ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Norm-based deontic logic for access control, some computational results

Xin Sun [a,b,*], Xishun Zhao [a], Livio Robaldo [c]

[a] Institute of Logic and Cognition, Sun Yat-sen University, Guangzhou, China

[b] Department of Foundations of Computer Science, Faculty of Philosophy, The John Paul II Catholic University of Lublin, Lublin, Poland

[c] University of Luxembourg, Luxembourg

ARTICLE INFO

Article history:
Received 11 September 2016
Received in revised form
20 December 2016
Accepted 22 January 2017
Available online xxxx

Keywords:
Deontic logic
Access control
Computational complexity

ABSTRACT

In this paper we study the complexity of deontic logics grounded on *norm-based* semantics and apply norm-based deontic logic to access control. Four principal norm-based deontic logics have been proposed so far: imperative logic, input/output logic, deontic default logic and deontic defeasible logic. We present the readers that imperative logic is complete for the 2ed level of the polynomial hierarchy and deontic default logic is located in the 3ed level of the polynomial hierarchy. We then show how it is possible to impose restrictions to imperative logic such that the complexity goes down to be tractable, allowing the logic to be used in practical applications. We focus on a specific application: access control.

1. Introduction

Deontic logic is a formal study of normative reasoning and norms. In 1951, the philosopher and logician Georg von Wright wrote a paper called "Deontic Logic" [1], which subsequently became the name of the research area. Von Wright's deontic logic is exactly the same as the modal logic KD. Such logic is later called standard deontic logic (SDL). With the work of Meyer [2], deontic logic became a part of computer science. SDL has been a useful tool in the specification and reasoning of access control policies because key notions in access control such as permission, prohibition and obligation are exactly the subjects of SDL [3–5].

Deontic logic provides a mathematically rigorous language for modeling access control policies. The vagueness and ambiguity of informal language disappear in the formal language of deontic logic. Deontic logic is also associated with a sound and complete axiomatic characterization. The interpretation of the normative concepts is axiomatically constructed in deontic logic. As a consequence of completeness, the framework is guaranteed to be consistent. Without consistency, the move to the implementation level would be meaningless.

Different approaches of deontic logic, alternative to SDL, have been studied in the past 6 decades including imperative logic [6,7], dynamic deontic logic [2,8], deontic STIT logic [9,10], input/output logic [11], deontic default logic [12,13] and deontic defeasible logic [14,15]. Those results are summarized in the handbook of deontic logic [16,17]. In imperative logic, input/output logic, deontic default logic and deontic defeasible logic, norms are explicitly represented. The truth value of deontic propositions in those logics are explained not by some set of possible worlds, but with references to a set of given norms. Such a non-possible world semantics has been originally termed in Hansen [18] as 'norm-based semantics'. We then use norm-based deontic logic as a general term to refer input/output logic, imperative logic, deontic default logic and deontic defeasible logic and use deontic modal logic to refer those approaches which adopt possible world semantics such as SDL.

Norms are the first class citizens in norm-based deontic logic. Norms are everywhere in our daily life and also in access control. For example:

- You **should** drive on the right side.
- Alice is **permitted** to read file-1 on Mondays.
- Bob is **forbidden** to write on file-2.
- Carol is **obliged** to delete all related files when he finishes his task.

In general, we view norms as normative rules which are used to regulated agent's behavior. A norm is a rule in the sense that it contains both a **premise** and a **consequence**. The premise describes the situation in which it is triggered, while the consequence prescribes the demand of the norm. Norms are normative in the sense

* Corresponding author at: Institute of Logic and Cognition, Sun Yat-sen University, Guangzhou, China.
E-mail address: xin.sun.logic@gmail.com (X. Sun).

http://dx.doi.org/10.1016/j.future.2017.01.028

that they classify what is obligatory, permitted or forbidden. An access control policy is a set of norms defining which user is to be granted access to which resource under which circumstances. Compared to SDL, norm-based deontic logic has the following advantages.

1. Norm-based deontic logic solves the contrary-to-duty paradox. The contrary-to-duty paradox is the most notorious paradox in deontic logic. The original phrasing of the paradox requires a formalization of the following scenario in which the sentences are mutually consistent and logically independent [19].
   (a) It ought to be that John goes to help his neighbors.
   (b) It ought to be that if John goes to help his neighbors, then he tells them he is coming.
   (c) If John does not go to help his neighbors, then he ought not to tell them he is coming.
   (d) John does not go to help.

   But the formalization of the above scenario using SDL is either inconsistent or not logically independent. Being not able to solve the contrary-to-duty paradox is seen as one of the most serious limitations of SDL. The contrary-to-duty scenario is also found in access control and it is called the "violation of obligation" in Benferhat et al. [20] and corresponds to the policy of reaction to new intrusions in Cuppens [5].

   Norm-based deontic logic, on the other hand, gives consistent and logically independent formalization of the above scenario, therefore solves the contrary-to-duty paradox. In general, norm-based deontic logic provides correct prescriptions in situations where some norms are already violated [21].
2. Norm-based deontic logic offers a formal mechanism to deal with normative conflicts.

   Consider the following scenario taken from Hansen [7], which is sometimes called the 'order puzzle': before you go to a party, you become the recipient of various imperative sentences:
   (a) Your mother says: if you drink anything, then do not drive.
   (b) Your best friend says: if you go to the party, then you drive.
   (c) Some acquaintance says: if you go to the party, then have a drink with me.

   Assume mother is more important than best friend, who is more important than acquaintance. What will you do? Intuitively, you should obey your mother and your best friend, and hence do the driving and not accept your acquaintance's invitation. However, it is not so clear what formal mechanism could explain this reasoning. Handling normative conflicts is also an important issue in access control and is discussed in Benferhat et al. [20]. SDL is unable to handle such conflicting imperatives. On the other hand, norm-based deontic logic appears as suitable tools to formalize such reasoning.
3. Norm-based deontic logic characterizes various notions of permission.

   Permission is probably the most important notion in the specification of an access control policy [22,23]. Philosophically, it is common to distinguish between two kinds of permission: negative permission and positive permission. Negative permission is straightforward to describe: something is negatively permitted according to certain norms iff it is not prohibited by those norms. That is, iff there is no obligation to the contrary. Positive permission is more elusive. Intuitively, something is positively permitted according to certain norms iff it can be derived from those norms. But what exactly does "derive" mean? In mathematics we can derive theorems in a "straight" way or by contradiction. These two methods of derivation give two different notions of positive permission. Makinson and van der Torre [24] introduces these two types of positive permission as static and dynamic permission. Other notions of permission, such as permission as *exception*, have been studied in [25,26]. All these notions of permission are useful in access control and can be captured by norm-based deontic logics, while SDL is only able to capture negative permission.

The above advantages of norm-based deontic logic shows that comparing to SDL, norm-based deontic logic is a better tool to be applied in the specification and reasoning of access control policies. Among those existing norm-based deontic logics, imperative logic is the most suitable for access control because different notions of permission can be uniformly expressed in imperative logic.

For the existing norm-based deontic logics, the computational complexity of input/output logic and deontic defeasible logic is studied in [27,26]. In this paper, we study the complexity of imperative logic and deontic default logic (Sections 3 and 5.1). We show that both imperative logic and deontic default logic are decidable but computationally intractable. We then impose restrictions to obtain some tractable imperative logic such that we can practically apply them to access control (Section 4). For the sake of readability, we put all proofs in the Appendix.

2. Background: complexity theory

We assume the readers are familiar with notions like Turing machines and the complexity classes $\mathtt{P}$, $\mathtt{NP}$ and $\mathtt{coNP}$. Oracle Turing machines and some complexity classes related to oracle Turing machines will be used in this paper.

Definition 1 (*Oracle Turing Machine [28]*)**.** An oracle for a language L is a device that is capable of reporting whether any string w is a member of L. An oracle Turing machine M^L is a modified Turing machine that has the additional capability of querying an oracle. Whenever M^L writes a string on a special oracle tape it is informed whether that string is a member of L, in a single computation step.

$\mathtt{P}^{\mathtt{NP}}$ is the class of problems solvable by a deterministic polynomial time Turing machine with an $\mathtt{NP}$ oracle. $\mathtt{NP}^{\mathtt{NP}}$ is the class of problems solvable by a non-deterministic polynomial time Turing machine with an $\mathtt{NP}$ oracle. Σ_2^p is another name for $\mathtt{NP}^{\mathtt{NP}}$. Π_2^p is another name for $\mathtt{coNP}^{\mathtt{NP}}$. Δ_{i+1}^p is $\mathtt{P}^{\Sigma_i^p}$ and Σ_{i+1}^p is $\mathtt{NP}^{\Sigma_i^p}$.

3. Imperative logic

If some given norms come into conflict, the best an agent can be expected to do is to follow a maximal subset of those norms. Intuitively, a priority ordering over the norms can be helpful in resolving conflicts, but a formal resolution mechanism has been difficult to provide. In particular, reasoning about prioritized norms is overshadowed by problems such as the order puzzle that are not satisfactorily resolved by many existing approaches such as Brewka [29], Marek and Truszczynski [30]. Based on input/output logic [11], Hansen [7] develops prioritized imperative logic which overcomes those difficulties.

3.1. Input/output logic

Input/output logic takes its origin in the study of conditional norms. The basic idea is: norms are conceived as a deductive machine, like a black box which produces normative statements as output, when we feed it factual statements as input. In input/output logic, a norm is an ordered pair of formulas $(a, x) \in L_{\mathbb{P}} \times L_{\mathbb{P}}$, where $L_{\mathbb{P}}$ is the language of propositional logic build from the set of propositional atoms $\mathbb{P}$. There are two types of norms which are used in input/output logic, mandatory norms and permissive norms. A mandatory norm $(a, x) \in O$ is read as "given a, x is obligatory". A permissive norm $(a, x) \in P$ is read as "given a, x is permitted". Mandatory norms are called commands or imperatives in imperative logic, while permissive norms are called licenses or authorizations. To distinguish these two types of norms in notation, we may represent commands as $a \Rightarrow_o x$ and licenses as $a \Rightarrow_p x$. In this paper, we will however stick the notation

Please cite this article in press as: X. Sun, et al., Norm-based deontic logic for access control, some computational results, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.01.028

used in input/output logic because no confusion will arise in our presentation.

Commands O can be viewed as a function from $2^{L_{\mathbb{P}}}$ to $2^{L_{\mathbb{P}}}$ such that for a set A of formulas, $O(A) = \{x \in L_{\mathbb{P}} : (a, x) \in O$ for some $a \in A\}$. Makinson and van der Torre [11] define the semantics of input/output logic for commands as follows:

$out(O, A) = Cn(O(Cn(A)))$.

Here Cn is the classical consequence operator of propositional logic.[1] Intuitively, $x \in out(O, A)$ means that given a set of commands O and facts A, x is obligatory.

Example 1. Let $O = \{(a, x), (a \vee b, y)\}$. Put $A = \{a\}$.

A	$Cn(A)$	$O(Cn(A))$	$out(O, A)$
a	$Cn(a)$	$\{x, y\}$	$Cn(\{x, y\})$

The proof system of input/output logic is build on derivations of commands. We say that a command (a, x) is derivable from a set O iff (a, x) is in the least set that extends $O \cup \{(\top, \top)\}$ and is closed under a number of derivation rules. Here $\top$ is an arbitrary tautology. The following are the derivation rules which are used by Makinson and van der Torre [11] to construct the proof systems of input/output logic:

- SI (strengthening the input): from (a, x) to (b, x) whenever $b \vdash a$.
- WO (weakening the output): from (a, x) to (a, y) whenever $x \vdash y$.
- AND (conjunction of output): from (a, x) and (a, y) to $(a, x \wedge y)$.

$deriv(O)$ is the smallest set that extends $O \cup \{(\top, \top)\}$ and is closed under the rules of SI, WO and AND.

Example 2. Let $O = \{(a \vee b, x)\}$. We have $(b, x \vee y) \in deriv(O)$. Indeed:

1. $(a \vee b, x)$ Assumption
2. (b, x) 1, SI
3. $(b, x \vee y)$ 2, WO.

In Makinson and van der Torre [11], the following soundness and completeness theorem is given:

Theorem 1 (*[11]*)**.** *Given a set of commands O and formula a,*

$x \in out(O, \{a\})$ *iff* $(a, x) \in deriv(O)$.

3.2. Prioritized imperative logic

3.2.1. Reasoning about obligation

Hansen introduces *preferred obeyable maximal family* (*pomfamily*) to characterize those commands which are still functioning in a given situation where not all commands can be obeyed. Given a finite set of prioritized commands $O^> = (O, >)$, where $>$ is an irreflexive and transitive relation between commands. Here $(a, x) > (b, y)$ means (a, x) has higher priority than (b, y). A prioritization of $>$ is a strict linear order $\gtrsim$ such that if $(a, x) \gtrsim (b, y)$ then $(a, x) > (b, y)$ for all $(a, x), (b, y) \in O$. The materialization of O is $m(O) = \{a \to x : (a, x) \in O\}$, which transforms a command to a material implication.

Definition 2 (*Preferred Obeyable Maximal Family [7]*)**.** Given a finite set of prioritized commands $O^>$ and a set of formulas A. $O' \in pomfamily(O^>, A)$ if there is a $\gtrsim$ which is a prioritization of $>$ such that $O' = \bigcup_{i=0}^{n} O_i$ where O_i is constructed as follows. We list $\gtrsim$ by $(a_1, x_1), \ldots, (a_n, x_n)$ such that $(a_i, x_i) \gtrsim (a_{i+1}, x_{i+1})$.

1. $O_0 = \emptyset$,
2. $O_i = O_{i-1} \cup \{(a_i, x_i)\}$ if $A \cup m(O_{i-1} \cup \{(a_i, x_i)\})$ is consistent. Otherwise $O_i = O_{i-1}$.

Here note that every prioritization induces an element of the *pomfamily*. The result after resolving normative conflicts is characterized by the following output operator:

$$x \in out^p(O^>, A) \text{ iff } x \in \bigcap\{out(O', A) : O' \in pomfamily(O^>, A)\}.$$

Example 3 (*Order Puzzle Formalized*)**.** Let the three commands in the order puzzle be represented respectively by $(p_1, \neg p_2)$, (p_3, p_2), (p_3, p_1). Let the set of prioritized commands $O = \{(p_1, \neg p_2), (p_3, p_2), (p_3, p_1)\}$ and let $>$ be $(p_1, \neg p_2) > (p_3, p_2) > (p_3, p_1)$. Let the set of facts be $\{p_3\}$. Then $pomfamily(O^>, \{p_3\}) = \{(p_1, \neg p_2), (p_3, p_2)\}$ and $out^p(O^>, \{p_3\}) = Cn(p_2)$, which means you should do the drive but not drink.

A natural decision problem in the imperative logic framework is the obligation-checking problem: given a set of prioritized commands $O^>$, input A and target x, decide if $x \in out^p(O^>, A)$. The following theorem reveals the complexity of the obligation-checking problem.

Theorem 2. *Given $O^> = (O, >)$ where $>$ is irreflexive and transitive. Let A be a set of formulas and x be a formula. Deciding if $x \in out^p(O^>, A)$ is Π_2^p-complete.*

3.2.2. Reasoning about permission

Permission is probably the most important notion in the specification of an access control policy. Several notions of permission are introduced in norm-based deontic logic [24–26,18]. Hansen [18] gives a unified presentation of different notions of permission in the setting of imperative logic. However, the norms studied in Hansen [18] are unconditional norms. Combining the ideas from Makinson and van der Torre [24] and Hansen [18], we define negative and positive permission in the setting of conditional norms.

Definition 3. Given a normative system $N = (O, P, >)$ where P is a finite set of licenses and $> \subseteq (O \cup P) \times (O \cup P)$ is a priority relation over norms. Let A be a set of formulas representing factual statements.

1. $NegPerm(N, A) = \{x \in L_{\mathbb{P}} : \neg x \notin out^p(O^>, A)\}$.
2. $PosPerm(N, A) = \{x \in L_{\mathbb{P}} : x \in out^p((O \cup \{(a', x')\})^>, A)$, for some $(a', x') \in P\}$, if $P \neq \emptyset$. Otherwise $PosPerm(N, A) = out^p(O^>, A)$.

Here $(O \cup \{(a', x')\})^>$ is an ordered set with the set being $O \cup \{(a', x')\}$ and the ordering is obtained by restricting $>$ to $O \cup \{(a', x')\}$.

Intuitively, x is negatively permitted iff x is not forbidden. Since something is forbidden iff its negation is obligatory, x is not forbidden iff $\neg x$ is not obligatory. Licenses play no role in negative permission but they are treated like weak commands in positive permission. The only difference between licenses and commands in positive permission is that while the latter may be used jointly, the former may only be applied one by one. It is well acknowledged in the deontic logic literature that permission cannot be used jointly [24]. As an illustration of such difference, imaging a situation in which a man is permitted to date either one of two girls, but not both of them.

[1] In Makinson and van der Torre [11], this semantics is called *simple-minded* input/output logic. Several different input/output logics are introduced in Makinson and van der Torre [11] as well.

Please cite this article in press as: X. Sun, et al., Norm-based deontic logic for access control, some computational results, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.01.028

If a license (a, x) has higher priority than a command $(\top, \neg x)$, positive permission can be understood as *exception* which says although x is forbidden in general, there is an exception which allows x, when a is the case. Detailed discussions of exception as a notion of permission can be found in Stolpe [25] and Governatori et al. [26]. On the other hand, if a license $(\top, x)$ has lower priority than a command $(a, \neg x)$, positive permission can be understood as *access denial* which says although x is granted in general, such permission is canceled when a is the case.

Taking licenses into consideration, natural decision problems in the imperative logic framework includes the following: given a normative system $N = (O, P, >)$, a set of factual statements A and a target formula x,

- negative permission-checking: decide if $x \in NegPerm(N, A)$.
- positive permission-checking: decide if $x \in PosPerm(N, A)$.

The following theorem reveals the complexity of those permission-checking problems.

Theorem 3. *Given $N = (O, P, >)$ where $>$ is irreflexive and transitive. Let A be a set of formulas and x be a formula,*

- *to decide if $x \in NegPerm(N, A)$ is Σ_2^p-complete.*
- *to decide if $x \in PosPerm(N, A)$ is Π_2^p-complete.*

3.2.3. Reasoning about prohibition

Based on the two notions of permission, we now introduce two corresponding notions of prohibition: explicit prohibition and implicit prohibition.

Definition 4. Given a normative system $N = (O, P, >)$, a set of input A and a target x,

1. $ExProhi(N, A) = \{x \in L_{\mathbb{P}} : \neg x \in out^p(O^>, A)\}$.
2. $ImProhi(N, A) = \{x \in L_{\mathbb{P}} : x \notin PosPerm(N, A)\}$.

Intuitively, x is explicitly prohibited if $\neg x$ is obligatory. On the other hand, x is implicitly prohibited if x is not positively permitted, which means there is no explicit command or license supporting x.

Corollary 1. *Given $N = (O, P, >)$ where $>$ is irreflexive and transitive, a set of input A and a target x,*

- *to decide if $x \in ExProhi(N, A)$ is Π_2^p-complete.*
- *to decide if $x \in ImProhi(N, A)$ is Σ_2^p-complete.*

Now we have a bundle of notions about permission and prohibition. Those notions are suitable for different applications. In the case of access control, we believe positive permission and its complement, implicit prohibition, are most useful. The reason is that the function of licenses are ignored in negative permission and explicit prohibition whereas in positive permission and implicit prohibition licenses play an important role. Moreover, the notion of access denial, which can hardly be modeled by most existing access control logics, can be modeled using positive permission and implicit prohibition. We illustrate this point by formalizing a scenario taken from van Hertum et al. [31] and show how imperative logic allows to correctly handle statements whose goal is to deny access rights.

Example 4. Suppose Ann is a professor with control over a resource r, Bob is a Ph.D. student of Ann who needs access to r, and Charles is a postdoc of Ann supervising Bob. Ann wants to grant Bob access to r, but wants to grant Charles the right to deny Bob's access to r.

A natural way for Ann to do this is to issue the following access control policy: Let $N = (O, P, >)$, where $O = \{(\neg Bob\_approve, \neg access(Charles, r))\}$, $P = \{(\top, access(Charles, r))\}$, and the command has higher priority than the license.

Now we have $access(Charles, r) \in PosPerm(N, \{\top\})$ and $access(Charles, r) \in ImProhi(N, \{\neg Bob\_approve\})$, which means this access control policy has the effect that Charles has access to r unless Bob denies his access.

Another advantage of positive permission is that it also captures the idea of permission as *exception*, which is supported by Stolpe [25]. According to Stolpe, permission must "denote the elimination of a norm from a normative system". According to the definition of positive permission, if a license has higher priority than commands, then it eliminates all those commands which are not consistent with it. Therefore characterize permission as exception.

4. Tractable imperative logic for access control

Results from the above section show that although imperative logic is decidable, the complexity is however intractable. In order to practically use imperative logic in access control, we have to lighten the complexity. This section shows that under reasonable restrictions on the priority ordering and the syntax of language, the complexity turns out to be tractable.

For the priority relation $>$, we restrict it to be a relation such that the restriction of $>$ on O is a strict linear order. Such restriction ensures that the commands are strictly stratified, although different license can still be incomparable or of the same priority.

Concerning the syntax, we impose the following restrictions. Let $Lit_{\mathbb{P}} = \mathbb{P} \cup \{\neg p : p \in \mathbb{P}\}$ be the set of literals build from $\mathbb{P}$. Let $L_{\mathbb{P}}^{cnl}$ be the conjunctions of literals of $\mathbb{P}$. A strict Horn clause is a non-empty disjunction of exactly one propositional atom and zero or more negated atoms. A Horn clause is a non-empty disjunction of at most one propositional atom and zero or more negated atoms.

Theorem 4. *Let $N = (O, P, >)$ be a normative system where for each $(a, x) \in P$, the restriction of $>$ on $O \cup \{(a, x)\}$ is a strict linear order. Let A be a set of strict Horn clauses. For every norm (b, y), let b be a conjunction of atoms and x be an atom. Let $x \in L_{\mathbb{P}}^{cnl}$. Then the following decision problems can be solved in polynomial time:*

1. *obligation-checking: $x \in out^p(O^>, A)$;*
2. *negative permission-checking: $x \in NegPerm(N, A)$;*
3. *positive permission-checking: $x \in PosPerm(N, A)$;*
4. *explicit prohibition-checking: $x \in ExProhi(N, A)$;*
5. *implicit prohibition-checking: $x \in ImProhi(N, A)$.*

5. Related work and extension

Many cryptographic solutions to the problem of access control have been proposed [32–35]. Multiple logics have been proposed for access control [36–38]. Most of these logics use a modality $says_k$ indexed by an agent k. *says*-based access control logics are designed for systems in which different agents can issue statements that become part of the access control policy. In contrast to the tractability of imperative logic, Garg and Abadi [39] show that the provability problem for *says*-based access control logic is PSPACE-complete.

Van Hertum et al. [31] have recently proposed a multi-agent variant of autoepistemic logic, called *distributed autoepistemic logic with inductive definitions* (dAEL(ID)), to be used as a says-based access control logic. By applying the semantic principles of autoepistemic logic to characterize the says-modality, dAEL(ID) allows us to derive a statement of the form $says_{\neg k}\phi$ on the basis of the observation that k has not issued statements implying ϕ. Supporting reasoning about such negated says-statements allows dAEL(ID) to straightforwardly model access denials. A major difference between imperative logic and dAEL(ID) is that the former is able to handle the conflicts and priority between norms, which is not addressed in the latter.

5.1. Deontic default logic

Horty's deontic default logic [13], which can be viewed as an attempt to reconstruct Reiter's default logic to normative

Please cite this article in press as: X. Sun, et al., Norm-based deontic logic for access control, some computational results, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.01.028

reasoning, is another representative norm-based deontic logic. In this section we present some complexity results of deontic default logic. Taken from Parent [40], now we concisely introduce deontic default logic.

Using notation of imperative logic, a prioritized default theory is a triple $(O, >, A)$ where O is a set of defaults/commands/mandatory norms and $>$ a priority relation over O which is irreflexive and transitive. The key concept in deontic default logic is that of "proper scenario" based on a default theory. A proper scenario is a subset of O satisfying certain conditions. The function of a proper scenarios is similar to that of an elements of a *pomfamily* in imperative logic. Intuitively, the defaults in a proper scenario tell us what counts as a binding (good, satisfactory, etc.) reason for what. Thus, if (a, x) is in the proper scenario O' based on a given default theory, then O' is said to provide a as a binding reason for x. The idea is to assume that the agent derives its obligations from justifications or reasons for those obligations: in particular, that the agent is bounded by an obligation if it possesses a binding reason for that obligation.

Given $O' \subseteq O$, let $Conclusion(O') = \{x : (a, x) \in O'\}$. Formally, the notion of proper scenario is defined using three other notions. Each corresponds to a condition that a default must meet in order to be binding. The first notion is that of a default being triggered in O', noted as $Triggered_{(O,>,A)}(O')$. The definition runs as follows:

$$Triggered_{(O,>,A)}(O') = \{(a, x) \in O : A \cup Conclusion(O') \vDash a\}.$$

The second notion is that of a default being conflicted in O'. Let $Conflicted_{(O,\geq,A)}(O')$ denote the set of all such defaults. The definition reads:

$$Conflicted_{(O,>,A)}(O') = \{(a, x) \in O : A \cup Conclusion(O') \vDash \neg x\}.$$

The third notion is that of a default being defeated in O'. For $O_1, O_2 \subseteq O$, let $O_1 \succ O_2$ if for all $(a_1, x_1) \in O_1$, $(a_2, x_2) \in O_2$, $(a_1, x_1) > (a_2, x_2)$. Let $O^{O_1/O_2} = (O - O_1) \cup O_2$.

$$Defeated_{(O,>,A)}(O_1) = \{(a, x) \in O : \text{ there exists } O_2 \subseteq Triggered_{(O,>,A)}(O_1) \text{ such that}$$

1. $O_2 \succ \{(a, x)\}$
2. there exists $O_3 \subseteq O_1$ with $O_2 \succ O_3$ such that
   (a) $A \cup Conclusion(O^{O_3/O_2})$ is consistent
   (b) $A \cup Conclusion(O^{O_3/O_2}) \vDash \neg x\}$.

Here O_2 can be called a defeating set while O_3 can be called an accommodation set. The idea is that a default (a, x) is defeated by a set of defaults O_1 if we can find a set of defeating default O_2 which is triggered by O_1 and we can find an accommodation set O_3 in O_1 such that if we replace O_3 by O_2, then the resulting set of defaults is consistent and implies $\neg x$. These three concepts are used to define the notion of a proper scenario.

Definition 5 (*Proper Scenario [13]*). Let $O' \subseteq O$ be a scenario based on the prioritized default theory $(O, >, A)$. Then O' is a proper scenario based on $(O, >, A)$, noted as $O' \in propScenario(O, >, A)$, just in case $O' = \bigcup_{i\geq 0} O'_i$ where

- $O'_0 = \emptyset$,
- $O'_{i+1} = \{(a, x) \in O :$
  $(a, x) \in Triggered_{(O,>,A)}(O'_i)$,
  $(a, x) \notin Conflicted_{(O,>,A)}(O')$,
  $(a, x) \notin Defeated_{(O,>,A)}(O')\}$.

Combining Horty's framework with imperative logic, we define proper output with the idea of viewing proper scenario as something similar to *pomfamily*.

Definition 6. $x \in out^d(O^>, A)$ iff $x \in \bigcap\{out(O', A) : O' \in propScenario_i(O, >, A)\}$.

Another technical contribution in this paper is the following complexity result of deontic default logic.

Theorem 5. *Given $O^> = (O, >)$ where O is finite and $>$ is irreflexive and transitive. Let A be a finite set of formulas and x be a formula. Deciding if $x \in out^d(O^>, A)$ is Δ_3^p-hard and in Π_3^p.*

6. Conclusion

In this paper we study the complexity of deontic logics grounded on *norm-based* semantics and apply imperative logic to access control. We present the readers that prioritized imperative logic is complete for the 2ed level of the polynomial hierarchy. To apply imperative logic to access control, restrictions are imposed such that the complexity turns out to be tractable. We also show that deontic default logic is located in the 3ed level of the polynomial hierarchy.

A natural future work is to build imperative logic based on logics which have stronger expressive power than propositional logic (such as description logic) and use it to model access control policies. Implementing imperative logic by a logical programming language to build a deontic machine to perform automatic reasoning is another interesting future work. A third direction of future work is to generalize imperative logic for role-based access control. In role-based access control, permissions are associated with roles, and users are assigned to appropriate roles. From the perspective of deontic logic, roles are institutional facts which are created by constitutive norms. The logic of constitutive norms has been well studied [41] and the combination of constitutive and mandatory norms has been explored using input/output logic in Sun and van der Torre [42]. We estimate that by combining the logic of constitutive norms and imperative logic we can obtain a logical framework to model role-based access control and leave it for future work.

Acknowledgments

Xin Sun and Xishun Zhao have been supported by the National Social Science Foundation of China grant "Researches into Logics and Computer Simulations for Social Games" under grant No. 13&ZD186. Xin Sun has been supported by the National Science Centre of Poland (BEETHOVEN, UMO-2014/15/G/HS1/04514). Livio Robaldo has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 661007 for the project "ProLeMAS: PROcessing LEgal language in normative Multi-Agent Systems".

Appendix

Theorem 2. *Given $O^> = (O, >)$ where $>$ is irreflexive and transitive. Let A be a set of formulas and x be a formula. Deciding if $x \in out^p(O^>, A)$ is Π_2^p-complete.*

Proof. Concerning the Π_2^p hardness, we show that the validity problem of 2-QBF$^\forall$ can be reduced to our problem.

Let $\forall p_1 \dots p_m \exists q_1 \dots q_n \Phi$ be a 2-QBF$^\forall$ where Φ is a propositional formula with variables in $\{p_1, \dots, p_m, q_1, \dots, q_n\}$. Let $A = \emptyset$, $O = \{(\top, p_1), \dots, (\top, p_m), (\top, \neg p_1), \dots, (\top, \neg p_m), (\top, \Phi)\}$, $\geq \emptyset$. Our aim is to show that this 2-QBF$^\forall$ is valid iff $\Phi \in out^p(O^>, A)$.

- If $\forall p_1 \dots p_m \exists q_1 \dots q_n \Phi$ is valid, then for all valuation V for $\{p_1, \dots, p_m\}$ there is a valuation V' for $\{q_1, \dots, q_n\}$ such that $V \cup V'$ gives truth value 1 to Φ and 0 to $\neg\Phi$.
  Let $O' = \{(\top, p'_1), \dots, (\top, p'_m), (\top, \Phi)\}$ be an arbitrary set such that each p'_i is either p_i or $\neg p_i$. Then it can be verified that $O' \in pomfamily(O^>, A)$. Indeed, let $>$ be a strict linear order over O such that $(\top, p'_1) > \dots > (\top, p'_m) > (\top, \Phi) > (\top, \sim p'_1) > \dots > (\top, \sim p'_m)$. Then O' is a preferred obeyable maximal family generated by $>$. By the construction we can further verify that O' ranges over all elements of $pomfamily(O^>, A)$. Note that $out(O', A) = Cn(\{p'_1 \dots, p'_m, \Phi\})$. Therefore $\Phi \in out(O', A)$. Then we conclude $\Phi \in out^p(O^>, A)$.

Please cite this article in press as: X. Sun, et al., Norm-based deontic logic for access control, some computational results, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.01.028

- If $\forall p_1 \dots p_m \exists q_1 \dots q_n \Phi$ is not valid, then there is a valuation V for $\{p_1, \dots, p_m\}$ such that for all valuations V' for $\{q_1, \dots, q_n\}$, $V \cup V'$ gives truth value 0 to Φ and 1 to $\neg\Phi$.

  Let $O' = \{(\top, p'_1), \dots, (\top, p'_m)\}$, where each p'_i is p_i if $p_i \in V$ and it is $\neg p_i$ if $p_i \notin V$. Then $O' \in pomfamily(O^>, A)$ because $A \cup m(O') = \{p'_1, \dots, p'_m\}$ is consistent and adding anything from $m(\{(\top, \neg p_1), \dots, (\top, \neg p_m), (\top, \Phi)\})$ will destroy the consistency. Note that $\neg\Phi \in Cn(\{p'_1, \dots, p'_m\})$ by the construction of $\{p'_1, \dots, p'_m\}$. Therefore $\Phi \notin out(O', A)$, which further implies that $\Phi \notin out^p(O^>, A)$.

So, we have reduced the validity problem of 2-QBF$^\forall$ to our problem, which shows the latter is Π_2^p-hard.

Concerning the Π_2^p-membership, we prove by giving the following algorithm on a non-deterministic Turing machine with an NP oracle to solve the complement of our problem.

1. Guess a subset $O' \subseteq O$.
2. Guess a strict extension of $>$.
3. Use the NP oracle to test if $O' \in pomfamily(O^>, A)$. If no, return "reject" on this branch. Otherwise continue.
4. Use the NP oracle to test if $x \notin out(O', A)$. If $x \notin out(O', A)$, then return "accept" on this branch. Otherwise return "reject" on this branch.

It can be verified that $x \notin out^p(O^>, A)$ iff the non-deterministic Turing machine returns "accept" on some branches. Step 3 can be done in polynomial time steps because the *pomfamily* membership can be decided in P^{NP}. Step 4 can also be done in polynomial time steps on a Turing machine with an ^{NP} oracle because the obligation-checking problem of input/output logic is also in P^{NP} (see chapter 6 of Sun [43]). Therefore the time complexity of this non-deterministic Turing machine is polynomial. □

Theorem 3. *Given $N = (O, P, >)$ where $>$ is irreflexive and transitive. Let A be a set of formulas and x be a formula,*

1. *to decide if $x \in NegPerm(N, A)$ is Σ_2^p-complete.*
2. *to decide if $x \in PosPerm(N, A)$ is Π_2^p-complete.*

Proof. 1. The negative permission-checking is complement to the obligation-checking problem.
2. Let $P = \{(a_1, x_1), \dots, (a_n, x_n)\}$. Then $x \in PosPerm(N, A)$ iff $x \in out^p((O \cup \{(a_1, x_1)\})^>, A) \cup \dots \cup out^p((O \cup \{(a_n, x_n)\})^>, A)$ iff $x \in out^p((O \cup \{(a_1, x_1)\})^>, A)$ or $x \in out^p((O \cup \{(a_2, x_2)\})^>, A)$ or ... or $x \in out^p(((O \cup \{(a_n, x_n)\})^>, A)$. Since the Π_2^p class is closed under union, we know that the static permission checking problem is in Π_2^p. The Π_2^p hardness can be proved by setting $P = \emptyset$ and reducing the obligation-checking problem to the static permission-checking problem. □

Theorem 4. *Let $N = (O, P, >)$ be a normative system where for each $(a, x) \in P$, the restriction of $>$ on $O \cup \{(a, x)\}$ is a strict linear order. Let A be a set of strict Horn clauses. For every norm (b, y), let b be a conjunction of atoms and x be an atom. Let $x \in L_{\mathbb{P}}^{cnl}$. Then the following decision problems can be solved in polynomial time:*

1. *obligation-checking: $x \in out^p(O^>, A)$;*
2. *negative permission-checking: $x \in NegPerm(N, A)$;*
3. *positive permission-checking: $x \in PosPerm(N, A)$;*
4. *explicit prohibition-checking: $x \in ExProhi(N, A)$;*
5. *implicit prohibition-checking: $x \in ImProhi(N, A)$.*

Proof (*Sketch*)**.** Here we only prove that obligation-checking is tractable. First, we show that *pomfamily* is a singleton and can be find in polynomial time. It is unique due to that $>$ is a strict linear on O. Now the main source of complexity in the computation of *pomfamily* is that we have to decide if $A \cup m(O_i \cup \{(a_i, x_i)\})$ is consistent. The readers are invited to verify that under our syntactical restriction this decision can be made in polynomial time. Finally, for *pomfamily* $= \{O'\}$, whether $x \in out(O', A)$ can be computed in polynomial time, due to the syntactical restriction. □

Lemma 1. *Given a prioritized default theory $(O, >, A)$, a scenario O' and a default (a, x)*

1. *deciding if $(a, x) \in Triggered_{(O,>,A)}(O')$ is coNP-complete.*
2. *deciding if $(a, x) \in Conflicted_{(O,>,A)}(O')$ is coNP-complete.*
3. *deciding if $(a, x) \in Defeated_{(O,>,A)}(O')$ is in Σ_2^p.*

Proof. Item 1 and 2 are trivial. Item 3 can be proved by a simple guess and check procedure on a non-deterministic Turing machine with an NP oracle. Here we omit the details. □

We will prove that deontic default logic is Δ_3^p-hard and in Π_3^p. [44] shows that the following problem is Δ_3^p-complete:

> Maximum 2-QBF: given an arbitrary 2-QBF$^\exists$ $\exists p_1 \dots p_m \forall q_1 \dots q_n \Phi$, decide if $V_1(p_m) = 1$ where V_1 is the **lexicographically maximal** valuation of $\{p_1, \dots, p_m\}$ such that for all valuation V_2 of $\{q_1, \dots, q_n\}$, $V_1 \cup V_2 \vDash \Phi$.

Here for two valuation of $\{p_1, \dots, p_m\}$, V_1 is lexicographically larger than V_2 iff there exists i such that $V_1(p_i) = 1$, $V_2(p_i) = 0$ and for all $j \in \{1, \dots, i-1\}$, $V_1(p_j) = V_2(p_j)$.

Theorem 5. *Given $O^> = (O, >)$ where O is finite and $>$ is irreflexive and transitive. Let A be a finite set of formulas and x be a formula. Deciding if $x \in out^d(O^>, A)$ is Δ_3^p-hard and in Π_3^p.*

Proof. We prove the Δ_3^p hardness by reducing Maximum 2-QBF to our problem. Given an arbitrary 2-QBF$^\exists$ $\exists p_1 \dots p_m \forall q_1 \dots q_n \Phi$, we construct $O = \{(\top, p_1), (\top, \neg p_1), \dots, (\top, p_m), (\top, \neg p_m), (\Phi, x)\}$, where x is a formula contains no propositional variable from $\{p_1, \dots, p_m, q_1, \dots, q_n\}$. We further let the priority relation be the universal relation, *i.e.* $\geq \emptyset$. Our aim is to show that to decide if $V_1(p_m) = 1$ where V_1 is the lexicographically maximal valuations of $\{p_1, \dots, p_m\}$ such that for all valuation V_2 of $\{q_1, \dots, q_n\}$ it holds that $V_1 \cup V_2 \vDash \Phi$, we only need to decide if $p_m \in out^d(O^>, \emptyset)$.

We first show that the following are equivalent: for arbitrary $O' \subseteq O - \{(\Phi, x)\}$ and $P' \subseteq \{p_1, \dots, p_m\}$ satisfying that $(\top, p_i) \in O'$ iff $p_i \in P'$ and $(\top, \neg p_i) \in O'$ iff $p_i \notin P'$,

1. $O' \cup \{(\Phi, x)\} \in propScenario(O, >, \emptyset)$ and $x \in Cn(Conclusion(O' \cup \{(\Phi, x)\}))$.
2. P' is a lexicographically maximal valuation for $\{p_1, \dots, p_m\}$ such that for all $Q' \subseteq \{q_1, \dots, q_n\}$, $P' \cup Q' \models \Phi$.

Assume P' is the lexicographically maximal valuation for $\{p_1, \dots, p_m\}$ such that for all $Q' \subseteq \{q_1, \dots, q_n\}$, $P' \cup Q' \models \Phi$. We show that $O' \cup \{(\Phi, x)\} \in propScenario(O, >, \emptyset)$. Indeed, we can construct $[O' \cup \{(\Phi, x)\}]_0 = \emptyset$, $[O' \cup \{(\Phi, x)\}]_1 = \{(a, x) \in O : (a, x) \in Triggered_{(O,>,A)}([\{O' \cup (\Phi, x)\}]_0), (a, x) \notin Conflicted_{(O,>,A)}(O' \cup \{(\Phi, x)\}), (a, x) \notin Defeated_{(O,>,A)}(O' \cup \{(\Phi, x)\})\}$. Here we have $O' \subseteq [O' \cup \{(\Phi, x)\}]_1$ because for all $(\top, l_i) \in O'$,

1. $(\top, l_i) \in Triggered_{(O,>,A)}(\emptyset)$.
2. $(\top, l_i) \notin Conflicted_{(O,>,A)}(O' \cup \{(\Phi, x)\})$.
3. $(\top, l_i) \notin Defeated_{(O,>,A)}(O' \cup \{(\Phi, x)\})$ because $(\top, \sim l_i) \not> (\top, l_i)$ since $\geq \emptyset$.

We further have $[O' \cup \{(\Phi, x)\}]_2 = \{(a, x) \in O : (a, x) \in Triggered_{(O,>,A)}([O' \cup \{(\Phi, x)\}]_1), (a, x) \notin Conflicted_{(O,>,A)}(O' \cup \{(\Phi, x)\}), (a, x) \notin Defeated_{(O,>,A)}(O' \cup \{(\Phi, x)\})\}$. Now we prove $[O' \cup \{(\Phi, x)\}]_2 = O' \cup \{((\Phi, x))\}$. This is because

1. for all $(\top, l_i) \notin O'$, $(\top, l_i) \in Conflicted_{(O,>,A)}(O' \cup \{(\Phi, x)\})$.
2. $O' \subseteq [O' \cup \{(\Phi, x)\}]_1 \subseteq [O' \cup \{(\Phi, x)\}]_2$.
3. $(\Phi, x) \in [O' \cup \{(\Phi, x)\}]_2$. The reason is: from $Cn(P') \models \Phi$ we derive $Consequence(O') \models \Phi$. Then we know $(\Phi, x) \in Triggered_{(O,>,A)}([O' \cup \{(\Phi, x)\}]_1)$. Meanwhile, $(\Phi, x) \notin Conflicted_{(O,>,A)}(O' \cup \{(\Phi, x)\})$ and $(\Phi, x) \notin Defeated_{(O,>,A)}(O' \cup \{(\Phi, x)\})$.

We further have $[O' \cup \{(\Phi, x)\}]_i = [O' \cup \{(\Phi, x)\}]_2$, for all $i \geq 3$. Therefore $O' \cup \{(\Phi, x)\} = \bigcup_{i \geq 0} [O' \cup \{(\Phi, x)\}]_i$, which proves

$O' \cup \{(\Phi, x)\} \in propScenario(O, >, \emptyset)$. Then trivially we have $x \in Cn(Conclusion(O' \cup \{(\Phi, x)\}))$.

Assume $O' \cup \{(\Phi, x)\} \in propScenario(O, >, \emptyset)$ and $x \in Cn(Conclusion(O' \cup \{(\Phi, x)\}))$. Then we know $(\Phi, x) \in [O' \cup \{(\Phi, x)\}]_i$ for some i. It cannot be that $(\Phi, x) \in [O' \cup \{(\Phi, x)\}]_0$ because $[O' \cup \{(\Phi, x)\}]_0 = \emptyset$.

- If $(\Phi, x) \in [O' \cup \{(\Phi, x)\}]_1$, then $(\Phi, x) \in Triggered_{(O,>,A)}(\emptyset)$, which means $\emptyset \vDash \Phi$. Then we know $P' \cup Q' \models \Phi$, where P' is the lexicographically maximal valuation for $\{p_1, \dots, p_m\}$ such that for all $Q' \subseteq \{q_1, \dots, q_n\}$.
- If $(\Phi, x) \notin [O' \cup \{(\Phi, x)\}]_1$ but $(\Phi, x) \in [O' \cup \{(\Phi, x)\}]_2$, then $[O' \cup \{(\Phi, x)\}]_1 = O'$ and $(\Phi, x) \in Triggered_{(O,>,A)}(O')$. Therefore $Conclusion(O') \vDash \Phi$. Then by the relationship between O' and P', we know that for all $Q' \subseteq \{q_1, \dots, q_n\}$, $P' \cup Q' \models \Phi$.
- If $(\Phi, x) \notin [O' \cup \{(\Phi, x)\}]_2$, then $[O' \cup \{(\Phi, x)\}]_2 = [O' \cup \{(\Phi, x)\}]_1 = O'$. Moreover $\bigcup_{i \geq 0}[O' \cup \{(\Phi, x)\}]_i = [O' \cup \{(\Phi, x)\}]_1 = O'$, which contradicts to $\bigcup_{i \geq 0}[O' \cup \{(\Phi, x)\}]_i = O' \cup \{(\Phi, x)\}$.

Then we can conclude that $P' \cup Q' \models \Phi$, where P' is the lexicographically maximal valuation for $\{p_1, \dots, p_m\}$ such that for all $Q' \subseteq \{q_1, \dots, q_n\}$.

Now we finish our reduction: given an arbitrary 2-QBF$^\exists$ $\exists p_1 \dots p_m \forall q_1 \dots q_n \Phi$, if V_1 is the lexicographically maximal valuations of $\{p_1, \dots, p_m\}$ such that for all valuation V_2 of $\{q_1, \dots, q_n\}$, $V_1 \cup V_2 \vDash \Phi$, to decide if $V_1(p_m) = 1$, we only need to decide if $p_m \in out^d(O^>, \emptyset)$. Such reduction is polynomial in the size of $\exists p_1 \dots p_m \forall q_1 \dots q_n \Phi$, which proves the Δ_3^p hardness.

Now for the Π_3^p membership. With Lemma 1 at hand. This theorem can be proved by a simple guess and check procedure on a non-deterministic Turing machine with an Σ_2^p oracle. Here we omit the details. □

References

[1] G. von Wright, Deontic logic, Mind 60 (1951) 1–15.
[2] J.J. Meyer, A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic, Notre Dame J. Form. Log. (1988) 109–136.
[3] G. Kołaczek, Model of role based access control mechanism (Ph.D. thesis), Technical University of Wrocław, 2001.
[4] G. Kołaczek, Application of deontic logic in role–based access control, Int. J. Appl. Math. Comput. Sci. 12 (2) (2002) 269–275.
[5] N. Cuppens-Boulahia, F. Cuppens, Specifying intrusion detection and reaction policies: An application of deontic logic, in: R. van der Meyden, L.W.N. van der Torre (Eds.), Deontic Logic in Computer Science, 9th International Conference, DEON 2008, Luxembourg, Luxembourg, July 15–18, 2008. Proceedings, in: Lecture Notes in Computer Science, vol. 5076, Springer, 2008, pp. 65–80. URL http://dx.doi.org/10.1007/978-3-540-70525-3_7.
[6] B. van Fraassen, Values and the heart's command, J. Philos. 70 (1973) 5–19.
[7] J. Hansen, Prioritized conditional imperatives: problems and a new proposal, Auton. Agents Multi-Agent Syst. 17 (1) (2008) 11–35. URL http://dx.doi.org/10.1007/s10458-007-9016-7.
[8] R. van der Meyden, The dynamic logic of permission, J. Logic Comput. 6 (1996) 465–479.
[9] J. Horty, Agency and Deontic Logic, Oxford University Press, New York, 2001.
[10] X. Sun, Conditional ought, a game theoretical perspective, in: J. L. H. van Ditmarsch, S. Ju (Eds.), Logic, Rationality, and Interaction: Proceedings of the Thire International Workshop, Guangzhou, China, 2011, pp. 356–369.
[11] D. Makinson, L. van der Torre, Input–output logics, J. Philos. Logic 29 (2000) 383–408.
[12] J. Horty, Reasoning with moral conflicts, Nous 37 (2003) 557–605.
[13] J. Horty, Reasons as Defaults, Oxford University Press, 2012.
[14] G. Antoniou, N. Dimaresis, G. Governatori, A system for modal and deontic defeasible reasoning, in: M.A. Orgun, J. Thornton (Eds.), AI 2007: Advances in Artificial Intelligence, 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, December 2–6, 2007, Proceedings, in: Lecture Notes in Computer Science, vol. 4830, Springer, 2007, pp. 609–613. URL http://dx.doi.org/10.1007/978-3-540-76928-6_62.
[15] G. Antoniou, N. Dimaresis, G. Governatori, A modal and deontic defeasible reasoning system for modelling policies and multi-agent systems, Expert Syst. Appl. 36 (2) (2009) 4125–4134. URL http://dx.doi.org/10.1016/j.eswa.2008.03.009.
[16] D. Gabbay, J. Horty, X. Parent, R. van der Meyden, L. van der Torre (Eds.), Handbook of Deontic Logic and Normative Systems, College Publications, London, 2013.
[17] D. Gabbay, J. Horty, R. van der Meyden, L. van der Torre (Eds.), Handbook of Deontic Logic and Normative Systems, Vol. 2, College Publications, London, 2017, (in press).
[18] J. Hansen, David Makinson on Classical Methods for Non-Classical Problems, Springer, Netherlands, Dordrecht, 2014, pp. 287–333. http://dx.doi.org/10.1007/978-94-007-7759-0_14, (Chapter) Reasoning about permission and obligation.
[19] R. Chisholm, Contrary-to-duty imperatives and deontic logic, Analysis 24 (1963) 33–36.
[20] S. Benferhat, R.E. Baida, F. Cuppens, A stratification-based approach for handling conflicts in access control, in: SACMAT 2003, 8th ACM Symposium on Access Control Models and Technologies, June 2–3, 2003, Villa Gallia, Como, Italy, Proceedings, ACM, 2003, pp. 189–195. URL http://doi.acm.org/10.1145/775412.775437.
[21] D. Makinson, L. van der Torre, Constraints for input/output logics, J. Philos. Logic 30 (2) (2001) 155–185.
[22] V. Genovese, D. Garg, New modalities for access control logics: Permission, control and ratification, in: C.A. Meadows, M.C.F. Gago (Eds.), Security and Trust Management - 7th International Workshop, STM 2011, Copenhagen, Denmark, June 27–28, 2011, Revised Selected Papers, in: Lecture Notes in Computer Science, vol. 7170, Springer, 2011, pp. 56–71. URL http://dx.doi.org/10.1007/978-3-642-29963-6_6.
[23] N. Dinesh, A.K. Joshi, I. Lee, O. Sokolsky, Permission to speak: A logic for access control and conformance, J. Log. Algebr. Program. 80 (1) (2011) 50–74. URL http://dx.doi.org/10.1016/j.jlap.2009.12.002.
[24] D. Makinson, L. van der Torre, Permission from an input/output perspective, J. Philos. Logic 32 (2003) 391–416.
[25] A. Stolpe, A theory of permission based on the notion of derogation, J. Appl. Log. 8 (1) (2010) 97–113. URL http://dx.doi.org/10.1016/j.jal.2010.01.001.
[26] G. Governatori, F. Olivieri, A. Rotolo, S. Scannapieco, Computing strong and weak permissions in defeasible logic, J. Philos. Logic 42 (6) (2013) 799–829. URL http://dx.doi.org/10.1007/s10992-013-9295-1.
[27] X. Sun, D.A. Ambrossio, Computational complexity of input/output logic, in: A. Bikakis, X. Zheng (Eds.), Multi-disciplinary Trends in Artificial Intelligence - 9th International Workshop, MIWAI 2015, Fuzhou, China, November 13–15, 2015, Proceedings, in: Lecture Notes in Computer Science, vol. 9426, Springer, 2015, pp. 72–79. URL http://dx.doi.org/10.1007/978-3-319-26181-2_7.
[28] S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, New York, USA, 2009.
[29] G. Brewka, Preferred subtheories: An extended logical framework for default reasoning, in: N.S. Sridharan (Ed.), Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989, Morgan Kaufmann, 1989, pp. 1043–1048. URL http://ijcai.org/Proceedings/89-2/Papers/031.pdf.
[30] V.W. Marek, M. Truszczynski, Nonmonotonic Logic - Context-dependent Reasoning, in: Artificial intelligence, Springer, 1993.
[31] P.V. Hertum, M. Cramer, B. Bogaerts, M. Denecker, Distributed autoepistemic logic and its application to access control, in: S. Kambhampati (Ed.), Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, IJCAI/AAAI Press, 2016, pp. 1286–1292. URL http://www.ijcai.org/Abstract/16/186.
[32] S.G. Akl, P.D. Taylor, Cryptographic solution to a problem of access control in a hierarchy, ACM Trans. Comput. Syst. 1 (3) (1983) 239–248. URL http://doi.acm.org/10.1145/357369.357372.
[33] N. Nagy, S.G. Akl, A quantum cryptographic solution to the problem of access control in a hierarchy, Parallel Process. Lett. 20 (3) (2010) 251–261. URL http://dx.doi.org/10.1142/S012962641000020X.
[34] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 27 (2) (2016) 340–352. URL http://dx.doi.org/10.1109/TPDS.2015.2401003.
[35] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, K. Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing, IEEE Trans. Inf. Forensics Secur. 11 (11) (2016) 2594–2608. URL http://dx.doi.org/10.1109/TIFS.2016.2590944.
[36] M. Abadi, Logic in access control, in: 18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22–25 June 2003, Ottawa, Canada, Proceedings, IEEE Computer Society, 2003, p. 228. URL http://dx.doi.org/10.1109/LICS.2003.1210062.
[37] D. Garg, F. Pfenning, Stateful authorization logic - proof theory and a case study, J. Comput. Secur. 20 (4) (2012) 353–391. URL http://dx.doi.org/10.3233/JCS-2012-0456.
[38] V. Genovese, Modalities in access control: Logics, proof-theory and application (Ph.D. thesis), University of Luxembourg, 2012.
[39] D. Garg, M. Abadi, A modal deconstruction of access control logics, in: R.M. Amadio (Ed.), Foundations of Software Science and Computational Structures, 11th International Conference, FOSSACS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29 - April 6, 2008. Proceedings, in: Lecture Notes in Computer Science, vol. 4962, Springer, 2008, pp. 216–230. URL http://dx.doi.org/10.1007/978-3-540-78499-9_16.
[40] X. Parent, Moral particularism in the light of deontic logic, Artif. Intell. Law 19 (2–3) (2011) 75–98. URL http://dx.doi.org/10.1007/s10506-011-9108-1.
[41] D. Grossi, A. Jones, Constitutive norms and counts-as conditionals, in: J. Horty, D. Gabbay, X. Parent, R. van der Meyden, L. van der Torre (Eds.), Handbook of Deontic Logic and Normative Systems, College Publications, London, 2013, pp. 407–441.

[42] X. Sun, L.W.N. van der Torre, Combining constitutive and regulative norms in input/output logic, in: F. Cariani, D. Grossi, J. Meheus, X. Parent (Eds.), Deontic Logic and Normative Systems - 12th International Conference, DEON 2014, Ghent, Belgium, July 12–15, 2014. Proceedings, in: Lecture Notes in Computer Science, vol. 8554, Springer, 2014, pp. 241–257. URL http://dx.doi.org/10.1007/978-3-319-08615-6_18.
[43] X. Sun, Logic and games of norms, a computational perspective (Ph.D. thesis), University of Luxembourg, 2016.
[44] M.W. Krentel, Generalizations of opt P to the polynomial hierarchy, Theoret. Comput. Sci. 97 (2) (1992) 183–198. URL http://dx.doi.org/10.1016/0304-3975(92)90073-O.

Xin Sun earned his master degree from Tsinghua University in China in 2012 and his Ph.D. degree from the University of Luxembourg in 2016. His current research interests include logic, game theory, computational complexity and quantum computation.

Xishun Zhao is a full professor at Institute of Logic and Cognition, Sun Yat-sen University, China. He received his Ph.D. degree in mathematics from Nanjing University, China in 1999. His current research interests include computational complexity in logic and analysis, knowledge representation and reasoning, expressive power and complexity of logic systems.

Livio Robaldo is post-doctoral fellows at the University of Luxembourg. His research interests mainly concern Legal Informatics, Natural Language Semantic, Deontic Logic and Multi-agent systems. He participated in eleven research projects, two of which in the role of principal investigator. He published his results in 72 locations, among which 12 journals. He is the single author of 4 journal papers and several conference workshop papers. Google scholar reports 987 citations.