Modalities Over Actions, I. Model Theory

L. Thorne McCarty
Computer Science Department
Rutgers University
New Brunswick, NJ 08903, USA

Abstract

This paper analyzes a language for actions and the deontic modalities over actions — i.e., the modalities permitted, forbidden and obligatory. The work is based on: (1) an action language that allows the representation of concurrent and repetitive events; (2) a deontic language that allows the representation of “free choice permissions”; (3) a proof procedure that admits a logic programming style of computation; and (4) a facility for nonmonotonic inference based on negation-as-failure. Applications of the language to several problems of common sense reasoning are also discussed. In particular, by imposing a “causal assumption” on the deontic modalities, we obtain an interesting solution to the frame problem and the ramification problem. This first part of the paper includes a model theory, and a sequel will include a proof theory, with soundness and completeness results for various fragments of the language.

1 Introduction

A standard approach to the representation of actions is to take the state of the world as primary, and to encode actions as transformations on states. McCarthy’s situation calculus [15] is the best known historical example, but classical dynamic logic [33, 10] adopts essentially the same ontology. There are several problems with this approach, not the least of which is the difficulty of extending the formalism to complex actions. To be sure, recent work has shown how to extend the situation calculus to represent continuous and partially ordered events [9], and concurrent, possibly conflicting, actions [14, 3]. But these are heroic efforts, and the complex machinery that seems to be required for these extensions is strong evidence of a recalcitrant ontology.

Another alternative approach is to take actions as primary, and to treat the state of the world as a derivative notion. The event calculus of Kowalski and Sergot [11] is one example of this approach, as is the work of Allen, et al. [2, 1] on action and time. Within the framework of dynamic logic, Pratt’s early work on process logic [34] and his more recent work on action logic [35] reflect a similar shift in perspective from a world consisting of a sequence of states to a world consisting of a set of actions. One justification for this shift in perspective is the fact that the most important properties of ordinary actions — e.g., concurrency and nondeterminism — are easier to express in a language in which actions are first-class objects.

But let us look more closely at the concept of an action. In our ordinary experience, actions have agents, and agents have choices. Are there constraints on the choices of agents? Indeed, there are, and in common sense reasoning these constraints are often expressed by the deontic modalities: We say that actions are either permitted (P), forbidden (F), or obligatory (O). Shouldn’t these modalities also be incorporated into our representation language?

In this paper, we investigate a language for actions and the deontic modalities over actions, and we show how this language can be used to model various aspects of common sense reasoning. Our work is related to recent work on the relationship between deontic logic and dynamic logic [30, 42, 27, 28], but it is based on the following developments:

1: The action language is based on [25]. In [25], McCarty and van der Meyden treat actions as predicates over a linear temporal order, and draw a distinction between basic actions and defined actions. Defined actions are represented by Horn clauses with (optional) linear order constraints, and the defined predicates are circumscribed to capture the intended interpretation of these Horn clauses as definitions. With this device, it is easy to represent concurrent and repetitive events, and to construct partially ordered plans. It is also easy to select an appropriate temporal on-
tology for a particular application, since discrete time and continuous time are just minor variations of a single logical language.

2: The deontic language is based on much earlier work in [16, 17]. In fact, the deontic modalities in the present paper are essentially the same as the deontic modalities in [16, 17], but applied to the action language in [25]. One significant feature of this early work is the fact that \( P \) is a “free choice permission”, and this interpretation is carried forth into the present paper, with one modification (see Definition 4.2 and related text). Another modification in the present paper involves the deontic conditional. The system in [16, 17] was a dyadic deontic logic, with a counterfactual conditional, whereas the present paper uses intuitionistic implication [39] for the deontic conditional. This change was motivated by a desire to be able to answer queries in the deontic language and the action language, uniformly, in the style of a logic program [18, 19], particularly where negation is involved.

3: Since the action definitions in [25] make use of circumscription, entailment in the action language is not recursively enumerable, and this property is inherited by the deontic language. Nevertheless, there are proof methods for important special cases. If the definitions are nonrecursive, then circumscription is equivalent to predicate completion [6, 37, 12]: A PROLOG-style interpreter for this case is discussed in [23]. If the definitions are linear recursive, then a special inductive proof method based on second-order intuitionistic logic may be applicable: A PROLOG-style interpreter for this case is discussed in [22]. The combination of these two special cases is illustrated in [24] for a first-order language without a linear temporal order, and in [25] for the action language. Also, for yet another special case, circumscriptive inference in the action language is actually decidable [29]. One of the objectives of the present work is to extend these special proof techniques from the first-order language and the action language to the full deontic language.

4: Finally, since deontic rules take the form of Horn clause logic programs, we can represent nonmonotonicity by negation-as-failure. There are numerous proposals for the semantics of negation-as-failure, of course, but the approach that works best in our system is based on partial intuitionistic circumscription [26], or PIC, for short. The PIC semantics agrees with the perfect model semantics [36] for stratified rules, and for unstratified rules it is strictly stronger than the well-founded semantics [41] and strictly weaker than the stable model semantics [8]. Most importantly, the PIC semantics is based on a version of circumscription that can be immediately generalized to the present system of deontic logic. This means that we can apply negation-as-failure directly to the deontic modalities, \( P, F \) and \( O \), with interesting consequences. For example, we can say that the actions of a particular agent are permitted unless they are explicitly forbidden; or, conversely, we can say that the actions of a particular agent are forbidden unless they are explicitly permitted.

This first part of the paper analyzes the action language and develops a model theory for the deontic language, essentially covering points (1) and (2) above. A discussion of the proof theory (3) and the semantics of negation-as-failure (4) is reserved for a sequel.

One of the main features of deontic logic is the fact that actors do not always obey the law. Indeed, it is precisely when a forbidden action occurs, or an obligatory action does not occur, that we need the machinery of deontic logic, to detect a violation and to take appropriate action. On the other hand, for purposes of planning, it is often useful to assume that actors do obey the law. We call this the causal assumption, since it enables us to predict the actions that will occur by reasoning about the actions that ought to occur. As we will see, it is straightforward to incorporate the causal assumption as an additional constraint in the deontic semantics.

Moreover, if we adopt the causal assumption, we can use the machinery of deontic logic to reason about the physical world. The slogan is simple:

\[ \text{Causation is Divine Obligation.} \]

Coupled with negation-as-failure, this principle provides an interesting solution to both the frame problem and the ramification problem. The basic idea is to posit an actor named ‘nature’, who always obeys the law, but can be overridden in particular situations by the actions of other agents. We will see how this works in Section 2.

Following an informal exposition and several examples in Section 2, we discuss events and actions in Section 3, and then develop a semantics for the deontic modalities in Section 4. Several properties of the semantics are established in this section, including two theorems that will be needed in the proof theory. Section 5 then concludes with a discussion of related work.

2 Intuitions and Applications

The fundamental assumption underlying the action language in [25] is the idea that, at some level of detail, we can identify a set of basic events. This level of detail might be quite coarse in a particular application. For example, we could take ‘SwimLap(\( x, t_1, t_2 \) \( \land \) \( t_1 < t_2 \)’ to be a basic event in which the actor \( x \) swims one lap of the pool between the time \( t_1 \) and the time \( t_2 \), ignoring the finer details about how this action is accomplished. We can then define...
more complex events by a set of Horn clauses. For example, the event in which \( x \) swims some finite number of laps between \( t_1 \) and \( t_2 \) could be defined by the recursive Horn clauses:

\[
\text{SwimLaps}(x, t_1, t_2) \iff \text{SwimLap}(x, t_1, t_2) \land t_1 < t_2,
\]

\[
\text{SwimLaps}(x, t_1, t_2) \iff \text{SwimLap}(x, t_1, t_3) \land \text{SwimLaps}(x, t_3, t_2) \land t_1 < t_3 < t_2,
\]

and the event in which \( x \) and \( y \) each swim some finite number of laps, with \( x \) finishing first, could be defined by the conjunction:

\[
\text{SwimLaps}(x, t_1, t_2) \land \text{SwimLaps}(y, t_1, t_3) \land t_1 < t_2 < t_3.
\]

In principle, if this representation turns out to be too coarse for a particular application, we can increase the level of detail, and define ‘SwimLap’ itself in terms of more basic predicates, such as the position of the swimmer over time, or even the movement of the swimmer’s arms and legs.

An important property of basic events is that they are definite relative to the chosen level of detail, whereas a defined event could be indefinite. If we only know ‘SwimLaps(\( a_1 \), \( t_1 \), \( t_2 \))’ for a particular actor \( a_1 \) and the particular times \( t_1 \) and \( t_2 \), then we don’t know how many laps \( a_1 \) swam, or how fast she swam them. But basic events should not have this kind of ambiguity. Formally, basic events should have the disjunctive and existential properties, that is, a disjunction \( A \lor B \) should be entailed by a basic event only if either \( A \) is entailed or \( B \) is entailed, and an existentially quantified proposition \( \exists \theta \, P(\theta) \) should be entailed by a basic event only if \( P(\theta) \) is entailed for some substitution \( \theta \). (See Section 3 below for a more detailed discussion of this requirement.)

Now, there is one important class of basic events that involve changes in the state of the world, and this fact has consequences for our choice of a logic. Suppose \( B \) is a first-order predicate with a single argument. We might want to define an event in which the state of the world changes from a time \( t_1 \) at which \( B(x) \) is true to a time \( t_2 \) at which \( \neg B(x) \) is true. As a convenient notation for such an event, we write:

\[
B(x)[t_1] \circ \neg B(x)[t_2] \land t_1 < t_2.
\]

For this event to be basic, however, we require: (i) that \( B \) is itself a base (i.e., undefined) predicate in the first-order language, and (ii) that the overall logic is intuitionistic rather than classical. These two conditions will guarantee that basic events generate only definite changes in the state of the world [18]. Similarly, suppose \( C \) is a first-order predicate with two arguments. We might want to define an event:

\[
C(x, y)[t_1] \circ (\forall w) \neg C(w, y)[t_2] \land t_1 < t_2.
\]

The same conditions will guarantee that this event, too, generates only definite changes in the state of the world. (Note also that \( B(x) \) and \( C(x, y) \) are neither true nor false in this example when \( t \) is between \( t_1 \) and \( t_2 \), as long as we use intuitionistic rather than classical logic.) Finally, a useful class of events consists of the nonevents, such as \( B(x)[t_1, t_2] \) or \( \neg B(x)[t_1, t_2] \), which assert that \( B(x) \) remains true, or remains false, respectively, over the interval from \( t_1 \) to \( t_2 \). Again, our conditions guarantee that these are definite events.

The concept of an event does not itself include the concept of agency, but if we pair an event \( \alpha(t_1, t_2) \) with an agent \( x \) we have an action. We write \( \text{DO}(\alpha(t_1, t_2), x) \) to say that \( x \) is (somehow) responsible for the occurrence of \( \alpha(t_1, t_2) \), or that \( \alpha(t_1, t_2) \) is (somehow) carried out by the agent \( x \). Such statements are veridical. That is, if \( x \) does \( \alpha(t_1, t_2) \), then \( \alpha(t_1, t_2) \) is true. The converse does not hold, of course. We can observe an event \( \alpha(t_1, t_2) \) without concluding that there exists an \( x \) such that \( \text{DO}(\alpha(t_1, t_2), x) \). (If we wanted to make such an inference, and ascribe responsibility to a particular agent, we would have to do so by abduction [31], not by deduction.)

In our language, the deontic modalities have a syntax similar to the syntax of DO. We write:

\[
\text{F}(\alpha(t_1, t_2), x), \quad \text{O}(\alpha(t_1, t_2), x) \quad \text{and} \quad \text{O}(\alpha(t_1, t_2), x), \quad \text{where} \quad \alpha(t_1, t_2) \quad \text{is an event and} \quad x \quad \text{is an agent.}
\]

The intuitive reading of \( \text{O}(\alpha(t_1, t_2), x) \) is straightforward. It means: “The actor \( x \) is obligated to perform the action \( \alpha(t_1, t_2) \).” The intuitive reading of \( \text{F} \) and \( \text{O} \) is more subtle, since actions can be indefinite. To say that \( x \) is forbidden, \( \text{F} \), to perform the action \( \alpha(t_1, t_2) \) means that all the ways of performing \( \alpha(t_1, t_2) \) are forbidden. Analogously, \( \text{P} \) is a “free choice permission”. To say that \( x \) is permitted, \( \text{P} \), to perform the action \( \alpha(t_1, t_2) \) means that all the ways of performing \( \alpha(t_1, t_2) \) are permitted. This is different from the weaker form of permission, \( \neg \text{F} \), that is usually studied in deontic logic. To say that \( x \) is not forbidden, \( \neg \text{F} \), to perform the action \( \alpha(t_1, t_2) \) means simply that there is some way of performing \( \alpha(t_1, t_2) \) that is permitted. These distinctions are difficult to express in standard deontic logic [5], but they are easy to express in our language because of the sharp contrast between definite and indefinite actions.

Finally, to complete the system, we write deontic conditionals as Horn clause logic programs, using any formulae we want from the action language, including the constraints on time. We also allow negation-as-failure to appear in the deontic conditionals, so that the following expressions are possible:

\[
\text{F}(\alpha(t_1, t_2), x) \iff \gamma(x, t_1) \land \neg \beta(x, t_1, t_2) \land t_1 < t_2,
\]

\[
\text{O}(\alpha(t_1, t_2), x) \iff \neg \text{F}(\alpha(t_1, t_2), x) \land t_1 < t_2,
\]
where $\alpha$, $\beta$ and $\gamma$ are events, and the symbol ‘$\sim$’ denotes negation-as-failure. Notice that the first rule here applies negation-as-failure to $\beta$, an expression in the action language, while the second rule applies negation-as-failure to an expression in the deontic language. The second rule says, intuitively, that $\alpha$ is obligatory if it is not forbidden.

There are obvious applications of these ideas in legal domains [20]. For example, an adequate representation of the concept of a corporation requires a representation of the “bundle of rights” available to the corporation’s creditors and the owners of its stock. Here is a simplified illustration:

**Example 2.1:** Consider a closely held corporation, $c$, with three stockholders, $a_1$, $a_2$ and $a_3$, and a bank, $b$, as its sole creditor. The corporation is obligated to pay interest to the bank, in some fixed amount of dollars, $d$, per month. Assuming that we have properly defined the action ‘TransferDollars ($x, y, n, t$)’, we could write this obligation as follows:

$$O \langle \text{TransferDollars} (c, b, d, t, c) \rangle \Leftarrow \text{Month} (t),$$

where the predicate ‘Month’ simply tests that $t$ is the beginning of the month. This is what we mean when we say that the bank has a “right” to receive ‘d’ dollars in interest from the corporation each month. The stockholders might also have a “right” to receive dividends each quarter, but the specification of this right would be quite different. First, we would write the following free choice permission:

$$P \langle (\exists n) \text{TransferDollars} (c, a_i, n, t, c) \rangle \Leftarrow \text{Quarter} (t),$$

for $i = 1, \ldots, 3$. By itself, this rule allows the corporation to select a stockholder, $a_i$, and a dollar amount, $n_i$, in making the transfer, but such a completely free choice would then be constrained by the following obligation:

$$O \langle \text{TransferDollars} (c, a_i, n_i, t, c) \rangle \Leftarrow \text{TransferDollars} (c, a_j, n_j, t) \land n_i \times r_j = n_j \times r_i,$$

for $i \neq j$. Here, $r_i$ is the fraction of total stock owned by $a_i$, so this rule expresses the fact that a dividend must be a *pro rata* distribution of assets to all stockholders. □

The deontic modalities are also useful for expressing planning problems. The following example is adapted from [25].

**Example 2.2:** Imagine two actors, $a_1$ and $a_2$, moving around in a suite of rooms, $r_1, \ldots, r_n$. We define an action ‘Move ($x, y, z, t_1, t_2$)’ as follows:

$$\text{Move} (x, y, z, t_1, t_2) \Leftarrow \text{In} (x, y) [t_1, t] \land \text{In} (x, z) [t, t_2] \land$$

$$\text{Connected} (y, z) \land t_1 < t < t_2,$$

Move ($x, y, z, t_1, t_2$) \Leftarrow \text{In} (x, y) [t_1, t] \land \text{In} (x, z) [t, t_2] \land$$

$$\text{Connected} (y, z) \land t_1 < t < t_2,$$

where ‘\text{In} (x, y)$’ asserts that actor $x$ is in room $y$, and ‘\text{Connected} (y, z)$’ asserts that room $y$ is connected by a doorway to room $z$. Suppose the actors $a_1$ and $a_2$ are forbidden to be in the same room at the same time. We could represent this forbidden event by:

$$\text{InSameRoom} (t_0, t_3) \Leftarrow$$

$$\text{In} (a_1, y) [t_1, t_2] \land t_0 \leq t_1 \leq t \leq t_2 \leq t_3 \land$$

$$\text{In} (a_2, y) [t', t''_2] \land t_0 \leq t' \leq t \leq t''_2 \leq t_3,$$

and then assert:

$$F \langle \text{InSameRoom} (t_0, t_3), a_1 \& a_2 \rangle \quad (1)$$

to represent the prohibition itself. (In this encoding, we are treating $a_1 \& a_2$ as a single agent, and ignoring the difficult problems of joint agency.) Given (1), we would like to know if the following action is also forbidden:

$$\text{Swap} (t_0, t_3) \Leftarrow \text{Move} (a_1, r_i, r_j, t_0, t_3) \land$$

$$\text{Move} (a_2, r_j, r_i, t_0, t_3),$$

where $r_i$ and $r_j$ are any two arbitrary rooms. In other words, we would like to know whether (1) implies:

$$F \langle \text{Swap} (t_0, t_3), a_1 \& a_2 \rangle, \quad (2)$$

taking the universal closure over $t_0$ and $t_3$. The answer depends on the connectivity of the rooms, and the solutions are discussed in [25]. Semantically, however, if (1) does not imply (2), then there exists *some* way to swap the positions of the two actors without violating the deontic constraints — i.e., there exists a *permitted plan*. □

In the previous example, we were able to formulate a planning problem without reference to the frame problem because our two agents controlled all the actions relevant to the plan. We can use the same idea to formulate a classical STRIPS planning problem. In this case, however, it is necessary to modify the representation somewhat. In Example 2.2, we specified all possible actions by definitions, and then asserted that certain actions were forbidden. We assumed, in effect, that all actions are permitted unless they are explicitly forbidden. In the blocks world, we will turn this around, and assume that actions are forbidden unless they are explicitly permitted. (Think of this modality as “can do” rather than “may do”.) Here is a simple example:

**Example 2.3:** Consider a blocks world with a base predicate ‘On ($x, y)$’, in which $x$ is a block and $y$ is either another block or a location on the table. Define the following
action:

\[\text{Move}(x, y, z, t_1, t_2) \iff (\forall w)\neg\text{On}(w, x)[t_1, t_2] \land \text{On}(x, y)[t_1] \land \text{On}(w, y)[t_2] \land (\forall w)\neg\text{On}(w, z)[t_1] \land \text{On}(x, z)[t_2] \land t_1 < t_2\]

This action simultaneously (i) maintains the fact that there is nothing on \(x\), (ii) clears \(x\) from its location on \(y\), and (iii) puts \(x\) on \(z\). To handle the frame problem, we also define the following actions:

\[\text{HoldOn}(x, y, t_1, t_2) \iff \text{On}(x, y)[t_1, t_2] \land t_1 < t_2,\]

which maintains the fact that \(x\) is on \(y\), and

\[\text{HoldClear}(x, t_1, t_2) \iff (\forall w)\neg\text{On}(w, x)[t_1] \land t_1 < t_2,\]

which maintains the fact that there is nothing on \(x\). We now assert that a particular actor, \(a_1\), is permitted to perform any bounded set of these actions concurrently, as long as the preconditions are satisfied:

\[
P(\text{Move}(x, y, z, t_1, t_2)) \land \\
\bigwedge_{i=1}^{m} \text{HoldOn}(u_{i1}, u_{i2}, t_1, t_2) \land \\
\bigwedge_{j=1}^{n} \text{HoldClear}(u_j, t_1, t_2, a_1) \iff \\
(\forall w)\neg\text{On}(w, x)[t_1] \land \text{On}(x, y)[t_1] \land \\
(\forall w)\neg\text{On}(w, z)[t_1] \land \\
\bigwedge_{i=1}^{m} \text{On}(u_{i1}, u_{i2})[t_1] \land \\
\bigwedge_{j=1}^{n} (\forall w)\neg\text{On}(w, u_j)[t_1] \land \\
t_1 < t_2\]

This is a free choice permission, and if we assert this permission for all \(m \leq M\) and \(n \leq N\), then \(a_1\) is allowed to select any blocks that it wants to ‘HoldOn’ and ‘HoldClear’. To turn this into a planning problem, assume that we have also specified a ‘Start’ condition and a ‘Goal’ condition. To show that there is no way for \(a_1\) to get from ‘Start’ to ‘Goal’, we try to prove:

\[
P(\text{Start}[t_0] \circ \text{Goal}[t_k], a_1) \iff t_0 < t_k,\]

under the assumption that all actions are forbidden unless they are explicitly permitted. If this proof fails, we will have a sequence of permitted actions that achieves the goal.

Notice that this is a “monotonic” solution to the frame problem, in the spirit of [13]. There are no frame axioms, just a collection of actions that explicitly change the state of the world. Unlike the solution in [13], however, our solution does not require us to update the complete state of the world, since the robot only needs to “hold onto” those blocks that are relevant to the planning problem. This may seem unrealistic (for example, it requires the robot to have \(M + N + 1\) arms!), but this is just the first step in our analysis. It is easy to modify the example so that the inertial actions occur by default:

**Example 2.4:** Let ‘Move’, ‘HoldOn’ and ‘HoldClear’ be defined as in Example 2.3. For convenience, we define the following actions as the opposites of ‘HoldOn’ and ‘HoldClear’, respectively:

\[
\text{ChangeOn}(x, y, t_3) \iff \text{On}(x, y)[t_3] \land (\forall w)\neg\text{On}(w, y)[t_4] \land t_3 < t_4
\]

\[
\text{ChangeClear}(x, t_3) \iff (\forall w)\neg\text{On}(w, x)[t_3] \land \text{On}(x, y)[t_4] \land t_3 < t_4
\]

Intuitively, we want to assert that ‘On’ and ‘Clear’ persist unless some specific action changes them. We can write this as follows:

\[
O(\text{HoldOn}(x, y, t_1, t_2), \text{nature}) \iff \\
\text{On}(x, y)[t_1] \land t_1 < t_2 \land \neg((\exists a, x_t)
\]

\[
O(\text{HoldClear}(x, t_1, t_2), \text{nature}) \iff \\
(\forall w)\neg\text{On}(w, x)[t_1] \land t_1 < t_2 \land \neg((\exists a, x_t)
\]

where the symbol ‘\(\neg\)’ denotes negation-as-failure. We thus postulate an actor named ‘nature’ who is obligated to perform the actions ‘HoldOn’ and ‘HoldClear’ unless some other action intervenes. (Remember; ‘nature’ always obeys the law!) We can now simplify the free choice permission for \(a_1\):

\[
P(\text{Move}(x, y, z, t_1, t_2), a_1) \iff \\
(\forall w)\neg\text{On}(w, x)[t_1] \land \text{On}(x, y)[t_1] \land \\
(\forall w)\neg\text{On}(w, z)[t_1] \land t_1 < t_2,
\]

and let ‘nature’ do some of the work.

The encoding in Example 2.4 suggests that we might encounter problems with the ramifications of an action, but these problems can also be solved using deontic modalities. Here is a standard example, from [38]:

**Example 2.5:** Imagine a university database system, in which the relation ‘\(\text{In}(s, c)\)’ means that student \(s\) is enrolled in course \(c\). The possible actions are ‘Add’ and
'Drop', which are defined as follows:

$$\text{Add}(s, c, t_1, t_2) \iff \neg \text{In}(s, c)[t_1] \circ \text{In}(s, c)[t_2] \land t_1 < t_2$$

$$\text{Drop}(s, c, t_1, t_2) \iff \text{In}(s, c)[t_1] \circ \neg \text{In}(s, c)[t_2] \land t_1 < t_2$$

(In this example, time is usually discrete, and $t_2$ is usually the immediate successor of $t_1$. ) Assume the existence of two deontic rules that permit ($P$) a student to add a course if she is not already enrolled in it, and to drop any course in which she is currently enrolled. There are also two deontic rules that obligate ($O$) the university to maintain a student’s enrollment status unless the student herself adds or drops a course. These rules are similar to the inertial rules in Example 2.4. Finally, assume that a student is required to take 'chem120', a laboratory course, whenever she takes 'chem110'. We could represent this requirement as follows:

$$\text{In}(s, \text{chem120})[t_1, t_2] \lor \text{Add}(s, \text{chem120}, t_1, t_2), s)$$

$$\text{O}(\neg \text{In}(s, \text{chem110})[t_1, t_2] \lor \text{Drop}(s, \text{chem120}, t_1, t_2), s)$$

Thus, if a student always obeys the law and is initially enrolled in no courses at all, she will always satisfy the university’s enrollment status unless the student herself adds or drops a course. We assume that the object arguments always precede the order arguments in the signature of the language, so that the predicate $P(x_1, \ldots, x_n, t_1, \ldots, t_m)$ has arity $\langle n, m \rangle$. A common device in the interpretation of intuitionistic logic (see, e.g., [7]) is to extend the original language by a new set of constants, which can then be used to specify a total domain for a Kripke structure. Accordingly, let $L(c, t)$ be the language $L$ augmented by an arbitrary set of constants $c$ for the object sort, and an arbitrary (but disjoint) set of constants $t$ for the order sort. We assume that $t$ is isomorphic to the rational numbers, and therefore countable, and for simplicity we assume that $c$ is countable as well. Moreover, we will use the natural order on the rationals to interpret the special order atoms in $L$, i.e., the atoms $t_1 < t_2$. Thus the total domain $t$ is a dense linear order.

The main idea underlying the Kripke semantics of intuitionistic logic, however, is that we should work with partial domains and partial models. We thus write a Kripke structure for $L$ as a quintuple $(W, \sqsubseteq, h, u, o)$, where $W$ is a nonempty set of worlds, $\sqsubseteq$ is a partial order on $W$, $u$ is a monotonic mapping from the worlds of $W$ to nonempty sets of object constants in $L(c, t)$, and $o$ is a monotonic mapping from the worlds of $W$ to nonempty sets of order constants in $L(c, t)$. (Note that $o(w)$, for a particular world $w \in W$, could be a finite linear order, or a singleton, or an infinite linear order that is not dense.) Intuitively, the third component of the Kripke structure, $h$, tells us the ground atomic formulae that are true at each $w \in W$. Formally, we first define an intuitionistic relation $R$ of arity $\langle n, m \rangle$ to be a function that assigns to every world $w \in W$ a subset of the Cartesian product $u(w)^n \times o(w)^m$, subject to the requirement that $R(w_1) \subseteq R(w_2)$ whenever $w_1 \sqsubseteq w_2$. We then define $h$ to be a mapping from the predicate constants in $L$ to the set of intuitionistic relations on $W$. The atomic clause of the “forcing” relation [7] is thus:

$$w, W \models P(c_1, \ldots, c_n, t_1, \ldots, t_m) \iff \langle c_1, \ldots, c_n, t_1, \ldots, t_m \rangle \in h(P)(w),$$

for $P$ a predicate constant of arity $\langle n, m \rangle$. For the order atoms: $w, W \models t_1 < t_2$ if and only if $t_1$ and $t_2$ are both in $o(w)$ and $t_1$ is less than $t_2$ in the natural order on the rationals. Among the compound formulae, the definitions of forcing for conjunction, disjunction and existential quantification depend only on a single world, $w$, as in classical logic, but the definitions for implication and universal quantification are nonclassical:

$$w, W \models \mathcal{B} \iff \mathcal{A} \text{ iff } w', W \models \mathcal{A} \text{ implies } w', W \models \mathcal{B} \text{ for every } w' \sqsupseteq w \text{ in } W, \text{ and the constants in } \mathcal{A} \text{ and } \mathcal{B} \text{ are in } u(w) \text{ or } o(w),$$

$$w, W \models (\forall x)\mathcal{A}(x) \iff w', W \models \mathcal{A}(c) \text{ for every } w' \sqsupseteq w \text{ in } W, \text{ and for all object constants } c \text{ in } u(w'),$$

Let $L$ be a function-free first-order language with two sorts: an object sort and an order sort. The object sort includes constants and variables, written $x_1, x_2, \ldots$, but the order sort includes only variables, written $t_1, t_2, \ldots$. We will assume that the object arguments always precede the order arguments in the signature of the language, so that the predicate $P(x_1, \ldots, x_n, t_1, \ldots, t_m)$ has arity $\langle n, m \rangle$. A common device in the interpretation of intuitionistic logic (see, e.g., [7]) is to extend the original language by a new set of constants, which can then be used to specify a total domain for a Kripke structure. Accordingly, let $L(c, t)$ be the language $L$ augmented by an arbitrary set of constants $c$ for the object sort, and an arbitrary (but disjoint) set of constants $t$ for the order sort. We assume that $t$ is isomorphic to the rational numbers, and therefore countable, and for simplicity we assume that $c$ is countable as well. Moreover, we will use the natural order on the rationals to interpret the special order atoms in $L$, i.e., the atoms $t_1 < t_2$. Thus the total domain $t$ is a dense linear order.

The main idea underlying the Kripke semantics of intuitionistic logic, however, is that we should work with partial domains and partial models. We thus write a Kripke structure for $L$ as a quintuple $(W, \sqsubseteq, h, u, o)$, where $W$ is a nonempty set of worlds, $\sqsubseteq$ is a partial order on $W$, $u$ is a monotonic mapping from the worlds of $W$ to nonempty sets of object constants in $L(c, t)$, and $o$ is a monotonic mapping from the worlds of $W$ to nonempty sets of order constants in $L(c, t)$. (Note that $o(w)$, for a particular world $w \in W$, could be a finite linear order, or a singleton, or an infinite linear order that is not dense.) Intuitively, the third component of the Kripke structure, $h$, tells us the ground atomic formulae that are true at each $w \in W$. Formally, we first define an intuitionistic relation $R$ of arity $\langle n, m \rangle$ to be a function that assigns to every world $w \in W$ a subset of the Cartesian product $u(w)^n \times o(w)^m$, subject to the requirement that $R(w_1) \subseteq R(w_2)$ whenever $w_1 \sqsubseteq w_2$. We then define $h$ to be a mapping from the predicate constants in $L$ to the set of intuitionistic relations on $W$. The atomic clause of the “forcing” relation [7] is thus:

$$w, W \models P(c_1, \ldots, c_n, t_1, \ldots, t_m) \iff \langle c_1, \ldots, c_n, t_1, \ldots, t_m \rangle \in h(P)(w),$$

for $P$ a predicate constant of arity $\langle n, m \rangle$. For the order atoms: $w, W \models t_1 < t_2$ if and only if $t_1$ and $t_2$ are both in $o(w)$ and $t_1$ is less than $t_2$ in the natural order on the rationals. Among the compound formulae, the definitions of forcing for conjunction, disjunction and existential quantification depend only on a single world, $w$, as in classical logic, but the definitions for implication and universal quantification are nonclassical:

$$w, W \models \mathcal{B} \iff \mathcal{A} \text{ iff } w', W \models \mathcal{B} \text{ for every } w' \sqsupseteq w \text{ in } W, \text{ and the constants in } \mathcal{A} \text{ and } \mathcal{B} \text{ are in } u(w) \text{ or } o(w),$$

$$w, W \models (\forall x)\mathcal{A}(x) \iff w', W \models \mathcal{A}(c) \text{ for every } w' \sqsupseteq w \text{ in } W, \text{ and for all object constants } c \text{ in } u(w'),$$

Let $L$ be a function-free first-order language with two sorts: an object sort and an order sort. The object sort includes constants and variables, written $x_1, x_2, \ldots$, but the order sort includes only variables, written $t_1, t_2, \ldots$. We will assume that the object arguments always precede the order arguments in the signature of the language, so that the predicate $P(x_1, \ldots, x_n, t_1, \ldots, t_m)$ has arity $\langle n, m \rangle$. A common device in the interpretation of intuitionistic logic (see, e.g., [7]) is to extend the original language by a new set of constants, which can then be used to specify a total domain for a Kripke structure. Accordingly, let $L(c, t)$ be the language $L$ augmented by an arbitrary set of constants $c$ for the object sort, and an arbitrary (but disjoint) set of constants $t$ for the order sort. We assume that $t$ is isomorphic to the rational numbers, and therefore countable, and for simplicity we assume that $c$ is countable as well. Moreover, we will use the natural order on the rationals to interpret the special order atoms in $L$, i.e., the atoms $t_1 < t_2$. Thus the total domain $t$ is a dense linear order.
Now let $A$ be a closed sentence in $L$. If $w, W \models A$ for every $w \in W$, we say that $(W, \subseteq, h, u, o)$ satisfies $A$. If $w, W \models A$ for every $w \in W$ such that the constants in $A$ are in $\text{in}(w)$, then we say that $A$ is true in $(W, \subseteq, h, u, o)$. Finally, if $\phi$ is a set of sentences and $\psi$ is a sentence, we write $\phi \models \psi$ if and only if $\psi$ is true in every Kripke structure that satisfies $\phi$.

We now apply the machinery of [21], which was developed for a one-sorted (atemporal) language, to the two-sorted language $L$. First, to state the circumscription axiom, we need to extend the concept of a final Kripke model. Let $J_1 = \langle W_1, \subseteq_1, h_1, u_1, o_1 \rangle$ and $J_2 = \langle W_2, \subseteq_2, h_2, u_2, o_2 \rangle$ be two Kripke structures for $L$. We say that a mapping $\tau : W_1 \rightarrow W_2$ is a homomorphism from $J_1$ into $J_2$ if and only if (i) it preserves $\subseteq$, and (ii) it preserves $h$, $u$ and $o$ relative to some fixed (but arbitrary) domain isomorphism $i$. (Note that this is possible only because we have assumed that the total domains $c$ and $t$ are countable.) Now let $K$ be an arbitrary class of Kripke structures and assume that $K = \langle W, \subseteq, h, u, o \rangle$ is a member of $K$.

**Definition 3.1** $K$ is a final Kripke structure for $K$ if and only if, for every $J \in K$ and every domain isomorphism $i$, there exists a unique homomorphism from $J$ into $K$.

It is easy to see that two final Kripke structures for $K$ are isomorphic, and thus either one could be designated as "the" final Kripke structure for $K$.

We typically use Definition 3.1 as follows: We take $K$ to be the class of Kripke structures that satisfy some set of first-order rules $R$, and we try to find a final Kripke structure, $K$, for $K$. If such a $K$ exists, we call it the final Kripke model for $R$. It then turns out (see Proposition 3.7 in [21]) that a universally quantified implication is entailed by $R$ if and only if it is true in $K$. Notice, however, that Definition 3.1 is not restricted to first-order theories, and could apply equally well to second-order theories, such as the circumscription axiom. On the other hand, not every instance of the circumscription axiom has a final Kripke model (see Section 5 of [21] for the discussion of a counterexample). In fact, the existence of a final Kripke model seems to be one measure of the coherence of a circumscribed theory in intuitionistic logic.

It is thus of some interest that we can always construct a final Kripke model for $\text{Circ}(R(P); P)$ when $R$ determines the class of events discussed in Section 2. These rules look like Horn clauses, but they include special expressions like: $B(x) \![t_1] \circ \neg B(x) \![t_2]$ and $\forall w \neg C(w, y) \! [t_1, t_2]$. Our approach is to treat these special expressions as atomic predicates in an extended language $L'$, and to define their meaning by a set of rules outside the scope of the circumscription axiom. Thus, if $B(x, t)$ and $C(x, y, t)$ are base predicates in $L$, i.e., predicates in $L$ that are not defined in $R$, we would adopt the following definitions:

**Definition 3.2**

\[
B(x) \![t_1] \circ \neg B(x) \![t_2] \iff B(x, t_1) \land [\bot \iff B(x, t_2)]
\]

\[
\neg B(x) \![t_1] \circ B(x) \![t_2] \iff [\bot \iff B(x, t_1)] \land B(x, t_2)
\]

\[
C(x, y) \! [t_1] \circ (\forall w) \neg C(w, y) \! [t_2] \iff C(x, y, t_1) \land (\forall w) [\bot \iff C(w, y, t_2)]
\]

\[
(\forall w) \neg C(w, y) \! [t_1] \circ C(x, y) \! [t_2] \iff (\forall w) [\bot \iff C(w, y, t_1)] \land C(x, y, t_2)
\]

\[
B(x) \! [t_1, t_2] \iff B(x, t_1) \land B(x, t_2) \land (\forall t) [B(x, t) \iff t_1 < t < t_2]
\]

\[
\neg B(x) \! [t_1, t_2] \iff \neg B(x, t_1) \land \neg B(x, t_2) \land (\forall t) [B(x, t) \iff t_1 < t < t_2]
\]

\[
C(x, y) \! [t_1, t_2] \iff C(x, y, t_1) \land C(x, y, t_2) \land (\forall t) [C(x, y, t) \iff t_1 < t < t_2]
\]

\[
(\forall w) \neg C(w, y) \! [t_1, t_2] \iff (\forall w) \neg C(w, y, t_1) \land (\forall w) \neg C(w, y, t_2) \land (\forall w) [\bot \iff C(w, y, t_1) \land t_1 < t < t_2]
\]

(We could extend this list in obvious ways, but this is sufficient for the examples in Section 2.) Let $D$ be the set of all such definitions. Even though our special expressions are "defined" in $D$, we will treat them as base predicates in $L'$, since they are not defined in $R$. We now proceed to the construction of the final Kripke model.

Let $c$ be any countable set of object constants distinct from the constants in $L$, and let the total order domain, $t$, be the rational numbers. Let $H$ be the set of all triples $\langle B, U, Q \rangle$,
where $U$ is any nonempty set of object constants in $L$ or $c$ that includes the constants in $R$, $Q$ is any nonempty set of rational numbers, and $B$ is any Herbrand interpretation for the base predicates in $L'$ over the universes $U$ and $Q$. Set $⟨B_1, U_1, Q_1⟩ ⊆ ⟨B_2, U_2, Q_2⟩$ if and only if $B_1 ⊆ B_2$ and $U_1 ⊆ U_2$ and $Q_1 ⊆ Q_2$. We note that $H \cup \{\{0,0,0\}\}$ is a complete lattice under this order, and we use ‘∩’ and ‘∪’ to denote the meet and join, respectively, in this lattice. It is obvious that any subset of $H$ could be interpreted as a Kripke structure for the base predicates in $L'$. Simply define $u(⟨B, U, Q⟩) = U$ and $o(⟨B, U, Q⟩) = Q$, and define:

$$h(P)(⟨B, U, Q⟩) = \{(c^n, t^m) | P(c^n, t^m) \in B\}$$

for every base predicate $P$ in $L'$. Now, using the techniques in [18], let $W^*$ be the largest subset of $H$ such that $\{W^*,\subseteq, h, u, o\}$ satisfies the definitions in $D$. (This set can always be constructed as the greatest fixed point of the “deletion” transformation associated with $D$.) For the defined predicates in $L'$, let $S_{U,Q}^T \cap (B)$ be the least fixed point of the van Emden-Kowalski “one-step consequence” operator for $R$ [40] over the universes $U$ and $Q$ that includes $B$. (For the details of this construction, see [21].) We can now define:

$$h^*(P)(⟨B, U, Q⟩) = \{(c^n, t^m) | P(c^n, t^m) \in S_{U,Q}^T \cap (B)\}$$

for every predicate $P$ in $L'$. Note that $h^*(P) = h(P)$ whenever $P$ is a base predicate. Our main result is:

**Theorem 3.3** Let $R$ be a set of Horn clauses in $L'$ with defined predicates $P$, and let $D$ be given by Definition 3.2. Then $\{W^*,\subseteq, h^*, u, o\}$ is the final Kripke model for Circ($R(P); P) \cup D$.

This result follows by a minor modification of Theorem 4.7 in [21].

Now that we have constructed the final Kripke model for Circ($R(P); P) \cup D$, we can discuss the concept of definite and indefinite events. Consider the following:

**Definition 3.4** Let $T$ be a theory in a language $L$, and let $A, B$ and $A(x)$ be atomic formulae in $L$. We say that $T$ has the disjunctive property if:

$$T \models A \lor B \iff T \models A \lor T \models B.$$  

We say that $T$ has the existential property if:

$$T \models (\exists x)A(x) \iff T \models A(x)\theta \text{ for some } \theta.$$  

It should be apparent that $R \cup D$ would not have these properties if entailment were interpreted classically, even in a simple language without linear order constraints. For example, if ‘$Q(a) \iff \neg B(b)$’ is in $D$, then $Q(a) \lor B(b)$ is entailed in classical logic, but neither atom is entailed by itself. If ‘$P(x) \iff Q(x)$’ and ‘$P(x) \iff B(x)$’ are also in $R$, then $(\exists x)P(x)$ is entailed in classical logic, but there is no substitution $\theta$ such that $P(x)\theta$ is entailed. However, in intuitionistic logic, the disjunctive and existential properties would hold. In fact, we can make a stronger statement. Let $R$ be a set of Horn clauses and let $D$ be a set of definitions including “embedded implications” and “embedded negations,” as in Definition 3.2. Let $\phi$ be a conjunction of atomic formulae, and assume that $\psi_1, \psi_2$ and $\psi(x)$ are also atomic. Then:

$$R \cup D \models \psi_1 \lor \psi_2 \iff \phi \quad (3)$$

and

$$R \cup D \models (\exists x)\psi(x) \iff \phi \quad (4)$$

as long as entailment is interpreted intuitionistically. (If $\phi$ or $\psi$, or both, have free variables in these entailments, we simply take the universal closure on the right-hand side.) These results follow from the fact that the final Kripke model for $R \cup D$ has an intersection property analogous to the model intersection property of Horn clause logic (see [18]). Intuitively, (3) and (4) show that we can assert the conjunction $\phi$ in the context of a theory $R \cup D$ and the disjunctive and existential properties will still hold.

However, the situation becomes more complicated when we add circumscription and an order sort. Let us say that $\phi$ is a definite event if (3) and (4) still hold for the two-sorted language $L'$ when the theory $R \cup D$ is strengthened to the theory Circ($R(P); P) \cup D$. It is now insufficient that $\phi$ is a conjunction of atomic formulae. For example, if $R$ includes:

$$P \iff Q(a) \text{ and } P \iff Q(b),$$

then $Q(a) \lor Q(b) \iff P$ is entailed by Circ($R(P); P)$, but neither $Q(a) \iff P$ nor $Q(b) \iff P$ is entailed. In general, the disjunctive and existential properties will not hold in the context of a circumscribed theory if defined predicates are included in $\phi$. Even if $\phi$ consists entirely of base predicates, the disjunctive and existential properties are not guaranteed. For example, suppose $B(t_1, t_2)$ is a base predicate and $R$ includes the rules:

$$P \iff B(t_1, t_2) \land t_1 < t_2,$$

$$Q \iff B(t_1, t_2) \land t_1 \geq t_2.$$  

Then, because $t_1$ and $t_2$ are interpreted over a linear order, the universal closure of $P \lor Q$ is $B(t_1, t_2)$ is entailed by $R$, but neither $P \iff B(t_1, t_2)$ nor $Q \iff B(t_1, t_2)$ is entailed.
Obviously, we can correct this problem if we specifically assert the order relations on \( t_1 \) and \( t_2 \). This will give us the class of \textit{basic events}. If \( B(x_1, \ldots, x_n, t) \) is a base predicate with only one order argument, then \( B(x_1, \ldots, x_n, t) \) is a basic event. If \( B(x_1, \ldots, x_n, t_1, t_2) \) is a base predicate with two order arguments, including the predicates in Definition 3.2, then

\[
B(x_1, \ldots, x_n, t, t),
\]

\[
B(x_1, \ldots, x_n, t_1, t_2) \land t_1 < t_2,
\]

\[
B(x_1, \ldots, x_n, t_1, t_2) \land t_1 > t_2,
\]

are all basic events. (Again, these definitions, as well as Definition 3.2, could be extended, but this is sufficient for the examples in Section 2.) The following proposition now follows easily from an analysis of the final Kripke model for \( \text{Circ}(\mathcal{R}(P); P) \cup D \).

**Proposition 3.5** If \( \phi \) is a basic event, then the disjunctive and existential properties stated in (3) and (4) hold for the strengthened theory \( \text{Circ}(\mathcal{R}(P); P) \cup D \) in the two-sorted language \( L' \) with linear order constraints.

Intuitively, basic events are the minimal (nonempty) definite events, and they correspond to the minimal (nonempty) worlds of \( (W^*, \sqsubseteq, h^*, u, o) \). We will use this fact in the following section.

## 4 Deontic Modalities

This section develops the semantics of \( P(\alpha, x) \), \( F(\alpha, x) \), and \( O(\alpha, x) \), as well as the semantics of \( \text{DO}(\alpha, x) \). The basic idea is to use the final Kripke model \( (W^*, \sqsubseteq, h^*, u, o) \) to define two concepts: (i) the denotation of an action \( \alpha \), written \( \langle \alpha \rangle \); and (ii) the Grand Permitted Set, written \( P \). The deontic modalities are then interpreted as statements about the possible relationships between \( \langle \alpha \rangle \) and \( P \). This is a variant of the deontic semantics first presented in [16, 17].

Consider a basic event, such as ‘SwimLap\( (x, t_1, t_2) \land t_1 < t_2 \)’. To assert that a particular agent, \( c_0 = x\theta \), performs this event at the particular times \( t_1 = t_1\theta < t_2 = t_2 \) is to assert that \( c_0 \) determines the evolution of a very small piece of the world between \( t_1 \) and \( t_2 \). A natural way to represent this fact in \( W^* \) is to point to the set:

\[
V = \{ w | w, W^* \models \text{SwimLap}(x, t_1, t_2)\theta \}.
\]

This set has a least element, \( \cap V \), corresponding to the small piece of the world controlled by \( c_0 \), but it also includes all possible completions of \( \cap V \), corresponding to the myriad ways that other agents (as well as \( c_0 \) herself, wearing a different hat!) could determine the evolution of the world, at \( t_1 \) and \( t_2 \), and at all other times \( t \). When we associate such a set with a particular agent, we think of it as a \textit{basic action}. Since all actions are defined by Horn clauses from basic actions, we adopt the following:

**Definition 4.1** The denotation of a ground action \( \alpha \), written \( \langle \alpha \rangle \), is defined recursively as follows:

- If \( \alpha = B\theta \) for some base predicate \( B \) and some ground substitution \( \theta \), then
  \[
  \{ w | w, W^* \models B\theta \} \in \langle \alpha \rangle
  \]

- If \( \alpha = R\theta \) for some ground substitution \( \theta \), where \( R \) is defined by the rule
  \[
  R \leftarrow Q_1 \land \ldots \land Q_k \land O,
  \]
  and if \( V_i \in \{ Q_i\theta \} \) for \( i = 1, \ldots, k \), and if the order constraints \( O\theta \) are satisfied in the natural order on the rationals, then
  \[
  V_1 \cap \ldots \cap V_k \in \langle \alpha \rangle.
  \]

Because of the way \( (W^*, \sqsubseteq, h^*, u, o) \) is constructed, every \( V \in \langle \alpha \rangle \) is a \textit{principal filter} in \( W^* \), i.e., it is an upward closed subset with a least element. We want the Grand Permitted Set, \( P \), to have the same property, so we simply define \( P \) for the agent \( c_0 \) to be any arbitrary set of principal filters in \( W^* \). Intuitively, \( P \) represents the \textit{permissible} ways that \( c_0 \) can determine the evolution of the world.

We are now ready to define the modalities, \( P, F, \) and \( O \). Given a Kripke structure \( (W, \sqsubseteq, h, u, o) \) for the action language, we extend it to a \textit{deontic structure} \( (W, P, \sqsubseteq, h, u, o) \) by adding a specification of the Grand Permitted Set. More precisely, there may be many Grand Permitted Sets, at least one for each agent, \( c \), and each of these may vary with \( w \) in \( W \), since we are still working with intuitionistic logic. Let us denote these sets by \( P_c(w) \), and include them all in the specification \( P \). In general, the domains of \( (W, \sqsubseteq, h, u, o) \) may be different from the domains of \( (W^*, \sqsubseteq, h^*, u, o) \), but they are isomorphic, and it simplifies our notation if we assume that they are identical. Thus, although \( \langle \alpha \rangle \) and \( P \) are actually defined on \( W^* \), we will write them as if they used the constants in \( (W, \sqsubseteq, h, u, o) \).

**Definition 4.2** Let \( (W, P, \sqsubseteq, h, u, o) \) be a deontic structure, and let \( \alpha \) be a ground event. The \textit{forcing conditions} for \( P, F, O \) are:

- \( w, W \models P(\alpha, c) \) iff, for all \( w' \sqsupseteq w \) in \( W \), the following condition holds:
  \[
  \forall V : V \in \langle \alpha \rangle \implies \exists V' : V \subseteq V \land V' \in P_c(w').
  \]
• \(w, W \models F(\alpha, c)\) iff, for all \(w' \supseteq w\) in \(W\), the following condition holds:

\[
\forall V, V' : V \in [\alpha] \land V' \subseteq V \quad \rightarrow \quad V' \not\in \mathcal{P}(w').
\] (6)

• \(w, W \models O(\alpha, c)\) iff, for all \(w' \supseteq w\) in \(W\), the following condition holds:

\[
\forall V' : V' \in \mathcal{P}(w') \quad \rightarrow \quad \exists V : V' \subseteq V \land V \in [\alpha].
\] (7)

To understand these definitions, recall that \(V' \subseteq V\) means that \(V'\) says more about the world than \(V\), since \(\cap V' \supseteq \cap V\). Also note that \(\mathcal{P}(w)\) is allowed to vary arbitrarily, i.e., nonmonotonically, over \(\subseteq\), but the deontic atoms \(P(\alpha, c), F(\alpha, c)\) and \(O(\alpha, c)\) increase only monotonically over \(\subseteq\) by virtue of the definition of forcing. (This will turn out to be a useful feature when we subsequently investigate negation-as-failure.)

Definition 4.2 can be simplified in the case of \(F\) and \(O\), as shown by the following:

**Lemma 4.3** Condition (6) in Definition 4.2 is equivalent to:

\[
\forall w : w, W^* \models \alpha \quad \rightarrow \quad \{w' \in W^* \mid w' \supseteq w\} \not\in \mathcal{P},
\] (8)

and condition (7) in Definition 4.2 is equivalent to:

\[
\forall w : \{w' \in W^* \mid w' \supseteq w\} \in \mathcal{P} \quad \rightarrow \quad w, W^* \models \alpha.
\] (9)

**Proof:** We outline the proof for \(F(\alpha, c)\). Assume (6), and choose a \(w\) such that \(w, W^* \models \alpha\). Since \((W^*, \subseteq, h^*, u, o)\) is the final Kripke model for \(\text{Circ}(R(P) ; P) \cup D\), this means that \(\alpha \in S_{P, Q} \uparrow (B)\) for some \((B, U, Q)\). Thus there exists a \(V \in [\alpha]\) such that \(\{w' \in W^* \mid w' \supseteq w\} \subseteq V\), and from (6) we conclude that \(\{w' \in W^* \mid w' \supseteq w\} \not\in \mathcal{P}\).

Conversely, assume (8) and choose any \(V \in [\alpha]\) and any \(V' \subseteq V\). If \(V' \in \mathcal{P}\), then \(V'\) has a least element \(w_0\), and \(V' = \{w' \in W^* \mid w' \supseteq w_0\}\). But then \(w_0, W^* \models \alpha\), and (8) implies that \(V' \not\in \mathcal{P}\), a contradiction. Thus \(V' \not\in \mathcal{P}\).

The proof for \(O(\alpha, c)\) is similar. \(\square\)

This result shows that our definition of \(F\) and \(O\) is basically the same as the definition in [16, 17]. The differences are minor: \(P\) was defined in [16, 17] on the least elements of the principal filters in \(W^*\), whereas here it is defined on the principal filters themselves. With this translation, all of the results in [16, 17] for \(F\) and \(O\) carry over to the present work. Note also the rough intuitive reading of (9): “\(\alpha\) is obligatory if it is true in all permitted worlds.” This means that our logic, for \(F\) and \(O\), is similar to standard deontic logic [5].

However, for \(P\), Definition 4.2 differs both from standard deontic logic and from the logic presented in [16, 17]. The contrast with standard deontic logic has already been discussed: It should be clear from condition (5) that \(P\) is a “free choice permission.” The other difference is more subtle. The definition of \(P\) in [16, 17] would be translated as follows:

\[
\forall V : V \in [\alpha] \quad \rightarrow \quad V \subseteq \mathcal{P}(w').
\] (10)

This is a plausible definition, but it has a curious consequence: \(P(\alpha, c) \Rightarrow \neg O(\beta, c)\) when \(\alpha\) and \(\beta\) are disjoint actions. (This fact was first pointed out by Ron van der Meyden.) For example: “If you are permitted to sell your house this year, then you are not obligated to pay your taxes.” If \(P\) is defined using condition (5), however, permission and obligation are independent (see Theorem 4.8, below). Under the current interpretation, it is possible to say that you are “permitted to sell your house this year”; if, in fact, you are “permitted to sell your house and pay your taxes this year.” Moreover, that would be a necessary implication, since \(P(\alpha \land \beta, c) \Rightarrow P(\alpha, c)\) becomes a valid formula when \(P\) is defined using condition (5). Although there are situations in which condition (10) seems appropriate (for example, if we wanted to focus very narrowly on a specific planning problem, and ignore all extraneous obligations), the current version of Definition 4.2 seems closer to our common sense intuitions about permission and obligation.

Now that we understand the semantics of \(P, F\) and \(O\), it is simple to define a compatible semantics for \(DO\). Since the permissible actions available to an agent, \(c\), are represented in our system by \(\mathcal{P}(w)\), which is a set of principal filters in \(W^*\), the action actually taken by an agent, \(c\), should be represented by a single principal filter in \(W^*\). Let us write this as \(D_c(w)\). We need a veridicality assumption, of course, and this can be conveniently encoded by the unique homomorphism \(\tau\) from \(\langle W, \subseteq, h, u, o \rangle\) into \(\langle W^*, \subseteq, h^*, u, o \rangle\).

We thus stipulate that:

\[
\tau(\{w' \in W \mid w' \supseteq w\}) \subseteq D_c(w),
\]

for all \(w \in W\), which implies that \(\tau(w) \supseteq \cap D_c(w)\).

Finally, we include the sets \(D_c(w)\) in a new component, \(D\), of our deontic structure:

**Definition 4.4** Let \(\langle W, D, P, \subseteq, h, u, o \rangle\) be a deontic structure, and let \(\alpha\) be a ground action. The forcing condition for \(DO\) is:

• \(w, W \models DO(\alpha, c)\) iff, for all \(w' \supseteq w\) in \(W\),

\[
\exists V : D_c(w') \subseteq V \land V \in [\alpha].
\]

It should now be clear what it means to say that an agent always “obeys the law,” namely: \(D_c(w)\) must be an element of the Grand Permitted Set!
Definition 4.5 \( \langle W, D, P, \sqsubseteq, h, u, o \rangle \) is causal for the agent \( c \) iff \( D_c(w) \in P_c(w) \) for all \( w \in W \).

Proposition 4.6 If \( \langle W, D, P, \sqsubseteq, h, u, o \rangle \) is causal for the agent \( c \), then

\[
\vdash \bot \iff DO(\alpha, c) \land F(\alpha, c),
\]

\[
\vdash DO(\alpha, c) \iff O(\alpha, c).
\]

Proof: Immediate, from Definitions 4.2, 4.4 and 4.5. \( \Box \)

We conclude this section with two theorems that are useful for the development of a proof theory. The first theorem states that certain inferences involving \( O \) and \( F \) can be reduced to inferences in the action language.

Theorem 4.7 Assume that \( O(\alpha_i) \), \( F(\beta_i) \), \( O(\gamma) \) and \( F(\gamma) \) are deontic atoms with identical agents, \( x \), and let \( f \) denote a special nullary predicate that does not appear in \( R \) or \( D \). Then

\[
\vdash O(\gamma) \iff \bigwedge_{i=1}^{n} O(\alpha_i) \land \bigwedge_{i=1}^{m} F(\beta_i)
\]

iff

\[
\text{Circ}(R(P);P) \cup D \models \gamma \lor f \iff \bigwedge_{i=1}^{n} \alpha_i \land \bigwedge_{i=1}^{m} f \iff \beta_i.
\]

Similarly,

\[
\vdash F(\gamma) \iff \bigwedge_{i=1}^{n} O(\alpha_i) \land \bigwedge_{i=1}^{m} F(\beta_i)
\]

iff

\[
\text{Circ}(R(P);P) \cup D \models f \iff \gamma \land \bigwedge_{i=1}^{n} \alpha_i \land \bigwedge_{i=1}^{m} f \iff \beta_i.
\]

Proof: We outline the proof for (11), and note that the proof for (13) is similar. Assume that (11) is false. Then, by Definition 4.2 and Lemma 4.3, there exists a Grand Permitted Set \( P \) such that condition (8) holds for all \( \beta_i \), and condition (9) holds for all \( \alpha_i \) but fails for \( \gamma \). Thus, for \( \gamma \), there exists some \( w_0 \) such that \( \{ w' \in W^* \mid w' \equiv w_0 \} \in P \) but \( w_0, W^* \not\models \gamma \). Suppose we add the special nullary predicate \( f \) to the worlds of \( W^* \), so that

\[
h^*(f)(w) = \{ \} \iff \{ w' \in W^* \mid w' \equiv w \} \notin P.
\]

Then \( w_0, W^* \models \alpha_i \) by (9), and \( w_0, W^* \not\models F(\beta_i) \) by (8), but \( w_0, W^* \not\models \gamma \lor f \). Since \( f \) does not appear in \( R \) or \( D \), however, we still have a Kripke model for \( \text{Circ}(R(P);P) \cup D \), and hence a countermodel to (12).

Conversely, assume that (12) is false. Then there exists a Kripke structure, \( J \), in a language including \( f \) that satisfies \( \text{Circ}(R(P);P) \cup D \) but falsifies the implication in (12) at some world \( w_0 \). Let \( \tau \) be the unique homomorphism from \( J \), with the \( f \)'s removed, into \( \langle W^*, \sqsubseteq, h^*, u, o \rangle \), and then add the \( f \)'s back to the world of \( W^* \) so that \( \tau \) preserves \( h(f) \) as well. In addition, for those worlds of \( W^* \) that are not in the image of \( \tau \), add \( f \) to \( \tau \) whenever \( w, W^* \models \beta_i \) for some \( i \). (Note that this cannot affect \( \tau(w_0) \), since \( \tau(w_0), W^* \not\models f \) and \( \tau(w_0), W^* \not\models \beta_i \) for all \( i \).) Now define a Grand Permitted Set \( P \) on \( W^* \) by setting

\[
\{ w' \in W^* \mid w' \equiv w \} \in P \iff w, W^* \models \alpha_i \text{ for } i = 1, \ldots, n, \text{ and } w, W^* \not\models f.
\]

It is straightforward to verify that this \( P \) provides a counter-model to (12) at \( \tau(w_0) \). In particular, we have \( \{ w' \in W^* \mid w' \equiv w_0 \} \in P \) by construction but \( \tau(w_0), W^* \not\models \gamma \), and thus condition (9) fails for \( \gamma \). \( \Box \)

The second theorem states that inferences about \( O \) are independent from inferences about \( P \).

Theorem 4.8 Assume that \( O(\alpha_i) \), \( P(\beta_i) \) and \( O(\gamma) \) are deontic atoms with identical agents, \( x \). Then

\[
\vdash O(\gamma) \iff \bigwedge_{i=1}^{n} O(\alpha_i) \land \bigwedge_{i=1}^{m} P(\beta_i)
\]

iff

\[
\text{Circ}(R(P);P) \cup D \models f \iff \gamma \land \bigwedge_{i=1}^{n} \alpha_i \land \bigwedge_{i=1}^{m} f \iff \beta_i.
\]

Proof: It is obvious that (16) implies (15). For the converse, assume that (16) is false. Then there exists a Grand Permitted Set \( P_2 \) such that condition (7) in Definition 4.2 holds for all \( \alpha_i \) but fails for \( \gamma \). Thus, for \( \gamma \), we have:

\[
\exists V' : V' \in P_1 \land \forall V : V' \subseteq V \rightarrow V \notin [\gamma].
\]

Construct a new Grand Permitted Set by defining

\[
P_2 = \{ V_1 \cap V_2 \mid V_1 \in P_1, V_2 \in [\beta_i] \text{ for some } i \},
\]

and then setting \( P = P_1 \cup P_2 \). (Note: It is necessary to verify that \( P_2 \) is a set of principal filters in \( W^* \).) Using this new \( P \), we note that condition (5) in Definition 4.2 now holds for all \( \beta_i \) and condition (7) still holds for all \( \alpha_i \). Moreover, condition (7) still fails for \( \gamma \), since (17) obviously remains true when we replace \( P_1 \) by \( P_1 \cup P_2 \). We have thus shown that (15) is false. \( \Box \)

These two theorems show that it is possible to construct a simplified proof theory for certain fragments of our language. If we use \( O \) and \( F \), but not \( P \), Theorem 4.7 shows that the most important inferences in the deontic language can be reduced to inferences in the action language, which
can then be solved using the techniques of intuitionistic logic programming, as in [22, 23]. If we use $O$ and $P$, but not $F$, Theorem 4.8 shows that we can compute the obligations first, and then compute the permissions independently, which turns out to be very simple. On the other hand, if we mix all three modalities in a single system, the situation is more complex. It is interesting to note that all the examples in Section 2 fall into one category or another, and there may be cognitive (and computational) reasons why this is so. We will discuss these observations on the proof theory in the second part of this paper.

5 Discussion

In standard deontic logic [5], modalities are applied to propositions, but there have been several attempts to develop a deontic logic in which modalities are applied to actions, as in the present work. A natural approach is to combine deontic logic with dynamic logic [33, 10] or process logic [34], and the earliest example of this approach seems to be [16, 17]. Meyer, et al., have investigated a variant of dynamic logic in which a special atom, $v$, is added to the language to denote a violation of the norms [30, 42], and they have shown that this system has some attractive properties as a deontic logic. A system proposed by van der Meyden, and shown to have a sound and complete proof theory in [27], is interesting because of the way it combines weak permission with free choice permission, but it does not seem to offer a natural concept of obligation. Another system, proposed by van der Meyden in [28], applies the techniques of logic programming to a deontic specification language, including the idea of a minimal model semantics and a least fixed-point operator.

An alternative approach, which is prominent in the philosophical literature, is to add the deontic operators to a language that already includes the modal action operator “see to it that,” or $stit$ [32, 4]. In such a language, we can say things like “the agent $x$ is obligated to see to it that $p$,” where $p$ is any arbitrary proposition. The intuitions underlying such a modality are very different from the intuitions underlying dynamic logic, and even further removed from the intuitions underlying process logic. The focus, in $stit$ theory, is on the goals that an agent is trying to achieve, whereas the focus in dynamic logic and process logic is on the trajectories of possible actions. The present paper clearly belongs in the latter camp: We can talk about agents achieving goals in our language, as in Examples 2.3 and 2.4, but these are derived concepts, not primitives.

It is unlikely that the merits of the various proposals for deontic logic will be judged on their semantics alone. More likely is an evaluation based on the utility of a particular formalism for the pragmatic tasks of common sense reasoning. For this, at a minimum, we need to look at the proof theory, and at the facilities for drawing nonmonotonic inferences. These topics will be discussed in the sequel.

References


