J Philos Logic (2018) 47:579–609
DOI 10.1007/s10992-017-9440-3

Obligation as Optimal Goal Satisfaction

Robert Kowalski[1] **· Ken Satoh**[2]

Received: 21 October 2016 / Accepted: 23 May 2017 / Published online: 5 July 2017

Abstract Formalising deontic concepts, such as obligation, prohibition and permission, is normally carried out in a modal logic with a possible world semantics, in which some worlds are better than others. The main focus in these logics is on inferring logical consequences, for example inferring that the obligation $\mathbf{O}\ q$ is a logical consequence of the obligations $\mathbf{O}\ p$ and $\mathbf{O}\ (p \rightarrow q)$. In this paper we propose a non-modal approach in which obligations are preferred ways of satisfying goals expressed in first-order logic. To say that p is obligatory, but may be violated, resulting in a less than ideal situation s, means that the task is to satisfy the goal $p \vee s$, and that models in which p is true are preferred to models in which s is true. Whereas, in modal logic, the preference relation between possible worlds is part of the semantics of the logic, in this non-modal approach, the preference relation between first-order models is external to the logic. Although our main focus is on satisfying goals, we also formulate a notion of logical consequence, which is comparable to the notion of logical consequence in modal deontic logic. In this formalisation, an obligation $\mathbf{O}\ p$ is a logical consequence of goals G, when p is true in all best models of G. We show how this non-modal approach to the treatment of deontic concepts deals with problems of contrary-to-duty obligations and normative conflicts, and argue that the approach is useful for many other applications, including abductive explanations, defeasible reasoning, combinatorial optimisation, and reactive systems of the production system variety.

✉ Robert Kowalski
rak@doc.ic.ac.uk

Ken Satoh
ksatoh@nii.ac.jp

[1] Imperial College London, Kensington, London SW7 2AZ, UK

[2] National Institute of Informatics, Tokyo, Japan

Keywords Deontic logic · Abductive logic programming · Normative conflicts · Contrary-to-duty obligations · Goals · Preferences

1 Introduction

There are two ways to understand such natural language sentences as *birds can fly*. One is to understand them literally, but only as defeasible assumptions. The other is to understand them as approximations to more precisely stated sentences, such as *a bird can fly if the bird is normal,* with an extra condition *the bird is normal,* which is defeasible, but is assumed to hold by default.

In this paper, we explore the second approach, applied to natural language sentences involving deontic attitudes. In contrast to modal approaches, which aim to stay close to the literal expression of natural language sentences, our approach uses a non-modal logic, in which implicit alternatives are made explicit. For example, instead of understanding the sentence *you should wear a helmet if you are driving a motorcycle* as it is expressed literally, we understand it instead as saying that you have a choice: *if you are driving a motorcycle, then you will drive with a helmet or you will risk suffering undesirable consequences that are worse than the discomfort of wearing a helmet.*

This is not an entirely new idea. Herbert Bohnert [8] suggested a similar approach for imperative sentences, treating the command *do a*, for example, as an elliptical statement of a non-modal, declarative sentence *either you will do a or else s*, where *s* is a sanction or "some future situation of directly unpleasant character". Alan Ross Anderson [2] built upon Bohnert's idea, but reformulated it in alethic modal logic, reducing deontic sentences of the form $\mathbf{O}\ p$ (meaning p is obligatory) to alethic sentences of the form $\mathbf{N}\ (\neg p \rightarrow s)$ (meaning it is necessarily the case that if p does not hold, then s holds). A similar reduction of deontic logic to alethic logic was also proposed by Stig Kanger [35]. Our non-modal approach, using abductive logic programming (ALP) [34], is similar in spirit, in the sense that goals in ALP - whether they represent the personal goals of an individual agent, the social goals of a society of agents, the dictates of a powerful authority, or physical constraints - are *hard* constraints that *must* be satisfied.

In the simplified variant of ALP that we use in this paper, an *abductive framework* is a triple $\langle$P, G, A$\rangle$, where P is a logic program representing an agent's beliefs, G is a set of sentences in FOL (first-order logic) representing the agent's goals, and A is a set of atomic sentences representing candidate assumptions. The logic program P serves as an intensional definition (or representation) of an incomplete model of the world, which can be extended by adding assumptions $\Delta \subseteq$ A, to obtain a more complete model represented by P $\cup\ \Delta$. The *abductive task* is to *satisfy* the goals G, by generating some $\Delta \subseteq$ A, such that:

G is true in the model represented by P $\cup\ \Delta$.

For simplicity, we consider only logic programs, which are sets of definite clauses of the form $conclusion \leftarrow condition_1 \wedge \ldots \wedge condition_n$, where *conclusion* and each $condition_i$ is an atomic formula, and all variables are universally quantified. Any logic program P (or P $\cup\ \Delta$) of this form has a unique minimal model [17]. The logic

program can be regarded as a definition of this model, and the model can be regarded as the intended model of the logic program.

In ordinary abduction, the goals G represent a set of observations, and Δ represents external events that explain G. In deontic applications, the goals G represent obligations, augmented if necessary with less desirable alternatives, and Δ represents actions and possibly other assumptions that together with P satisfy G.

In general, there can be many $\Delta \subseteq A$ that satisfy the same goals G. In some cases, the choice between them may not matter; but in other cases, where some Δ are better than others, it may be required to generate some best Δ. For example, in ordinary abduction, it is normally required to generate the best explanation of the observations. In deontic applications, it is similarly required to generate some best more complete model of the world. However, due to practical limitations of incomplete knowledge and lack of computational resources, it may not always be feasible to generate a best Δ. In some cases, it may not even be possible to decide whether one Δ is better than another. It other cases, it may be enough simply to satisfy the goals [63] without attempting to optimise them. Nonetheless, the aim of generating a best solution represents a normative ideal, against which other, more practical solutions can be compared.

For this purpose, we extend the notion of an abductive framework $\langle P, G, A\rangle$ to that of a *normative abductive framework* $\langle P, G, A, <\rangle$, where $<$ is a strict partial ordering among the models represented by extended logic programs $P \cup \Delta$, where $\Delta \subseteq A$.

> The *normative abductive task* is to *satisfy* G by generating *some* $\Delta \subseteq A$, such that G is true in the model M represented by $P \cup \Delta$ and there does not exist any $\Delta' \subseteq A$ such that G is true in the model M' represented by $P \cup \Delta'$ and $M < M'$.

This focus on satisfying goals in ALP contrasts with the more common focus on determining logical consequence in most formal logics. We will argue that some of the problems that arise in deontic logic in particular are due to this focus on logical consequence, and that they can be solved by shifting focus to goal satisfaction. To facilitate the argument, we employ the following definition, adapted from [31]:

> An obligation **O** p is a *logical consequence* of a normative abductive framework $\langle P, G, A, <\rangle$ if and only if p is true in *all* best models of G.

In this paper, we show how ALP deals with contrary-to-duty obligations, which arise when the violation of a primary obligation p invokes a secondary obligation q. We represent such contrary-to-duty obligations by means of a goal of the form $\neg p \rightarrow q$ (or equivalently $p \vee q$), together with an indication that models in which p is true are better than models in which q is true.

We also address the problem of reasoning with conflicting norms, which arise when two obligations p and q are incompatible. We represent such conflicting norms by goals of the form $p \vee$ *sanction1* and $q \vee$ *sanction2*, where models in which p is true are better than models in which *sanction1* is true, models in which q is true are better than models in which *sanction2* is true, but models in which both p and *sanction2* are true and models in which both q and *sanction1* are true may be incomparable.

At various places in this paper, we compare the ALP approach with that of standard deontic logic and some of its variants, production systems, Horty's default

deontic logic, constrained optimisation, SBVR (Semantics of Business Vocabulary and Rules [61] and SCIFF [1]. The comparison with constrained optimisation shows that the separation in ALP between goals and preferences is a standard approach in practical problem solving systems. One of the advantages of the separation is that it shows how the normative ideal of generating a best solution can be approximated in practice by strategies designed to find the best solution possible within the computational resources available. The comparison with SBVR, on the other hand, shows that the syntactic limitations of ALP compared with modal deontic logics do not seem to be a limitation in practice, because they are shared with other approaches, such as SBVR, developed for applying deontic reasoning to practical applications.

The application of ALP to deontic reasoning has previously been explored in [38], which applies ALP to deontic interpretations of the Wason selection task [70] and to moral dilemmas in so-called trolley problems [65]. However, the approach most closely related to the one of this paper is that of SCIFF [1], which uses ALP to specify and verify interaction in multi-agent systems. Alberti et al. [1] compare SCIFF with modal deontic logics, but do not discuss the treatment of conflicting obligations or contrary-to-duty obligations.

Although we compare our approach with standard deontic logic, we do not compare it in detail with the myriad of other logics that have been developed for deontic reasoning. These other logics include defeasible deontic logics [48], stit logics of agency [30], input-output logics [43, 44] and preference-based deontic logics, such as [26] and [66]. As Paul Bartha [7] puts it, "Attempts to address these problems have resulted in an almost bewildering proliferation of different systems of deontic logic - at least one per deontic logician, as some have quipped - so that innovation inevitably meets with a certain amount of skepticism and fatigue." Instead, we broaden our comparison to include such related work as production systems, constrained optimisation and SBVR, which have received little attention in the literature on deontic logic.

Many of the issues addressed in this paper are controversial, for example whether first-order logic is adequate for knowledge representation and problem solving or whether other logics are necessary; and whether a single universal logic is possible for human reasoning or whether many logics are needed for different purposes. Although it is possible to address these issues with theorems and their proofs, we pursue a more informal approach in this paper. We assume only that the reader has a general background in formal logic, but no specific knowledge of deontic logic or ALP. The next two sections provide brief introductions to both deontic logic and ALP.

2 Deontic Logic

Deontic logic is concerned with representing and reasoning about norms of behaviour. However, many authors have denied "the very possibility of the logic of norms and imperatives" [27]. Makinson [44], in particular, states that "there is a singular tension between the philosophy of norms and the formal work of deontic logicians. .... Declarative statements may bear truth-values, i.e. are capable of being true or false, whilst norms are items of another kind. They assign obligations, permissions, and prohibitions. They may be applied or not, respected or not... But it

makes no sense to describe norms as true or as false." However Jørgensen [33], while acknowledging this distinction between norms and declarative sentences, noted that there are "inferences in which one or both premises as well as the conclusion are imperative sentences, and yet the conclusion is just as inescapable as the conclusion of any syllogism containing sentences in the indicative mood only". The resulting conundrum has come to be known as *Jørgensen's dilemma*.

Despite these philosophical concerns, deontic logic has been a thriving research area, owing in large part to its formalisation in modal logic by von Wright [69]. The best known formalisation, which is commonly used as a basis for comparison with other deontic logics, is standard deontic logic (SDL). SDL is a propositional logic with a modal operator **O** representing obligation, where **O** p means that p is obligatory.

SDL can be formalised by adding to non-modal propositional logic the following axiom schemas and inference rule:

D: $\neg (\mathbf{O}\ p \wedge \mathbf{O} \neg\ p)$
K: $\mathbf{O}\ p \wedge \mathbf{O}\ (p \rightarrow q) \rightarrow \mathbf{O}\ q$
NEC: If p is a theorem, then **O** p is a theorem.

Modal operators representing prohibition **F** and permission **P** can be defined in terms of obligation **O**:

$$\mathbf{F}p \leftrightarrow \mathbf{O} \neg\ p$$
$$\mathbf{P}p \leftrightarrow \neg\ \mathbf{O} \neg p$$

The semantics of SDL is defined in terms of models $M = <W, R>$, where W is a set of possible worlds and R is a binary relation over possible worlds, where the intention of $(w, w') \in R$ is that w' is a world where everything obligatory in w holds. For simplicity, and to aid comparison with the semantics of ALP, possible worlds $w \in W$ can be represented by sets of (non-modal) atomic sentences (which do not include any Boolean connectives or the modal operator **O**). The definition of $M, w \mid = p$, expressing that p is true in $w \in W$, is then just:

$M, w \models p$ iff $p \in w$, where p is an atomic sentence
$M, w \models \neg p$ iff it is not the case that $M, w \models p$
$M, w \models p \wedge q$ iff $M, w \models p$ and $M, w \models q$
(and similarly for the other Boolean connectives)
$M, w \models \mathbf{O}p$ iff $M, w' \models p$ for all w' such that $(w, w') \in R$.

The proof theory of SDL is sound and complete with respect to this semantics.

There are many, well-known problems with SDL and its variants. In Section 5, we will see how ALP deals with the problems presented in this section. A major source of these problems is the following inference pattern, which follows from K and NEC:

RM: If $p \rightarrow q$ is a theorem, then **O** $p \rightarrow$ **O** q is a theorem.

Thus if you have an obligation p, and if q is any consequence of p, then you also have the obligation q.

2.1 Ross's Paradox

RM entails, for example, Ross's Paradox [58]:

It is obligatory that the letter is mailed.
If the letter is mailed, then the letter is mailed or the letter is burned.
Therefore, it is obligatory that the letter is mailed or the letter is burned.
i.e. **O** *mail*, $mail \rightarrow mail \vee burn$. Therefore **O** ($mail \vee burn$).

Thus, it seems that, if you are obliged to mail a letter, then you can satisfy the obligation either by mailing it or by burning it.

2.2 The Good Samaritan Paradox

RM also entails the Good Samaritan Paradox [54]:

It ought to be the case that Jones helps Smith who has been robbed.
If Smith has been robbed and Jones helps Smith, then Smith has been robbed.
Therefore, it ought to be the case that Smith has been robbed.
i.e. **O** ($rob \wedge help$), $rob \wedge help \rightarrow rob$**.** Therefore **O** *rob*.

But concluding that a person ought to be robbed if the person ought to be helped when he is robbed seems hardly good advertising for being a Good Samaritan.

2.3 Chisholm's Paradox

In his history of deontic logic, McNamara [46] identifies Chisholm's Paradox [12] as "the booster rocket" that propelled deontic logic into a distinct specialization. The Paradox can be represented as follows:

It ought to be that Jones goes to assist his neighbours.
It ought to be that, if Jones goes, then he tells them he is coming.
If Jones doesn't go, then he ought not tell them he is coming.
Jones doesn't go.
i.e. **O** *go*, **O** ($go \rightarrow tell$), $\neg go \rightarrow$ **O** $\neg$ *tell*, $\neg go$.

Much of the discussion concerning the Paradox concerns the representation of conditional obligations of the kind involved in the second and third sentences. For example, the second and third sentences can be represented in the alternative forms $go \rightarrow$ **O** *tell* and **O** ($\neg go \rightarrow \neg tell$), respectively. Different representations lead to different problems. See, for example, the discussion in [11].

McNamara [46] claims, in the context of discussing the Paradox, that there is nearly universal agreement that such conditional obligations cannot be faithfully represented "by a composite of some sort of unary deontic operator and a material conditional". One of the most common responses to the problem is to employ a dyadic deontic logic, like that of [26], in which conditional obligations are expressed using a binary deontic operator **O** (q/p), representing that the obligation q is conditional on p.

Another reaction to the Paradox is to formalise it in an action or temporal logic, e.g. [47], so that the obligation for John to assist his neighbours holds only until he doesn't go, at which time he has the new obligation not to tell that he is coming. However, as [55] points out, the solution doesn't work for contrary-to-duty obligations not involving change of state, as in Forrester's paradox.

2.4 Forrester's Paradox

Forrester's [19] Paradox of Gentle Murder has been called "the deepest paradox of all" [22]. Here is a common formulation:

It is forbidden for a person to kill.	i.e. $\mathbf{O} \neg \mathit{kill}$
But if a person kills, the killing ought to be gentle.	i.e. $\mathit{kill} \rightarrow \mathbf{O}\, \mathit{kill\ gently}$
If a person kills gently, then the person kills.	i.e. $\mathit{kill\ gently} \rightarrow \mathit{kill}$

Suppose, regrettably, that Smith kills Jones. Then he ought to kill him gently. But, by RM, Smith ought to kill Jones, which contradicts the first obligation, that Smith ought not to kill Jones.

2.5 Conflicting Obligations

Whereas RM seems to allow too many inferences, axiom schema D, because it prevents conflicting obligations, seems to be too restrictive. In particular, it cannot deal with the conflicts that arise in such famous examples as Sartre's Dilemma, where a young man during the Second World War in occupied France is torn between two conflicting obligations:

Join the French resistance.	i.e. $\mathbf{O}\, \mathit{join}$
Stay at home and look after his aged mother.	i.e. $\mathbf{O}\, \mathit{stay}$
Joining and staying are incompatible.	i.e. $\neg$ (join $\wedge$ *stay*)

Together with RM, D implies that these obligations are inconsistent. But as Hilpinen and McNamara [27] put it, such moral dilemmas "seem not only logically coherent but all too familiar".

Sartre's Dilemma is a hard case, but conflicting obligations also arise in more mundane cases. For example:

Don't eat with your fingers.
If you are served cold asparagus, eat it with your fingers.
You are served cold asparagus.
i.e. $\mathbf{O} \neg \mathit{fingers}, \quad \mathit{asparagus} \rightarrow \mathbf{O}\, \mathit{fingers}, \quad \mathit{asparagus}.$

In SDL and most other modal deontic logics, it follows that you should both eat with your fingers and not eat with your fingers, which is clearly impossible. However, intuitively, the first obligation is a general rule, which is defeated by the second obligation, which is an exception to the rule. Horty [32] shows how to formalise such defeasible rules using default logic [57]. Our ALP representation of conflicting obligations, in Section 5, can be viewed, in part, as a variant of Horty's solution,

using Poole's [2] transformation of default rules into strict rules with abductive hypotheses.

3 Abductive Logic Programming

Abduction was identified by Charles Sanders Peirce [49] as a form of reasoning in which assumptions are generated in order to deduce conclusions - for example to generate the assumption q, to deduce the conclusion p, using the belief $q \rightarrow p$. Peirce focused on the use of abductive reasoning to generate explanations q for observations p. In Artificial Intelligence, abduction has also been used for many other purposes, including natural language understanding [28], fault diagnosis [53] and planning [18].

Poole et al. [50] developed a form of abduction, called *Theorist*, and showed that it can be used for non-monotonic, default reasoning - for example, to generate the assumption *normal-bird(tweety),* to deduce *can-fly(tweety),* using the beliefs *bird(tweety)* and $\forall X$ *(bird(X)* $\wedge$ *normal-bird(X)* $\rightarrow$ *can-fly(X))*. Poole [51] showed, more generally, that, by making implicit assumptions, like *normal-bird(X),* explicit, default rules in default logic [57] can be translated into "hard" or "strict" rules in an abductive framework.[1] Bondarenko et al. [9] showed that abduction with an argumentation interpretation can be used to generalize many other existing formalisms for default reasoning. Poole [52] showed that, by associating probabilities with assumptions, abductive logic programs can also represent Bayesian networks.

Abductive logic programming (ALP) [34] is a variant of Theorist, in which:

> the task is to extend a "theory" P, which is a logic program,
> with a set of assumptions $\Delta \subseteq$ A,
> which are *ground* (i.e. variable-free) atomic sentences,
> so that the extended logic program P $\cup \Delta$ both
> solves a goal G and satisfies integrity constraints I.

This characterisation of ALP distinguishes between goals G, which are "one off" and integrity constraints I, which are "persistent". It reflects the historical origins of ALP, in which logic programs are used to solve existentially quantified goals, but are extended with assumptions, which are restricted by integrity constraints, which are universally quantified.

However, in this paper, we employ a variant of ALP in which the emphasis is shifted from logic programs to integrity constraints, which can be arbitrary sentences of first-order logic (FOL), in the spirit of the related framework FO(ID) [16], in which FOL is extended with logic programs, viewed as inductive definitions. Moreover, we do not distinguish formally between goals and integrity constraints and between solving a goal and satisfying integrity constraints.

[1] This translation is similar to the use of an abnormality predicate in circumscription [45], expressing the default rule in the form $\forall X$ *(bird(X)* $\wedge \neg$ *abnormal-bird(X)* $\rightarrow$ *can-fly(X))*.

In this simplified variant of ALP, an *abductive framework* is a triple ⟨P, G, A⟩, where P is a definite clause logic program, G is a set of sentences in FOL, A is set of atomic sentences, and:

the task is to *satisfy* G, by generating *some* $\Delta \subseteq$ A such that
G is true in the minimal model *min*(P $\cup$ Δ) defined by P $\cup$ Δ.[2]

It is the task of satisfying G that gives G its goal-like nature. As mentioned in the Introduction, a sentence in G can represent a personal goal of an individual agent, a social goal of a society of agents, a dictate of a powerful authority, or a physical or logical constraint. It can also represent an observation to be explained. Despite these different uses of sentences in G, they all have the same formal properties; and we use the two terms *goal* and *integrity constraint* interchangeably.

In this paper, we understand the term *goal satisfaction* in a model-theoretic sense, which contrasts with the theorem-proving view in which *goal satisfaction* means that G is a theorem that is a logical consequence of P $\cup$ Δ or of the completion of P $\cup$ Δ [13]. These two different uses of logic, for theorem-proving and for satisfiability, have analogues in modal logic, where there has also been a shift away from theorem-proving to model checking [24] and to model generation [6]. The corresponding shift from a theorem-proving semantics for ALP to a model generation semantics plays an important role in the ALP approach to reasoning about obligations.

3.1 Logic Programs (LP) as Definitions of Minimal Models

In this paper, we restrict attention to simplified *logic programs* that are sets of *definite clauses* of the form:

$$conclusion \leftarrow condition_1 \wedge \ldots \wedge condition_n$$

where *conclusion* and each $condition_i$ is an atomic formula. All variables are implicitly universally quantified with scope the entire clause. If the clause contains no variables and $n = 0$, then the clause is called a *fact* (and written simply as *conclusion*). Otherwise, the clause is called a *rule*. For example, let P1 be the logic program:

P1 :
$$threat(E, T) \leftarrow fire(E, T)$$
$$threat(E, T) \leftarrow flood(E, T)$$
$$fire(e1, 11)$$
$$flood(e2, 13)$$

where the variable E and the constants *e1* and *e2* represent events, and the variable T and the constants 11 and 13 represent time points. P1 consists of two rules and three facts. The first rule is shorthand for the sentence:

$$\forall E, T\,(threat(E, T) \leftarrow fire(E, T))$$

[2]This is similar to the minimisation of abnormality predicates in circumscription. However, circumscription is a sceptical approach, in which the task is to derive sentences that are true in *all* minimal models.

The set of all the facts that can be derived[3] from a definite clause program P represents a unique model *min*(P) of P. For example:

$$min(\text{P1}) = \{fire(e1, 11),\ flood(e2, 13),\ threat(e1, 11),\ threat(e2, 13)\}$$

is a model of P1. In general, a set M of facts (ground atomic sentences) can be viewed as a model-theoretic structure, representing all and only the ground atomic sentences that are true in M. So any fact not in M is false in M. In general, such model-theoretic structures represented by sets of facts are called *Herbrand interpretations*. If P is a definite clause program, then the Herbrand interpretation *min*(P) is a model of P, because every clause in P is true in *min*(P).

Not only is the Herbrand model *min*(P) a model of P, but it is the unique *minimal model* of P, in the sense that *min*(P) $\subseteq$ M' for any other Herbrand model M' of P [17].

We also say, somewhat loosely, that P is a set of *beliefs*, and that *min*(P) is a *model of the world*. This is a different notion from the notion of belief in epistemic logic, in which it is possible to distinguish between a statement p about the world and a statement **B** p of a belief about the world. In our simplified approach, there is no distinction between the "world" and "beliefs" about the world, which are represented in the unadorned form p rather than **B** p.

3.2 Goals

In ordinary logic programming, definite clause programs P are used to solve *definite goals* G, which are existentially quantified conjunctions of atoms. For example, consider the goal:

G1: $\exists$ *E,T threat(E, T).*

Then P1 above solves G1, because $\exists$ *E,T threat(E, T)* is true in *min*(P1). Moreover, G1 is solved by "computing" one or more of the instances *threat(e1,* 11*), threat(e2,* 13*)* of G1 that are true in *min*(P1) and that justify the solution.

In this paper, we exploit the interpretation of logic programs as definitions of minimal models, to generalise goals to arbitrary sentences of FOL. For example, let G2 and G3 be the goals:

G2: $\neg\ \exists E, T(threat(E, T) \wedge (T \neq 11 \vee T \neq 13))$
G3: $\forall E, T(threat(E, T) \rightarrow fire(E, T) \vee flood(E, T))$.

Then P1 satisfies both G2 and G3, because G2 and G3 are both true in *min*(P1). Notice that we use the forward arrow "$\rightarrow$" for implication in FOL, and the backward arrow "$\leftarrow$" for implication in logic programs.

In our application of ALP to the treatment of obligations, we need the expressive power of FOL to represent such goals as:

G4: $\neg\ \exists$ *Agent1, Agent2, Action* (*happens(do(Agent1, Action))* $\wedge$ *harms(Action, Agent2)*)
G5: $\forall$ *Agent, Action* (*promise(do(Agent, Action))*$\rightarrow$ *happens(do(Agent, Action))*)

[3] In general, *min*(P) is the set of all facts derived by exhaustively applying *modus ponens* to the *ground* program obtained from P by replacing all variables in P by variable-free terms.

Here, except for the lack of any explicit representation of less desirable alternatives, G4 expresses a prohibition from doing anyone any harm, and G5 expresses an obligation to keep one's promises.

Although *goals* can have the form of any sentence of FOL, they often have the form:

$$\forall X\ [antecedent \rightarrow \exists Y\ [consequent_1 \vee ... \vee consequent_n]]$$

where X is the set of all variables, including time or action variables, that occur in *antecedent* and Y is the set of all variables, including time or action variables, that occur only in $consequent_1 \vee ... \vee consequent_n$. If $n = 0$, the goal is equivalent to a denial:

$$\forall X \neg\, antecedent$$

Informally speaking, *antecedent* is typically a conjunction of conditions about the past or the present, and each $consequent_i$ is a conjunction of conditions about the present or the future. Goals of this form are a non-modal version of sentences in the temporal modal logic MetateM [6]. They have the desirable property that they can be satisfied without generating complete models, simply by performing actions to make *consequents* true whenever *antecedents* become true [40–42]. For example, the goal:

G6: $\forall E, T1\ [threat(E, T1) \rightarrow \exists\, T2\ [eliminate(E, T1, T2) \wedge T1 < T2 < T1 + 3]]$

expresses that, whenever a threat E is observed at a time *T1*, you eliminate the threat at some future time *T2,* within 3 units of time after *T1*. Notice that, in theory, the number of threats could be unbounded, and the set Δ of actions needed to satisfy G6 could be infinite. This would be difficult for a theorem-proving semantics, but is unproblematic for model generation.

3.3 Integrity Constraints

Logic programs can be used both as programs for performing computation and as databases for query-answering. When a logic program P is used as a database, then query-answering is the task of determining whether a query Q expressed as a sentence in FOL is true in the minimal model of the database, or of generating instances of Q that are true in the minimal model. Such queries do not involve a commitment to the truth of Q. In contrast, *integrity constraints* specify *necessary* properties of the database. In this respect, integrity constraints are like necessary truths in alethic modal logic, and the database P is like a set of contingent truths.

In our application of ALP to the treatment of obligations, we treat a logic program P, describing actions, external events and the consequences of actions and other events, in effect, as a database, and we treat obligations, augmented if necessary with less desirable alternatives, as integrity constraints. These integrity constraints G can be used not only to check whether P complies with G, but also to actively generate an update Δ, so that $P \cup \Delta$ complies with G. There may be many such Δ, and some Δ may be better than others. Consider for example the database/program:

P2 : $threat(E, T) \leftarrow fire(E, T)$
$threat(E, T) \leftarrow flood(E, T)$
$fire(e1, 11)$

Assume that P2 also contains a definition of $<$, implying such facts as $0 < 1$, ..., $11 < 12$, $11 < 13$, etc. Then P2 does not comply with G6 above, because G6 is false in the minimal model of P2:

$$min(\text{P2}) = \{fire(e1, 11), threat(e1, 11), 0 < 1, \ldots, 11 < 12, 11 < 13, \ldots\}$$

However, P2 can be updated, either with $\Delta 1 = \{$*eliminate(e1*, 11, 12*)*$\}$ or with $\Delta 2 = \{$*eliminate(e1*, 11, 13*)*$\}$. Both P2 $\cup\, \Delta 1$ and P2 $\cup\, \Delta 2$ satisfy G6. Intuitively, P2 $\cup\, \Delta 1$ is somewhat better than P2 $\cup\, \Delta 2$, because, everything else being equal, it is better to deal with a problem earlier rather than later.

3.4 Abductive Explanations

The use of abduction to generate explanations of observations treats observations as goals, rather than as facts. Consider, for example, the following simplified causal theory about some of the possible causes of smoke:

$$smoke(E, T + 1) \leftarrow fire(E, T)$$
$$smoke(E, T + 1) \leftarrow prank(E, T)$$

Suppose that *fire* and *prank* are abducible predicates (whose ground instances constitute the set A). Then an observation of *smoke* at time 12 can be represented by a goal such as *smoke(e3*, 12*)* and can be explained either by $\Delta 1 = \{$*fire(e3*, 11*)*$\}$ or by $\Delta 2 = \{$*prank(e3*, 11*)*$\}$. Given no other information, it may be hard to decide whether one explanation is better than the other. In practice, if the situation warrants it, it might be desirable to actively obtain additional observations, to distinguish between the different hypotheses, because the more observations a hypothesis explains the better.

3.5 Reduction of Soft Constraints to Hard Constraints

Data base integrity constraints can be *hard constraints*, which represent physical or logical properties of the application domain, or *soft constraints*, which represent ideal behaviour and states of affairs, but which may nonetheless be violated. However, in ALP, all integrity constraints are hard constraints. Soft constraints need to be represented as hard constraints, by including less desirable alternatives explicitly. This reformulation of soft constraints as hard constraints in ALP is like the Andersonian reduction of deontic logic to alethic modal logic [2], but with the obvious difference that in ALP hard constraints are represented in FOL.

For example, a typical library database [62] might contain facts about books that are held by the library, about eligible borrowers, and about books that are out on loan. Some integrity constraints, for example that a book cannot be simultaneously out on loan and available for loan, are hard constraints, which reflect physical reality. Other constraints, for example that a person is not allowed to keep a book after the return date, are soft constraints, which may be violated in practice.

Because in ALP all integrity constraints are hard constraints, the soft constraint about not keeping a book after the return date:

$$\forall Person, Book, T \neg [overdue(Book, T) \wedge has(Person, Book, T)]$$

needs to be reformulated as a hard constraint, by specifying what happens if the soft constraint is violated. For example:

$$\forall Person, Book, T\ [overdue(Book, T) \wedge has(Person, Book, T) \rightarrow liable\text{-}to\text{-}fine(Person, Book, T)].$$

Reformulated in this way, if some instance of *has(Person, Book, T)* becomes true, then the hard constraint can be made true either by making the corresponding instance of *overdue(Book, T)* false, or by making the corresponding instance of *liable-to-fine(Person, Book, T)* true. Which of the two alternatives is preferable depends on the person and the circumstances.

Similarly, if the integrity constraint G6 above is understood as a soft constraint, then it needs to be reformulated as a hard constraint, by adding one or more additional alternatives. For example:

$$\begin{aligned} \text{G7}: \forall E, T1\ [threat(E, T1) \rightarrow \exists T2\ & [eliminate(E, T1, T2) \wedge T1 < T2 < T1 + 3] \\ & \vee [escape(E, T1, T2) \wedge T1 < T2 < T1 + 5] \\ & \vee [submit(E, T1, T2) \wedge T1 + 4 < T2]] \end{aligned}$$

Somewhat better from a knowledge representation point of view is to rewrite G7 in a more general form, with the aid of an auxiliary predicate, say *deal-with-threat(E, T1)*:

$$\begin{aligned} &\text{G8:}\quad \forall E, T1\ [threat(E, T1) \rightarrow deal\text{-}with\text{-}threat(E, T1)] \\ &\text{P3:}\quad deal\text{-}with\text{-}threat(E, T1) \leftarrow eliminate(E, T1, T2) \wedge T1 < T2 < T1 + 3 \\ &\qquad\quad deal\text{-}with\text{-}threat(E, T1) \leftarrow escape(E, T1, T2) \wedge T1 < T2 < T1 + 5 \\ &\qquad\quad deal\text{-}with\text{-}threat(E, T1) \leftarrow submit(E, T1, T2) \wedge T1 + 4 < T2 \end{aligned}$$

The more general representation G8 is more flexible than G7, because additional alternatives can be added separately as additional sentences to P3, without changing the goal G8.

This way of representing alternatives is similar to the way in which defeasible rules, such as *can-fly(X)* ← *bird(X)*, are turned into strict rules by adding a single extra defeasible condition, such as *normal-bird(X)* or ¬ *abnormal-bird(X)*. The various alternative ways in which a bird can fail to be normal can be represented separately.

The reformulation of soft constraints as hard constraints is also like the Andersonian reduction. However, while the Andersonian reduction employs a single propositional constant *s*, representing a single, general, abstract sanction, the ALP reductions of both soft constraints to hard constraints and of default rules to strict rules, employ an additional condition, such as *deal-with-threat(E, T1)* or *normal-bird(X),* which is specific to the constraint or rule to which it is added.

We maintain that obligations and prohibitions are similar. They can be hard constraints, which are inviolable, or they can be soft constraints, whose violations are represented explicitly as less desirable alternatives. For example, the prohibition "do not steal" can be represented literally as a hard constraint, which admits no alternatives:

$$\forall\ Agent, T \neg steal(Agent, T)$$

But if instead the prohibition is understood as a soft constraint, acknowledging that stealing can happen but should be discouraged, then it needs to be represented as a

hard constraint, with additional, less desirable alternatives represented explicitly. For example:

$$\forall\, Agent, T1\ [\neg steal(Agent, T1) \vee \exists\, T2\ [punished(Agent, T2) \wedge T1 < T2]]$$

equivalently:

$$\forall\, Agent, T1\ [steal(Agent, T1) \rightarrow \exists\, T2\ [punished(Agent, T2) \wedge T1 < T2]]$$

In contrast with the Andersonian reduction, which applies to all obligations, whether they can be violated or not, the ALP reduction applies only to obligations that can be violated, and to other soft constraints more generally. Hard constraints, whether they represent necessary properties of the problem domain or inviolable patterns of behaviour, are represented literally, without the addition of any less desirable alternatives.

4 The Separation of Goals from Preferences in ALP

Not only does the Andersonian reduction, $\mathbf{O}\ p \leftrightarrow \mathbf{N}\ (\neg\ p \rightarrow\ s) \leftrightarrow \mathbf{N}\ (p \vee s)$, treat the disjunction $p \vee s$ as a hard constraint, but by defining $\mathbf{O}\ p$ in terms of $p \vee s$ it also builds into the semantics a preference for p over s. Van Benthem et al. [66] generalises this simple preference into a more general binary relation M1 $\leq$ M2 between possible worlds M1 and M2, representing that M2 is at least as good as M1.

Our model-theoretic semantics of ALP, when applied to normative tasks, similarly employs a preference ordering among minimal models, which are like possible worlds, but the ordering is separate from and external to the logic. This separation of goals from preferences is an inherent feature of abductive reasoning, where generating possible explanations is a distinct activity from preferring one explanation to another. It is also a feature of most problem-solving frameworks in Artificial Intelligence and constrained optimization e.g. [15], where it is standard practice to separate the specification of constraints from the optimisation criteria. In ALP, this separation has the advantage of simplifying the logic, because the semantics does not need to take preferences into account.

4.1 The Map Colouring Problem

The following variation of the map-colouring problem illustrates the separation of goals from preferences in constrained optimisation. The problem can be formulated in deontic terms, say, as instructions to a person colouring a map. Given a map of countries and an assortment of possible colours:

Every country **ought** to be assigned a colour.
It is **forbidden** to assign two different colours to the same country.
It is **forbidden** to assign the same colour to two adjacent countries.

For simplicity, assume that these are hard constraints, so we don't have to worry about how to deal with failures of compliance.

In ALP, the map can be represented by a logic program P, defining the predicates *country(X)* and *adjacent(X, Y)*. The possible actions of assigning a colour C to a country X can be represented by a set A of candidate assumptions, represented by atomic sentences of the form *colour(X, C)*. The goal G is a set (or conjunction) of first-order sentences:

$$\forall X\ [country(X) \rightarrow \exists\, C\ colour(X, C)]$$
$$\forall X, C1, C2\ [colour(X, C1) \wedge colour(X, C2) \rightarrow C1 = C2]$$
$$\forall X, Y, C\ \neg\ [adjacent(X, Y) \wedge colour(X, C) \wedge colour(Y, C)]$$

In addition, P needs to include a definition $X = X$ of the identity relation. A solution is a set $\Delta \subseteq$ A, assigning colours to countries, such that G is true in *min*(P $\cup\, \Delta$). There are exactly two such minimal models for a simple map with two adjacent countries *iz* and *oz* and two colours *red* and *blue*, where A = {*colour(iz, red), colour(iz, blue), colour(oz, red), colour(oz, blue)*}. Ignoring the extension of the identity relation, the two models are:

$$\text{M1} = \{country(iz), country(oz), adjacent(iz, oz), colour(iz, red), colour(oz, blue)\}$$
$$\text{M2} = \{country(iz), country(oz), adjacent(iz, oz), colour(iz, blue), colour(oz, red)\}$$

For a more complicated map with many countries and many colours there would be many more solutions.

So far, there is not much difference between the modal and the ALP representations. But now suppose that it is deemed desirable to colour the map using as few colours as possible. In ALP and other problem-solving frameworks, this optimisation criterion could be formalised by means of a cost function, which is represented separately, possibly in a metalanguage, as in [59, 60]. Such cost functions are employed in search strategies such as branch and bound, to generate solutions incrementally by successive approximation. Suboptimal solutions found early in the search are used as a bound to abandon partial solutions that are already worse than the best solution found so far. (For example, if a solution has already been found using five colours, then there is no point trying to extend a partial solution that already uses five colours.) Once a solution has been found, whether it is optimal or not, the search can be terminated with the best solution found so far. (Or it can be acted upon tentatively, until a better solution has been found.) Such *anytime* problem solving strategies are essential for practical applications.

Using deontic logic, it would be necessary to incorporate the optimisation criterion (fewest colours, in this example) into the object level statement of the goal (colour the map, subject to the constraints). It is hard to see how this could be done; and, even if it could, it is hard to see how the resulting deontic representation would then be used to find solutions for difficult problems in practice.

4.2 Decision Theory

The separation of goals from preferences, which is inherent in ALP, is also built into the foundations of decision theory, which treats reasoning about goals and beliefs as

a separate activity from making decisions about actions. As Jonathan Baron [5] puts it in his textbook, *Thinking and Deciding* (page 6):

> Decisions depend upon beliefs and goals, but we can think about beliefs and goals separately, without even knowing what decisions they will affect.

Conversely, classical decision theory is concerned with choosing between alternative actions, without even considering the goals that motivate the actions and the beliefs that imply their possible consequences. Normative decision theory, which is concerned with maximising the utility (or goodness) of the expected consequences of actions, is a theoretical ideal, against which other, more practical, prescriptive approaches can be evaluated. Baron [5, page 231] argues that the fundamental normative basis of decision making (namely, maximising the utility of consequences) is the same, whether it is concerned with the personal goals of individual agents or with moral judgements concerning others.

Arguably, classical decision theory, which not only separates thinking about goals and beliefs from deciding between actions but ignores the relationship between thinking and deciding, is too extreme. Deontic logic is the opposite extreme, entangling in a primary obligation $\mathbf{O}\ p$ and a secondary obligation (in one or other of the forms $\mathbf{O}\ (\neg p \rightarrow q)$, $\neg p \rightarrow \mathbf{O}\ q$ or $\mathbf{O}\ (q\ /\ \neg\ p)$) the representation of a goal $p \vee q$ together with a preference for one alternative, p, over another, q. In contrast with these two extremes, ALP, like practical decision analysis [25, 36] separates thinking about goals and beliefs from deciding between alternative actions, but without ignoring their relationship.

4.3 Algorithm = Logic + Control

The separation of goals from preferences in the ALP approach to reasoning about norms is analogous to the separation of logic from control in the logic programming approach to reasoning about algorithms [37]. Consider, for example, the English language procedure for alerting the driver of a train to an emergency on the London underground [38]:

> *In an emergency, press the alarm signal button, to alert the driver.*

It might be tempting to represent the sentence as an anankastic conditional [67] in a modal logic, for example as:

> *If there is an emergency and you **want** to alert the diver,*
> *then you **should** press the alarm signal button.*

However, the same procedure can also be understood both logically and more simply as a definite clause:

> *The driver is alerted to an emergency, if you press the alarm signal button.*

together with an indication that the clause should be used *backwards* to reduce a goal matching the conclusion of the clause to the sub-goals corresponding to the conditions of the clause. The use of the imperative verb *press* in the English sentence suggests that the belief represents the preferred method for achieving the conclusion.

There are of course other ways of trying to alert the driver, like crying out loud, which might also work, and which might even be necessary if the preferred method fails. For example:

A person is alerted to a possible danger,
if you cry out loud and the person is within earshot.

In ALP agents [40], logic programs are used both backwards, to reduce goals to subgoals, and forwards, to infer logical consequences of candidate actions.

For example, the following English sentence can also be read as a definite clause:

You are liable to a fifty pound penalty,
if you use the alarm signal button improperly.

The clause can be used backwards or forwards. But read as an English sentence, its clear intention is to be used in the forward direction, to derive the likely, undesirable consequence of using the alarm signal button when there isn't an emergency. However, there is nothing to prevent a person from using the clause backwards, if he perversely wants to incur a penalty, or if he wants to use the penalty for some other mischievous purpose.

Different logic programming languages employ different control strategies. Some logic programming formalisms, including Datalog and Answer Set Programming, are entirely declarative, leaving the issue of control to the implementation, beyond the influence of the "programmer". Prolog, on the other hand, uses clauses backwards as goal-reduction procedures, and tries them one at a time, sequentially, in the order in which they are written. By determining the order in which clauses are written, the programmer can impose a preference for one goal-reduction procedure over another. For example, the order in which the clauses are written in the earlier program P3 prefers eliminating a threat, over escaping from the threat, over submitting to the threat.

The sequential ordering of alternatives, as in Prolog, is sufficient for many practical applications, and it has been developed in one form or another in many other logical frameworks. For example, Brewka et al. [10] employ a non-commutative form of disjunction to indicate "alternative, ranked options for problem solutions". In the domain of deontic logic, Governatori and Rotolo [23] employ a similar, non-commutative modal connective $a \otimes b$, to represent a as a primary obligation and b as a secondary obligation if a is violated. Sequential ordering is also a common conflict resolution strategy in many production system languages.

4.4 Production Systems

Production systems have been used widely for modelling human thinking in cognitive science, and were popular for implementing expert systems in the 1980s. In recent years, they have been used in many commercial systems for representing and executing business rules.

A production system is a set of condition-action rules (or production rules) of the form *IF conditions THEN actions*, which can be understood either imperatively as expressing that if the *conditions* hold then **do** the actions, or in deontic terms as expressing that if the *conditions* hold then the *actions* **should** be performed.

The *IF-THEN* form of production rules is not the *if-then* of logical implication. Part of the reason for this is that contrary *actions* may be required when the *conditions* of different rules hold simultaneously, as in the case of the rules:

IF there is a threat THEN eliminate the threat.
IF there is a threat THEN escape from the threat.
IF there is a threat THEN submit to the threat.

If *IF-THEN* were logical *if-then*, then this would be logically equivalent to:

IF there is a threat THEN
eliminate the threat AND escape from the threat AND submit to the threat.

which is not physically possible. Production systems use "conflict resolution" strategies, to decide between such contrary actions.

In this example, the production rules can be reformulated in logical terms by replacing *AND* by *OR*, and by treating the resulting sentence as a goal to be made true. Conflict resolution then becomes a separate strategy for choosing between alternatives [40]. The ALP reconstruction of production systems in [40] is similar to the ALP reconstruction of deontic logic proposed in this paper.

Production rules of the form *IF conditions THEN actions* are purely reactive. The *actions* are performed only after the *conditions* have been recognised. But in ALP goals of the logical form *antecedent*→*consequent* can be made true in any way that conforms to the truth table for material implication. They can be made true reactively, by making *consequent* true when *antecedent* becomes true; preventatively, by making *antecedent* false, avoiding the need to make *consequent* true; or proactively, by making *consequent* true without waiting for *antecedent* to become true [38, 41, 42]. These alternative ways of making *antecedent*→*consequent* goals true is once again a separate matter of preferring one model over another.

In many cases, it is possible to identify the best way of making goals true, at "compile time", before they need to be considered in practice; and, for this purpose, it is often sufficient to order the rules sequentially in order of preference. But in other cases, it is better to decide at "run time", taking all the facts about the current situation into account. Separating goals from preferences, as in ALP, leaves these options open, whereas combining goals and preferences inextricably into the syntax and semantics of the logic, as in modal deontic logic (and Prolog), forces decisions to be made at compile time and closes the door to other, more flexible possibilities.

4.5 Normative ALP Frameworks

In this paper, we are neutral about the manner in which preferences are specified, and assume only that the specification of preferences induces a strict partial ordering $<$ between models, where M $<$ M' means that M' is *better than* M. The ordering can be defined directly on the models themselves, or it can be induced more practically by a cost function, by an ordering of clauses or rules, or by a priority ordering of candidate assumptions A. In particular, it can take into account that "other things being equal" it is normally better to make the consequents of conditional goals true earlier rather than later.

Given such an ordering $<$, we can define what it means to satisfy an abductive framework as well as possible. A *normative ALP framework* is a tuple of the form $\langle$P, G, A, $<\rangle$, where P is a set of definite clauses, G is a set of sentences in FOL, A is set of ground atomic sentences and $<$ is a partial ordering between Herbrand interpretations, where:

the task is to *satisfy* $\langle$P, G, A, $<\rangle$ by generating *some* $\Delta \subseteq$ A such that
G is true in M = *min*(P $\cup$ Δ) and there does not exist any Δ' $\subseteq$ A
such that G is true in M' = *min*(P $\cup$ Δ') and M $<$ M'.

If there are several such best Δ that satisfy $\langle$P, G, A, $<\rangle$, then an agent can choose freely between them. But, if because of limited computational resources the agent is unable to generate a best Δ, then the definition can nonetheless serve as a normative ideal, against which other more practical solutions can be compared.[4]

5 ALP as a Deontic Logic

The ALP distinction between logic programs, representing beliefs, and integrity constraints, representing goals, can be viewed as a weak modal logic, in which beliefs p are expressed without a modal operator (and are not distinguished from what is actually the case), and goals p are implicitly prefixed with a modal operator, **O** p, expressing that p *must* be true. Viewed in this way, there are no nested modalities, and there are no mixed sentences, such as $p \rightarrow$ **O** q.

Although these syntactic restrictions may seem very limiting, they are shared with several other approaches to deontic logic, such as that of Horty [29, 31]. Moreover, they are also shared with the deontic modal logic of SBVR (Section 6), which has been developed specifically to deal with practical applications.

Arguably, the syntactic restrictions of ALP have an advantage over the more liberal syntax of modal deontic logics, because there is no need to choose between different ways of representing conditional obligations, which in effect are all represented implicitly in the same form **O** $(p \rightarrow q)$.

Although the main focus in ALP is on satisfying goals as well as possible, we can define a notion of logical consequence, following the lead of Horty [29, 31] building on van Fraassen [68], and referred to as vFH below.

5.1 The van Fraassen-Horty (vFH) Non-Modal Deontic Logic

As in the ALP approach, vFH restricts attention to obligations of the form **O** p, where p does not include the modal operator **O**. According to the basic version of vFH [29, Theorem 3], if G is a set of sentences of ordinary classical logic, then:

O p is a logical consequence of {**O** q | $q \in$ G} if and only if,

[4] As in preference-based deontic logics, we may want to exclude infinite sequences of increasingly better Δ. Alternatively, we may accept that there is no absolutely best Δ, and simply generate the best Δ possible in the given circumstances.

there exists *some* G' $\subseteq$ G such that
G' is consistent and p is a non-modal logical consequence of G',
i.e. p is true in all classical models of G'.

This is a "credulous semantics", (because of the qualification *some* G'), which is more like the notion of satisfaction than like the usual notion of logical consequence. However, Horty [29, 31] significantly extends this basic semantics, reformulating it in default logic with priorities between default rules, and considering both credulous and sceptical variants. Our notion of logical consequence in ALP, presented in the next section, 5.2, is an adaptation of Horty's "sceptical semantics".

Horty [29] claimed that, at the time, the van Fraassen [68] proposal was "the only intuitively adequate account of reasoning in the presence of normative conflicts". Horty [31] illustrates the treatment of normative conflicts in the prioritised default logic version of vFH with the example of Section 2.3, where *antecedent* $\Rightarrow$ *consequent* represents a default rule, whose meaning is that if *antecedent* holds and *consequent* is consistent then *consequent* holds by default:

Don't eat with your fingers.
If you are served cold asparagus, eat it with your fingers.
i.e *true* $\Rightarrow \neg$ *fingers*, *asparagus* $\Rightarrow$ *fingers*

Here the second rule has priority over the first. Horty shows that, in both its credulous and sceptical versions, the default theory implies both **O** $\neg$ *fingers* and **O** (*fingers* /*asparagus*). Both of these logical consequences also hold when the same example is formulated in dyadic deontic logics. But the deontic logic formulations also imply the intuitively unintended consequence **O** $\neg$ *asparagus*, which is not implied by the default theory with the vFH semantics. The default logic is defeasible, because, given the additional, "hard" information *asparagus*, the obligation **O** $\neg$ *fingers* no longer holds, and the contrary obligation **O** *fingers* holds instead. We give an ALP representation of the example in Section 5.3.

5.2 Normative ALP Frameworks and Implied Obligations

The vFH semantics can be adapted to ALP by defining **O** p to be a logical consequence of a normative abductive framework $\langle$P, G, A, $<\rangle$ as meaning that p is true in *all* best models of G. More formally:

O p is a *logical consequence* of $\langle$P, G, A, $<\rangle$ if and only if, for *all* $\Delta \subseteq$ A,
if G is true in M = *min*(P $\cup$ Δ) and there does not exist any $\Delta' \subseteq$ A
such that G is true in M' = *min*(P $\cup$ Δ') and M $<$ M',
then p is true in M.

The modal operators **F** for prohibition and **P** for permission can be defined in terms of obligation **O**:

$\langle$P, G, A, $<\rangle$ *implies* **F** p if and only if G implies **O** $\neg$ p.
$\langle$P, G, A, $<\rangle$ *implies* **P** p if and only if G does not imply **O** $\neg$ p.

Viewed in vFH terms, this is a *sceptical* semantics, because, for **O** p to be a logical consequence of $\langle$P, G, A, $<\rangle$, p must be true in *all* best models of G. In contrast, the

semantics of satisfying obligations is *credulous*, because, to satisfy ⟨P, G, A, <⟩ it suffices to generate *some* best model of G.

The main difference between the vFH and ALP approaches are that in vFH obligations are soft constraints, but in ALP they are hard constraints. In addition, vFH represents conditional obligations in the dyadic form $\mathbf{O}\ p/q$, but ALP represents them, in effect, in the form $\mathbf{O}\ (q \rightarrow p)$ with ordinary material implication.

Prakken [56] proposes an alternative approach, which is also based on default logic, but is combined with SDL. He argues that, by comparison, the vFH approach has several limitations. Perhaps the most serious is that all defaults in vFH are deontic defaults, but that "factual" defaults are also necessary. This limitation does not apply to the ALP approach, because ALP combines goals/constraints, to represent deontic defaults, with logic programs, to represent factual defaults.

Prakken also points out that the vFH approach can represent only weak permissions $\mathbf{P}\ p$, which hold implicitly when $\mathbf{O} \neg p$ does not hold. SDL and many other deontic logics can also represent strong permissions. The difference is that, if $\mathbf{P}\ p$ is a weak permission, and $\neg p$ later becomes obligatory, then there is no conflict, because the weak permission $\mathbf{P}\ p$ simply no longer holds. But if $\mathbf{P}\ p$ is a strong permission, then the presence or introduction of the obligation $\mathbf{O} \neg p$ introduces a normative conflict.

This limitation of vFH can be avoided in ALP by treating a strong permission as an obligation not to apply a sanction, as proposed by Anderson [3] and developed further by Asher and Bonevac [4]. For example, we can represent the situation in which no vehicles are allowed in the park, but authorised vehicles are permitted. by the goals:

$$\textit{vehicle} \rightarrow \textit{liable to fine}$$
$$\textit{vehicle} \wedge \textit{authorized} \rightarrow \neg\ \textit{liable to fine}$$

This captures the normative conflict between an obligation and a permission, which is the defining characteristic of strong permission, but it does not capture the more common use of strong permission to override an obligation. For this, we need to represent the obligation and permission as a rule and an exception:

$$\neg\ \textit{exception} \wedge \textit{vehicle} \rightarrow \textit{liable to fine}$$
$$\textit{vehicle} \wedge \textit{authorized} \rightarrow \neg\ \textit{liable to fine}$$
$$\textit{exception} \leftarrow \textit{authorized}$$

In the remainder of this section we briefly show how normative ALP frameworks deal with the other problems of deontic logic presented earlier in the paper.

5.3 Normative Conflicts in ALP Frameworks

Here is a normative framework ⟨P, G, A, <⟩ corresponding to the vFH example in Section 5.1:

P = {*exception* ← *asparagus*},
G = {¬ *exception* ∧ *fingers* → *sanction*, *asparagus* → *fingers*}
A = {*fingers, asparagus, sanction*}
M < M' if *sanction* ∈ M and *sanction* ∉ M'.

In this representation, the soft obligation not to eat with fingers is reformulated as a hard obligation by adding both a sanction and an extra condition ¬ *exception*. But for simplicity the obligation to eat asparagus with fingers is treated as a hard obligation, without the addition of any sanctions or exceptions. The only best models that satisfy G are M1 = {} and M2 = {*asparagus, exception, fingers*}. Consequently, as in the vFH representation, ⟨P, G, A, <⟩ implies both **O** (¬ *exception* →¬ *fingers*) and **O** (*asparagus* → *fingers*), but not **O** (¬ *asparagus*).

Suppose, however, that eating cold asparagus with fingers is not an obligation, but simply a strong permission overriding the obligation not to eat with fingers. Then it suffices to replace the goal *asparagus* → *fingers* by the goal *asparagus* → ¬ *sanction.* There are then three best models that satisfy the new goals, the two models, M1 and M2, above plus the additional model M3 = {*asparagus, exception*}, in which the strong permission is not exercised.

5.4 Sartre's Dilemma

In the previous example, the obligation to eat asparagus with fingers is an exception to an obligation that holds as general rule. But in Sartre's Dilemma, the two obligations are incomparable. This can be represented by ⟨P, G, A, <⟩ where:

P = {}, G = {*join* ∨ *sanction1,* *stay* ∨ *sanction2*, ¬ (*join* ∧ *stay*)}
A = {*join, stay, sanction1, sanction2*}
M < M' if M contains more of the sanctions, *sanction1* and *sanction2*, than M'.

There are three models that satisfy the goals: M1 = {*join, sanction2*}, M2 = {*stay, sanction1*} and M3 = {*sanction1, sanction2*}. All of these models involve sanctions, so are less than ideal. But none the less, there are two equally best models, M1 and M2. *join* ∨ *stay* is true in both of these. So **O** (*join* ∨ *stay*) is a logical consequence.

5.5 Ross's Paradox

Suppose that *mail* is a hard constraint in the framework ⟨P, G, A, <⟩ where:

P = {}, G = {*mail,* ¬ (*mail* ∧ *burn*)},
A = {*mail, burn*}, and < = {}.

Then M = {*mail*} is the only minimal model that satisfies G. But *mail* ∨ *burn* is true in M. So ⟨P, G, A, <⟩ implies both **O** *mail* and **O** (*mail* ∨ *burn*) as logical consequences. But it is not possible to make *mail* true by making *burn* true, because there is no model that satisfies ⟨P, G, A, <⟩ and also contains the action *burn*.

Suppose, more realistically, that *mail* is really a soft constraint, which is represented as a hard constraint *mail* ∨ *sanction* in the framework ⟨P', G', A', <'⟩ where:

P' = {}, G' = {*mail* ∨ *sanction*, ¬ (*mail* ∧ *burn*)},
A' = {*mail, burn, sanction*} and
M <' M' if *sanction* ∈ M and *sanction* ∉ M'.

There is only one best model that satisfies the modified framework, namely the same model M = {*mail*}, as before. So, as in the simpler framework ⟨P, G, A, <⟩, the more realistic framework ⟨P', G', A', <'⟩ implies both **O** *mail* and **O** (*mail* ∨ *burn*). But it is not possible to satisfy the obligation *mail* by performing the action *burn*, because there is no best model that contains the action *burn*.

So, no matter whether *mail* is regarded as a hard or soft constraint, **O** (*mail*) implies **O** (*mail* ∨ *burn*), but in neither case does generating a model that makes *burn* true satisfy the obligation **O** (*mail*). Viewed in this way, Ross's Paradox is not a paradox at all, but rather, as Fox [20] also argues, a confusion between satisfying an obligation and implying the obligation as a logical consequence of other obligations. Arguably, the "paradox" also suggests that the focus in deontic logic on inferring logical consequences is misdirected, and that it should be directed towards satisfying obligations instead.

5.6 The Good Samaritan Paradox

Suppose, as do Hilpinen and McNamara [27], that the obligation to help Smith, who has been robbed, is represented as **O** (*rob* ∧ *help*). We can represent this by the framework ⟨P, G, A, <⟩ where:

$$\mathrm{P} = \{\}, \mathrm{G} = \{rob \wedge help\}, \mathrm{A} = \{rob, help\} \text{ and } < = \{\}.$$

There is only one minimal model M = {*rob, help*} that satisfies ⟨P, G, A, <⟩. It follows that ⟨P, G, A, <⟩ implies **O** *rob*, which is the root of the paradox.

But surely this is a misrepresentation. As Forrester [19] and others have pointed out, the obligation to help Smith, who has been robbed, is more faithfully represented by *rob* ∧ **O** *help*. This can be represented in turn by the framework ⟨P', G', A', <'⟩:

$$\mathrm{P'} = \{rob\}, \mathrm{G'} = \{help\}, \mathrm{A'} = \{help\} \text{ and } <' = \{\}.$$

It is still the case that M = {*rob, help*} is the only minimal model that satisfies the representation, and it is still the case that the representation implies **O** *rob*. However, it is not the case that it is necessary to generate a model of the form *min*(P' ∪ Δ) in which *rob* ∈ Δ, in order to satisfy ⟨P', G', A', <'⟩.

But arguably even this improved representation is inadequate. A truly good Samaritan is one who comes to a person's aid whenever a person needs it. Sticking to a propositional representation for simplicity and for ease of comparison, this can be represented by the framework ⟨P", G", A", <"⟩ where:

$$\mathrm{P"} = \{\}, \mathrm{G"} = \{rob \rightarrow help\}, \mathrm{A"} = \{rob, help\} \text{ and}$$
$$\mathrm{M} <" \mathrm{M'} \text{ if } rob \in \mathrm{M} \text{ and } rob \notin \mathrm{M'}$$

There are three minimal models that satisfy this framework, corresponding to the three ways of making the material implication *rob* → *help* true:

$$\mathrm{M1} = \{\}, \mathrm{M2} = \{help\}, \mathrm{M3} = \{rob, help\},$$

With the given preference relation, both M1 and M2 are equally best. So the revised framework ⟨P", G", A", <"⟩ implies **O** (*rob* → *help*) and even **O** (¬ *rob*), but does not imply either **O** (*rob*) or **O** (*help*).

Suppose, however, that we now observe that Smith has just been robbed, and we treat the observation simply as a fact to be accepted, adding *rob* to P", obtaining the updated framework ⟨P" ∪ {*rob*}, G", A", <"⟩. M3 is now the only (and best) minimal model that satisfies the updated framework, which implies **O** (*rob* → *help*), **O** (*rob*) and **O** (*help*).

The implication **O** (*rob*) is undoubtedly unintuitive. None the less, it faithfully reflects the definition of logical consequence, which restricts the set of possible models satisfying G in a framework ⟨P, G, A, <⟩ to models that also satisfy P by construction. But if, as in this case, some sentence $p \in$ P is already true, then it is not necessary to satisfy G by generating a model that *makes* p true, by adding p to Δ. So although p is true in all models that satisfy G and therefore **O** (p) is a logical consequence, it is not the case that it is necessary or obligatory to make p true.

The moral of the story is that, as in Ross's Paradox, goal satisfaction is more appropriate than logical consequence for reasoning about obligations. Moreover, the moral does not depend upon possible sanctions or exceptions. So it applies as much to deontic logic as it does to ALP.

5.7 Chisholm's Paradox

We will consider two representations. The first representation ⟨P, G, A, <⟩ does not involve any sanctions or exceptions, but represents the obligation for Jones to go to the assistance of his neighbour simply as a preference for making the material implication *go* → *tell* true by making both its antecedent and consequent true, over making its antecedent false:

$$\text{P} = \{\}, \text{G} = \{go \rightarrow tell, \quad \neg go \rightarrow \neg tell\}, A = \{go, tell\} \text{ and}$$
$$\text{M} < \text{M'} \text{ if } go \notin \text{M and } go \in \text{M'}.$$

M1 = {} and M2 = {*go, tell*} are the only minimal models that make G true, but M2 is better than M1. So ⟨P, G, A, <⟩ implies **O** (*go*) and **O** (*tell*), as is intuitively correct.

Now suppose we observe that Jones doesn't go. We can represent this simply by removing *go* from the set of candidate assumptions, obtaining the updated framework ⟨P, G, A – {*go*}, <⟩. The only (and best) minimal model that satisfies the updated framework is the less than ideal model M1 = {}, which implies **O** (¬ *go*) and **O** (¬ *tell*).

A similar result is obtained by systematically transforming the modal representation from soft constraints into hard constraints by introducing Bohnert-Andersonian-style sanctions, obtaining the framework ⟨P', G', A', <'⟩ where:

$$\text{P'} = \{\}, \text{G'} = \{go \vee sanction1,\ go \rightarrow tell \vee sanction2,\ \neg go \rightarrow \neg tell \vee sanction3\}$$
$$\text{A'} = \{go, tell, sanction1, sanction2, sanction3\} \text{ and}$$
$$\text{M} <' \text{ M' if M' contains fewer sanctions } \{sanction1, sanction2, sanction3\} \text{ than M.}$$

There is only one best minimal model, M1 = {*go, tell*}, which contains no sanctions; and **O** (*go*) and **O** (*tell*), as before. If we now observe that Jones doesn't go, then we update the framework to ⟨P,' G', A' – {*go*}, <'⟩. The only best model that satisfies the updated framework is the less than ideal model {*sanction1*}, which implies **O** (¬ *go*) and **O** (¬ *tell*) as before.

5.8 Forrester's Paradox

As in the Chisholm paradox, we consider two propositional representations, one with sanctions and the other without. The simpler representation ⟨P, G, A, <⟩ without sanctions represents the obligation not to *kill* as a preference for making the goal *kill* → *kill gently* true by making *kill* false. In addition, we need to express that *killing gently* and *killing violently* are mutually exclusive alternative ways of making *kill* true:

$$\begin{aligned} \text{P} &= \{\textit{kill} \leftarrow \textit{kill gently}, \quad \textit{kill} \leftarrow \textit{kill violently}\} \\ \text{G} &= \{\textit{kill} \rightarrow \textit{kill gently}, \quad \neg(\textit{kill gently} \wedge \textit{kill violently})\} \\ \text{A} &= \{\textit{kill gently}, \quad \textit{kill violently}\} \text{ and} \\ \text{M} &< \text{M' if } \textit{kill} \in \text{M and } \textit{kill} \notin \text{M'}. \end{aligned}$$

There is only one best minimal model M1 = {} that makes G true. So ⟨P, G, A, <⟩ implies **O** (*kill* → *kill gently*) and **O** (¬ *kill*), as is intuitively correct.

However, if we observe *kill* and update the framework to ⟨P ∪ {*kill*}, G, A, <⟩, then M2 = {*kill gently, kill*} is now the only (and best) model that makes G true, and **O** (*kill gently*) is a consequence. If instead we observe *kill violently* and update the framework to ⟨P ∪ {*kill violently*}, G, A, <⟩, then the goals are unsatisfiable, because there is no longer any model that makes the goals true.

The situation is similar if we represent the example with sanctions by the framework ⟨P', G', A', <'⟩ where:

$$\begin{aligned} \text{P'} &= \text{P} = \{\textit{kill} \leftarrow \textit{kill gently}, \quad \textit{kill} \leftarrow \textit{kill violently}\} \\ \text{G'} &= \{\neg \textit{kill} \vee \textit{penalty}, \quad \textit{kill} \rightarrow \textit{kill gently} \vee \textit{severe penalty}, \\ &\quad \neg(\textit{kill gently} \wedge \textit{kill violently})\} \\ \text{A'} &= \{\textit{kill gently}, \quad \textit{kill violently}, \quad \textit{penalty}, \quad \textit{severe penalty}\} \\ \text{M} &<' \text{M' if } \textit{penalty} \in \text{M and } \textit{penalty} \notin \text{M'} \\ \text{M} &<' \text{M' if } \textit{severe penalty} \in \text{M and } \textit{penalty} \in \text{M'}. \end{aligned}$$

There is only one best minimal model, M1 = {}, which contains no penalties. So both **O** (*kill* → *kill gently*) and **O** (¬ *kill*), as before. If we now observe *kill* and update the framework to ⟨P' ∪ {*kill*}, G', A', <'⟩, then the only best model is now the less than ideal model M2 = {*kill gently*, *kill*, *penalty*}. So the framework implies **O** (*kill gently*), as before.

If instead we observe *kill violently*, then there is a best model, M3 = {*kill violently*, *kill*, *penalty, severe penalty*}, and **O** (*kill violently*) is a logical consequence. But this

doesn't mean it is necessary to *satisfy* the goals by making *kill violently* true, because *kill violently* is already and unavoidably true.

6 Comparison with SBVR (Semantics of Business Vocabulary and Rules)

In Section 5, we considered some of the problems of deontic logic that have been investigated in the field of philosophical logic. In this section, we consider some of the issues that arise when deontic logic is applied to practical applications. SBVR, which is based on predicate logic extended with deontic and alethic modal operators, has been adopted by the Object Management Group (OMG) for specifying the vocabulary and rules of complex organisations, "for business purposes, independent of information systems designs" [61, page 3]. Despite its use of modal logic, SBVR has many properties that are similar to the ALP approach in this paper. In SBVR:

> "most statements of business rules include only one modal operator, and this operator is the main operator of the whole rule statement. For these cases, we simply tag the constraint as being of the modality corresponding to its main operator, without committing to any particular modal logic" [61, p. 108].

This simplified, tagged form of modal logic in SBVR is similar to the "tagging" of sentences in ALP as either goals or beliefs. Sentences tagged as obligations in SBVR correspond to goals in ALP, and sentences tagged as necessities in SBVR correspond in ALP to beliefs (representing definitions). The correspondence is not exact because "goals" in ALP include some integrity constraints that would be tagged in SBVR by an alethic modal operator representing necessity.

The authors of the SBVR standard observe that sentences that are not naturally expressed in this simplified, tagged form of modal logic can often be rewritten in this form. For example:

> "For each Person, it is obligatory that that Person is a husband of at most one Person" can be rewritten as "It is obligatory that each Person is a husband of at most one Person" [61, p. 109].

> "For each Invoice, if that Invoice was issued on Date1 then it is obligatory that that Invoice is paid on Date2 where Date2 $<=$ Date1 $+$ 30 days" can be rewritten as "It is obligatory that each Invoice that was issued on Date1 is paid on Date2 where Date2 $<=$ Date1 $+$ 30 days"[5] [61, p. 116].

This similar use of tagging in both ALP and SBVR supports the thesis that goals in non-modal ALP are adequate for representing the goal component of obligations in many practical applications.

[5]To be more precise, the quantification of Date1 and Date2 should be specified, i.e. **It is obligatory that for each** Invoice **and for each** Date1, if the Invoice is issued on Date1, then **there exists some** Date2 on which the Invoice is paid **where** Date2 $<=$ Date1 $+$ 30 days.

In addition to tagging sentences with modal operators, SBVR also specifies levels of enforcement to deal with violations:

> "Depending on enforcement level, violating the rule could well invite response, which might be anything from immediate prevention and/or severe sanction, to mild tutelage" [61, page 171].

These enforcement levels are:

> "a position in a graded or ordered scale of values that specifies the severity of action imposed in order to put or keep an operative business rule in force" [61, page 176].

To the best of our knowledge, such responses to violations are not represented in the SBVR formalism. This avoids the problems associated with contrary-to-duty obligations, which arise in ordinary deontic logics, and which are addressed with ALP in this paper.

7 Abductive Expectations in SCIFF

The approach that is most closely related to the one in this paper is that of Alberti et al. [1], which maps deontic operators into abducible deontic predicates in ALP, using a proof procedure SCIFF, based on the IFF proof procedure of [21]. Here is a simplified variant of an example from [1], representing an obligation and a prohibition as integrity constraints:

$$\begin{aligned} &\forall A, B, Info, D, T1\,[\mathbf{H}(query(A, B, Info, D), T1) \\ \rightarrow\ &\exists Answer, T2\,[[\mathbf{E}(inform(B, A, Info, Answer), T2) \wedge T2 < T1 + D] \vee \\ &\qquad [\mathbf{E}(refuse(B, A, Info), T2) \wedge T2 < T1 + D]]] \end{aligned}$$

$$\begin{aligned} &\forall A, B, Info, Answer, T1, T2\,[\mathbf{H}(inform(A, B, Info, Answer), T1) \\ \rightarrow\ &\qquad \mathbf{EN}(refuse(A, B, Info), T2) \end{aligned}$$

The first constraint means that, if agent A sends to agent B a *query* for *Info* at time *T1* for response with maximum delay D, then B is expected to reply with either an *inform* or a *refuse* message by D time units later. The second constraint means that, if A sends an *inform* message to B, then A is expected not to send a *refuse* message to B at any time.

$\mathbf{H}(e, t)$ expresses that an event e happens at a time t. $\mathbf{E}(e, t)$ is an abducible predicate representing an obligation that e happens at t. $\mathbf{EN}(e, t)$ is an abducible predicate representing a prohibition that e happens at t. Abductive solutions are restricted to those whose obligations actually happen and whose prohibitions do not happen.

In contrast with SCIFF, we do not employ separate predicates $\mathbf{H}(e, t)$, $\mathbf{E}(e, t)$ and $\mathbf{EN}(e, t)$, but employ only a single abducible predicate $\mathbf{H}(e, t)$ (or $happens(e, t)$). Events that are given as happening are included in the set of beliefs P. Positive and negative expectations are both expressed in the same form as given events, but are

included in the set of candidate assumptions A. The obligation and prohibition of the example above can then be represented by the goals:

$$\forall A, B, Info, T1, D\ [happens(query(A, B, Info, D), T1)$$
$$\rightarrow \exists Answer, T2\ [[happens(inform(B, A, Info, Answer), T2) \wedge T2 < T1 + D]$$
$$\vee\ [happens(refuse(B, A, Info), T2) \wedge T2 < T1 + D]]]$$

$$\forall A, B, Info, Answer, T1, T2 \neg\ [happens(inform(A, B, Info, Answer), T1)$$
$$\wedge\ happens(refuse(A, B, Info), T2)]$$

SCIFF focuses on specifying and verifying interaction in multi-agent systems. Alberti et al. [1] compare the SCIFF approach with modal deontic logics, but do not discuss the treatment of conflicting obligations or contrary-to-duty obligations.

SCIFF uses a theorem-proving view of goal satisfaction, which is adequate when the sequence of interactions between agents is finite, and logic programs are written in if-and-only-if form [13]. This contrasts with the model-generation view in this paper. For our intended applications, in which the sequence of interactions is conceptually never-ending, the model-generation view determines the truth value of any goal expressed in FOL, but the theorem-proving view is incomplete.

The situation is analogous to that of arithmetic, where the standard model of arithmetic is the minimal model of the definite clause definitions of addition and multiplication [14, 39]. The model-generation view of goal satisfaction determines the truth value of any sentence of arithmetic in this minimal model, but the theorem-proving view is incomplete.

8 Conclusions

This paper concerns the more general controversy about the adequacy of classical first-order logic compared with other formal logics, and compared with modal logics in particular. We have argued that, in the case of representing and reasoning about deontic attitudes, the use of FOL for representing goals in ALP is a viable alternative to the use of modal logics. We have seen that the ALP approach is related to the vFH non-modal representation of obligations in default logic, and that, like the default logic approach, the ALP approach also tolerates normative conflicts. We have argued that, although the syntax of the deontic operators is restricted in comparison with that of modal deontic logics, it is nonetheless adequate both for many practical applications and for many of the problematic examples that have been studied in the philosophical literature.

The ALP approach represents obligations as hard goals or constraints, by representing sanctions and exceptions as additional, explicit alternatives. This is similar to the way in which abduction in Theorist turns defeasible rules into strict rules, by turning assumptions about normality into explicit defeasible conditions. We have argued that the ALP approach has the advantage that similar techniques apply to a wide range of applications, not only to satisfying obligations, but also to default reasoning, explaining observations and combinatorial optimisation.

In general, the ALP approach focuses on solving or satisfying goals, in contrast with modal deontic logics, which focus on inferring logical consequences. However, we have defined a notion of logical consequence for obligations in ALP, by adapting the sceptical version of Horty's definition of logical consequence; and we have applied the definition to some of the examples that have proved problematic for modal deontic logic. We have argued that the ALP approach provides a satisfactory solution of the problems, and that, in cases where the ALP solution may not seem entirely intuitive, it is rather that logical consequence is a less appropriate consideration than goal satisfaction. Moreover, the goal satisfaction semantics makes it possible to distinguish between the normative ideal of satisfying goals in the best way possible, and the more practical objective of satisfying the goals in the best way possible given the resources that are available at the time.

The ALP approach of this paper is similar to the ALP approach of SCIFF. The main difference is between the theorem-proving semantics of SCIFF and the model generation semantics that we use in this paper.

This paper also concerns the controversy about whether a single logic, such as ALP, might be adequate for formalising human reasoning, or whether many logics are needed for different purposes. One of the strongest arguments for ALP is that it subsumes production systems [40], which have been widely promoted as a general-purpose theory of human thinking [64]. Other arguments include its use for abductive reasoning, default reasoning [50] and probabilistic reasoning, with the power of Bayesian networks [52].

The application of ALP to deontic reasoning in this paper is a further test of its generality. Conversely, the generality of ALP is an argument for its application to deontic reasoning. Of course, both of these claims – for ALP as a general-purpose logic, and for ALP as a logic for deontic reasoning – need further testing. For this purpose, extending the ALP approach from single agent to multi-agent systems, where different agents have different goals and beliefs, is perhaps the most important challenge for the future.

Acknowledgements Many thanks to Christoph Benzmueller, Harold Boley, Tom Blackson, Hendrik Decker, Guido Governatori, Henry Prakken, Fariba Sadri, Giovanni Sartor, Markus Schacher, Marek Sergot, Leon Van Der Torre, as well as the anonymous referees, for helpful comments on earlier drafts of the paper. Kowalski also thanks the Japanese Society for the Advancement of Science for its support in the initial phase of this work.

References

1. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P., & Sartor, G. (2006). Mapping deontic operators to abductive expectations. *Computational & Mathematical Organization Theory*, *12*(2-3), 205–225.

2. Anderson, A.R. (1958). A reduction of deontic logic to alethic modal logic. *Mind*, *67*(265), 100–103.
3. Anderson, A.R. (1966). The formal analysis of normative systems. In Rescher, N. (Ed.) *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press.
4. Asher, N., & Bonevac, D. (2005). Free choice permission is strong permission. *Synthese*, *145*(3), 303–323.
5. Baron, J. (2008). *Thinking and deciding, 3th edn*. Cambridge University Press.
6. Barringer, H., Fisher, M., Gabbay, D., Owens, R., & Reynolds, M. (1996). *The imperative future: principles of executable temporal logic*. Wiley.
7. Bartha, P. (2002). Review of agency and deontic logic. In *Notre Dame philosophical reviews*.
8. Bohnert, H.G. (1945). The semiotic status of commands. *Philosophy of Science*, *12*, 302–315.
9. Bondarenko, A., Dung, P.M., Kowalski, R.A., & Toni, F. (1997). An abstract, argumentation theoretic approach to default reasoning. *Artificial Intelligence*, *93*(1), 63–101.
10. Brewka, G., Benferhat, S., & Le Berre, D. (2004). Qualitative choice logic. *Artificial Intelligence*, *157*(1), 203–237.
11. Carmo, J., & Jones, A.J. (2002). Deontic logic and contrary-to-duties. *Handbook of philosophical logic* (pp. 265–343). Springer.
12. Chisholm, R.M. (1963). Contrary-to-duty imperatives and deontic logic. *Analysis*, *24*(2), 33–36.
13. Clark, K.L. (1978). Negation as failure. In *Logic and data bases* (pp. 293–322). Springer.
14. Davis, M. (1980). The mathematics of non-monotonic reasoning. *Artificial Intelligence*, *13*(1), 73–80.
15. Dechter, R. (2003). *Constraint processing*. Morgan Kaufmann.
16. Denecker, M. (2000). Extending classical logic with inductive definitions. In *Computational logic—CL 2000* (pp. 703–717). Springer.
17. Van Emden, M.H., & Kowalski, R.A. (1976). The semantics of predicate logic as a programming language. *Journal of the ACM (JACM)*, *23*(4), 733–742.
18. Eshghi, K. (1988). Abductive planning with event calculus. In *ICLP/SLP* (pp. 562–579).
19. Forrester, J.W. (1984). Gentle murder, or the adverbial Samaritan. *Journal of Philosophy*, *81*(4), 193–197.
20. Fox, C. (2015). The semantics of imperatives. *The Handbook of Contemporary Semantic Theory*, *3*, 314.
21. Fung, T.H., & Kowalski, R. (1997). The IFF proof procedure for abductive logic programming. *The Journal of logic programming*, *33*(2), 151–165.
22. Goble, L. (1991). Murder most gentle: the paradox deepens. *Philosophical Studies*, *64*(2), 217–227.
23. Governatori, G., & Rotolo, A. (2006). Logic of violations: A Gentzen system for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, *4*, 193–215.
24. Halpern, J.Y., & Vardi, M.Y. (1991). Model checking vs. theorem proving: a manifesto. *Artificial Intelligence and Mathematical Theory of Computation*, *212*, 151–176.
25. Hammond, J.S., Keeney, R.L., & Raiffa, H. (1999). *Smart choices: a practical guide to making better life choices*. Boston: Harvard Business School Press.
26. Hansson, B. (1969). An analysis of some deontic logics. *Nous*, 373–398.
27. Hilpinen, R., & McNamara, P. (2013). Deontic logic: a historical survey and introduction. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, L. van der Torre (Eds.), *Handbook of deontic logic and normative systems* (pp. 3–136). College Publications.
28. Hobbs, J.R., Stickel, M., Martin, P., & Edwards, D. (1988). Interpretation as abduction. In *Proceedings of the 26th annual meeting on association for computational linguistics* (pp. 95–103).
29. Horty, J.F. (1993). Deontic logic as founded on nonmonotonic logic. *Annals of Mathematics and Artificial Intelligence*, *9*(1-2), 69–91.
30. Horty, J.F. (2001). *Agency and Deontic Logic*. Oxford: Oxford University Press.
31. Horty, J.F. (2012). *Reasons as defaults*. Oxford University Press.
32. Horty, J.F. (2014). Deontic Modals: why abandon the classical semantics?. *Pacific Philosophical Quarterly*, *95*(4), 424–460.
33. Jørgensen, J. (1937). Imperatives and logic. *Erkenntnis*, *7*(1), 288–296.
34. Kakas, A.C., Kowalski, R., & Toni, F. (1998). The role of logic programming in abduction. In *Handbook of logic in artificial intelligence and programming 5* (pp. 235–324). Oxford University Press.
35. Kanger, S. (1971). New foundations for ethical theory. In Hilpinen, R. (Ed.) *Deontic logic: introductory and systematic readings* (pp. 36-58). Dordrecht: Reidel, D.

36. Keeney, R.L. (1992). *Value-focused thinking. A path to creative decision making*. Harvard University Press.
37. Kowalski, R. (1979). Algorithm= logic + control. *Communications of the ACM*, *22*(7), 424–436.
38. Kowalski, R. (2011). *Computational logic and human thinking. How to be artificially intelligent* Cambridge University Press.
39. Kowalski, R. (2014). Logic programming, volume 9, computational logic. In Siekmann, J. (Ed.) *The history of logic series, edited by Dov Gabbay & John Woods* (pp. 523–569). Elsevier.
40. Kowalski, R., & Sadri, F. (2009). Integrating logic programming and production systems in abductive logic programming agents. In *Proceedings of the third international conference on web reasoning and rule systems*. Chantilly.
41. Kowalski, R., & Sadri, F. (2014). A logical characterization of a reactive system language. In *Proceedings of RuleML 2014*. Springer Verlag.
42. Kowalski, R., & Sadri, F. (2016). Programming in logic without logic programming. *Theory and Practice of Logic Programming*, *16*(3), 269–295.
43. Makinson, D., & Van Der Torre, L. (2000). Input/output logics. *Journal of Philosophical Logic*, *29*(4), 383–408.
44. Makinson, D. (1999). On a fundamental problem of deontic logic. *Norms, logics and information systems. New studies on deontic logic and computer science* (pp. 29–54).
45. McCarthy, J. (1986). Applications of circumscription to formalising common sense knowledge. *Artificial Intelligence*, *28*(1), 89–116.
46. McNamara, P. (2006). Deontic logic. *Handbook of the History of Logic*, *7*, 197–289.
47. Meyer, J.J.C. (1988). A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, *29*(1), 109–136.
48. Nute, D. (Ed.) (1997). *Defeasible deontic logic. Essays in Nonmonotonic Normative Reasoning*. Boston: Kluwer Academic Publishers.
49. Peirce, C.S. (1931). Collected papers. In Hartshorn, C., & Weiss, P. (Eds.) Cambridge: Harvard University Press.
50. Poole, D., Goebel, R., & Aleliunas, R. (1987). *Theorist: a logical reasoning system for defaults and diagnosis*, (pp. 331–352). New York: Springer.
51. Poole, D. (1988). A logical framework for default reasoning. *Artificial intelligence*, *36*(1), 27–47.
52. Poole, D. (1997). The independent choice logic for modelling multiple agents under uncertainty. *Artificial intelligence*, *94*(1), 7–56.
53. Pople, H.E. (1973). On the mechanization of abductive logic. In *IJCAI* (Vol. 73, pp. 147-152).
54. Prior, A.N. (1958). Escapism: the logical basis of ethics. In *Essays in moral philosophy*.
55. Prakken, H., & Sergot, M. (1996). Contrary-to-duty obligations. *Studia Logica*, *57*(1), 91–115.
56. Prakken, H. (1996). Two approaches to the formalisation of defeasible deontic reasoning. *Studia Logica*, *57*(1), 73–90.
57. Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, *13*(1), 81–132.
58. Ross, A. (1941). Imperatives and logic. *Theoria*, *7*, 53–71.
59. Satoh, K. (1990). Formalizing soft constraints by interpretation ordering. In *Proceedings of the ninth European conference on artificial intelligence* (pp. 585–590). Stockholm.
60. Satoh, K., & Aiba, A. (1993). Computing soft constraints by hierarchical constraint logic programming. *Transactions of Information Processing Society of Japan*, *34*(7), 1555–1569.
61. SBVR (Semantics of Business Vocabulary and Business Rules), Version 1.2. (2013). OMG Document Number: formal/2013-11-04. Standard document http://www.omg.org/spec/SBVR/1.2/PDF/.
62. Sergot, M.J. (1982). Prospects for representing the law as logic programs. *Logic Programming*, 33–42.
63. Simon, H.A. (1972). Theories of bounded rationality. *Decision and Organization*, *1*(1), 161–176.
64. Thagard, P. (1996). *Mind: introduction to cognitive science*. Cambridge: MIT press.
65. Thomson, J.J. (1985). Double effect, triple effect and the trolley problem: squaring the circle in looping cases. *Yale Law Journal*, *94*, 6.
66. Van Benthem, J., Grossi, D., & Liu, F. (2014). Priority structures in deontic logic. *Theoria*, *80*(2), 116–152.
67. Fintel, V., Kai, & Iatridou, S. (2005). What to do if you want to go to Harlem: Anankastic conditionals and related matters. Manuscript MIT.
68. Van Fraassen, B. (1973). Values and the heart's command. *The Journal of Philosophy*, *70*, 5–19.
69. Von Wright, G.H. (1951). Deontic logic. *Mind*, *60*(237), 1–15.
70. Wason, P.C. (1968). Reasoning about a rule. *The Quarterly Journal of Experimental Psychology*, *20*(3), 273–281.