ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA250988

Reasoning with Epistemic Rights and Duties: Automating a Dynamic Logic of the Right to Know in LogiKEy

Lara Lawniczak[a], Luca Pasetto [b,*], Christoph Benzmüller[a,c], Xu Li[b] and Réka Markovich[b]

[a]Otto-Friedrich-Universität Bamberg
[b]University of Luxembourg
[c]Freie Universität Berlin

Abstract. It is not straightforward to reason about specific legal concepts such as *epistemic rights and duties*, which are crucial in AI systems that have to make autonomous decisions based on who knows what, who is entitled to know, and under what conditions information should be shared or withheld. Such issues are central to responsible AI, data governance, and regulatory compliance. A concrete application arises is in the context of the GDPR, where a data subject has a *right to know* whether and for what purpose her personal data is being processed, creating a *duty to tell* for the controller when asked. On the other hand, if the software used for the processing is proprietary, the data subject does not have the right to know its exact mechanisms, so her asking to know them does not create a corresponding duty for the data controller.

In this paper, a shallow semantical embedding (SSE) of the Dynamic Logic of the Right to Know (LRK) in Higher-Order Logic is presented. The embedding is proven faithful, and it is encoded and experimented with in the *Isabelle/HOL* proof assistant. The SSE is then used to reason with the GDPR example encoded in LRK.

The embedding of LRK differs from existing ones in how it represents the dynamic updating of the model: instead of performing changes on the domain of possible worlds, the provided SSE maintains the accessibility and neighborhood relations within the context of a formula. Updates are then handled by updating the relations, while the domain of possible worlds stays the same.

The work presented in this paper contributes to the LogiKEy knowledge engineering methodology and framework, which enables experimentation with logics and logic combinations, with general and domain knowledge, and with concrete use cases.

1 Introduction

Epistemic rights, such as the *right to know*, concern an individual's ability to access, receive, and control information, playing a crucial role in our information-centric society [62]. Such rights raise critical questions about transparency, accountability, and compliance with legal frameworks and are increasingly relevant to AI systems that handle personal data, make autonomous decisions, or mediate access to information [59]. For example, the General Data Protection Regulation (GDPR) [26] establishes that the data subject has a right to (or right of) access to some specific information, such as information about whether her personal data is being processed, and if so, for what purpose. As Article 15 puts it, the data subject has a *right to obtain* this information by asking for it. However, we also know from Recital 63 [26] that this right should not adversely affect the rights or freedoms of others, including trade secrets or intellectual property and in particular the copyright protecting the software, meaning that the data subject is not allowed to know this information. Reasoning with such an epistemic right is not straightforward, as it requires formally representing dynamic knowledge changes, contextual dependencies, and the interplay between obligations, permissions, and prohibitions.

In this example, the right to know is a *power* as described in the theory of normative positions based on the work of Hohfeld [34], where power is characterized as the potential of the agent to execute an action resulting in a change in the counterparty's normative positions, for instance by creating a duty [46]. The Dynamic Logic of the Right to Know (LRK) [42] is a logic that explicitly deals with the right to know as a power to know whether something is the case. LRK allows to represent and reason with scenarios where agents are entitled to access certain information under specific conditions, an issue highly relevant to the GDPR and to AI regulation. These regulations concern hundreds of thousand of entities for the procedures of whom automation and transparent, explainable computational solutions are highly valuable, and hence subject to very active research within the field of AI and Law.

The objective is to enable *machines* to reason correctly and verifiably on right to know issues, and to develop practical means to support this. However, neither LRK nor any other logic with a similar purpose has been automated or applied yet. In this paper, we provide a Shallow Semantic Embedding (SSE) of LRK and demonstrate that, and how, this logic can be seen and elegantly handled as a fragment of Classical Higher-Order Logic (HOL) [21, 8]. We show that our embedding is faithful, that is, sound and complete, and we use it to successfully encode and reason with the above GDPR example.

This work constitutes an important addition to the LogiKEy [15] logic-pluralistic knowledge representation and reasoning methodology represented in Fig. 1. LogiKEy's unifying formal framework is based on SSEs of 'object' logics (and their combinations) in HOL, enabling the provision of powerful tool support: off-the-shelf theorem provers and model finders for HOL [9], as provided, e.g., in the proof assistant system *Isabelle/HOL* [50], are assisting the LogiKEy knowledge engineer to flexibly experiment with underlying logics and their combinations, with general and domain knowledge, and with concrete use cases–all at the same time. The approach is capable of supporting model checking, model finding and theorem proving,

* Corresponding Author. Email: Luca.Pasetto@uni.lu.

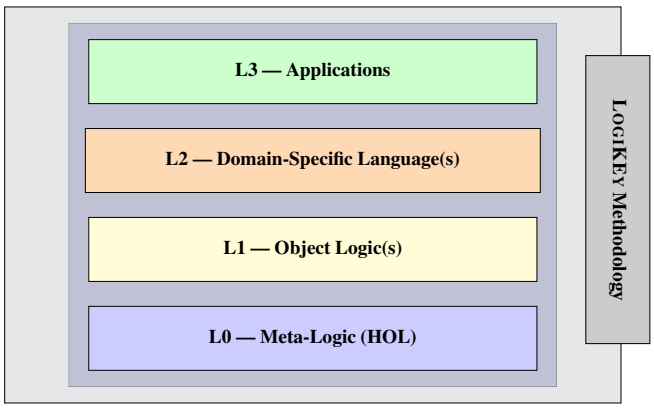

Figure 1. LogiKEy KR&R methodology

while continuous improvements of off-the-shelf provers and model finders boost the reasoning performance in LogiKEy without further ado. One particular focus of the framework is on ethico-legal applications [17] and normative reasoning [52]. At layer L0 of LogiKEy, HOL is serving as the underlying metalogic to encode (combinations of) the object logics of layer L1, allowing for the expression of domain theories at layer L2, which enables experimentation with concrete applications and examples at layer L3. While the LogiKEy methodology has been applied successfully to automate a number of logics over the years, in this work we correctly encode a non-trivial logic like LRK and we automate it for the first time.

The structure of the paper is as follows: Section 2 briefly recaps HOL, and Sect. 3 sketches LRK. In Sect. 4 a shallow semantical embedding of LRK in HOL is presented and discussed, and faithfulness of the embedding is proved. Section 5 tests the embedding and uses it for automating the above GDPR example. Section 6 discusses related work and concludes the paper.

2 Classical Higher-Order Logic

Classical Higher-Order Logic (HOL), a polymorphic version of Church's *type theory* [21, 8], is sketched below (adapted from [5]).

Syntax of HOL. The *syntax* of HOL is defined by the grammar

$$s,t := P_\alpha \mid x_\alpha \mid (\lambda x_\alpha s_\beta)_{\alpha\Rightarrow\beta} \mid (s_{\alpha\Rightarrow\beta} t_\alpha)_\beta$$

where $\alpha,\beta,o \in \mathcal{T}$ and with $\mathcal{T}$ being a set of *simple types* defined by $\alpha,\beta := o \mid i \mid (\alpha\Rightarrow\beta)$. Type o denotes truth values, i individuals, and $\Rightarrow$ is the function type constructor. The P_α are typed constants symbols in the signature of HOL, and the x_α are typed variable symbols (distinct from P_α). Complex HOL terms are constructed from given HOL terms via λ-abstraction $(\lambda x_\alpha s_\beta)_{\alpha\Rightarrow\beta}$ and function application $(s_{\alpha\Rightarrow\beta} t_\alpha)_\beta$, which both involve type constraints. HOL is thus a logic of terms defined on top of the simply typed λ-calculus, and terms of type o are called *formulas*. The type of each term is given as a subscript and may be omitted if obvious in context. As *primitive logical connectives* we choose $\neg_{o\Rightarrow o}, \vee_{o\Rightarrow o\Rightarrow o}, =_{\alpha\Rightarrow\alpha\Rightarrow o}$ (short: $=^\alpha$) and $\Pi_{(\alpha\Rightarrow o)\Rightarrow o}$ (short: Π^α). Other logical connectives can be introduced as abbreviations resp. shorthand notations; e.g. $\forall x_\alpha\, \varphi_o = \Pi^\alpha(\lambda x_\alpha\, \varphi)$ and $\longrightarrow_{o\Rightarrow o\Rightarrow o} = \lambda x_o \lambda y_o (\neg x \vee y)$.[1]

[1] Dot-notation is used in the remainder, where the scope opened by . is reaching as far to the right as consistent with the formula structure. Moreover, types may be omitted if obvious in context; we may thus write $\lambda x\, y.\, \neg x \vee y$ instead of $\lambda x_o \lambda y_o (\neg x_o \vee y_o)$.

Semantics of HOL. A *frame* $\mathcal{D}$ for HOL is a collection $\{\mathcal{D}_\alpha\}_{\alpha\in\mathcal{T}}$ of nonempty sets $\mathcal{D}_\alpha$, such that $\mathcal{D}_o = \{T, F\}$ (for true and false). $\mathcal{D}_i$ is chosen freely and $\mathcal{D}_{\alpha\Rightarrow\beta}$ are collections of functions mapping $\mathcal{D}_\alpha$ into $\mathcal{D}_\beta$. A *model* for HOL is a tuple $\mathcal{M} = (\mathcal{D}, I)$, where $\mathcal{D}$ is a frame, and I is a family of typed interpretation functions mapping constant symbols P_α to appropriate elements of $\mathcal{D}_\alpha$, called the *denotation* of P_α. The logical connectives $\neg, \vee, \Pi$ and $=$ are always given their expected standard denotations: $I(\neg) = \mathit{not} \in \mathcal{D}_{o\Rightarrow o}$ s.t. $\mathit{not}(T) = F$ and $\mathit{not}(F) = T$; $I(\vee) = \mathit{or} \in \mathcal{D}_{o\Rightarrow o\Rightarrow o}$ s.t. $\mathit{or}(a,b) = T$ iff $(a = T$ or $b = T)$; $I(=^\alpha) = \mathit{id} \in \mathcal{D}_{\alpha\Rightarrow\alpha\Rightarrow o}$ s.t. $\forall a,b \in \mathcal{D}_\alpha, \mathit{id}(a,b) = T$ iff a is identical to b; $I(\Pi^\alpha) = \mathit{all} \in \mathcal{D}_{(\alpha\Rightarrow o)\Rightarrow o}$ s.t. $\forall s \in \mathcal{D}_{\alpha\Rightarrow o}, \mathit{all}(s) = T$ iff $s(a) = T$ for all $a \in \mathcal{D}_\alpha$.

A *variable assignment* g maps variables x_α to elements in $\mathcal{D}_\alpha$. $g[d/x]$ denotes the g', that is identical to g, except for variable x, which is now mapped to d. The *denotation* $[\![s_\alpha]\!]^{M,g}$ of an HOL term s_α on a model $\mathcal{M} = \langle \mathcal{D}, I\rangle$ under assignment g is an element $d \in \mathcal{D}_\alpha$ defined in the following way:

$$\begin{array}{lcl} [\![P_\alpha]\!]^{\mathcal{M},g} & = & I(P_\alpha) \\ [\![x_\alpha]\!]^{\mathcal{M},g} & = & g(x_\alpha) \\ [\![(s_{\alpha\Rightarrow\beta} t_\alpha)_\beta]\!]^{\mathcal{M},g} & = & [\![s_{\alpha\Rightarrow\beta}]\!]^{\mathcal{M},g}([\![t_\alpha]\!]^{\mathcal{M},g}) \\ [\![(\lambda x_\alpha s_\beta)_{\alpha\Rightarrow\beta}]\!]^{\mathcal{M},g} & = & \text{the } f{:}\mathcal{D}_\alpha \mapsto \mathcal{D}_\beta \text{ s.t. for all } d \in \mathcal{D}_\alpha \\ & & f(d) = [\![s_\beta]\!]^{\mathcal{M},g[d/x_\alpha]} \end{array}$$

It follows: $[\![\forall x_\alpha \varphi_o]\!]^{\mathcal{M},g} = T$ iff $[\![\varphi_o]\!]^{\mathcal{M},g[d/x_\alpha]} = T$ for all $d \in \mathcal{D}_\alpha$.

In a *standard model* a domain $\mathcal{D}_{\alpha\Rightarrow\beta}$ is defined as the set of all total functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$: $\mathcal{D}_{\alpha\Rightarrow\beta} = \{f \mid f : \mathcal{D}_\alpha \mapsto \mathcal{D}_\beta\}$. In a *Henkin model* (or general model) [33] function spaces are not necessarily required to be the full: $\mathcal{D}_{\alpha\Rightarrow\beta} \subseteq \{f \mid f : \mathcal{D}_\alpha \mapsto \mathcal{D}_\beta\}$. However, it is required that every term still denotes.

Term s_o is *valid in* $\mathcal{M}$ *under assignment* g, denoted as $\mathcal{M}, g \models^{\text{HOL}} s_o$, iff $[\![s_o]\!]^{\mathcal{M},g} = T$. s_o is *valid in* $\mathcal{M}$, denoted as $\mathcal{M} \models^{\text{HOL}} s_o$, iff $\mathcal{M}, g \models^{\text{HOL}} s_o$ for all assignments g, and s_o is *valid*, denoted as $\models^{\text{HOL}} s_o$, iff s_o is valid in all Henkin models $\mathcal{M}$.

Due to Gödel [31] a sound and complete mechanization of HOL with standard semantics cannot be achieved. For HOL with Henkin semantics sound and complete calculi exist, cf. [9, 12] and the references therein. Each standard model is obviously also a Henkin model. Consequently, when a HOL formula is Henkin-valid, it is also valid in all standard models.

3 A Dynamic Logic of the Right to Know

The Dynamic Logic of the Right to Know (LRK), introduced in [42], can be used to represent scenarios involving communication between two agents, where information is communicated directly from the sender to the receiver (indicated by s and r, respectively). Announcements can only be made by the sender, and the receiver may ask questions. The behavior of the sender is subject to additional restrictions, for instance coming from information security policies. The main objective of LRK is to characterize and reason about the receiver's so-called *power* [46] type of right to know whether φ, which means that the sender is obliged to announce the answer *if* the receiver asks the question whether φ is the case—the question itself creating the duty which did not exist before. The main notions of LRK are introduced as required for the rest of the paper; for details see [42].

Syntax of LRK. The syntax of LRK is defined by the grammar (where $p \in$ PROP are atomic propositional symbols):

$$\varphi,\psi := p \mid \neg\varphi \mid \varphi \to \psi \mid U\varphi \mid Q\varphi \mid K_r\varphi \mid \mathbb{O}_s\varphi \mid [\varphi?]\psi \mid [\varphi!]\psi$$

Other Boolean connectives can be defined as usual, and the following abbreviations are introduced: $R_r\varphi := U(Q\varphi \vee Q\neg\varphi)$, $[r:\varphi?]\psi := (\neg R_r\varphi \wedge \psi) \vee (R_r\varphi \wedge [\varphi?]\psi)$, and $[s:\varphi!]\psi := (\varphi \to [\varphi!]\psi)$.

In LRK, $U\varphi$ is the familiar universal modality expressing that "φ is true in all worlds", while $Q\varphi$ is a technical modality that first appeared in [60]. The fundamental idea that "The receiver has the power to know the answer to the question φ?" is captured by the formula $R_r\varphi := U(Q\varphi \vee Q\neg\varphi)$. The expression $K_r\varphi$ means "The receiver knows φ", while $\mathbb{O}_s\varphi$ stands for "The sender is obliged to announce φ". The technical operators $[\varphi?]\varphi$ and $[\varphi!]\varphi$ are used to define more complex operators. The concept that "After the receiver has asked the question $\varphi?$, ψ holds" is represented by $[r:\varphi?]\psi := (\neg R_r\varphi \wedge \psi) \vee (R_r\varphi \wedge [\varphi?]\psi)$. Finally, the notion of truthful announcements is expressed by $[s:\varphi!]\psi := (\varphi \rightarrow [\varphi!]\psi)$, which reads as "After the sender truthfully announces φ, it follows that ψ holds".

Semantics of LRK. An *LRK model* is a tuple $M = (W, \sim, \approx, N, V)$, where W is a non-empty set of possible worlds, $\sim$ and $\approx$ are two equivalence relations on W, $N : W \mapsto \wp(\wp(W))$ is such that $w \in X$ for all $w \in W$ and $X \in N(w)$, and $V : \text{PROP} \mapsto \wp(W)$ is a valuation function that assigns a set of worlds to each atomic proposition. Vice versa, each world can be identified with the set of propositions that are validated in it. A pointed model is a pair M, w such that w is a state of M. For every state $w \in W$, $\sim(w)$ denotes the set $\{v \in W \mid w \sim v\}$, and analogously for $\approx(w)$. The relation $\sim$ is the epistemic indistinguishability relation of the receiver. In what follows, the partition generated by the equivalence classes of $\approx$ will also be denoted by $\approx$, and this partition $\approx$ encodes the set of questions to which the receiver has the power to know the answers, as prescribed for instance by some given information security policies. N is a neighborhood function and each subset $X \in N(w)$ is an *ideal epistemic state* for the receiver at w, i.e., the epistemic state X is compliant with the given policies specified at w.

Given a model $M = (W, \sim, \approx, N, V)$, for all $w \in W$ and $\varphi \in \mathcal{L}$, the satisfaction relation $M, w \models \varphi$ is inductively defined as:

$M, w \models p$ iff $w \in V(p)$
$M, w \models \neg\varphi$ iff $M, w \not\models \varphi$
$M, w \models \varphi \rightarrow \psi$ iff $M, w \not\models \varphi$ or $M, w \models \psi$
$M, w \models U\varphi$ iff $\forall v{\in}W$, $M, v \models \varphi$
$M, w \models Q\varphi$ iff $\forall v{\in}W$, $w \approx v$ implies $M, v \models \varphi$
$M, w \models K_r\varphi$ iff $\forall v{\in}W$, $w \sim v$ implies $M, v \models \varphi$
$M, w \models \mathbb{O}_s\varphi$ iff $\forall X{\in}N(w)$, $X \subseteq \sim(w)$ implies $X \subseteq [\![\varphi]\!]_M$
$M, w \models [\varphi?]\psi$ iff $M_{\varphi?}, w \models \psi$
$M, w \models [\varphi!]\psi$ iff $M_{\varphi!}, w \models \psi$

where $[\![\varphi]\!]_M = \{x \in W \mid M, x \models \varphi\}$ and $M_{\varphi?}$ and $M_{\varphi!}$ are defined as follows:

$M_{\varphi?} := (W, \sim, \approx, N_{\varphi?}, V)$ where for all $x \in W, N_{\varphi?}(x) = \{X \in N(x) \mid X \subseteq [\![\varphi]\!]_M \text{ or } X \subseteq [\![\neg\varphi]\!]_M\}$; and

$M_{\varphi!} := (W, \sim_{\varphi!}, \approx, N, V)$ where $\sim_{\varphi!} = \{(u, v) \in \sim \mid M, u \models \varphi \text{ iff } M, v \models \varphi\}$.

The notion of validity, written as $\models^{LRK} \varphi$, is defined as usual (cf. Sect. 2). In addition, if the above conditions on $\sim$ and $\approx$ to be equivalence relations are dropped and the extra condition on the neighborhood function N is removed, $M = (W, \sim, \approx, N, V)$ is called an LRK$^-$ model; all other definitions apply analogously. An LRK$^-$ model thus constitutes a basic model structure, analogous to basic logic K in the well-known modal logic cube, which leaves the accessibility relation between worlds unrestricted with regard to further conditions such as reflexivity, symmetry, or transitivity.

The convention adopted by the authors of [42] is that an operator without an index serves only as a technical modality, such as U or Q. The LRK semantics of $R_r\varphi$ is non-standard but straightforward: the receiver has the power to know the answer to φ? if and only if the partition $\approx$ "settles" the question, meaning that each cell in $\approx$ lies entirely within either the truth set of φ or $\neg\varphi$. Since $Q\varphi$ behaves as a normal modality for $\approx$, this idea is precisely captured by $U(Q\varphi \vee Q\neg\varphi)$. The formula $[r:\varphi?]\psi$ expresses that "after the receiver asks $\varphi?$, ψ holds." If the receiver does not have the power to know the answer, nothing changes after $[r:\varphi?]$. Otherwise, the sender is obliged to answer, and model updating ensures that non-answering epistemic states cease to be ideal. The dynamic operator $[\varphi?]\psi$ captures what holds after this update, leading to the formalization of $[r:\varphi?]\psi$ as $(\neg R_r\varphi \wedge \psi) \vee (R_r\varphi \wedge [\varphi?]\psi)$. The semantics of $K_r\varphi$ and $[s:\varphi!]\psi$ is standard, except that in $M_{\varphi!}$ links are deleted between φ and $\neg\varphi$-states rather than removing $\neg\varphi$-states from the model. Note that N assigns, to each possible world w, a set of ideal epistemic states $N(w)$: for $\mathbb{O}_s\varphi$, the sender is obliged to announce φ if it is "known" in all ideal epistemic states achievable through further announcements, as $X \subseteq \sim(w)$ says that the ideal epistemic state X must be achievable from the current epistemic state $\sim(w)$.

4 Modeling LRK as a Fragment of HOL

Before providing an embedding of LRK into HOL, a few words about the SSE approach employed here are in order. An SSE of a target logic into HOL provides a translation between the two logics in such a way that the former is identified and characterized as a proper fragment of the latter.[2] Once such an SSE is obtained, all that is needed to prove (or refute) conjectures in the target logic is to provide the SSE, encoded in an input file, to the HOL prover or model finder in addition to the encoded conjecture. The HOL tool can then be used as-is to solve problems in the target logic, i.e. without making any changes to its source code.

4.1 Shallow Semantical Embedding of LRK in HOL

In this subsection an SSE for target logic LRK in HOL is presented and faithfulness of the embedding is proved. To define our SSE, the type of LRK propositions is mapped to HOL type $\tau := \gamma{\Rightarrow}\gamma{\Rightarrow}\upsilon{\Rightarrow}\sigma$, where the following (additional) type abbreviations are used: $\gamma := i{\Rightarrow}i{\Rightarrow}o$, $\upsilon := i{\Rightarrow}((i{\Rightarrow}o){\Rightarrow}o)$ and $\sigma := i{\Rightarrow}o$. Note how HOL type τ captures the dependencies of LRK formulas on two accessibility relations (of type γ), a neighborhood function (of type υ) and on possible worlds (of type i). These dependencies are visible in the definition of their semantical evaluation in Sect. 3, and in principle we could avoid the dependency of the second type γ, since only one of the accessibility relations is updated during semantical evaluation while the other one remains fixed. In general, this idea to explicitly capture and maintain relevant dependencies is analogous to previous work [6, 7, 11], but a bit more complicated here, because more dependencies need to be maintained and dynamically updated.

For each propositional symbol p^k of LRK, the corresponding HOL signature is assumed to contain a corresponding predicate constant symbol $p^k_{i\Rightarrow o}$, which is (rigidly) denoting the set of all those worlds in which p^k holds. The mapping $\lfloor . \rfloor$ translates a LRK formula φ into a HOL term $\lfloor\varphi\rfloor$ of type τ. As first and second arguments such a mapped HOL term $\lfloor\varphi\rfloor$ accepts indistinguishability relation terms t_γ and q_γ, and as third argument a neighborhood function term n_υ. Finally, it takes a current world term w_i with respect to which the evaluation is performed. These dependencies are passed recursively through the recursive evaluation structure captured in the definitions below, and can be modified/updated on the fly. Such updates occur when either the current world in the evaluation changes (this happens for the modal connectives U, Q, Kr, and $\mathbb{O}_s$), or when either

[2] The SSE technique is not be confused with higher-order abstract syntax [54]. For differences between shallow and deep embeddings see [29, 19].

the neighborhood function (see $[?\varphi]\psi$) or the indistinguishability relation $\sim$ (see $[!\varphi]\psi$) is modified. Since only one of the two indistinguishability relations is modified recursively, while the other is always kept fixed, the captured dependencies are not minimal and could be reduced to just one indistinguishability relation of type τ, but we decided for the sake of clarity to represent both of them explicitly. As mentioned in Sect. 3, the connective Q is a technical (normal) modality that is needed to define R_r. We keep this approach for the SSE: since the semantics for $R_r\varphi$ is highly non-standard, the embedding and proof would be much harder if $R_r\varphi$ were primitive in the language.

The mapping $\lfloor . \rfloor$ is defined recursively as follows:

$\lfloor p^k \rfloor = A_{\sigma\Rightarrow\tau}(p^k_\sigma)$ with $A_{\sigma\Rightarrow\tau} = \lambda p_\sigma t_\gamma q_\gamma n_\upsilon w_i.\, pw$

$\lfloor \neg\varphi \rfloor = \neg_{\tau\Rightarrow\tau}\ \lfloor\varphi\rfloor$ with $\neg_{\tau\Rightarrow\tau} = \lambda\varphi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \neg(\varphi tqnw)$

$\lfloor \varphi \rightarrow \psi \rfloor = \rightarrow_{\tau\Rightarrow\tau\Rightarrow\tau}\ \lfloor\varphi\rfloor\lfloor\psi\rfloor$ with $\rightarrow_{\tau\Rightarrow\tau\Rightarrow\tau} = \lambda\varphi_\tau\psi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \neg\varphi tqnw \vee \psi tqnw$

$\lfloor U\varphi \rfloor = U_{\tau\Rightarrow\tau}\lfloor\varphi\rfloor$ with $U_{\tau\Rightarrow\tau} = \lambda\varphi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \forall v_i.\, \varphi tqnv$

$\lfloor Q\varphi \rfloor = Q_{\tau\Rightarrow\tau}\lfloor\varphi\rfloor$ with $Q_{\tau\Rightarrow\tau} = \lambda\varphi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \forall v_i.\, qwv \longrightarrow \varphi tqnv$

$\lfloor Kr\varphi \rfloor = Kr_{\tau\Rightarrow\tau}\lfloor\varphi\rfloor$ with $Kr_{\tau\Rightarrow\tau} = \lambda\varphi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \forall v_i.\, twv \longrightarrow \varphi tqnv$

$\lfloor \mathbb{O}_s\varphi \rfloor = \mathbb{O}_{s\,\tau\Rightarrow\tau}\lfloor\varphi\rfloor$ with $\mathbb{O}_{s\,\tau\Rightarrow\tau} = \lambda\varphi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \forall H_\sigma.\ nwH \longrightarrow (\forall v_i.\, Hv \longrightarrow twv) \longrightarrow (\forall u_i.\, Hu \longrightarrow \varphi tsnu)$

$\lfloor [\varphi?]\psi \rfloor = ?_{\tau\Rightarrow\tau\Rightarrow\tau}\lfloor\varphi\rfloor\lfloor\psi\rfloor$ with $?_{\tau\Rightarrow\tau\Rightarrow\tau} = \lambda\varphi_\tau\psi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \psi tq(\text{UPDATEN}\varphi tqn)w$

$\lfloor [\varphi!]\psi \rfloor = !_{\tau\Rightarrow\tau\Rightarrow\tau}\lfloor\varphi\rfloor\lfloor\psi\rfloor$ with $!_{\tau\Rightarrow\tau\Rightarrow\tau} = \lambda\varphi_\tau\psi_\tau t_\gamma q_\gamma n_\upsilon w_i.\, \psi(\text{UPDATET}\varphi tqn)qnw$

where $\text{UPDATEN}\varphi tqn = \lambda w_i X_\sigma.\, nwX \wedge ((\forall v_i.\, Xv \longrightarrow \varphi tqnv) \vee (\forall v_i.\, Xv \longrightarrow \neg\varphi tqnv))$ and $\text{UPDATET}\varphi tqn = \lambda u_i v_i.\, tuv \wedge (\varphi tqnu \longleftrightarrow \varphi tqnv)$.

We first prove faithfulness of the embedding for the more general LRK$^-$ models and then extend the result to LRK models. This two step approach is not only technically simpler, it also shows that faithfulness results can be obtained for a wider class of models, independent of the particular constraints assumed for $\sim$, $\approx$ and N.

Lemma 1 (Faithfulness: LRK$^-$ models)**.** *Consider an LRK$^-$ model $M = (W, \sim, \approx, N, V)$ and a HOL model $\mathcal{M} = (\mathcal{D}, I)$ such that*

1. *the set of worlds W in M is identified with the domain $\mathcal{D}_i$ in $\mathcal{M}$,*
2. *for each $p^k \in$ PROP of LRK, there is a constant symbol p^k_σ in HOL s.t. for all $w \in W = \mathcal{D}_i$: $I(p^k_\sigma)(w) = T$ iff $w \in V(p^k)$,*
3. *the HOL signature contains $\sim_\gamma$, $\approx_\gamma$ and $\boldsymbol{N}_\upsilon$ s.t. $I(\sim_\gamma) = \sim$, $I(\approx_\gamma) = \approx$ and $I(\boldsymbol{N}_\upsilon) = N$; $\sim$ and $\approx$ are the mentioned LRK equivalence relations, and N is s.t. $w \in X$ for all $w \in W$ and $X \in N(w)$, resp. $I(\boldsymbol{N}_\upsilon)$ satisfies $\forall w_i.\forall X_\sigma.(NwX \longrightarrow Xw)$.*

Then, for all $w \in W$, and X_i not occurring free in $\lfloor\varphi\rfloor$, we have:

$$M, w \models^{LRK} \varphi \text{ iff } [\![\lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon X_i]\!]^{\mathcal{M}, g[w/X_i]} = T \text{ in HOL.}$$

Proof. By induction over LRK formula φ; we only sketch it here.

In the base case, where φ is $p^k \in$ PROP, the statement follows from semantical evaluation and assumption 2.

The cases for $\neg\varphi$ and $\varphi \vee \psi$ follow from the induction hypothesis.

In the case for $U\varphi$ we need to show $M, w \models U\varphi$ in LRK iff $[\![\lfloor U\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon X_i]\!]^{\mathcal{M}, g[w/X_i]} = T$ in HOL. Since $\lfloor U\varphi\rfloor$ λ-converts into $\lambda t_\gamma q_\gamma n_\upsilon w_i.\forall v_i.\lfloor\varphi\rfloor tqnv$, the latter is equivalent to $[\![\forall v_i.\, \lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon v_i]\!]^{\mathcal{M}, g[w/X_i]} = T$, respectively to $[\![\forall v_i.\, \lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon v_i]\!]^{\mathcal{M}, g} = T$, since X_i is not occuring in $\lfloor\varphi\rfloor$ by assumption. This is equivalent to $[\![\lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon X_i]\!]^{\mathcal{M}, g[w/X_i]} = T$ for all $w \in D_i$ by the semantics of HOL, which by induction hypothesis, and since $D_i = W$ by assumption 1, is equivalent to $M, w \models \varphi$ for all $w \in W$ in LRK. By the semantics of LRK it follows $M, w \models U\varphi$ as intended.

The cases for $Q\varphi$, $Kr\varphi$, and $\mathbb{O}_s\varphi$ follow analogously by semantic evaluation, induction and the application of the assumptions.

In the case for $[\varphi!]\psi$ we need to show $M, w \models [\varphi!]\psi$ in LRK iff $[\![\lfloor[\varphi!]\psi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon X_i]\!]^{\mathcal{M}}, g[w/X_i] = T$ in HOL. This is equivalent to $[\![\lfloor\psi\rfloor(\text{UPDATET}\lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon) \approx_\gamma \boldsymbol{N}_\upsilon X_i]\!]^{\mathcal{M}, g[w/X_i]} = T$, since $\lfloor[\varphi!]\psi\rfloor$ λ-converts into $\lambda t_\gamma q_\gamma n_\upsilon w_i.\, \lfloor\psi\rfloor(\text{UPDATET}\lfloor\varphi\rfloor tqn)qnw$. By definition, $\text{UPDATET}\lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon$ λ-converts into $\lambda u_i v_i.\, \sim_\gamma uv \wedge (\lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon u \longleftrightarrow \lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon v)$, which by the induction hypothesis on φ and the assumptions corresponds to the updated accessibility relation $\sim_{\varphi!} = \{(u, v) \in \sim \mid M, u \models \varphi \text{ iff } M, v \models \varphi\}$. By this, the assumptions and the induction hypothesis on ψ we have $[\![\lfloor\psi\rfloor(\text{UPDATET}\lfloor\varphi\rfloor \sim_\gamma \approx_\gamma \boldsymbol{N}_\upsilon) \approx_\gamma \boldsymbol{N}_\upsilon X]\!]^{\mathcal{M}, g[w/X_i]} = T$ equivalent to $M_{\varphi!}, w \models \psi$.

The case for $[\varphi?]\psi$ follows analogously. □

Theorem 2 (Faithfulness: LRK models)**.** *Suppose that the requirements given for Lemma 1 are satisfied for all considered LRK models M and HOL models $\mathcal{M}$. Furthermore, let* Ax *be a set of HOL axioms postulating the relation symbols $\sim_\gamma$ and $\approx_\gamma$ (the HOL counterparts of $\sim$ and $\approx$) to denote equivalence relations, and $\boldsymbol{N}_\upsilon$ (the HOL counterpart of N) to satisfy the condition $\forall w_i.\forall U_\sigma.\, (nwU \longrightarrow Uw)$. Then, we have: $\models^{LRK} \varphi$ iff* Ax $\models^{HOL} \lfloor\varphi\rfloor$

Proof. Corollary of Lemma 1; the additional LRK-conditions on $\sim$, $\approx$ and N are enforced by the axioms Ax postulated in HOL. □

4.2 Encoding into Isabelle/HOL

The presented SSE of LRK in HOL has been encoded in the proof assistant *Isabelle/HOL* to enable automated reasoning with LRK for the first time.[3] All necessary types can be modeled in a straightforward way. We declare type i (using **typedecl** in Isabelle) to denote possible worlds and then introduce type aliases (using **type_synonym**) for σ, γ, υ and τ, as mentioned in Sect.4.1. Type bool represents (the bivalent set of) truth values. Type σ is the type for formulas independent from a context, and type τ relates formulas to their context, which consists of an accessibility and a neighborhood relation. γ is the type of the functions that are used to update the accessibility and neighborhood relations.

The relations $\approx$, $\sim$, and N are declared as infix constant symbols. $\approx$ defines the set of questions the receiver is allowed to know the answers to, whereas $\sim$ is the usual indistinguishability relation for the receiver. The ideal epistemic state of the receiver is captured via the relation N.

In order for $\approx$ and $\sim$ to be equivalence relations, we constrain them to be reflexive, transitive, and symmetric:

axiomatization where til_ref: "∀ w. w ∼ w" **and** til_sym: "∀ w v. w ∼ v ⟶ v ∼ w" **and** til_trans: "∀ w v u. w ∼ v ∧ v ∼ u ⟶ w ∼ u" **and** dtil_ref: "∀ w. w ≈ w " **and** dtil_sym: "∀ w v. w ≈ v ⟶ v ≈ w" **and** dtil_trans: "∀ w v u. w ≈ v ∧ v ≈ u ⟶ w ≈ u" **and** Nax: "∀ w. ∀ H. (N w H ⟶ H w)"

The axiom Nax constraints the neighborhood relation N by stating that all sets of worlds that are in the set of sets of worlds returned by (N w) contain w itself. This implies that none of the sets is empty.

The operator for atomic propositions ${}^A(\cdot)$ of type $\sigma\Rightarrow\tau$ checks whether an atom p holds in a given world w:

abbreviation atom::"$\sigma\Rightarrow\tau$" ("^{A}_") **where** "^{A}p ≡ λt q n w. p w"

[3] The full sources of our encoding can be found at http://logikey.org in subfolder LRK (https://github.com/cbenzmueller/LogiKEy/tree/master/LRK).

The formulas $\top$ and $\bot$ are defined represent truth and falsity (indendent of the given world):

abbreviation lrkTop::τ ("$\top$") **where** "$\top \equiv \lambda$t q n w. True"
abbreviation lrkBot::τ ("$\bot$") **where** "$\bot \equiv \lambda$t q n w. False"

We then define the Boolean connectives that are primitive in LRK. To distinguish between HOL connectives (e.g., $\neg$) and the lifted LRK connectives (e.g., $\neg_{\tau\Rightarrow\tau}$), we use boldface fonts:

abbreviation lrkNot::"$\tau\Rightarrow\tau$" ("$\neg$_")
where "$\neg\varphi \equiv \lambda$t q n w. $\neg(\varphi$ t q n w)"
abbreviation lrkImp::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("_$\rightarrow$_")
where "$\varphi\rightarrow\psi \equiv \lambda$t q n w. $(\varphi$ t q n w) $\longrightarrow (\psi$ t q n w)"

For convenience, we also define the remaining Boolean connectives as abbreviations.

abbreviation lrkAnd::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("_$\wedge$_")
where "$\varphi\wedge\psi \equiv \lambda$t q n w. $(\varphi$ t q n w) $\wedge$ $(\psi$ t q n w)"
abbreviation lrkOr::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("_ $\vee$ _")
where "$\varphi\vee\psi \equiv \lambda$t q n w. $(\varphi$ t q n w) $\vee$ $(\psi$ t q n w)"
abbreviation lrkEquiv::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("_$\leftrightarrow$_")
where "$\varphi\leftrightarrow\psi \equiv \lambda$t q n w. $(\varphi$ t q n w)$\longleftrightarrow (\psi$ t q n w)"

To build formulas in LRK, embeddings of the specific connectives of the logic are needed (see Sect. 3). Here, U is the universal modality, Rr represents the right to know of the receiver, Kr the knowledge of the receiver, Os the obligation of the sender to announce something truthfully, Q is a technical modality.

abbreviation lrkU::"$\tau\Rightarrow\tau$" ("U _")
where "U $\varphi \equiv \lambda$t q n w. $\forall$ v. φ t q n v"
abbreviation lrkQ::"$\tau\Rightarrow\tau$" ("Q _")
where "Q $\varphi \equiv \lambda$t q n w. $\forall$ v. (q w v) $\longrightarrow (\varphi$ t q n v)"
abbreviation lrkRr::"$\tau\Rightarrow\tau$" ("Rr _")
where "Rr $\varphi \equiv \lambda$t q n w. $\forall$ va. $\forall$ v. (q va v) $\longrightarrow$
$(\varphi$ t q n v) $\vee$ ($\forall$ v. q va v $\longrightarrow (\neg\varphi$ t q n v))"
abbreviation lrkKr::"$\tau\Rightarrow\tau$" ("Kr _")
where "Kr $\varphi \equiv \lambda$t q n w. $\forall$ v. (t w v) $\longrightarrow (\varphi$ t q n v)"
abbreviation lrkO::"$\tau\Rightarrow\tau$" ("Os _") **where**
"Os $\varphi \equiv \lambda$t q n w. $\forall$ H. (n w H) $\longrightarrow$
($\forall$ v. H v $\longrightarrow$ (t w v)) $\longrightarrow$ ($\forall$ i. H i $\longrightarrow (\varphi$ t q n i))"

Formulas $[\varphi?]\varphi$ and $[\varphi!]\varphi$ are defined using the dynamic updates mentioned above, and they are then used to encode $[r : \varphi?]\psi$ and $[s:\varphi!]\psi$:

abbreviation lrkQuestion::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("[_?]_")
where "$[\varphi?]\psi \equiv \lambda$t q n w. $(\psi$ t q (update_N φ n t q) w)"
abbreviation lrkExclamation::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("[_!]_")
where "$[\varphi!]\psi \equiv \lambda$t q n w. $(\psi$ (update_t φ n t q) q n w)"
abbreviation lrkAfterQuestion::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("[r:_?]_")
where "$[r:\varphi?]\psi \equiv$ $(((\neg$ Rr $\varphi) \wedge \psi) \vee ($ (Rr $\varphi) \wedge [\varphi?]\psi$))"
abbreviation lrkAfterExclamation::"$\tau\Rightarrow\tau\Rightarrow\tau$" ("[s:_!]_")
where "$[s:\varphi!]\psi \equiv \varphi \rightarrow [\varphi!]\psi$"

Update of the accessibility relation and neighborhood function is:

abbreviation update_t::"$\tau\Rightarrow\upsilon\Rightarrow\gamma\Rightarrow\gamma\Rightarrow\gamma$"
where "update_t φ n t q $\equiv \lambda$u v. t u v $\wedge$ (φ t q n u $\longleftrightarrow \varphi$ t q n v)"
abbreviation update_N::"$\tau\Rightarrow\upsilon\Rightarrow\gamma\Rightarrow\gamma\Rightarrow\upsilon$"
where "update_N φ n t q $\equiv \lambda$w.λX. n w X $\wedge$
((($\forall$ v. (X v $\longrightarrow (\varphi$ t q n v))) $\vee$ ($\forall$ v. (X v $\longrightarrow$ (($\neg\varphi$) t q n v)))))"

Finally, the notion of validity is embedded, and the initial accessibility relations $\approx$, $\sim$ and neighborhood function N are used for this. Their value is then updated when the operators $[\varphi?]\varphi$ and $[\varphi!]\varphi$ are used in the evaluated formula.

abbreviation lrkValidLocal::"$\tau\Rightarrow$i$\Rightarrow$bool" ("$\lfloor$_$\rfloor$_")
where "$\lfloor\varphi\rfloor_w \equiv \varphi$ til dtil N w"
abbreviation lrkValidGlobal::"$\tau\Rightarrow$bool" ("$\lfloor$_$\rfloor$")
where "$\lfloor\varphi\rfloor \equiv \forall$ w. φ til dtil N w"

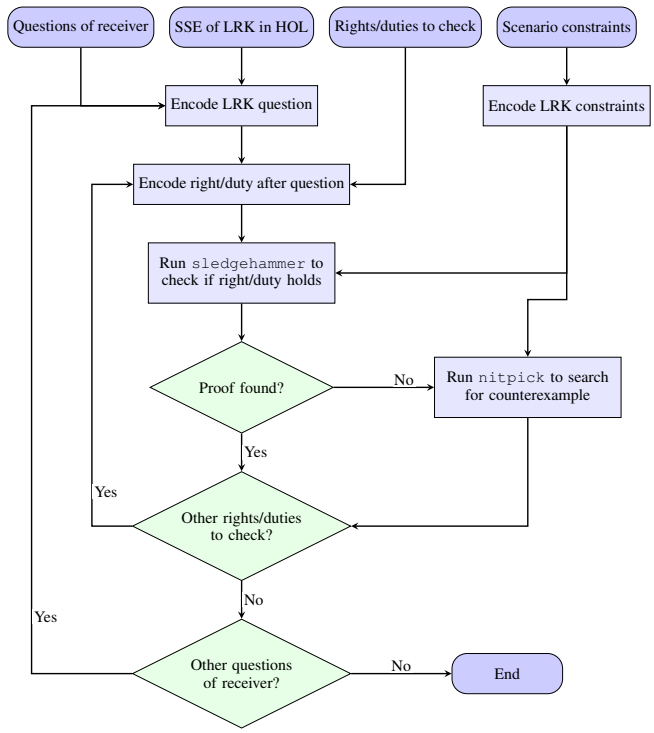

Figure 2. Workflow diagram

5 Case Study: GDPR Scenario

In a first experiment, the SSE of LRK in HOL has been tested by checking that relevant propositions, semantic results and the axioms postulated for LRK in [42] are actually implied in the embedded logic as expected; see the online repository mentioned in Footnote 3.

A second experiment demonstrates the use of the embedding on the GDPR example introduced in Sect. 1. This example has been represented in LRK and encoded in *Isabelle/HOL* using the shallowly embedded logic. Recall that according to the GDPR, a data subject has the right to access some specific information: for instance, whether her personal data is being processed, and if so, with what purpose [26]. However, this right should not adversely affect the rights or freedoms of others, such as the copyright protecting the software used to process data.

Consider the following situation: a data controller processes the personal data of a data subject with the purpose of optimizing advertisement placement. In doing so, the controller uses proprietary software using mechanism X. When the data subject asks the controller whether her personal data is processed and if so, whether this happens in order to optimize what advertisement she sees on the website, the questions create the controller's duty to answer them. However, due to the software being proprietary, the controller has a duty to not tell whether it applies mechanism X. Hence when the data subject asks about it, according to the GDPR, this question does not establish the controller's duty to tell.

The workflow is portrayed in Fig. 2: for each question of the data subject ds to the data controller dc, we check which rights or duties hold using the SSE of LRK in HOL on the encoded scenario. Results of proven and disproven rights and duties after each new question are reported in Table 1. The proposition "the data is processed by dc" is represented by propositional variable pa; "the data is processed with the purpose of optimizing advertisement placement" is represented by pb; and the proposition "the software used by dc uses mechanism

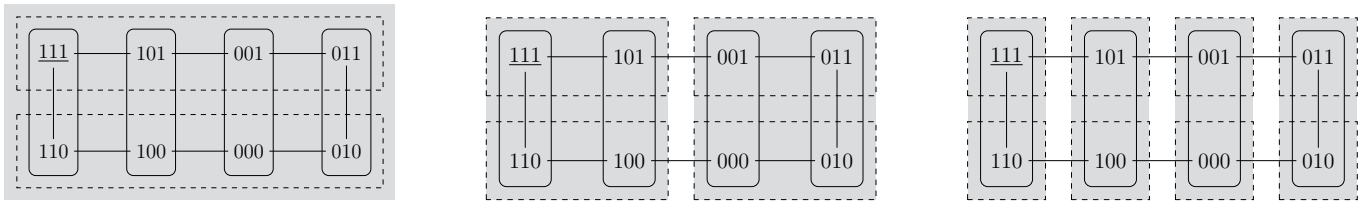

Figure 3. Initial model (left) — pa? Updated Model (middle) — pa?pb? Updated Model (right)

X" is represented by pc. The situation is displayed in Fig. 3 (left): states are identified by 3-bits sequences, corresponding to the value of pa, pb, and pc, respectively. We are in state 111, where all three propositions are true. The indistinguishability relation $\sim$ is indicated by the straight line, whereas the equivalence relation $\approx$ (the right of ds to know whether pa, and whether pb) is pictured by the rectangles with the rounded corners. Finally, for every state s, neighborhood function Ns contains every subset H such that: (1) H contains s itself; (2) H is contained in a shaded area (corresponding to what the dc is obligated to announce); and (3) H is not contained in one of the dashed rectangles (ds is prohibited to know pc).

We start to encode the scenario by declaring constants of type σ for the three propositions and constants of type i for the eight different worlds, each representing one combination of values of pa, pb, and pc. The worlds are named with letters from a to h, where a corresponds to assignment $\{pa = 0,\ pb = 0,\ pc = 0\}$ and h corresponds to $\{pa = 1,\ pb = 1,\ pc = 1\}$.

consts pa::σ pb::σ pc::σ a::i b::i c::i d::i e::i f::i g::i h::i

Next, this initial model is represented in the embedded logic. This includes ensuring the distinctness of the eight worlds, as well as declaring the truth values of each of the propositions in each possible world (axioms aa–ah). Moreover, we define the elements that are included in the accessibility relations $\sim$ and $\approx$ (axioms $til1$–$til8$, $dtil1$–$dtil10$) and neighborhood function (axiom Neigh). The $\sim$ relation initially contains all possible pairs of worlds. Since $\sim$ is reflexive, transitive, and symmetric, not all pairs of worlds need to be listed explicitly. To define N and $\approx$, we rely on what ds is allowed to know, or forbidden to know.

axiomatization where

a0:"(∀ x. (x=a∨x=b∨x=c∨x=d∨x=e∨x=f∨x=g∨x=h))" **and**
a00:"(∀ x. (x=a ⟶ ¬(x=b∨x=c∨x=d∨x=e∨x=f∨x=g∨x=h)))" **and**
a01:"(∀ x. (x=b ⟶ ¬(x=a∨x=c∨x=d∨x=e∨x=f∨x=g∨x=h)))" **and**
a02:"(∀ x. (x=c ⟶ ¬(x=a∨x=b∨x=d∨x=e∨x=f∨x=g∨x=h)))" **and**
a03:"(∀ x. (x=d ⟶ ¬(x=a∨x=b∨x=c∨x=e∨x=f∨x=g∨x=h)))" **and**
a04:"(∀ x. (x=e ⟶ ¬(x=a∨x=b∨x=c∨x=d∨x=f∨x=g∨x=h)))" **and**
a05:"(∀ x. (x=f ⟶ ¬(x=a∨x=b∨x=c∨x=d∨x=e∨x=g∨x=h)))" **and**
a06:"(∀ x. (x=g ⟶ ¬(x=a∨x=b∨x=c∨x=d∨x=e∨x=f∨x=h)))" **and**
a07:"(∀ x. (x=h ⟶ ¬(x=a∨x=b∨x=c∨x=d∨x=e∨x=f∨x=g)))" **and**
aa:"¬⌊^{A}pa⌋$_a$ ∧ ¬⌊^{A}pb⌋$_a$ ∧ ¬⌊^{A}pc⌋$_a$" **and**
ab:"¬⌊^{A}pa⌋$_b$ ∧ ¬⌊^{A}pb⌋$_b$ ∧ ⌊^{A}pc⌋$_b$" **and**
ac:"¬⌊^{A}pa⌋$_c$ ∧ ⌊^{A}pb⌋$_c$ ∧ ¬⌊^{A}pc⌋$_c$" **and**
ad:"¬⌊^{A}pa⌋$_d$ ∧ ⌊^{A}pb⌋$_d$ ∧ ⌊^{A}pc⌋$_d$" **and**
ae:"⌊^{A}pa⌋$_e$ ∧ ¬⌊^{A}pb⌋$_e$ ∧ ¬⌊^{A}pc⌋$_e$" **and**
af:"⌊^{A}pa⌋$_f$ ∧ ¬⌊^{A}pb⌋$_f$ ∧ ⌊^{A}pc⌋$_f$" **and**
ag:"⌊^{A}pa⌋$_g$ ∧ ⌊^{A}pb⌋$_g$ ∧ ¬⌊^{A}pc⌋$_g$" **and**
ah:"⌊^{A}pa⌋$_h$ ∧ ⌊^{A}pb⌋$_h$ ∧ ⌊^{A}pc⌋$_h$"**and**
til1:"a ∼ b" **and** til2:"b ∼ c" **and** til3:"c ∼ d" **and**
til4:"d ∼ e" **and** til5:"e ∼ f" **and** til6:"f ∼ g" **and**
til7:"g ∼ h" **and** til8:"h ∼ a" **and** dtil1:"a ≈ b" **and**
dtil2:"c ≈ d" **and** dtil3:"e ≈ f" **and** dtil4:"g ≈ h" **and**
dtil5:"¬(a ≈ c)" **and** dtil6: "¬(a ≈ e)" **and** dtil7:"¬(a ≈ g)" **and**
dtil8:"¬(c ≈ e)" **and** dtil9:"¬(c ≈ g)" **and** dtil10: "¬(e ≈ g)" **and**
Neigh: "∀ w. ∀ H. ((N w) H ⟷
(H w ∧(¬ (check_subset H (λx. (x=a ∨ x=c ∨ x=e ∨ x=g)))) ∧
(¬ (check_subset H (λx. (x=b ∨ x=d ∨ x=f ∨ x=h))))))"

Questions	Proven rights/duties	Disproven rights/duties
–	$R_r pa$ $R_r pb$	$R_r pc$ $\mathbb{O}_s pa$ $\mathbb{O}_s pb$ $\mathbb{O}_s pc$
pa?	$[r:pa?]\ R_r pa$ $[r:pa?]\ R_r pb$ $[r:pa?]\ \mathbb{O}_s pa$	$[r:pa?]\ R_r pc$ $[r:pa?]\ \mathbb{O}_s pb$ $[r:pa?]\ \mathbb{O}_s pc$
pa?pb?	$[r:pb?]([r:pa?]\ R_r pa)$ $[r:pb?]([r:pa?]\ R_r pb)$ $[r:pb?]([r:pa?]\ \mathbb{O}_s pa)$ $[r:pb?]([r:pa?]\ \mathbb{O}_s pb)$	$[r:pb?]([r:pa?]\ R_r pc)$ $[r:pb?]([r:pa?]\ \mathbb{O}_s pc)$
pa?pc?	$[r:pc?]([r:pa?]\ R_r pa)$ $[r:pc?]([r:pa?]\ R_r pb)$ $[r:pc?]([r:pa?]\ \mathbb{O}_s pa)$	$[r:pc?]([r:pa?]\ R_r pc)$ $[r:pc?]([r:pa?]\ \mathbb{O}_s pb)$ $[r:pc?]([r:pa?]\ \mathbb{O}_s pc)$

Table 1. Proven and disproven rights/duties at each new question

In the initial model (Fig. 3 (left); Table 1 (first row)), the following LRK literals for rights hold in world h: $R_r\ pa$ (ds has the right to know whether pa) and $R_r\ pb$ (ds has the right to know whether pb), while $R_r\ pc$ does not hold (ds does not have the right to know pc). In the encoded SSE, we prove that $\lfloor Rr^A pa \rfloor_h$ and $\lfloor Rr^A pb \rfloor_h$ hold using `sledgehammer`, and using `nitpick` we find counterexamples for $\lfloor Rr^A pc \rfloor_h$, $\lfloor Os^A pa \rfloor_h$, $\lfloor Os^A pb \rfloor_h$ and $\lfloor Os^A pc \rfloor_h$ (dc has no obligation yet to announce anything).

Next, ds asks whether pa holds and we show that the updated model changes dynamically as expected; see Fig. 3 (middle) and Table 1 (second row). In the $pa?$ updated model, ds (still) has the right to know whether pa holds and now it is obligatory for dc to announce it. Also, ds (still) has the right to know whether pb holds, but not whether pc holds. In the encoded SSE, we use `sledgehammer` to prove that $\lfloor [r{:}^A pa?]\ Rr^A pa \rfloor_h$, $\lfloor [r{:}^A pa?]\ Rr^A pb \rfloor_h$, and $\lfloor [r{:}^A pa?]\ Os^A pa \rfloor_h$ hold.

Then, if now ds asks whether pb holds, the model is again updated; see Fig. 3 (right) and Table 1 (third row). In the $pb?$ (after $pa?$) updated model, ds still has the right to know pa and pb, and no right to know pc. The obligation for dc to announce pa remains, and now it is also obligatory for dc to announce pb. We find proofs for these formulas to hold in the encoded SSE: $\lfloor [r{:}^A pb?]\ \big([r{:}^A pa?]\ Rr^A pa\big) \rfloor_h$, $\lfloor [r{:}^A pb?]\ \big([r{:}^A pa?]\ Rr^A pb\big) \rfloor_h$, $\lfloor [r{:}^A pb?]\ \big([r{:}^A pa?]\ Os^A pa\big) \rfloor_h$, $\lfloor [r{:}^A pb?]\ \big([r{:}^A pa?]\ Os^A pb\big) \rfloor_h$.

However, if after asking whether pa holds, ds demands to know also pc, no obligation for dc to announce pc arises, since ds does not have the right to know pc. Hence, after this question the model is still as in Fig. 3 (middle); see also Table 1 (fourth row). We

show that asking whether pc holds does not result in an obligation for dc to announce it, while the right of ds to know pa and pb, but not pc, as well as the obligation for dc to announce pa remain. As expected, the LRK formula $[r : pc?](\langle r : pa? \rangle \mathbb{O}_s pc)$ stating that after such questions dc is obligated to announce pc is disproven: in the encoded SSE `nitpick` provides a counterexample to $\left\lfloor [r{:}^A pc?] \left(\langle r{:}^A pa? \rangle O s^A pc \right) \right\rfloor_h$, where $N\ h = \{g, h\}$. To understand this, notice that in LRK the formula unfolds to $(\neg R_r\ pc) \wedge ([r : pa?](\mathbb{O}_s pc)) \vee ((R_r\ pc) \wedge ([pc?]([r : pa?](\mathbb{O}_s pc))))$. Since ds does not have the right to know pc, $\neg R_r\ pc$ holds and we consider the first part of the disjunction $(\neg R_r\ pc) \wedge ([r : pa?](\mathbb{O}_s pc))$. Formula $([r : pa?](\mathbb{O}_s pc))$ corresponds to $((\neg R_r\ pa) \wedge (\mathbb{O}_s pc)) \vee (R_r\ pa \wedge [pa?]\mathbb{O}_s pc)$. $R_r\ pa$ holds, so we only need to check that $[pa?]\mathbb{O}_s pc)$ is falsified by the provided counterexample with $N\ h = \{g, h\}$. Notice that $\mathbb{O}_s pc$ would imply that since $g \sim h$, then pc holds for all worlds in $\{g, h\}$. However, pc does not hold in world g, which corresponds to the evaluation $pa = 1,\ pb = 1,\ pc = 0$. Consequently, the original LRK formula $[r : pc?](\langle r : pa? \rangle \mathbb{O}_s pc)$ does not hold in the current world.

6 Related Work and Conclusion

The right to know is a form of legal right called an *epistemic right* [62]. Implementing logics for epistemic rights is increasingly important for AI systems, as they now frequently operate in domains where access to information is governed by legal, ethical, or institutional norms [59]: an intelligent agent must not only manage and reason about knowledge but also respect who is entitled to know what and when, for example when handling health data [28]. Studies on the logical analysis of legal rights include [37, 44, 45, 36], with more recent works exploring power-type rights in [46, 58, 25]. These works focus on general rights, while a specific logic for epistemic rights is LRK, the Dynamic Logic of the Right to Know [42], that we mechanized in this paper. One reason such a specific logic is needed is that certain reasoning patterns are valid for epistemic rights but not for general rights.[4] As noted by [25], there are two main approaches to formalizing legal power: the first [37, 44] defines power in terms of obligations, permissions, and actions; while in the second approach [45, 36] power cannot be reduced to static normative positions, aligning with Hohfeld's original distinction (see [46]). In LRK, the second approach is followed, as the power to know is treated as a notion distinct from obligatory announcements.

Dynamic Epistemic Logic (DEL) is is a family of modal logics of model change, and the type of public announcements used in LRK, removing links and not states, originates from [27] and can be represented as arrow updates [38]. LRK is also related to logics of questions [32, 60] and inquisitive semantics [22]. Other works that use deontic logic for epistemic actions include [4, 2, 43].

The work presented in this paper follows the LogiKEy methodology outlined in [15, 14]. This approach has focused on using SSEs in HOL [6] to enable knowledge engineers to explore and experiment with different object logics and their combinations, integrating both general and domain-specific knowledge. Other works based on LogiKEy are for example [17, 53], and work in [52, 51, 16, 18, 13] specifically focuses on SSEs of deontic logics. Work in [11] presents an SSE of Public Announcement Logic (PAL), where the embedded formulas depend only on evaluation domains, while additional parameters that are more complex to maintain and update are needed for LRK. Another contribution of this paper has been to show that the LogiKEy approach scales well to support the embedding of a non-trivial logic like LRK. Of course, the automation of non-trivial object logics such as LRK with HOL argumentation tools will eventually reach its limits, while specialized LRK provers may ultimately still be responsive. However, it should be noted that developing powerful native LRK provers and model finders is a very ambitious task. And if such tools eventually exist, they can also be integrated into our LogiKEy framework as additional external tools.

The SSE approach has been specifically employed for the automation of quantified modal logics [10]. A variety of other methods are available for automated reasoning in modal logics. For propositional modal logic it is possible is to use encodings to SAT [57] or to SMT solvers [1]. Authors have provided resolution calculi for modal logics [47, 49] and corresponding model construction algorithms [35] that are implemented in the resolution-based multimodal logic prover KSP [48]. Algorithms for unification in modal logics exist [3], as well as syntactic abstraction methods [24] to reason about first-order modal logics. Authors in [61] studied how DEL models can be faithfully represented as knowledge structures for symbolic model checking with an approach based on Binary Decision Diagram reasoning.

There are also a number of tools that can be used for legal reasoning and compliance checking, for instance implementations of Answer Set Programming (ASP) [40], its extension implemented in DLV [41], implementations of defeasible deontic logic [39, 23], or PROLEG (PROlog based LEGal reasoning support system) [56]. A survey of automated reasoners for compliance checking can be found in [55]. However, none of these can explicitly represent epistemic rights such as the right to know.

Compared to other approaches for automating a logic, the SSE approach offers several advantages: (1) it handles all conversions directly within HOL, making the encoding concise and readable; (2) it naturally extends from propositional to first-order and higher-order logic, enabling more expressive reasoning; (3) it supports deep and shallow embeddings [19] and mechanized faithfulness proofs between them; and (4) it benefits from state-of-the-art theorem provers and model finders integrated with *Isabelle/HOL* [20]. In general, this approach may have a performance loss compared to domain-specific solutions, but not always [30].

To the best of our knowledge, this work presents the first automation of a logic capable of reasoning about the right to know. The SSE approach enables us to represent and experiment with the right to know by effectively handling model updates, as shown in the discussed GDPR use case. Directions for future research include applying the embedded logic to represent other use cases involving epistemic rights, and extending the embedding to multi-agent scenarios, with multiple sender and receiver agents, possibly with overlapping roles. Another important direction that is often required in practical applications is to combine LRK with other logics, such as doxastic or temporal logics, for more expressive reasoning: the meta-logical nature of the LogiKEy framework supports such combinations.

Acknowledgements

This work was supported by the Luxembourg National Research Fund (FNR) through the project Logical methods for Deontic Explanations (INTER/DFG/23/17415164/LODEX) and the project Deontic Logic for Epistemic Rights (OPEN O20/14776480). We also thank the anonymous reviewers for their fruitful feedback.

[4] As observed in [42], consider, e.g., $R_r p \rightarrow R_r \neg p$, which is valid in LRK, but would be counterintuitive for operator $R_r p$ interpreted as a general power-right.

References

[1] C. Areces, P. Fontaine, and S. Merz. Modal satisfiability via SMT solving. In R. Hennicker and R. de Nicola, editors, *Software, Services, and Systems*, volume 8950 of *LNCS*, pages 30–45. Springer, 2015.

[2] G. Aucher, G. Boella, and L. van der Torre. A dynamic logic for privacy compliance. *Artificial Intelligence and Law*, 19(2):187, 2011.

[3] F. Baader and S. Ghilardi. Unification in modal and description logics. *Logic Journal of the IGPL*, 19(6):705–730, 2010.

[4] P. Balbiani and P. Seban. Reasoning about permitted announcements. *Journal of Philosophical Logic*, 40(4):445–472, 2011.

[5] C. Benzmüller. Cut-elimination for quantified conditional logic. *Journal of Philosophical Logic*, 46(3):333–353, 2017.

[6] C. Benzmüller. Universal (meta-)logical reasoning: Recent successes. *Science of Computer Programming*, 172:48–62, 2019.

[7] C. Benzmüller. Universal (meta-)logical reasoning: The wise men puzzle (Isabelle/HOL Dataset). *Data in Brief*, 24(103823):1–5, 2019.

[8] C. Benzmüller and P. Andrews. Church's Type Theory. In E. N. Zalta and U. Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Stanford University, Spring 2024 edition, 2024.

[9] C. Benzmüller and D. Miller. Automation of higher-order logic. In D. M. Gabbay et al., editors, *Handbook of the History of Logic, Volume 9 — Computational Logic*, pages 215–254. North Holland, 2014.

[10] C. Benzmüller and L. C. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis*, 7(1):7–20, 2013.

[11] C. Benzmüller and S. Reiche. Automating public announcement logic with relativized common knowledge as a fragment of HOL in LogiKEy. *Journal of Logic and Computation*, 33(6):1243–1269, 2023.

[12] C. Benzmüller, C. Brown, and M. Kohlhase. Higher-order semantics and extensionality. *Journal of Symbolic Logic*, 69(4):1027–1088, 2004.

[13] C. Benzmüller, A. Farjami, and X. Parent. Åqvist's dyadic deontic logic E in HOL. *FLAP*, 6(5):733–755, 2019.

[14] C. Benzmüller, A. Farjami, D. Fuenmayor, et al. LogiKEy workbench: Deontic logics, logic combinations and expressive ethical and legal reasoning (Isabelle/HOL dataset). *Data in Brief*, 33(106409):1–10, 2020.

[15] C. Benzmüller, X. Parent, and L. van der Torre. Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. *Artificial Intelligence*, 287:103348, 2020.

[16] C. Benzmüller, A. Farjami, and X. Parent. Dyadic deontic logic in HOL: Faithful embedding and meta-theoretical experiments. In S. Rahman et al., editors, *New Developments in Legal Reasoning and Logic: From Ancient Law to Modern Legal Systems*, volume 23 of *Logic, Argumentation & Reasoning*. Springer Nature, 2022.

[17] C. Benzmüller, D. Fuenmayor, and B. Lomfeld. Modelling value-oriented legal reasoning in LogiKEy. *Logics*, 2(1):31–78, 2024.

[18] C. Benzmüller et al. I/O logic in HOL. *FLAP*, 6(5):715–732, 2019.

[19] C. Benzmüller. Faithful logic embeddings in HOL — deep and shallow. In *CADE-30*, volume 15943 of *LNCS*, pages 280–302. Springer, 2025.

[20] J. Blanchette, S. Böhme, and L. Paulson. Extending Sledgehammer with SMT solvers. *Journal of Automated Reasoning*, 51:116–130, 2011.

[21] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.

[22] I. A. Ciardelli and F. Roelofsen. Inquisitive dynamic epistemic logic. *Synthese*, 192:1643–1687, 2015.

[23] M. Cristani et al. The architecture of a reasoning system for defeasible deontic logic. *Procedia Comput. Sci.*, 225:4214–4224, 2023.

[24] D. Doligez, J. Kriener, L. Lamport, et al. Coalescing for reasoning in first-order modal logics. In C. Benzmüller et al., editors, *ARQNL 2014*, volume 33 of *EPiC Series in Computing*, pages 1–16. EasyChair, 2014.

[25] H. Dong and O. Roy. Dynamic logic of legal competences. *J. Log. Lang. Inf.*, 30(4):701–724, 2021.

[26] European Parliament and Council of the European Union. Regulation (EU) 2016/679 (GDPR), 2016.

[27] J. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6:147–169, 1997.

[28] S. Gerke, T. Minssen, and G. Cohen. Ethical and legal challenges of artificial intelligence-driven healthcare. In A. Bohr et al., editors, *Artificial Intelligence in Healthcare*, pages 295–336. Academic Press, 2020.

[29] J. Gibbons and N. Wu. Folding domain-specific languages: deep and shallow embeddings (functional pearl). In J. Jeuring and M. M. T. Chakravarty, editors, *ICFP 2014*, pages 339–347. ACM, 2014.

[30] T. Gleißner, A. Steen, and C. Benzmüller. Theorem provers for every normal modal logic. In T. Eiter et al., editors, *Proc. of LPAR-21*, volume 46 of *EPiC Series in Computing*, pages 14–30. EasyChair, 2017.

[31] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.

[32] J. Groenendijk and M. Stokhof. Chapter 19 - questions. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 1055–1124. North-Holland, Amsterdam, 1997.

[33] L. Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15(2):81–91, 1950.

[34] W. N. Hohfeld. Fundamental legal conceptions applied in judicial reasoning. In W. W. Cook, editor, *Fundamental Legal Conceptions Applied in Judicial Reasoning and Other Legal Essays*, pages 23–64. New Haven: Yale University Press, 1923.

[35] U. Hustadt et al. Model construction for modal clauses. In *Automated Reasoning*, volume 14740 of *LNCS*, pages 3–23. Springer, 2024.

[36] A. J. I. Jones and M. Sergot. A formal characterisation of institutionalised power. *Log. J. IGPL*, 4(3):427–443, 1996.

[37] S. Kanger and H. Kanger. Rights and parliamentarism. *Theoria*, 32(2): 85–115, 1966.

[38] B. Kooi and B. Renne. Arrow update logic. *The Review of Symbolic Logic*, 4(4):536–559, 2011.

[39] H.-P. Lam and G. Governatori. The making of SPINdle. In G. Governatori et al., editors, *Rule Representation, Interchange and Reasoning on the Web*, number 5858 in LNCS, pages 315–322. Springer, 2009.

[40] N. Leone and F. Ricca. Answer set programming: A tour from the basics to advanced development tools and industrial applications. In *Reasoning Web*, volume 9203 of *LNCS*, pages 308–326. Springer, 2015.

[41] N. Leone, G. Pfeifer, et al. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.*, 7(3):499–562, 2006.

[42] X. Li and R. Markovich. A dynamic logic of the right to know. *FLAP*, 12(2):221–250, 2025.

[43] X. Li, D. Gabbay, and R. Markovich. Dynamic Deontic Logic for Permitted Announcements. In *Proc. of the 19th Intl. Conf. on Principles of Knowledge Representation and Reasoning*, pages 226–235, 2022.

[44] L. Lindahl. *Position and change: A study in law and logic*. Synthese Library. Springer Dordrecht, 1 edition, 1977.

[45] D. Makinson. On the formal representation of rights relations. *Journal of Philosophical Logic*, 15(4):403–425, 1986.

[46] R. Markovich. Understanding Hohfeld and Formalizing Legal Rights: the Hohfeldian Conceptions and Their Conditional Consequences. *Studia Logica*, 108, 2020.

[47] C. Nalon and C. Dixon. Clausal resolution for normal modal logics. *Journal of Algorithms*, 62(3):117–134, 2007.

[48] C. Nalon, U. Hustadt, and C. Dixon. A resolution-based theorem prover for $\mathsf{K}_n$kn: Architecture, refinements, strategies and experiments. *Journal of Automated Reasoning*, 64, 2018.

[49] C. Nalon, C. Dixon, and U. Hustadt. Modal resolution: Proofs, layers, and refinements. *ACM Trans. Comput. Logic*, 20(4), 2019.

[50] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, Berlin, Heidelberg, 2002.

[51] X. Parent and C. Benzmüller. Automated verification of deontic correspondences in Isabelle/HOL – first results. In *Proc. of ARQNL 2022*, volume 3326, pages 92–108. CEUR-WS.org, 2023.

[52] X. Parent and C. Benzmüller. Conditional normative reasoning as a fragment of HOL. *J. Appl. Non-Class. Log.*, 34:561–592, 2024.

[53] L. Pasetto and C. Benzmüller. Implementing the Fatio protocol for multi-agent argumentation in LogiKEy. In *Proc. of ARQNL 2024*, volume CEUR-3875, pages 38–47. CEUR-WS.org, 2024.

[54] F. Pfenning and C. Elliott. Higher-order abstract syntax. In R. L. Wexelblat, editor, *Proc. of the ACM SIGPLAN'88 Conf. on Progr. Language Design and Implementation (PLDI)*, pages 199–208. ACM, 1988.

[55] L. Robaldo et al. Compliance checking on first-order knowledge with conflicting and compensatory norms: a comparison among currently available technologies. *Artif. Intell. Law*, 32(2):505–555, 2023.

[56] K. Satoh, K. Asai, T. Kogawa, et al. Proleg: An implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In T. Onada et al., editors, *New Frontiers in Artificial Intelligence*, pages 153–164. Springer Berlin Heidelberg, 2011.

[57] R. Sebastiani and M. Vescovi. Automated reasoning in modal and description logics via sat encoding: the case study of k(m)/alc-satisfiability. *J. Artif. Intell. Res.*, 35:343–389, 2009.

[58] G. Sileno and M. Pascucci. Disentangling deontic positions and abilities: a modal analysis. In F. Calimeri et al., editors, *Proc. of the 35th Edition of the Italian Conf. on Comp. Logic (CILC)*, pages 36–50, 2020.

[59] UNESCO. Recommendation on the ethics of ai. https://unesdoc.unesco.org/ark:/48223/pf0000381137, 2022. Accessed: 2025-05-06.

[60] J. van Benthem and Ş. Minică. Toward a dynamic logic of questions. *Journal of Philosophical Logic*, 41(4):633–669, 2012.

[61] J. van Benthem et al. Symbolic model checking for dynamic epistemic logic–S5 and beyond. *J.Log.Comp.*, 28(2):367–402, 2018.

[62] L. Watson. *The Right to Know: Epistemic Rights and Why We Need Them*. Routledge Focus on Philosophy. Routledge, 1 edition, 2021.