is not even known if the problems are decidable.

11.5 Reasoning about Strategies and Knowledge

So far, we have taken into account agents’ knowledge only semantically, by introducing epistemic indistinguishability relations in the models to capture agents’ uncertainty. However, we did not refer to knowledge in the object language through epistemic operators. In this section, we add knowledge operators to the language of strategic logic and show how interplay between knowledge and abilities can be specified.

11.5.1 Towards an Epistemic Extension of ATL

A natural extension of CL and ATL is to add epistemic operators to the languages, in order to formalise reasoning about the interaction of knowledge and abilities of agents and coalitions. A straightforward combination of ATL and the multi-agent epistemic logic MAEL, called Alternating-time Temporal Epistemic Logic (ATEL). ATEL enables specification of various modes and nuances of interaction between knowledge and strategic abilities, e.g.:

- $\langle A\rangle \varphi \rightarrow E_A \langle A\rangle \varphi$ (if group $A$ can bring about $\varphi$ then everybody in $A$ knows that they can);
- $E_A \langle A\rangle \varphi \land \neg C_A \langle A\rangle \varphi$ (the agents in $A$ have mutual knowledge but not common knowledge that they can enforce $\varphi$);
- $\langle i\rangle \varphi \rightarrow K_i \neg \langle \text{agt} \setminus \{i\}\rangle \neg \varphi$ (if agent $i$ can bring about $\varphi$ then she knows that the rest of agents cannot prevent it).
The combined language can be interpreted in *concurrent epistemic game models* (CEGM)

\[ S = (\text{Ag, St, } \{\sim_a | a \in \text{Ag}\}, \text{Act, act, out}), \]

combining the CGM-based models for ATL and the multi-agent epistemic models. That is, we interpret ATEL in the same structures as those used in the variants of ATL for incomplete information, presented in Section 11.4.2. The semantics of ATEL is given by union of the semantic clauses for epistemic logic and those for ATL. This straightforward combination gives a logic that is an independent fusion of the two parts. For example, the next-time fragment of ATL (i.e., coalition logic) combined with epistemic logic is completely axiomatised simply by adding the standard S5 axioms for knowledge to the axiomatisation of coalition logic described in Section 11.3.3. If common and/or distributed knowledge operators are included, adding standard axioms for group knowledge again results in a complete axiomatisation. Analogous axiomatisations can be obtained for the full fusion of ATL with MAEL, i.e. ATEL.

The independent product of ATL and epistemic logic is useful and allows capturing meaningful properties such as the ones mentioned above. Some interesting properties, however, turn out to be difficult to express, and some formulae turn out to have counterintuitive meaning. This is particularly true for the interplay between epistemic properties and long-term abilities of agents and coalitions, expressible in the full language of ATEL.

Importantly, one would expect that an agent’s ability to achieve property \( \varphi \) should imply that the agent has enough control and knowledge to *identify and execute* a strategy that enforces \( \varphi \). A number of more sophisticated approaches to combining strategic and epistemic operators have been proposed in order to be able to capture such issues. Most of the solutions agree that only uniform strategies are really executable, cf. our exposition of ATL variants for incomplete information in Section 11.4.2. However, in order to identify a successful strategy, the agents must consider not only the courses of action, starting from the current state of the system, but also from states that are indistinguishable from the current one. There are many cases here, especially when group epistemics is concerned: the agents may have common, ordinary, or distributed knowledge about a strategy being successful, or they may be hinted the right strategy by a distinguished member (the “leader”), a subgroup (“headquarters committee”) or even another group of agents (“consulting company”). We discuss the most interesting cases in the subsequent paragraphs.
11.5. REASONING ABOUT STRATEGIES AND KNOWLEDGE

11.5.2 Epistemic Levels of Strategic Ability

There are several possible interpretations of A’s ability to bring about property $\gamma$, formalized by formula $\langle A \rangle \gamma$, under imperfect information:

1. There exists a specification of A’s behavior $\sigma_A$ (not necessarily executable!) such that, for every execution of $\sigma_A$, $\gamma$ holds.

2. There is a uniform strategy $s_A$ such that, for every execution of $s_A$, $\gamma$ holds (*A have objective ability to enforce $\gamma$*).

3. A know (in one sense or another, see below) that there is a uniform $s_A$ such that, for every execution of $s_A$, $\gamma$ holds (*A have a strategy “de dicto” to enforce $\gamma$*).

4. There is a uniform $s_A$ such that A know that, for every execution of $s_A$, $\gamma$ holds (*A have a strategy “de re” to enforce $\gamma$*).

The above interpretations form a sequence of increasingly stronger levels of ability – each next one implies the previous ones. We observe that the case (1) above corresponds to formula $\langle A \rangle \gamma$ interpreted in the original semantics of ATL (or, alternatively, in ATEL). The other cases, however, are not expressible in straightforward combinations of ATL and epistemic logic such as ATEL, not even for the next-time fragment (i.e., Coalition Logic) and not even when there is only a single agent. We will show in Section 11.5.3 how they can be formally characterized with a suitable combination of strategic and epistemic modalities.

Note also that cases (2)-(4) come close to various philosophical notions of ability that we discussed in Sections 11.1 and 11.2. Cases (3) and (4) closely resemble Ryle’s distinction between “knowing that” and “knowing how”. In (3), A know that they can enforce $\gamma$. In (4), they know how to do it. Case (2) corresponds to the first level of ability according to McCarthy and Hayes (i.e., objective existence of a subprogram that achieves $\gamma$). Case (4) is analogous to the second level of McCarthy and Hayes (the subprogram exists, and there is enough information to realize it) as well as Moore’s formalization of ability. Finally, case (3) can be seen as Moore’s notion of ability with knowledge de re (of the right strategy to play) replaced with knowledge de dicto (that such a strategy exists).

Out of the four levels of ability, “knowing how to play” (4) is arguably most interesting. However, the statement “A know that every execution of $s_A$ satisfies $\gamma$” is precise only if A consists of a single agent $a$. Then, we take into account the paths starting from states indistinguishable from the current one according to $a$ (i.e., $\bigcup_{q' \in imgl(q,\sim a)}\text{out-plays}(q', s_a)$). In case of multiple agents, there are several different “modes” in which they can know the right strategy. That is, given strategy $s_A$, coalition $A$ can have:
• **Common knowledge** that $s_A$ enforces $\gamma$. This requires the least amount of additional communication when coordinating a joint strategy (agents from $A$ may agree upon a total order over their collective strategies at the beginning of the game and that they will always choose the maximal successful strategy with respect to this order).

• **Mutual knowledge** that $s_A$ enforces $\gamma$: everybody in $A$ knows that $s_A$ brings about $\gamma$.

• **Distributed knowledge** that $s_A$ enforces $\gamma$: if the agents share their knowledge at the current state, they can identify the strategy as successful.

• **“Leader”**: the strategy can be identified by an agent $a \in A$;

• **“Headquarters committee”**: $s_A$ can be identified by a subgroup $A' \subseteq A$.

• **“Consulting company”**: $s_A$ can be identified by another group $B$.

• Other variations are possible, too.

### 11.5.3 Expressing Epistemic Levels of Ability and Constructive Knowledge

The issue of expressing various knowledge-related levels of ability through a suitable combination of strategic and epistemic logics has attracted significant attention. Most extensions (or refinements) of ATL, proposed as solutions, cover only some of the possibilities, albeit in an elegant way. Others offer a more general treatment of the problem at the expense of an unnecessarily complex logical language. One of the main problems is how to capture the interplay between epistemic, strategic, and temporal aspects of play with the kind of quantifiers offered by ATL on hand, and epistemic logic on the other. For example, “$A$ have distributed knowledge about how to play to enforce $\gamma$” can be rephrased as “there is a strategy for $A$ such that, for every state that is indistinguishable from the current one for all agents in $A$, and for every path from that state, possibly resulting from execution of the strategy, $\gamma$ must hold on the path.” This, however, cannot be directly expressed in ATL which combines quantification over strategies and paths within a single operator $\langle A \rangle$. One way out is to use separate modal operators to quantify over strategies and paths, the way it is done e.g. in Strategic STIT and Strategy Logic. Indeed, “knowing how to play” in the simpler case of single-step games can be expressed using a combination of Chellas STIT and epistemic logic. One can speculate that a straightforward combination of epistemic operators with Strategic STIT (or Strategy Logic) should work
equally well for long-term abilities. However, such combinations have not been investigated yet.

Another solution is to define the success of a strategy from a set of states, instead of a single global state. This idea was used in Constructive Strategic Logic (CSL) which extends ATL with so called constructive knowledge operators. In CSL, each formula is interpreted in a set of states (rather than a single state). We write $M, Q \models \langle A \rangle \varphi$ to express the fact that $A$ must have a strategy which is successful for all “opening” states from $Q$. The new epistemic operators $K_i, E_A, C_A, D_A$ for “practical” or “constructive” knowledge yield the set of states for which a single evidence (i.e., a successful strategy) should be presented (instead of checking if the required property holds in each of the states separately, like standard epistemic operators do).

Formally, the semantics of CSL over concurrent epistemic game models is defined by the following clauses:

\[
M, Q \models p \text{ iff } p \in L(q) \text{ for every } q \in Q;
\]

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

\[
M, Q \models \varphi \land \psi \text{ iff } M, Q \models \varphi \text{ and } M, Q \models \psi;
\]

\[
M, Q \models \langle A \rangle \varphi \text{ iff } \text{there is a uniform strategy } s_A \text{ such that } M, \lambda \models \varphi \text{ for every } \lambda \in \bigcup_{q \in Q} \text{out}_{\text{plays}}(q, s_A);
\]

\[
M, Q \models K_i \varphi \text{ iff } M, \{q' \mid \exists q \in Q \; q \sim_i q'\} \models \varphi;
\]

\[
M, Q \models C_A \varphi \text{ iff } M, \{q' \mid \exists q \in Q \; q \sim_A^C q'\} \models \varphi;
\]

\[
M, Q \models E_A \varphi \text{ iff } M, \{q' \mid \exists q \in Q \; q \sim_A^E q'\} \models \varphi;
\]

\[
M, Q \models D_A \varphi \text{ iff } M, \{q' \mid \exists q \in Q \; q \sim_A^D q'\} \models \varphi.
\]

The semantic clauses for temporal operators are exactly as in ATL*. Additionally, we define that $M, q \models \varphi$ iff $M, \{q\} \models \varphi$.

A nice feature of CSL is that standard knowledge operators can be defined using constructive knowledge, see Section 11.5.4 for details. Thus, one can use formulae of CSL to express the following:

1. $\langle a \rangle \varphi$ expresses that agent $a$ has a uniform strategy to enforce $\varphi$ from the current state (but may not know about it);

2. $K_a \langle a \rangle \varphi$ refers to agent $a$ having a strategy “de dicto” to enforce $\varphi$ (i.e. knowing only that some successful uniform strategy is available);

3. $\mathbb{K}_a \langle a \rangle \varphi$ refers to agent $a$ having a strategy “de re” to enforce $\varphi$ (i.e. having a successful uniform strategy and knowing the strategy);
Again, this connects neatly to the fundamental work on ability in philosophy and AI that we discussed in Sections 11.1 and 11.2. It can be argued that standard knowledge operators \( K_a \) capture Ryle’s notion of know-that, whereas constructive knowledge operators \( \mathbb{K}_a \) refer to the notion of know-how. Also, formalisations (1) and (3) above roughly correspond to McCarthy and Hayes’s levels (1) and (3) of program ability. Finally, formulae \( \mathbb{K}_a \langle a \rangle \phi \) and \( K_a \langle a \rangle \phi \) capture formally Moore’s distinction between ability de re and de dicto for long-term strategies. This extends naturally to abilities of coalitions, with \( C_A \langle A \rangle \phi, E_A \langle A \rangle \phi, D_A \langle A \rangle \phi \) formalizing common, mutual, and distributed knowledge how to play, \( \mathbb{K}_a \langle A \rangle \phi \) capturing the “leader” scenario, and so on (and similarly for different levels of knowledge “de dicto”). We conclude the topic with the following example.

**Example 11.10 (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 (\textit{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 \( c_i \) \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.\(^5\)

Formally, the concurrent epistemic game model \( \text{Attack}^n_k \) is constructed as follows:

- \( \text{Ag} = \{1, \ldots, k\} \):
- \( \text{St} = Q \cup S \), where states in \( Q = \{(c_1, \ldots, c_k) \in \{0,1\}^k\} \) identify possible codes for the (closed) safe, and states in \( S = \{\text{open, alarm}\} \) represent the situations when the safe has been opened, or when the alarm has been activated;
- \( \text{PROP} = \{\text{open}\}; \quad L(\text{open}) = \{\text{open}\}; \quad \text{Act} = \{0, 1, \text{nop}\}; \)
- \( \text{act}(i, q) = \{0, 1, \text{nop}\} \) for \( q \in Q \), and \( \text{act}(i, q) = \{\text{nop}\} \) for \( q \in S \);

\(^5\) For the more seriously minded readers, we observe that the story is just a variant of coordinated attack.
11.5. REASONING ABOUT STRATEGIES AND KNOWLEDGE

- For all \( x \in S^t \): \( \text{out}(x, \text{nop}, \ldots, \text{nop}) = x \). For \( q \in Q \), and at least one \( \alpha_i \neq \text{nop} \): \( \text{out}(q, \alpha_1, \ldots, \alpha_k) = \text{open} \) if \( \alpha_j = c_j \) for at least \( n \) agents \( j \) and \( \alpha_i \notin \{c_i, \text{nop}\} \) for no \( i \); else, \( \text{out}(q, \alpha_1, \ldots, \alpha_k) = \text{alarm} \).

- \( q \sim_i q' \) iff \( q[i - 1] \oplus q'[i] = q'[i - 1] \oplus q[i] \oplus q[i + 1] = q'[i] \oplus q'[i + 1] \).

The following CSL formulae hold in every state \( q \in Q \) of model \( \text{Attack}^n_k \), assuming that \( k \geq 3 \):

- \( \langle \text{Ag} \rangle \text{Fopen} \land \neg \exists_{\text{Ag}} \langle \text{Ag} \rangle \text{Fopen} \): 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);

- \( \mathbb{D}_{\text{Ag}} \langle \text{Ag} \rangle \text{Fopen} \): if the agents share information they can recognize who should send what;

- \( \mathbb{D}_{\{1, \ldots, n-1\}} \langle \text{Ag} \rangle \text{Fopen} \): 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 \( \{k - n + 2, \ldots, k\} \);

- Still, \( \neg \mathbb{D}_{\{1, \ldots, n-1\}} \langle 1, \ldots, n-1 \rangle \text{Fopen} \): 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 \( \mathbb{D}_{\text{Ag}} \langle \text{Ag} \rangle \text{Fopen} \) says that if the agents in \( \text{Ag} \) 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 \( \varphi \); what might be missing is means of exploiting the potential (e.g., by communication). In the same way, \( D_{A} \langle \langle A \rangle \rangle \text{F}\varphi \) says that the epistemic potential for \( A \) to determine the right strategy for \( \text{F}\varphi \) is there, too. So, it might be profitable to design efficient communication mechanisms to make the most of it.

11.5.4 Closer Look at Constructive Knowledge

In order to “constructively know” that \( \varphi \), agents \( A \) must be able to find (or “construct”) a mathematical object that supports \( \varphi \). This is relevant when \( \varphi \equiv \langle B \rangle \psi \); in that case, the mathematical object in question is a strategy for \( B \) which guarantees achieving \( \psi \). The semantic role of constructive
knowledge operators is to produce sets of states that will appear on the left hand side of the satisfaction relation. In a way, these modalities “aggregate” states into sets, and sets into bigger sets.

Note that in CSL we can use two different notions of validity. We say that a formula is weakly valid (or simply valid) if it is satisfied individually by each state in every model, i.e., if $M, q \models \varphi$ for all models $M$ and states $q$ in $\mathcal{M}$. It is strongly valid if it is satisfied by all non-empty sets in all models; i.e., if for each $M$ and every non-empty set of states $Q$ it is the case that $M, Q \models \varphi$. We are ultimately interested in the former. The importance of strong validity, however, lies in the fact that strong validity of $\varphi$ makes $\varphi$ and $\psi$ completely interchangeable. That is, if $\varphi_1 \iff \varphi_2$ is strongly valid, and $\varphi'$ is obtained from $\psi$ through replacing an occurrence of $\varphi_1$ by $\varphi_2$, then $M, Q \models \psi$ if $M, Q \models \varphi'$. It is not difficult to see that the same is not true for weak validity.

Clearly, strong validity implies validity, but not vice versa.

Defining Standard Knowledge from Constructive Knowledge

In the semantics of CSL, formulae are interpreted in sets of states; in order for $\varphi$ to hold in $M, Q$, the formula must be “globally” satisfied in all states from $Q$ at once (i.e., with a single strategy). Notice, however, that $M, Q \models \langle \emptyset \rangle \psi \cup \varphi$ iff $M, q \models \varphi$ for every $q \in Q$. This can be used as a technical trick to evaluate $\varphi$ “locally” (i.e., in every state of $Q$ separately). In particular, we can use it to define standard knowledge from constructive knowledge as:

$$K_a \varphi \equiv \langle K_a \langle \emptyset \rangle \varphi \cup \psi, \text{ and analogously for group knowledge operators. It is not difficult to see that } M, q \models \langle K_a \langle \emptyset \rangle \varphi \cup \psi \text{ iff } M, q' \models \varphi \text{ for every } q' \text{ such that } q \sim_a q'. \text{ More generally, the following formula of CSL is strongly valid: } K_a \varphi \iff \langle \hat{K}_a \langle \emptyset \rangle \varphi \cup \psi, \text{ where } \hat{K} = C, E, D \text{ and } \hat{K}_a = C, E, D. \text{ In consequence, we obtain that standard knowledge can be seen as a special case of constructive knowledge.}

Properties of Constructive Knowledge

We believe that operators $C_A, E_A, D_A$ and $K_a$ do capture a special kind of knowledge of agents. An interesting question is: do these notions of knowledge have the properties usually associated with knowledge? In particular, do postulates $K, D, T, 4, 5$ of epistemic logic hold for constructive knowledge? Below, we list the constructive knowledge versions of the S5
11.5. REASONING ABOUT STRATEGIES AND KNOWLEDGE

axioms of individual knowledge. “Yes” means that the schema is strongly valid; “No” means that it is not even weakly valid (keep in mind that strong validity implies validity). Incidentally, none of the properties turns out to be weakly but not strongly valid.

Thus, in general, the answer is no; particularly, the truth axiom does not hold. Note, however, that the axiom does hold in the restricted case when constructive knowledge is applied to positive strategic formulae, i.e., $K_a (\varphi \rightarrow \psi) \rightarrow (K_a \varphi \rightarrow K_a \psi)$ is strongly valid in CSL. Moreover, the above results show that $K_a$ satisfies the standard postulates for beliefs: logical closure, consistency, and positive as well as negative introspection. This suggests that “knowing how to play” sits somewhere in between the realms of doxastic and epistemic logic: it is stronger than belief but not quite as strong as standard knowledge.

11.5.5 Public Announcements

The combination of ATL modalities and epistemic modalities can be seen as extending ATL to take imperfect and incomplete information into account, but it can also be seen as a way of making epistemic logic dynamic: to allow us to reason about how knowledge changes. The “dynamic epistemic logics”, dealing with how different types of actions and other events change knowledge, were discussed in Chapter 6. A difference between these logics and epistemic variants of ATL is, in addition to the important difference that the ATL language uses strategic/coalitional modalities, that the latter use a more abstract model of actions. But what if we restrict the actions considered in ATL to be one of the specific types of actions considered in dynamic epistemic logic? Here we will look at the case where actions are restricted to be public announcements.

For simplicity, we will only consider the language of Coalition logic CL (i.e., the next-step fragment of the ATL language), and we will consider a $(G)$ modality with a simpler semantics: we let $(G)\varphi$ mean that “there exists a joint public announcement by group $G$, such that $\varphi$ will be true”. The discussion that follows extend directly also to the cases where we allow the full ATL language, as well as the combined exists-forall interpretation “there exists a joint public announcement by group $G$ such that no matter what the other agents announce, $\varphi$ will be true”.
In the case that actions are public announcements we can interpret the language simply in a standard epistemic model, which indeed models both which actions (public announcements) are available to an agent, and what the consequences of a group making a joint action (joint public announcement) are. Regarding the former, an agent can truthfully announce something she knows. Regarding the latter, the consequence of an announcement by a single agent \( i \) is the model update by a formula of the form \( K_i \varphi \), assuming it is common knowledge that public announcements are truthful. The consequence of a joint announcement of a group \( G \) will then be a model update with a formula of the form \( V_i \cap \bigwedge_{i \in G} K_i \varphi_i \).

Formally, let \( M = (St, \sim_1, \ldots, \sim_n, V) \) be an epistemic model, where \( St \) is a set of states, \( \sim_i \subseteq St \times St \) is an epistemic indistinguishability relation and is assumed to be an equivalence relation for each agent \( i \in N \), and \( V : \Theta \to 2^{St} \) assigns primitive propositions to the states in which they are true. When \( q \) is a state in \( M \), we let:

\[
M, q \models (G) \varphi \iff \exists \{ \psi_i : i \in G \} \subseteq L_{el} \text{ such that } M, q \models \bigwedge_{i \in G} K_i \psi_i \text{ and } M|_{\bigwedge_{i \in G} K_i \psi_i, q} \models \varphi
\]

where \( L_{el} \) is the standard epistemic language and \( M|\psi = (St', \sim'_1, \ldots, \sim'_n, V') \) is the update of \( M \) by \( \psi \) by \( \psi' = \{ q' \in St : M, q' \models \psi \} \) and \( \sim'_i = \sim_i \cap (St' \times St') \); \( V'(p) = V(p) \cap St' \).

The consequences of restricting actions to truthful public announcements are (perhaps surprisingly) significant. This is due to the fact that now there is a very intimate relationship between knowledge and ability, since action is knowledge (the set of available actions is the set of known formulas). Let us revisit one of the key relationships between knowledge and ability we have considered in this chapter, in the case that actions are public announcements: the \textit{de dicto/de re} distinction.

It is easy to see that

\[
K_i(i) \varphi
\]

expresses the fact that agent \( i \) knows \textit{de dicto} that she can enforce \( \varphi \), i.e., that \( M, q \models K_i(i) \varphi \) iff for any \( t \) such that \( q \sim_i t \), there exists a \( \psi \in L_{el} \) such that \( M, t \models K_i \psi \) and \( M|K_i \psi, t \models \varphi \).

Consider now the formula

\[
(i) K_i \varphi
\]

obtained by swapping the two modalities. Intuitively, this formula says that agent \( i \) can perform some action such that she will know \( \varphi \) afterwards. Note that in coalition logic (or ATL), the corresponding formula does \textit{not} express the fact that agent \( i \) knows \textit{de re} that she can enforce \( \varphi \) (this is true also for
the single agent case, when the semantics of the $\langle i \rangle$ operator of coalition logic is “there exists an action by $i$ such that $\varphi$ will be true, like the $(i)$ operator introduced here). In fact, that claim is not expressible in ATL at all, as discussed before. However, it turns out that it is in fact expressed by (11.7) in the special case of public announcements.

Proposition 11.1
\[ M, q \models \langle i \rangle K_i \varphi \iff \text{there exists a } \psi \in L_{el} \text{ such that for any } t \text{ such that } q \sim_i t, \]
\[ M, t \models K_i \psi \text{ and } M[K_i \psi, t] \models \varphi. \]

Again, this fact is due to the intimate relationship between knowledge and ability in the case that actions are public announcements, and does not hold in general.

11.6 Concluding remarks

The development of a logical theory that adequately captures the subtle interactions between knowledge and ability has been a long-standing goal in philosophy and, more recently, artificial intelligence. While it seems relatively straightforward to develop a theory that captures some meaningful notion of ability, it seems surprisingly hard to develop a theory that explains the interaction between knowledge and ability. In this article, we have surveyed work on this problem in philosophy, artificial intelligence, and mainstream computer science.

11.7 Notes

The seminal work of Robert Moore on the relationship between knowledge and ability can be found in (Moore, 1977, 1990). Much of the discussion in Section 11.1 is inspired by (and modelled on) Stanley and Williamson (2001). Ryle’s argument that a reduction from know-how to know-that is not possible is found in (Ryle, 1949). Some other early philosophical approaches to logics of agency and ability are due to Chellas (1969), Brown (1988), and Belnap and Perloff (1988). See (Belnap, Perloff, and Xu, 2001, p.36) for comments on the semantics of Belnap and Perloff’s operator. For a simpler variant of STIT, see, e.g., the work of Broersen, Herzig, and Troquard (2006). Troquard (2007) surveys STIT axioms and logics. We refer the reader to this contribution of Troquard (2007) for a discussion of STIT logics in artificial intelligence, and to a paper by Belnap et al. (2001) for a detailed review from the perspective of philosophical logic. Broersen et al. (2006) show how STIT logics can be used to encode the logic ATL.
CHAPTER 11. KNOWLEDGE AND ABILITY


Alternating-time Temporal Logic was introduced by Alur, Henzinger, and Kupferman (1997, 2002), and Coalition Logic by Pauly (2001, 2002). See the book of Chellas (1980) for an introduction to neighbourhood semantics. Soundness and completeness of the ATL axioms and inference rules shown in Section 11.3.4 was proved by Goranko and van Drimmelen (2006). EXPTIME-completeness of ATL was shown by van Drimmelen (2003) and by Goranko and van Drimmelen (2006) using alternation tree automata, and a practically efficient tableau-based decision procedure for it was developed by Goranko and Shkatov (2009). The whole ATL\(^\ast\) was proved decidable and 2EXPTIME-complete by Schewe (2008) using alternating tree automata.

Different ways of incorporating agents’ incomplete information and use of memory into ATL\(^\ast\) have been studied by van der Hoek and Wooldridge (2003), Alur et al. (2002), Schobbens (2004), and by Jamroga and Ágotnes (2007). The four variants of the semantics of ATL\(^\ast\) with incomplete information and the i/I/r/R notation is due to Schobbens (2004). The classification of ability under incomplete information as objective and subjective is due to Jamroga and Bulling (2011). The alternative approach to modeling executable strategies, mentioned at the end of Section 11.4.2, is studied by Guelev, Dima, and Enea (2011) and by Diaconu and Dima (2012).

See the work of Bulling and Jamroga (2013, 2011) and of Bulling, Dix, and Jamroga (2010) for further details on the impact of knowledge on strategic ability. Readers interested in modal \(\mu\)-calculus and fixpoint operators can consult, e.g., the work of Bradfield and Stirling (2006). An introduction to alternation-free \(\mu\)-calculus is given by Alur et al. (2002), while Bulling et al. (2010) provide more details on how adding epistemic relations to the semantics of strategic operators increases the complexity of model checking.
for CL, ATL, and ATL*. ATEL was first studied by van der Hoek and Wooldridge (2002, 2003). Completeness of coalition logic extended with epistemic axioms was shown by Ågotnes and Alechina (2012). The different possible interpretations of “bringing about” under imperfect information were discussed by Jamroga and van der Hoek (2004). Variants of ATL that cover only some of the possible ways strategic and epistemic operators can interact include approaches by Schobbens (2004), van Otterloo and Jonker (2004), and by Ågotnes (2006), while a more general treatment is proposed by Jamroga and van der Hoek (2004). Strategic STIT is studied by Hory (2001), and by Broersen et al. (2006), and Strategy Logic by Mogavero, Murano, and Vardi (2010). The paper by Herzig and Troquard (2006) shows how “knowing how to play” in the simpler case of single-step games can be expressed using a combination of Chellas STIT and epistemic logic. Constructive Strategic Logic (CSL) was introduced and studied by Jamroga and Ågotnes (2007).

Interpretations of coalition operators where actions are restricted to public announcements, leading, e.g., to Group announcement logic, are further studied by Ågotnes and van Ditmarsch (2008) and by Ågotnes, Balbiani, van Ditmarsch, and Seban (2010).

**Acknowledgements** The authors would like to thank Jan Broersen, who provided extensive comments on earlier drafts of this chapter, which helped to improve it considerably. Wojciech Jamroga acknowledges the support of the National Research Fund (FNR) Luxembourg under the project GALOT (INTER/DFG/12/06), as well as the support of the 7th Framework Programme of the European Union under the Marie Curie IEF project ReVINK (PIEF-GA-2012-626398). Michael Wooldridge would like to acknowledge the support of the ERC under Advanced Grant 291528 ("RACE").
References


REFERENCES


CHAPTER 11. KNOWLEDGE AND ABILITY


