Capítulo 1

An educational epistemic model checker for Philosophy students

Carlos Aguilera-Ventura
IRIT, Université Paul Sabatier, Toulouse

Antonio Yuste-Ginel
Universidad Complutense de Madrid

1.1 Introduction and Context

Presentation and a brief history of the tool. This chapter reports on an open-ended project whose main aim is implementing a model checker for teaching and learning epistemic logic. It was designed to have Philosophy undergraduate students as target users. The project was born as the Bacherlors dissertation of Carlos, it was supervised by Alfredo and Antonio, and it is called *Epistemic Model Checker* (EMC). The first public version was released on GitHub on July 2020. A second version, containing some substantial improvements was presented at the *4th Conference of Tools for Teaching Logic.*[1] Finally, we developed some of its latest features in secret as a surprise for Alfredo in the present volume.

Epistemic logic and its philosophical importance. Epistemic logic can be broadly understood as the logical modelling of *epistemic attitudes*. Roughly, epistemic attitudes are mental states that cognitive agents hold toward propositions in order for them to have a (more or less accurate) representation of the world. This representation allows agents to intelligently interact with the environment and with other agents, among other things. The typical focus of epistemic logic, inherited from mainstream epistemology, is on two such attitudes: *knowledge* and *belief*. Despite its applications to computer science (e.g., in symbolic approaches to AI, epistemic game theory, or cryptography (see [5, 13]), epistemic logic was born as a project of philosophical inquiry [7], and it still attracts the attention of an important part of the philosophical academic community. It can be understood as a branch of the broader field of Formal Epistemology [1], which is, in turn, a branch of Formal Philosophy [6]. Hence, it is not unusual to find learning content about epistemic logic in the syllabus of different undergraduate courses for future philosophers worldwide. However, just as it happens with any other formal topic, epistemic logic poses a number of practical problems when understood as learning content for humanity students, mainly caused by the lack of formal background and the absence of habit of mathematical thinking.

The importance of model checking problems. Model-checking problems are a crucial learning activity to understand the meaning of epistemic operators: they train students in the very fundamental concept of the truth of a formula with respect to a model. Although they

[1] `https://toolsforteachinglogic23.weebly.com/`

are decision problems, the process of learning them requires more detailed feedback than a simple validation of solutions: students need to know *why* a certain formula is true or false at a given state of a model. Unfortunately, tutorial slots are limited and personalized training gets complicated when the number of students scales up. This translates into the call for tools that permit asynchronous and autonomous learning of model-checking problems, providing students not only with a way of validating their own solutions but also with appropriate explanations.

Contribution. In short, EMC provides a user-friendly solver of epistemic model-checking problems with additional textual explanations for the proposed solutions. Models are handled via a straightforward set-theoretic notation, and EMC automatically produces a graphical representation of the model, while formulas are handled using a straightforward syntax. Moreover, the software also offers a sequence of labelled graphs that represent each step of the model-checking algorithm, giving users a pictorial representation of how the procedure unfolds. In this chapter, we provide two main contributions with respect to previous versions of the tool: the inclusion of the *common knowledge operator* [15] in our language; and the release of the current version of the tool as a web application `https://emc-snowy.vercel.app/`, so as to spread its use and gather more feedback from instructors and students for further improvements.

Structure. The rest of this Chapter is organized as follows. Section 1.2 provides the necessary background on epistemic logic. In Section 1.3, we present the main features of EMC. Our software is compared to other existing solutions in Section 1.4. We conclude by giving some future research directions in Section 1.5.

1.2 Background

Let us first introduce the formal background notions that EMC is fed with. Any reader who is familiar with standard epistemic logic with common knowledge can skip this section.

1.2.1 Syntax

We assume as given from now on a countable set of atoms At and a finite, non-empty set of agents Ag. The **language of multi-agent epistemic logic with common knowledge**, denoted as $\mathcal{L}$, is parametrised by both sets and it is given by the following grammar:

$$A ::= \quad p \mid \neg A \mid (A \vee A) \mid (A \wedge A) \mid (A \rightarrow A) \mid \mathsf{K}_a A \mid \mathsf{C}_{\mathsf{B}} A$$

where p ranges over At, a ranges over Ag, and B ranges over $\wp(\mathsf{Ag})$. Individual epistemic formulas $\mathsf{K}_a A$ reads "agent a knows/believes that A" (we stick to knowledge from now on for the sake of brevity). The dual of K_a ($\hat{\mathsf{K}}_a =_{def} \neg\mathsf{K}_a\neg$) is interpreted as *epistemic possibility* (i.e., information that is consistent with a's knowledge). So, for instance, $\mathsf{K}_a(p \rightarrow \hat{\mathsf{K}}_b q)$ reads "a knows that if p is the case, then b considers q as epistemically possible". Finally, $\mathsf{C}_{\mathsf{B}} A$ reads "the group of agents B has common knowledge that A".

A couple of remarks are in order. First, the notion of common knowledge is not of exclusive interest to the epistemic logic community; it also has applications in game theory and non-formal epistemology, among others. The unfamiliar reader is referred to [15] for an overview of the concept. One of the characterisations of this notion is usually given in the following terms: a group of agents commonly knows that A if and only if everyone in the group knows, that everyone in the group knows... that A (where ... is an infinitely long chain of 'everyone in the group knows'). Second, our focus on knowledge (instead of belief) is not philosophically innocent. In fact, there is a deep and long-standing debate within the epistemic logic community (sometimes intersecting with non-formal epistemology) about what the properties of belief of knowledge are, and whether they are inter-definable notions. Although very interesting, this debate is not strictly needed for our presentation, so we just point out to work of [2] for an overview.

1.2.2 Semantics

Semantically, epistemic logics are interpreted using multi-relational models, where each relation is assigned to one different agent. Formally, a **multi-agent epistemic model** for Ag and At (or just a model, for short) is a tuple $M = (W, R, V)$ where:

- $W \neq \emptyset$ represents a set of *possible worlds*;
- $R : \mathsf{Ag} \to \wp(W \times W)$ is a function assigning an *epistemic accessibility relation* $R_a \subseteq W \times W$ to each agent a; and
- and V is an *atomic valuation* assigning to each atom p the set of worlds $V(p)$ where p is true.

A pointed model is a pair $((W, R, V), w)$, where $w \in W$. Given a relation $R \subseteq W \times W$, we use R^+ to denote its transitive closure, i.e., the smallest (w.r.t. set inclusion) transitive relation that contains R. This notion plays an important role in the formal interpretation of the common knowledge operator.

Formulas are interpreted at pointed models, where they can be either true or false. We denote by $\models$ the **truth relation**, and read $M, w \models A$ as "the formula A is true at the pointed model (M, w)" or "formula A is true at world w of model M". Let (M, w) be given, the relation is defined recursively on the structure of formulas:

$$
\begin{array}{rll}
M, w \models p & \text{iff} & w \in V(p) \\
M, w \models \neg A & \text{iff} & \text{it is not the case that } M, w \models A \\
M, w \models A \vee B & \text{iff} & \text{either } M, w \models A \text{ or } M, w \models B \\
M, w \models A \wedge B & \text{iff} & \text{both } M, w \models A \text{ and } M, w \models B \\
M, w \models A \rightarrow B & \text{iff} & \text{either not } M, w \models A \text{ or } M, w \models B \\
M, w \models \mathsf{K}_a A & \text{iff} & (w, u) \in R \text{ implies } M, u \models A \\
M, w \models \mathsf{C}_{\mathsf{B}} A & \text{iff} & (w, u) \in (\bigcup_{a \in \mathsf{B}} R_a)^+ \text{ implies } M, u \models A
\end{array}
$$

Give $M = (W, R, V)$ and $A \in \mathcal{L}$. We say that A is globally true at M iff $M, u \models A$ for every $u \in W$. See [3, Appendix A] for more details on relational semantics for epistemic logic.

Some intuitions behind the definitions. The formal notion of epistemic accessibility can be intuitively understood as follows, $(w, u) \in R_a$ means that, if w is the real world, then a considers u as a candidate for the real world. As an example, think of a as an agent that does not know whether Yakarta is the current capital of Indonesia, and think of w as a world where Yakarta is actually the capital of Indonesia, and u as a world where it is not. Then, we have that a considers both w and u as candidates for the real world (formally, this amounts to $(w, w) \in R_a$ and $(w, u) \in R_a$). Hence, the truth clause for K_a is telling us that a knows that A at w if and only if A is true in every possible world that a considers as a candidate for the real world from w.

Finally, **model checking problems** are computational decision problems of the following shape:

Input. A finite pointed model (M, w) and a formula $B \in \mathcal{L}$.
Query. Is B true at (M, w)?

Examples will follow shortly. Let's just note now that these problems are clearly decidable (they are actually P-complete problems [4]). A simple algorithm for solving these problems works as follows: (i) it computes the subformulas of A, (ii) it labels each world of the model with the subformulas that are true at it, starting with atoms and proceeding incrementally with longer formulas until arriving at A itself. Note that the definition of truth always uses shorter formulas on the right-hand side of each clause.

1.3 Features of the application

EMC is written in Javascript to improve its web performance. We have used the Tree-sitter[2] tool for the syntactic handling of the formulas, as well as the Cytoscape[3] library for graph visualization.

In short, our application permits the student to:

1. Introduce models using a straightforward set-theoretic notation and automatically generate their graphical representation.
2. Introduce formulas using either (i) a simple syntax or; (ii) a virtual keyboard.
3. Query the program about a model checking the problem and get the solution in a step-by-step and explained fashion.

We believe that these three features, especially the last one, facilitate students to spot and correct their own mistakes when solving epistemic model-checking problems. Figure 1.1 provides a view of the web environment, which is a brief explanation of what each field is used for.

1.3.1 Input method for models

In EMC, models are introduced via .set files. The syntax for describing them is just the standard set-theoretic notation. A

Let us consider, as an example, the epistemic model for the Muddy Children problem with three kids after the first announcement is made.

[2]`https://tree-sitter.github.io/tree-sitter/`
[3]`https://js.cytoscape.org/`

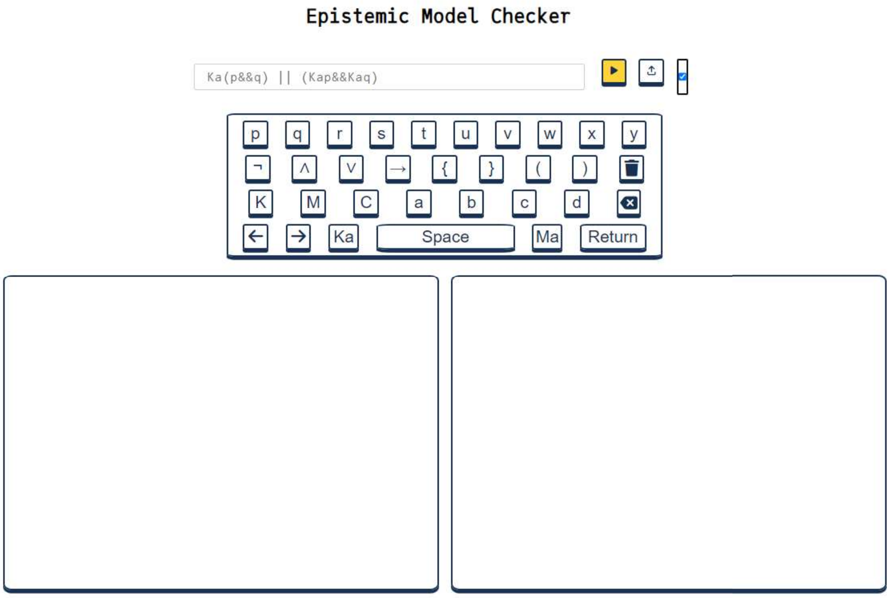


Figure 1.1: General view of EMC web app

For unfamiliar readers, the Muddy Children is a classical puzzle in the (dynamic) epistemic logic literature where a human progenitor poses to her/his offspring the problem of identifying who among them has their forehead spotted with mud. So, consider the version of the puzzle with three kids after the progenitor has declared "at least one of you has mud in her/his forehead".[4] Take $\{a, b, c\}$ as the set of the three kids, and p (resp. q and r) as the sentence "kid a (resp. b, c) has his/her forehead clean". The resulting model can be introduced into EMC by creating a .set file containing the following lines:

```
W={w0,w1,w2,w3,w4,w5,w6}

Ra={<w0,w0>,<w1,w1>,<w2,w2>,<w3,w3>,<w4,w4>,
<w5,w5>,<w6,w6>,<w0,w3>,<w3,w0>,<w1,w2>,
<w2,w1>,<w4,w6>,<w6,w4>}
```

[4]For further details and a full solution to the problem, the curious reader is referred to [3, Appendix B].

```

Rb={<w0,w0>,<w1,w1>,<w2,w2>,<w3,w3>,<w4,w4>,
<w5,w5>,<w6,w6>,<w0,w4>,<w4,w0>,<w3,w6>,
<w6,w3>,<w2,w5>,<w5,w2>}

Rc={<w0,w0>,<w1,w1>,<w2,w2>,<w3,w3>,<w4,w4>,
<w5,w5>,<w6,w6>,<w0,w1>,<w1,w0>,<w2,w3>,
<w3,w2>,<w5,w6>,<w6,w5>}

V(p)={w0,w1,w4}
V(q)={w4,w5,w6}
V(r)={w1,w2,w5}


```

Once the .set file is loaded in EMC's graphical interface, the graphical representation shown in Figure 1.2 is automatically generated by the software.

Note that the seven worlds of the model cover all possible valuations over $\{p, q, r\}$ except the one where the three statements are false (because the previous announcement of the progenitor has excluded this possibility).

1.3.2 Input method for formulas

Formulas are handled by EMC through the following simple syntax:

Input syntax	Standard notation
&&	$\wedge$
||	$\vee$
=>	$\rightarrow$
–	$\neg$
Ka	K_a
Ma	$\hat{\mathsf{K}}_a$
C{abc}	$\mathsf{C}_{\{a,b,c\}}$

For example, $\mathsf{C}_{\{b,c\}}\hat{\mathsf{K}}_b(q \wedge \neg\mathsf{K}_a p)$ is written C{b,c} Mb (q && – Ka p)

We utilized Tree-sitter as the parser, a robust parsing framework

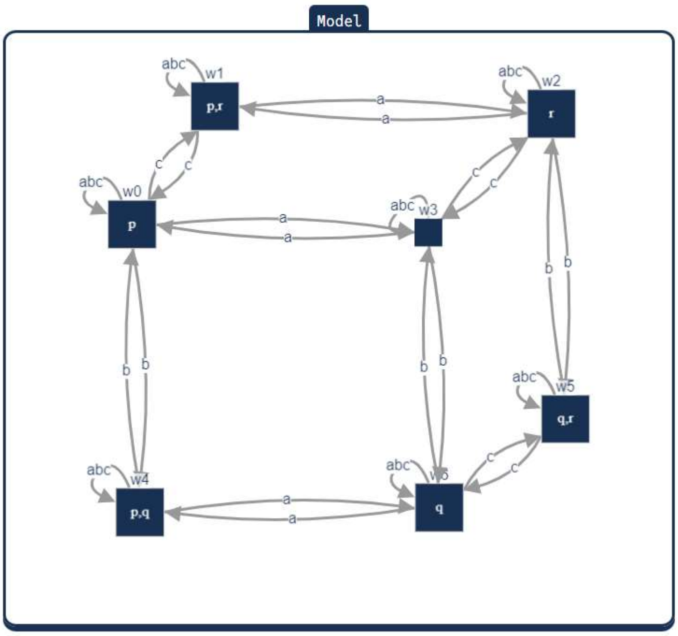


Figure 1.2: Graphical representation of the MuddyChildren scenario produced by EMC.

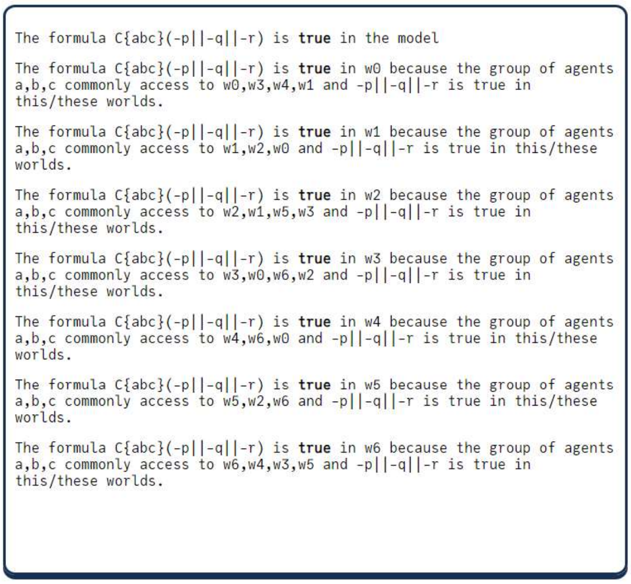
designed for creating and manipulating abstract syntax trees (ASTs). While primarily intended for programming language parsing, this tool can also parse text using custom CFGs (context-free grammars[5]). The resulting AST serves not only for syntax verification but also for extracting information. As an example, to obtain the type of a given formula, we have used the following query.

```
(formula
   operator:(or))@or_formula
(formula
   operator:(and))@and_formula
(formula
   operator:(iff))@iff_formula
(formula
   operator:(know))@know_formula
(formula
   operator:(eq))@eq_formula
(formula
   operator:(not))@not_formula
(atom) @atom_formula
```

1.3.3 Step-by-step display of the algorithm

Once the user queries the program whether a formula is true in a model, she gets the solution for each world, together with a brief explanation for each of them. Following the example, if we query EMC about the semantic status of $\mathsf{C}_{\mathsf{Ag}}(\neg p \vee \neg q \vee \neg r)$ (informally, it is common knowledge among the kids that at least one of them has a dirty forehead), then we will obtain the answer depicted in Figure 1.3. Moreover, the user also gets a list of graphs, each of them corresponding to one step of the model-checking algorithm. For instance, the graph corresponding to the third step of our example, where all Boolean subformulas of the queried formulas have already been evaluated in the model, is shown in Figure 1.4.

[5]The used grammar can be checked in Github



```
The formula C{abc}(-p||-q||-r) is true in the model

The formula C{abc}(-p||-q||-r) is true in w0 because the group of agents
a,b,c commonly access to w0,w3,w4,w1 and -p||-q||-r is true in
this/these worlds.

The formula C{abc}(-p||-q||-r) is true in w1 because the group of agents
a,b,c commonly access to w1,w2,w0 and -p||-q||-r is true in this/these
worlds.

The formula C{abc}(-p||-q||-r) is true in w2 because the group of agents
a,b,c commonly access to w2,w1,w5,w3 and -p||-q||-r is true in
this/these worlds.

The formula C{abc}(-p||-q||-r) is true in w3 because the group of agents
a,b,c commonly access to w3,w0,w6,w2 and -p||-q||-r is true in
this/these worlds.

The formula C{abc}(-p||-q||-r) is true in w4 because the group of agents
a,b,c commonly access to w4,w6,w0 and -p||-q||-r is true in this/these
worlds.

The formula C{abc}(-p||-q||-r) is true in w5 because the group of agents
a,b,c commonly access to w5,w2,w6 and -p||-q||-r is true in this/these
worlds.

The formula C{abc}(-p||-q||-r) is true in w6 because the group of agents
a,b,c commonly access to w6,w4,w3,w5 and -p||-q||-r is true in
this/these worlds.
```

Figure 1.3: Textual output of EMC

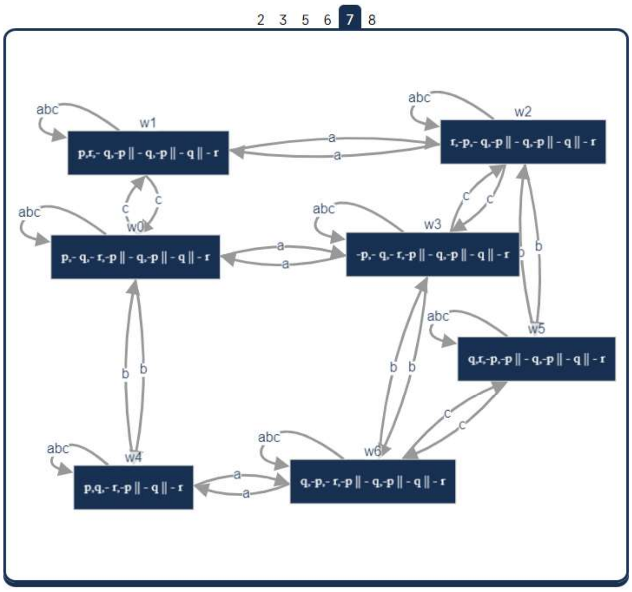


Figure 1.4: Intermediate graph for our example

1.4 Related work

There are many other epistemic model checkers available online. Here is a non-exclusive list of them:

- *Symbolic Model Checker for Dynamic Epistemic Logic* (SMCDEL)[6], by Malvin Gattinger and colleagues [11], is a powerful and efficient software based on the translation of (dynamic) epistemic logic model checking problems to the satisfiability problem of propositional logic (SAT), so as to make use of state-of-the-art efficient SAT solvers.

- *Model Checking Knowledge* (MCK), developed by a number of researchers from the University of New South Wales[7] (see e.g., [12] for an application). This project takes into account not only knowledge but also time and probabilistic aspects, broadening in this way the target community of users.

- *Epistemic Logic Visualiser* (ELVis)[8], by Shoshin Nomura includes the full toolkit of *Dynamic Epistemic Logic* (DEL),[9] and it additionally covers proof theoretical issues by the handling and automatic production of sequent-based proofs (see [8]).

- *DEL playground*[10], by Elliot Evans, which is in turn based on *Modal Logic Playground* by Ross Kirsling is a new and graphical epistemic model checker that also includes *public announcements* in its repertoire.[11]

[6] Available at `https://w4eg.de/malvin/illc/smcdelweb/index.html`

[7] Available at `http://www.cse.unsw.edu.au/~mck/pmck/`, please consult there the full list of contributors too.

[8] Available at `https://nomuras.github.io/ELVis/`

[9] DEL is the study of extensions of epistemic logic system with operators meant to capture different types of informational changes [3, 10, 14]

[10] Available at `https://vezwork.github.io/modallogic/?model=;AS?formula=`

[11] Public Announcement Logic (PAL) is the most well-known dynamic epistemic logic, first introduced by [9].

We focus our comparison of EMC with the last two programs since the other ones were developed for researchers and not for students as target users. The main advantages of ElVis and DEL playground over EMC are that:

- They include dynamic features (public announcements in the case of DEL playground, and the full power of event models in the case of ELViS);
- They allow for checking/forcing some binary property of accessibility relations; and
- They include a drawing tool for entering the model directly in a graphical setting (instead of describing it formally, as we have to do in EMC).

However, EMC has also some advantages over these two programs. First, it includes the common knowledge operator, whose formal meaning, according to our teaching experience, results in a significant challenge to undergraduate students (when compared to individual epistemic operators). More importantly, there are a couple of features that EMC has and other tools do not, and we think they are of great pedagogical relevance:

- the step-by-step decomposition of the algorithm execution together with its graphical representation; and
- the display of textual explanations for results.

As we have argued, these utilities let students not only validate their solutions but also spot, understand and self-correct their mistakes.

1.5 Conclusion and future steps

This chapter was devoted to the presentation EMC, a web model checker for multi-agent epistemic logic with common knowledge. We explained its main features and compared it with similar existing tools.

There are several possible directions for the future development of our model checker. Besides the inclusion of dynamic aspects and the possibility of inputting the model graphically, which we already mentioned, we consider it most important to expand the use of the tool among the target public and to gather feedback from them (perhaps in a systematic way), to adapt better the tool to their teaching-learning needs.

Bibliography

[1] Horacio Arló-Costa, Vincent F Hendricks, Johan Van Benthem, Henrik Boensvang, and Rasmus K Rendsvig. *Readings in formal epistemology*. Springer, 2016.

[2] Guillaume Aucher. Principles of knowledge, belief and conditional belief. In Manuel Rebuschi, Martine Batt, Gerhard Heinzmann, Franck Lihoreau, Michel Musiol, and Alain Trognon, editors, *Interdisciplinary Works in Logic, Epistemology, Psychology and Linguistics: Dialogue, Rationality, and Formalism*, pages 97–134. Springer, 2014.

[3] Alexandru Baltag and Bryan Renne. Dynamic Epistemic Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2016.

[4] Tristan Charrier, François Schwarzentruber, G Bezhanishvili, G D'Agostino, G Metcalfe, and T Studer. Complexity of dynamic epistemic logic with common knowledge. *Advances in Modal Logic*, 12:27–31, 2018.

[5] Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Vardi. *Reasoning about knowledge.* MIT press, 2004.

[6] Sven Ove Hansson, Vincent F Hendricks, and Esther Michelsen (editors) Kjeldahl. *Introduction to formal philosophy.* Springer, 2018.

[7] Jaakko Hintikka. *Knowledge and belief: an introduction to the logic of the two notions.* Cornell University Press, 1962.

[8] Shoshin Nomura, Hiroakira Ono, and Katsuhiko Sano. A cut-free labelled sequent calculus for dynamic epistemic logic. *Journal of Logic and Computation*, 30(1):321–348, 2020.

[9] Jan Plaza. Logics of public announcements. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216. Oak Ridge National Laboratory, 1989.

[10] Johan van Benthem. *Logical dynamics of information and interaction.* Cambridge University Press, 2011.

[11] Johan Van Benthem, Jan Van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic-s5 and beyond. *Journal of Logic and Computation*, 28(2):367–402, 2018.

[12] Ron van der Meyden. Optimizing epistemic model checking using conditional independence. In J Lang, editor, *Proceedings of the Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, EPTCS, pages 398–414. Open Publishing Association, 2017.

[13] Hans van Ditmarsch, Joseph Y. Halpern, Wiebe van der Hoek, and Barteld Kooi, editors. London: College Publications, 2015.

[14] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic epistemic logic.* Springer, 2007.

[15] Peter Vanderschraaf and Giacomo Sillari. Common Knowledge. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2022 edition, 2022.