Knowledge and games : theory and implementation
Witzel, S.A.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)

Download date: 27 Dec 2018
Introduction

Does she know what he knows? And if so, what is she going to do?
This dissertation takes a computer science perspective on questions of knowledge and interaction and presents approaches for endowing artificial agents with corresponding reasoning capabilities.

Background

Epistemic logic

Epistemic logic is the formal study of reasoning about knowledge, including knowledge about knowledge (so-called higher-order knowledge). The modern field of epistemic logic has been initiated by von Wright [158] and Hintikka [78]. More recent treatments include work by Fagin et al. [60] and Meyer and van der Hoek [100].

The fundamental idea for a formal model of agents’ knowledge is to consider a set of possible worlds, or states in which the actual world may be, together with indistinguishability relations between them, one for each agent. In each possible world, certain atomic statements, or propositions, hold. Intuitively, an agent knows whatever holds in all worlds which he cannot distinguish from (or which he considers possible in) the actual world. In order to properly reflect certain properties that the philosophical notion of knowledge is thought to possess, the indistinguishability relations are usually required to be equivalence relations.

This relational semantics is often called Kripke semantics, and the involved structures Kripke structures, due to the pioneering work in modal logic by Kripke [87].

For clarification, consider the following example.¹ Assume that for some reason, Ann knows Bob’s bank PIN code. It then depends on her higher-order knowledge

¹This example is taken from [17]. For simplicity, issues like possession of the physical bank pass or effects of involving additional agents are abstracted away from.
whether she can safely empty his bank account: she can do this only if she knows that he doesn’t know that she knows the code—otherwise he would immediately suspect her. Such a “safe” situation is modeled in Figure 1. In the actual world on the left, Bob’s PIN is 1234. Bob considers another world possible, where his PIN is still 1234 (he knows his PIN), but where Ann considers a world possible where his PIN is not 1234. In the actual world, however, Ann only considers the actual world possible. That is, Ann knows Bob’s PIN, Bob doesn’t know that she knows it, and Ann in turn knows that he doesn’t know that she knows it.

![Figure 1: Model of a situation where it is safe for Ann (A) to empty Bob’s (B) bank account.](image)

In order to talk about such models formally, an epistemic language is used, consisting of formulas with a clearly defined syntax. In the simplest case, the language consists of letters for the atomic statements, certain logical connectives to combine statements to more complex ones, and a knowledge operator for each agent. For example, if we use $p$ to denote that Bob’s PIN is 1234, the connectives $\land$ and $\neg$ to denote “and” and “not”, and $K_A$ and $K_B$ as knowledge operators for Ann and Bob, then the formula

$$K_A(p \land \neg K_B K_A p)$$

means that Ann knows Bob’s PIN and knows that he doesn’t know that she knows it. As we saw, this formula holds in the actual world in the model in Figure 1.

In general, one can reason, or deduce, in two fundamental ways. Semantically, one can say, for example, that a formula follows from another formula if the former holds in all models under consideration in which the latter holds. Syntactically, one can use certain axioms and inference rules to reach a desired conclusion. We are in this dissertation mostly concerned with semantic arguments.

Besides knowledge, one can also model the related notion of belief.\(^2\) The difference is that, philosophically speaking, the concept of knowledge implies correctness, while beliefs may be false. On the level of models, this is reflected by using more general accessibility relations, which need not be equivalence relations. For example, if we replace Bob’s indistinguishability between the actual and the middle world in Figure 1 by a one-way accessibility from the actual to the middle

\(^2\)In some contexts, one then speaks of doxastic logic, though in other contexts this term has a more narrow definition referring to frameworks formalizing how beliefs are revised.
world, we obtain the situation depicted in Figure 2 where Bob (mistakenly) believes that Ann does not know his PIN.

![Figure 2: Model of a situation where Bob (B) mistakenly believes that Ann (A) does not know his PIN.]

A concept of particular importance is **common knowledge** (or belief), which intuitively corresponds to the infinite iteration of **mutual knowledge** (or belief):

$$K_Bp, \ K_AK_Bp, \ K_BK_AK_Bp, \ K_AK_BK_AK_Bp, \ldots$$

and so on ad infinitum. We illustrate this limit case in the upcoming subsection.

**Distributed computing**

Distributed computing is the formal study of programs, or processes, running simultaneously on possibly different processing units. Typically, these processes can coordinate and communicate in some way, which necessitates a study of their joint behavior and the system they form as a whole. In particular, one can ask what processes can be said to “know” about each other, each other’s state, and each other’s knowledge. This is relevant, for example, for sharing resources or ensuring correct or secret transmission of information. Consequently, issues around knowledge in distributed systems have been studied using formalisms similar to the one described above, among others, by Parikh and Ramanujam [116], Chandy and Misra [39], and extensively by Fagin et al. [60].

Some intuitions about the concept of common knowledge can be drawn from a classic example in distributed computing, introduced by Akkoyunlu et al. [1] and Gray [70] and later independently considered by Rubinstein [129]. The example in its most well-known version features two generals who are positioned with their armies on two hilltops with no direct line of sight (and no mobile phones). They need to launch a **coordinated attack** on their enemy in the valley. So one general, let us call him A, sends a messenger to the other general, B, proposing a time for the attack. Unfortunately, A has no way of knowing whether and when his messenger will arrive to deliver the message. Since B is aware of this fact, upon receiving the proposal he sends the messenger back to confirm. So far so good, but B has no way of knowing whether and when his confirmation will be received. And as long as it has not been received, A will not know that B received the
original proposal, and thus \( A \) will not attack. Since \( A \) is aware of this fact, he sends the messenger back to confirm... and before they know it, the generals are trapped in an infinite loop.

It can be shown that what the generals need is common knowledge of the original message, no finite level of mutual knowledge will suffice for their purposes. Without compromises, and without sufficient time for exchanging infinitely many confirmations, common knowledge can only be attained by synchronous communication such as provided by a direct line of sight (or mobile phones).

**Game theory**

Game theory, a framework for the formal study of strategic interaction, was initiated by von Neumann and Morgenstern \([104]\). Many modern textbooks exist, e.g., Osborne \([106]\). Issues of (common) knowledge and belief are a subject of continuing interest in game theory, starting with work by Harsanyi \([76]\) and Aumann \([12]\). The interface of (epistemic) logic and game theory is currently being actively studied, for example by van Benthem \([18]\).

The basic idea of non-cooperative game theory is that agents, also called *players*, act and interact without any institutions that would allow them to form and enforce binding agreements for cooperation. At the heart of such a player lies a *payoff function*, which models the payoff, or *utility*, which the player derives from any given interaction. An interaction in this framework is taken to be a tuple of simultaneous *actions*, or *strategies*, one for each player. A payoff function then maps such *strategy profiles*, or *joint strategies*, to numbers representing the utility.\(^4\) A rational player is assumed to aim at maximizing his utility.

For example, assume that Ann loves Indian food and isn’t too fond of Mexican food, and that Bob likes Ann\(^5\) and really wants to eat in the same restaurant as she does. So consider the actions of going to an Indian restaurant or going to a Mexican restaurant. The according payoff functions can be specified using a *payoff matrix* as in Figure 3.

Clearly, such payoff structures induce (or reflect) *preferences*: Ann (unconditionally) prefers the Indian over the Mexican restaurant, while Bob prefers Indian over Mexican if Ann chooses Indian, and Mexican over Indian if she chooses Mexican. If Bob does end up in a different restaurant than Ann, though, then he would rather eat Mexican food than Indian.

\(^3\)There are some subtleties regarding just how synchronous communication can actually be, especially when mediated by communication devices. Monderer and Samet \([101]\) show how common belief can be seen to approximate common knowledge as uncertainty decreases. See Chapter 1, Section 1.5, for some discussion in our context.

\(^4\)We are not concerned with so-called *extensive-form* games, which consist of multiple actions taken in turns. Suffice it to say that they can be represented in the *normal form* we describe here.

\(^5\)He hasn’t found out about his bank account yet.
In this example, going to the Mexican restaurant is strictly dominated for Ann: no matter what Bob does, she always gets a strictly higher payoff from choosing the Indian restaurant. The usual definition of a rational player implies that such an action will not be chosen—it can be eliminated, leaving us with the game in Figure 4.

It turns out that in this reduced game, Mexican is dominated by Indian for Bob. Eliminating this strategy in turn, we end up with a trivial game where both Ann and Bob are left with the single choice of going to the Indian restaurant. This process, first considered by Dekel and Fudenberg [46], is known as iterated elimination of strictly dominated strategies (IESDS).

So far we have taken an omniscient point of view and simply manipulated the fully specified payoff matrix. If we put ourselves in a player’s shoes, however, who may have incomplete information about other players’ preferences and knowledge, then things get more involved. In our example, for Bob to safely perform the last elimination, he needs to know that Ann eliminated the Mexican restaurant choice—that is, he needs to know that Ann is rational and strictly prefers the Indian restaurant. If we introduce a third player, Carol, who likes Bob and wants to prevent him from being alone with Ann, then for her restaurant choice Carol needs to know whether Bob knows what Ann prefers. The whole elimination process then needs to be formulated in terms of knowledge, and epistemic logic lends itself to that end.

If the players obtain their knowledge through some sort of communication, then matters are somewhat simpler with synchronous communication, which, as we saw above, creates common knowledge, superseding intermediate levels of mutual knowledge and removing the need for confirmations. Such a setting is the

---

6Of course Ann could eliminate the choice for other reasons than rationality and dominance, but let us stick with this explanation here.
motivation for the main part of this dissertation, and we give some more intuition and justification in the appropriate places.

Outside the main part, we deal with two other topics from game theory: firstly **combinatorial auctions**, and secondly **coalition formation**.

Combinatorial auctions are auctions of multiple goods where the bidders can bid on *combinations* of these goods. As in conventional single-item auctions, the bidders transmit their bids to the auctioneer, who determines the winner of the auction. However, in contrast to single-item auctions, in a combinatorial setting bid representation and winner determination become rich and complex issues; see, e.g., the book compiled by Cramton et al. [43] for a recent overview of these and other topics. We consider a certain kind of combinatorial auction where services transforming goods are offered, rather than conventional atomic goods. These services then may be scheduled in such a way that the output of one transformation can be used as input for another transformation. We examine how preferences over such orderings of the offered transformations can be taken into account.

Coalition formation belongs to the field of **cooperative** game theory, which assumes that there is a way for players to form and enforce binding agreements. The **coalitions** resulting from such agreements constitute the focus of interest on two scales. From the individual player’s point of view, the questions concern what he can, or should, get out of joining a coalition, and what coalitions he prefers over others; and on a larger scale, the dynamics of coalitions are of interest, how they form and change, and when they might be considered stable. The recent textbook by Ray [123] provides an overview of these latter issues. We address them under an operational viewpoint of merging and splitting coalitions.

**Social networks**

In the research field of social networks, one studies *relationships* and *interdependencies* among individuals. This field has been highly active in recent years, see, e.g., the books by Jackson [83] and Goyal [69].

We are here especially interested in *communication networks*, that is, networks which determine the possibilities for communication among the individual agents. Sharing of information in social networks has been studied in probabilistic frameworks, e.g., by Chamley [38]. Within logic, the relevance of epistemic issues in communication networks has been recognized by a number of authors, e.g., by van Benthem [20] and Pacuit and Parikh [109].

To illustrate some of the issues and subtleties involved, we give here an informal instance of the framework we set up in the main part of this dissertation. We look at communication about *preferences* among *groups* of agents, and we are interested in the evolution of *knowledge* within such “group networks”, and in how the agents can use that knowledge in order to choose their *actions*.

We formalize the assumptions we make later on. For now consider Figure 5,
and remember that Bob likes Ann and Carol likes Bob, and how their choices depend on each other. Shyness and jealousy forbid them to speak to each other explicitly about coordination and preferences, so let us assume that communication is limited, for example only through observing behavior.

Figure 5 shows three possible group configurations of Ann, Bob and Carol. In (a), Bob shares a communication platform with Ann and another one with Carol. Through the former, he can learn about Ann’s preferences, and through the latter, Carol can learn about Bob’s preferences (e.g., through observations while pairwise eating at the same restaurant). However, Carol cannot learn about Ann’s preferences. In (b), Carol can learn about Ann’s preferences, but she cannot learn what Bob knows about Ann’s preferences (and she is too jealous to ask him directly). Only in (c), where they all share the same communication platform, they can all commonly learn about each other. In particular, Carol can then learn about Ann’s preferences, and also that Bob knows them. She will then be able to predict his choice of action, and choose her action accordingly.

Theory and Implementation

Epistemic logic and game theory have a strong focus on an external and descriptive point of view.

In epistemic logic, this is already reflected by the fact that there is one central model comprising all agents. It describes a modeler’s perspective on what the agents can be said to know in a philosophical sense, and what they indeed will be able to figure out if they are perfect reasoners; but it does not necessarily say exactly how the agents in actuality arrive at that knowledge.\footnote{There is a sizable but as yet somewhat inconclusive literature in computer science and game theory concerning modeling of bounded rationality and reasoning capabilities; see, e.g., \cite{59,130}. We discuss these and related issues at various places, notably in Chapter 3, Section 3.6.}

Similarly, many concepts in game theory describe what situations rational players will end up in, but not exactly how they might get there.
Without doubt, epistemic logic does provide elegant and general mechanisms to perform deductions and determine the truth value of statements. But those, again, feel more like a tool for the modeler, typically without any claim that they reflect what agents actually (can) do. So these mechanisms do not really try to take an agent’s point of view and can, without any qualms, be arbitrarily complex.

This kind of criticism is not new. For example, Parikh [114] aims to model more closely what and how we as humans actually know and believe. However, the mainstream research focus does not lie on such approaches, nor on concrete algorithmic realizations of epistemic reasoning, for example, within artificial agents.

Also in game theory, certain algorithmic characterizations exist, the procedure of IESDS explained above being one example. However, IESDS still takes a centralized perspective and operates on the fully specified payoff matrix. Tan and Werlang [144], Brandenburger [30] and Börgers [28] have characterized mutual beliefs about each other’s rationality and preferences that lead players to the outcome of IESDS, but so far it has not been examined just what players can do given some particular pieces of information, or how such information can be arrived at and processed.

In the main part of this dissertation, we want to take a procedural and subjective point of view: How might a player actually access his theoretically ascribed knowledge, and how might he obtain the game-theoretic solution?

To this end, we are interested in restricting the general theories and obtaining concrete implementations of simple fragments that can be proved correct with respect to the general theoretical foundations. The aim of this approach is to obtain practical implementations grounded in theory.

**Chapter overview**

Figure 6 shows how the chapters of this dissertation relate to its main concepts and to each other. The main part of the dissertation is formed by Chapters 1–3, which build upon each other and make a progression from knowledge and theory to games and implementation. The remaining satellite chapters have either been directly inspired by the main part (in the case of Chapter 4) or are related via the same main concepts (in the case of Chapters 5 and 6). In the following, we give more details on the contents of the chapters and the overall structure of the dissertation.

The basic idea of the main part (Chapters 1–3) is to view computer processes, or otherwise distributed programs, as players in a game-theoretic setting with incomplete information. As such, they should be able to communicate in order to obtain information, and to perform game-theoretic algorithms.

In particular, we focus on the IESDS algorithm described above, and a setting where each player initially knows only his own preferences. Players can communicate their preferences across a communication network, and as they
communicate, their knowledge changes and they can obtain more conclusive elimination outcomes.

In order to obtain an actual implementation, we need to enable players to actually compute their knowledge so that they can access what they theoretically can be said to know. In order to make these computations efficient, we are especially interested in a framework where knowledge is simple to maintain and process. We therefore impose clear restrictions on the allowed communication, and, for the reasons mentioned above, focus on synchronous communication.

We have illustrated the underlying approach, which we call explicit knowledge programming, in [155].

In Chapter 1 (Guarding common knowledge), we establish the technical foundations to support implementation of synchronous communication, and thus the attainment of common knowledge, among computer processes representing players. To this end, we examine dialects of a process calculus which is available in the form of programming languages.

We define a setting where processes are treated “on an equal footing”, in the sense that none of them takes a special role in initiating or coordinating communication. This is a prerequisite for interpreting processes as players in arbitrary games, since the differences among them should only depend on the strategies and preferences of a particular given game, and not be predetermined through some a priori roles.

We then show that a certain guard construct is needed in the language in order to implement correct programs in such a symmetric setting. Since

---

8Our term explicit knowledge programming is somewhat related to explicit knowledge and algorithmic knowledge [113, 60]; however, it is mostly meant to contrast with knowledge-based programs [61]. We give some more discussion on this in Chapter 3, Section 3.6.
this construct is not commonly provided, our result practically identifies a unique programming language suitable for our purposes.

*This chapter is an extended version of [154].*

In **Chapter 2** (Knowledge in interaction structures), we define what we call interaction structures, a concrete class of communication networks compatible with the findings from **Chapter 1**. We also specify what kind of communication scenario we focus on, consisting of possible initial situations and possible communication.

We then study properties of knowledge that results from such communication. In particular, we investigate what is the impact of common knowledge of the underlying interaction structure, and we establish that common knowledge distributes over disjunction for formulas without negation. These results can be used to simplify reasoning about knowledge in our setting.

*This chapter builds on parts of [9], joint work with Krzysztof R. Apt and Jonathan A. Zvesper.*

**Chapter 3** (Strategies in interaction structures) then turns towards the game-theoretic and implementation-oriented side.

We study games in the presence of an interaction structure, which allows players to communicate their preferences, assuming that each player initially only knows his own preferences. We study the outcomes of IESDS that can be obtained in any given state of communication.

The insights from **Chapter 2** are used in order to prove that the outcomes of IESDS which we establish indeed correctly reflect what the players know in any particular situation. Building upon **Chapter 1**, we then describe a distributed algorithm that implements IESDS locally in each player process.

*This chapter extends unpublished joint work with Krzysztof R. Apt and Jonathan A. Zvesper.*

This rounds off the main part of the dissertation and starts the more loosely related satellite chapters. The first of these has developed from the main part and is close to it in spirit, with the difference that it focuses on a centralized rather than a distributed approach, and that it considers computer games rather than games in the strict sense of game theory.

In **Chapter 4** (Epistemic reasoning in computer games), we argue that reasoning about knowledge, including about each other’s knowledge, plays a crucial role in real-life strategic and social interaction. We survey existing literature and games which simulate such interaction, and show that this issue is currently neglected.
We give concrete scenarios from existing computer games which could profit from incorporating such reasoning techniques and substantiate one of them by describing a simple implementation intended for experimental evaluation. Finally, we discuss a number of issues that arise when generalizing our approach, some of which go beyond the scope computer games and are of more general interest.

*This chapter extends joint work with Jonathan A. Zvesper and Ethan Kennerly, published as [156, 157].*

The last two satellite chapters return to game theory, and in particular to the areas of coalition formation and auctions.

**Chapter 5** (Coalition formation: A generic approach) proposes an abstract approach to coalition formation that focuses on simple merge and split rules transforming partitions of a group of players.

We identify conditions under which every iteration of these rules yields a unique partition. The main conceptual tool is a specific notion of a stable partition.

The results are parametrized by a preference relation between partitions of a group of players and naturally apply to coalitional TU-games, hedonic games and exchange economy games.

*This chapter is joint work with Krzysztof R. Apt, to appear as [8].*

In **Chapter 6** (Time constraints in mixed auctions), we extend the existing framework of mixed multi-unit combinatorial auctions to include time constraints, present an expressive bidding language, and show how to solve the winner determination problem for such auctions using an integer programming implementation.

Mixed multi-unit combinatorial auctions are auctions where bidders can offer combinations of transformations of goods rather than just simple goods. For example, a transformation might take dough and water and yield bread. This model has great potential for applications in the context of supply chain formation, which is further enhanced by the integration of time constraints.

We consider different kinds of time constraints: they may be based on either time points or intervals, they may determine a relative ordering of transformations, they may relate transformations to absolute time points, and they may constrain the duration of transformations.

*This chapter is based on unpublished joint work with Ulle Endriss.*

In **Chapter 7** we give an outlook on possible future directions for implementing epistemic logic.