PERMISSIONS AND OBLIGATIONS

L. Thorne McCarty
Computer Science Department
Rutgers University
New Brunswick, New Jersey

ABSTRACT: This article describes a formal semantics for the deontic concepts -- the concepts of permission and obligation — which arises naturally from the representations used in artificial intelligence systems Instead of treating deontic logic as a branch of modal logic, with the standard possible worlds semantics, we first develop a language for describing actions, and we define the concepts of permission and obligation in terms of these action descriptions. Using our semantic definitions, we then derive a number of intuitively plausible inferences, and we show generally that the paradoxes which are so frequently associated with deontic logic do not arise in our system*

I INTRODUCTION

The representation of deontic concepts — the concepts of permission and obligation -- has not yet been seriously addressed in the artificial intelligence literature, but there are numerous application areas in which these concepts seem to be required. In our work on the TAXMAN Project [1] [2], for example, we represent the characteristics of various kinds of stocks and bonds by describing the rules of permission and obligation which are binding, at any given time, on the corporation and its securityholders In our work on the "usufructuary" provisions of the Louisiana Civil Code [3], just recently initiated, we have encountered a similar need for the representation of complex permissions and obligations Nor are these examples confined to legal domains In the classical work on single agent planning systems, e.g., [4], the operators which change the state of the world can be interpreted as a set of "permitted" actions, but in a more realistic planning environment, with multiple agents, we would expect to see "obligatory" actions as well, and we would expect to see the actions of one agent produce modifications in the rules of permission and obligation binding upon another agent. Similar observations apply to the field of computer security, see, e.g., [5], where there has been extensive debate over the appropriate "authorization mechanisms" for a community of computer users. For all of these purposes, a formalization of the concepts of permission and obligation appears to be essential

Outside of the field of artificial intelligence, there exists an extensive literature on the deontic concepts, by logicians [6] [7], philosophers [8] [9], and lawyers [10] [11] [12], but the attempts to formalize these concepts have generally led to paradox Since the 1950s, deontic logic has been treated as a branch of modal logic, with the necessity' operator replaced by the obligation" operator, O. and the "possibility" operator replaced by the "permission" operator, P Many of the theorems of modal logic turn out to be intuitively correct under this translation For example, the dual relationship between necessity and possibility becomes a dual relationship between obligation and permission, Op = ~P~p. and this formula certainly seems plausible If it is false that you are permitted to do not-p, then you are obligated to do p. and vice versa The formula Op > p. which is valid in any modal system with a reflexive accessibility relation between possible worlds, would not be plausible in a deontic logic, since people in the actual world do not always abide by their obligations, but it can be replaced by the more plausible formula Op => Pp, which is valid as long as every possible world has some possible world accessible from it This point was first noted by Kripke, in one of his original papers on possible worlds semantics [13].

Despite these positive results, there are several other modal formulae which seem counterintuitive in a deontic logic, and which cannot be so easily modified. For example, the formula for disjunctive permission, Pp D Pip v q), contradicts our ordinary understanding of what it means to grant permission to do p v q, but this formula is valid even in the weakest modal systems Likewise, any formula containing an iterated operator, such as OPp or POp, seems anomalous in a deontic context, and yet the various modal systems are distinguished precisely by the way in which they handle these iterated modalities. Of course, it may make sense to say that you are permitted to impose a particular obligation upon someone else, or upon yourself, and we might conceivably write this as POp, but the inferences we would make about such statements do not correspond at all to the inferences which are valid in the standard possible world semantics. Finally, even the dual relationship between permission and obligation seems problematical if we cast it into the form Pp = ~0~p If it is false that you are obligated to do not-p, i.e., if it is false that you are forbidden to do p, does it follow that p is permitted ? Stringing together all of these questionable inferences, it is not surprising that we can generate a host of "deontic paradoxes," and the literature is full of them For a survey, see [14] and [15]

In this paper, we will develop a formal semantics for

**This article is based upon work supported by Grant No. MCS-82-03591 from the National Science Foundation, Washington, D.C., and by a grant from the Louisiana State Law Institute, Baton Rouge, Louisiana.

permissions and obligations which seems to avoid these difficulties, and we will do so in a way which is entirely natural for an artificial intelligence system Instead of representing the deontic concepts as operators applied to propositions, as in a standard modal logic, we will represent them as dyadic forms which take condition descriptions and action descriptions as their arguments The most important part of this representation is the use of action descriptions in the place of propositions Instead of granting permissions and imposing obligations on the state of the world itself, we will grant permissions and impose obligations on the actions which change the state of the world This is an approach long advocated by Castaneda [15], and pursued in various forms by von Wright [16] [14]. but to carry out this approach in full it seems necessary to establish a connection between the abstract description of an action and the concrete changes that occur in the world when the action takes place This has been a major concern of artificial intelligence research throughout its history [17] [18] [19], of course, and we will draw upon this earlier work in constructing our formalisms Although the actions that we actually discuss in this paper are fairly simple ones, intended to highlight the principal features of the deontic representation, the action descriptions themselves can be extended to more realistic situations, in several ways We will return to this point in our concluding remarks

II DEONTIC SEMANTICS

In this section we develop a formal semantic interpretation of the deontic concepts, using a variant of the possible worlds approach Our strategy proceeds in stages We start with an ordinary first-order language L and a set of states S. and we use these materials to construct a new language L_A in which we are able to describe actions The formulae of L_1 are evaluated with respect to the states in S. as usual, but the formulae of L_A are evaluated with respect to sequences of states, or worlds. We thus have a way of saying that an action is "true' in, or is satisfied" by, a particular world The details of these constructions are presented in Sections IIA and II.B below. Now consider a state r which is situated at the junction between a "past world v and a "future" world w We assume that there exists a set P_r which tells us, for each past world v, all the future worlds w which are "permitted." Working exclusively with this permitted set P_r, we construct three expressions which tell us whether an action at r is permitted, forbidden, or obligatory, respectively These expressions then become part of our deontic language L_D The details of these constructions are presented in Section IIC below This is not the end of the story, however. Since each deontic expression has a definite truth value at each state in S, it turns out that the language L_D can be embedded within our original first-order language L , and thus the process of linguistic construction we have outlined here becomes fully recursive This technique enables us to represent "dynamic" permissions and obligations, i.e., permissions and obligations which change over time, without the use of iterated modalities This latter point is developed in Section II.D

The principal technical difficulty in this development rises in connection with the definition of "satisfaction" for the language L_A Our initial approach is similar in spirit to the approach of Harel [20] and Rosenschein [21]: We define a primitive action to be a relation between two states, and we define the meaning of the more complex formulae of L_A by a set of recursive truth definitions on arbitrary sequences of states But the ordinary notion of satisfaction in L_1 which takes into account the complete state of the world at a given time, is too imprecise for our purposes here, and we will supplement it with a notion of strict satisfaction, which associates with each action in L_A the specific set of changes in the world attributable to that action It turns out that this notion of strict satisfaction is absolutely essential to the construction of the deontic language L_D Without it, our definition of a rule of permission simply would not work We will return to this point in Section IIIB

A. State Descriptions

Let L_1 be a many-sorted function-free first-order language with equality, and let S be a set of states with respect to which the formulae of L_1 are evaluated We will follow the standard procedures for specifying the syntax and the semantics of a first-order language Thus, if (Own x y) is a formula of L_1 with free variables x and y, and if o is an assignment of the variables x and y to elements in the domain of interpretation of L_1 and if Own(s) is the set of tuples defining the extension of the predicate Own for s € S, then we will say that (Own x y) is true m s under the assignment o" if and only if $\langle\sigma(x),o(y)\rangle$ E Own(s) We will write this in general as σ,& = (Own x y), but if the assignment σ is fixed and clear from the context, we will often omit it from the notation and write s = (Own x y) Truth conditions for the nonatomic formulae of L will also be defined in the standard way If there are constraints in our domain of interpretation, expressible as a finite set of formulae in L_1 we will simply assume that S has been restricted in advance to include only those states in which the constraints are conjunctively satisfied To avoid any mathematical complexities, however, and to reveal the points of greatest importance to the representational problems of artificial intelligence, we will also assume, whenever it is convenient to do so, that the relevant sets are finite Thus we may assume that the predicate symbols are finite, that the domain of interpretation is finite, and so on We will attempt to remove these restrictions at a later date

We plan to use the states in S to represent the world at different points in time, as in a standard temporal logic, but with one important modification Normally, each $s \in S$ is assumed to provide us with a *complete* specification of the state of the world at a particular time, but we wish to work primarily with *partial* specifications of the state of the world. Our approach is similar to the approach of Barwise and Perry [22]. Since we can think of the complete specification of a state $s \in S$ as a collection of *sets*, one set for each predicate in L_1, it is natural to think of a partial specification of s as a collection of *subsets*, one subset for each predicate and one subset for the complement of each predicate in L_1. Specifically, for $s \in S$, let *s* be an ordered collection of sets

$$\langle P_1, P^c_1, \ldots, \ldots, P_n, P^c_n \rangle$$

satisfying the requirement that

$$P_k(s) \subset P_k(s) \text{ and } P^c_k(s) \cap P_k(s) = \emptyset$$

for all predicates P_k in L_1. Under these conditions, we will say that s is a *substate* of $\mathbf{s}$, and we will denote the set of all such substates by $S(\mathbf{s})$. It is important to note that these definitions depend explicitly on $\mathbf{s}$. Within each $S(\mathbf{s})$, there exists a natural partial order, given by set inclusion, and an additive binary operation, given by set union, but this will not be the case in general for arbitrary substates. Specifically, for any s and t in $S(\mathbf{s})$, we define

$$s<t \leftrightarrow \text{for all } k,\ P_k(s) \subseteq P_k(t) \text{ and } P_k^c(s) \subseteq P_k^c(t),$$
$$\text{and for some } k,$$
$$\text{either } P_k(s) \subset P_k(t) \text{ or } P_k^c(s) \subset P_k^c(t).$$

Also, for any s and t in $S(\mathbf{s})$, we define $s+t$ to be the ordered collection of all sets of the form $P_k(s) \cup P_k(t)$ and $P_k^c(s) \cup P_k^c(t)$, so that $s+t$ is itself a member of $S(\mathbf{s})$. We now extend these ideas from states and substates, to *worlds* and *subworlds*. We define a world $\mathbf{w}$ to be a linear sequence of states $\{\mathbf{s}_i | \mathbf{s}_i \in \mathbf{S}\}$, and we denote the set of all such worlds by $\mathbf{W}$. Given any $\mathbf{w} \in \mathbf{W}$, we define a subworld w to be a linear sequence of substates $\{s_i | s_i \in S(\mathbf{s}_i)$ for all $\mathbf{s}_i$ in $\mathbf{w}\}$, and we denote the set of all such subworlds by $W(\mathbf{w})$. It is convenient to index these worlds and subworlds as follows:

$$<\ldots\ \mathbf{s}_{-2},\ \mathbf{s}_{-1},\ \mathbf{s}_0,\ \mathbf{s}_1,\ \mathbf{s}_2,\ \ldots>$$

and to define a shift operator T^n which takes $\mathbf{w} = \{\mathbf{s}_i\}$ into $T^n\mathbf{w} = \{\mathbf{s}_{i+n}\}$. We can then extend the partial order "<" and the binary operation "+" to subworlds in the obvious way, component by component, with the nonexistent substates in a subworld interpreted as existing, but null, substates. Finally, we define an operation of sequential composition in $W(\mathbf{w})$: if $w_1 \in W(\mathbf{w})$ is a subworld ending in s_n, and if $w_2 \in W(T^{-n}\mathbf{w})$ is a subworld beginning with s_0, then we set $w_1;w_2 = w_1 + T^n w_2$.

We are primarily interested in these substates and subworlds because they can be used to specify the precise content of an action. In particular, the concept of a *subworld* is the technical device which we will use to focus on a small set of changes in the world, and to designate these changes as the "meaning" of an action description, while ignoring everything else that is occurring (or not occurring) in the same world at the same time. We will thus be making extensive use of the subworld concept in the remainder of this paper. The concept of a *substate* will be used only briefly, to define the meaning of the primitive actions, and for this we need only work with the atomic formulae in L_1. There are two main issues which need to be addressed for both substates and subworlds, however, and these issues are simpler for substates:

1.) Satisfaction. What does it mean to say that a formula A of L_1 is *true* in a substate $s \in S(\mathbf{s})$? It is convenient to adopt the following *monotonicity* condition:

$$s \models A \leftrightarrow (\forall t)[t \in S(\mathbf{s}) \wedge t \geq s \rightarrow t \models A]$$

to insure that the truth value of A cannot change as we move along the partial order to a more "complete" substate in $S(\mathbf{s})$. However, if A is an atomic formula in L_1, or if A is the negation of an atomic formula in L_1, we can use the same truth definition here that we used to define truth in a complete state $\mathbf{s} \in \mathbf{S}$. For example, define:

$$\sigma, s \models (Own\ x\ y) \leftrightarrow <\sigma(x),\sigma(y)> \in Own(s)$$

and define:

$$\sigma, s \models \neg(Own\ x\ y) \leftrightarrow <\sigma(x),\sigma(y)> \in Own^c(s)$$

and it is then clear from our construction of $S(\mathbf{s})$ that the monotonicity condition holds. If A is nonatomic, the intuitionistic truth definitions due to Kripke [23] will give us the results we want. For example, the existential quantifier can be defined in the usual way in each substate, and the monotonicity condition will still hold, but the universal quantifier in a substate s must be defined for all $t \in S(\mathbf{s})$ such that $t \geq s$. These definitions suggest a model-theoretic version of some of the recent results in nonmonotonic logic [24] [25] [26], but we will not pursue this analysis in the present paper. It is sufficient for our present purposes to confine our attention to the atomic formulae in L_1.

2.) Strict Satisfaction. It is possible, indeed quite probable, for a substate $s \in S(\mathbf{s})$ to satisfy a formula A of L_1 and still contain a great deal of "irrelevant" detail. To avoid this situation, we need to develop the notion of *strict satisfaction*, which we write as $s \models_0 A$. The intuitive idea here is that a substate *strictly satisfies* a formula A if it contains those tuples necessary for the truth of A, but no more. For an atomic formula A, the substate which strictly satisfies A is simple: it consists of a single nonempty set containing a single tuple. For example, the formula $(Own\ x\ y)$ is strictly satisfied by the substate s for which $Own(s) = \{<\sigma(x),\sigma(y)>\}$, and the formula $\neg(Own\ x\ y)$ is strictly satisfied by the substate s' for which $Own^c(s') = \{<\sigma(x),\sigma(y)>\}$. Note, however, that s and s' cannot both be members of the same set $S(\mathbf{s})$. The nonatomic formulae can be handled in the same intuitive way. If A is a conjunction of two atomic formulae, then $\sigma, s \models_0 A$ if and only if s contains the tuples satisfying both of the conjuncts, but no more; and if A is a disjunction of two atomic formulae, then $\sigma, s \models_0 A$ if and only if s contains a tuple satisfying one of the disjuncts, but no more. Negation cannot be defined in this way, but we can always transform a formula of L_1 so that all negations have atomic scope. At any rate, the notion of strict satisfaction will be used primarily within the context of our language of actions, L_A, in which, as we shall see, there are no general negation operators. Thus, once again, it is sufficient for our present purposes to confine our attention to the atomic formulae in L_1.

B. Action Descriptions

Let us now extend the concepts of satisfaction and strict satisfaction to the language of actions: L_A. We begin by constructing a set of *primitive actions*, and defining the truth conditions for these actions on a subworld consisting of only two substates: $w = <s,t>$. Our primitive actions are thus elementary *statechanges*, a familiar construct in artificial intelligence. It is convenient to work with the following set of actions:

1.) Affirmative Statechanges. Let A and B be atomic formulae in L_1, and let E be a (possibly empty) conjunction of equality and inequality constraints on the variables appearing in A and B. Suppose we can show that $\sigma, \mathbf{s} \not\models (A \wedge B \wedge E)$ for all σ and for all $\mathbf{s} \in \mathbf{S}$. We will say in this case that A and B are *incompatible* in $\mathbf{S}$, given the constraints in E. A fact of this sort would generally be established, not by an inspection of $\mathbf{S}$, but by a deduction from the axiomatic constraints we have imposed on our domain. For example, if our language L_1 contains the predicate *Own*, we might naturally impose the constraint that a single property can be owned by only one actor:

$(\forall x_1)(\forall x_2)(\forall y)[(Own\ x_1\ y) \wedge (Own\ x_2\ y) \supset (x_1 = x_2)]$

in which case we could take $A = (Own\ x_1\ y)$ and $B = (Own\ x_2\ y)$ as atomic formulae which are incompatible given the constraint $E = (x_1 \neq x_2)$. Under these conditions we can construct an expression $(A|B)_E$ to describe the action in which we "change the world from a situation in which A is true to a situation in which B is true."

Definition 1: Let A and B be atomic formulae which are incompatible in S, given the constraints in E, and let s and t be substates of some states s and t in S. Then $(A|B)_E$ is *true* on $\langle s,t\rangle$ under the assignment σ if and only if $\sigma,s \models A$ and $\sigma,t \models B$ and $\sigma \models E$.

We will write this in general as $\sigma,\langle s,t\rangle \models (A|B)_E$, as long as it is understood that the symbol "$\models$" in this context designates the satisfaction of an *action description* by a particular *world*, rather than the satisfaction of a first-order formula by a particular state. Obviously, if $A = (Own\ x_1\ y)$ and $B = (Own\ x_2\ y)$, then the expression $(A|B)_E$ describes a "transfer of the property y from x_1 to x_2." The definition of strict satisfaction in this situation is straightforward. We simply replace the symbol "$\models$" by the symbol "$\models_o$" in the previous definition. Thus, in our example, the action $(A|B)_E$ would be strictly satisfied by the deletion of an ownership tuple from the first state, and the addition of an ownership tuple to the second state, and this is precisely the result we want.

2.) Creative and Destructive Statechanges. These are special cases of the previous actions, where we take either $A = \sim B$ or $B = \sim A$. For example, if $A = (Love\ x\ y)$ and $B = (Hate\ x\ y)$, then $(A|\sim A)$ represents the destruction of love, $(\sim B|B)$ represents the creation of hate, and we make no claims at all about the compatibility or the incompatibility of these two actions.

3.) Identities. Here, we construct the expression $(A|A)$, and define it to be true on $\langle s,t\rangle$ if and only if $s \models A$ and $t \models A$. A useful special case is the expression (A), which is defined to be true on $\langle s,s\rangle$ if and only if $s \models A$. Note that this is the only action which can be reflexively true on a single state. A similar device is used extensively by Harel [20] to represent the conditional test in a programming language.

We now specify the meaning of the compound actions in L_A, beginning with the definition of disjunction, parallel composition and sequential composition. We need to provide a separate recursive definition for "$\models$" and "$\models_o$" in each case.

Definition 2: Let α and β be actions in L_A, and let σ be an assignment of the free variables in α and β to elements in the domain of interpretation of L_1. Then, for all $w \in W$, and for all $w \in W(w)$, the truth conditions for the actions $\alpha \vee \beta$, $\alpha \wedge \beta$, and $\alpha;\beta$ are given as follows:

Disjunction:

$\sigma,w \models \alpha \vee \beta \leftrightarrow \sigma,w \models \alpha$ or $\sigma,w \models \beta$

$\sigma,w \models_o \alpha \vee \beta \leftrightarrow \sigma,w \models_o \alpha$ or $\sigma,w \models_o \beta$

Parallel Composition:

$\sigma,w \models \alpha \wedge \beta \leftrightarrow \sigma,w \models \alpha$ and $\sigma,w \models \beta$

$\sigma,w \models_o \alpha \wedge \beta \leftrightarrow$

for some w_1 and w_2 in $W(w)$,

$w = w_1 + w_2$ and $\sigma,w_1 \models_o \alpha$ and $\sigma,w_2 \models_o \beta$

Sequential Composition:

$\sigma,w \models \alpha;\beta \leftrightarrow \sigma,w \models \alpha$ and $\sigma,T^n w \models \beta$

$\sigma,w \models_o \alpha;\beta \leftrightarrow$

for some $w_1 \in W(w)$ and $w_2 \in W(T^n w)$,

$w = w_1;w_2$ and $\sigma,w_1 \models_o \alpha$ and $\sigma,w_2 \models_o \beta$.

Although we are using the familiar logical symbols "$\vee$" and "$\wedge$" here, the meaning of these connectives in L_A is slightly different from their meaning in L_1. When we perform the action $\alpha \vee \beta$, we are either performing the action α, or we are performing the action β, but we are *not* performing the actions α and β together, because of our definition of strict satisfaction. This interpretation has consequences for our analysis of disjunctive permissions, as we will see in Section III.B below. The notion of a conjunction of actions in L_A is ambiguous, and we have distinguished here only two simple cases, a strictly "parallel" conjunction, and a strictly "sequential" conjunction. In a more realistic language of actions, we would attempt to include some more complex methods of composition, using overlapping time intervals and uncertain endpoints [27], but the present language is complex enough to reveal the principal features of the deontic concepts.

We now define three additional operations on the actions in L_A: predicate restriction, which restricts one of the sorts in the action $\alpha(x)$ to a predicate R, and existential and universal quantification, which are likewise restricted to the individuals satisfying a predicate R. We will omit the definition of "$\models$" when it is identical to the definition of "$\models_o$".

Definition 3: Let $\alpha(x)$ be an action in L_A which contains the free variable x, and let R be a unary predicate in L_1. Let σ be an assignment of the free variables in α to elements in the domain of interpretation of L_1, and let $\sigma(x/u)$ be the assignment which is identical to σ except that it assigns the variable x to the individual u. Then, for all $w \in W$, and for all $w \in W(w)$, the truth conditions for the actions $\alpha_{Rx}(x)$, $(\exists:Rx)\alpha(x)$ and $(\forall:Rx)\alpha(x)$ are given as follows:

Predicate Restriction:

$\sigma,w \models_o \alpha_{Rx}(x) \leftrightarrow \sigma,w \models_o \alpha(x)$ and $\sigma,s_0 \models (R\ x)$

Existential Quantification:

$\sigma,w \models_o (\exists:Rx)\alpha(x) \leftrightarrow$

for some u, $\sigma(x/u),s_0 \models (R\ x)$ and $\sigma(x/u),w \models_o \alpha(x)$

Universal Quantification:

$\sigma,w \models (\forall:Rx)\alpha(x) \leftrightarrow$

for every u such that $\sigma(x/u),s_0 \models (R\ x)$,

$\sigma(x/u),w \models \alpha(x)$

$\sigma,w \models_o (\forall:Rx)\alpha(x) \leftrightarrow$

for every u_1 such that $\sigma(x/u_1),s_0 \models (R\ x)$,

there exists a w_1 in $W(w)$ such that

$$\sigma(x/u_j), w_i \models_o \alpha(x)$$

and $w = \Sigma w_i$

We can illustrate these definitions with several simple examples. Suppose $\alpha(x,y)$ represents the "transfer of a property y to an actor x," and $(S\ y)$ represents the fact that "y is a stock." Then the predicate restriction $\alpha_{Sy}(x,y)$ represents the "transfer of a *stock* y to an actor x." Suppose furthermore that $(C\ x)$ represents the fact that "x is a corporation." Then $(\exists{:}Cx)\alpha_{Sy}(x,y)$ represents the "transfer of a stock y to *some* corporation x," a disjunctive action, and $(\forall{:}Cx)\alpha_{Sy}(x,y)$ represents the "transfer of a stock y to *all* corporations x," a parallel composition of actions. Notice that it is necessary for our definitions to refer to the complete state s_0 in w, even though we are only specifying truth conditions for a subworld $w \in W(w)$.

C. Permissions and Obligations

We are now in a position to define the semantics of permission and obligation, relative to a state r. First, for any two worlds v and w, if r is both the last state in v and the first state in w, we will say that the pair <v,w> is "joined together" at r. Let us consider the set of pairs {<v,w>} where w is a subworld of w, and where <v,w> is joined together at r. We will assume that there exists, at each r, a subset P_r of the set {<v,w>} which informs us of our "permitted courses of action" in the following way: If v is the world up to the present state, then the subworld w is a permitted course of action *if and only if* $\langle v,w\rangle \in P_r$. It is helpful to think of the set P_r as an "oracle," and to imagine that we can consult with this oracle whenever we are contemplating a course of action w. We will never have full access to the set P_r or its complement, of course, but we will know some of its members, and some of its nonmembers, by virtue of the rules of permission and obligation.

We begin by defining the notion *permitted*. We write $\langle\phi|\alpha\rangle_P$ to represent the English sentence: "if ϕ is satisfied up to the present, then the action α is permitted."

> **Definition 4:** $\langle\phi|\alpha\rangle_P$ is *true* at r under the assignment σ if and only if, for all <v,w> joined together at r, and for all $w \in W(w)$
>
> $$[\sigma,v \models \phi \text{ and } \sigma,w \models_\sigma \alpha] \Rightarrow \langle v,w\rangle \in P_r$$

It is important to note that this definition incorporates the concept of strict satisfaction, instead of ordinary satisfaction, and that any subworld which strictly satisfies the action α belongs to the set P_r. Therefore, two separate rules of permission will cumulate by set theoretic union, and a course of action will be permitted if it is permitted by any *one* of these rules.

We now define, in a similar manner, the notion *forbidden*. We write $\langle\phi|\alpha\rangle_F$ to represent the English sentence: "if ϕ is satisfied up to the present, then the action α is forbidden."

> **Definition 5:** $\langle\phi|\alpha\rangle_F$ is *true* at r under the assignment σ if and only if, for all <v,w> joined together at r, and for all $w \in W(w)$:
>
> $$[\sigma,v \models \phi \text{ and } \sigma,w \not\models \alpha] \Rightarrow \langle v,w\rangle \notin P_r$$

Note that this definition incorporates the ordinary concept of satisfaction, instead of the concept of strict satisfaction. In effect, we are saying here that if α is forbidden, then it is not only the subworlds strictly satisfying α which are forbidden, but also all supersets of these subworlds.

To arrive at the definition of *obligation*, let us consider the preceding definition with the symbol "$\sim\alpha$" formally substituted for the action α. The final line of this definition could then be rewritten as:

$$\langle v,w\rangle \in P_r \Rightarrow [\sigma,v \models \phi \Rightarrow \sim\sigma,w \models \sim\alpha]$$

We will not actually construct "negative" actions in L_A, for both technical and philosophical reasons, but this formal manipulation suggests an appropriate step to take. We will replace the expression "$\sim\sigma,w \models \sim\alpha$" with the expression "$\sigma,w \models \alpha$." Accordingly, let us write $\langle\phi|\alpha\rangle_O$ to represent the English sentence: "if ϕ *is satisfied* up to the present, then the action α is obligatory," and let us recast our previous truth condition to read as follows:

> **Definition 6:** $\langle\phi|\alpha\rangle_O$ is *true* at r under the assignment σ if and only if, for all <v,w> joined together at r, and for all $w \in W(w)$:
>
> $$\langle v,w\rangle \in P_r \Rightarrow [\sigma,v \models \phi \Rightarrow \sigma,w \models \alpha]$$

This final definition has a structural similarity to the standard definition of modal logic. The action α is *obligatory* if and only if it is true in all *permitted* worlds.

D. Dynamic Permissions and Obligations

Notice that every expression in our deontic language L_D is either true or false, under the assignment σ, in each state r. If one of these expressions has free variables in it then it expresses a *relationship* among individuals in the domain of interpretation of L_1, and we can define a new predicate in L_1 which holds just in case the expression in L_D holds. Thus, we can *assert* the existence of permissions and obligations in the same way we assert any other first-order statement. Using the machinery for defining statechanges, we can then construct actions which modify the deontic assertions, we can make these actions permitted, or forbidden, or obligatory, and so on. In other words, the process of linguistic construction, from L_1 to L_A to L_D, is fully recursive.

Now, suppose the following formulae are true, in each state r, for a particular ϕ and α:

$$\sim [\ \langle\phi|\alpha\rangle_P \wedge \langle\phi|\alpha\rangle_F\ ]$$

$$\sim [\ \langle\phi|\alpha\rangle_O \wedge \langle\phi|\alpha\rangle_F\ ]$$

Then we have satisfied the conditions for constructing an affirmative statechange, and we can define a general action which makes α permitted, forbidden, or obligatory, in sequence, as we wish. But, as we shall see in Section III.D below, these formulae are true for *all* ϕ and α. This makes the management of deontic modifications particularly simple and elegant.

III DEONTIC AXIOMATICS

In this section we consider various formulae involving the deontic expressions, and we prove that these formulae are true in our system, for all r, and for all assignments σ. We thus present part of a *sound* axiomatization of our proposed deontic logic, and we indicate this fact by writing the formulae as deontic theorems, denoted "$\vdash_D$." We will not be able to present a *complete* axiomatization, however, since the rules of inference in our system are considerably more complex than the rules of inference in a standard

modal logic. This point will be discussed in Section III.F. There is a second purpose to our discussion of deontic axiomatics, however, and that is to demonstrate that the familiar paradoxes of deontic logic do not arise in our system. To this end, we will examine the intuitive interpretations of our deontic formulae in some detail.

A. Compound Obligations

Let us first consider a disjunctive obligation. We establish the following result:

$$\vdash_D \langle\phi|\alpha_1\rangle_O \vee \langle\phi|\alpha_2\rangle_O \supset \langle\phi|\alpha_1 \vee \alpha_2\rangle_O \quad (1)$$

Proof: Assume either $\langle\phi|\alpha_1\rangle_O$ or $\langle\phi|\alpha_2\rangle_O$ is true at r. Using Definition 6, we pick $\langle v,w\rangle$ in P_r and assume that $v \models \phi$. On this assumption, if the first disjunct on the left hand side of (1) is true, then $w \models \alpha_1$; and if the second disjunct on the left hand side of (1) is true, then $w \models \alpha_2$. Thus $w \models \alpha_1 \vee \alpha_2$. Since this result holds for arbitrary $\langle v,w\rangle$, it follows that $\langle\phi|\alpha_1 \vee \alpha_2\rangle_O$ is true at r.

Note that the converse implication is false. To see this, let $\langle v,w_1\rangle \in P_r$ be a permitted subworld for which $w_1 \models \alpha_1$ and $\sim w_1 \models \alpha_2$, and let $\langle v,w_2\rangle \in P_r$ be a permitted subworld for which $w_2 \models \alpha_2$ and $\sim w_2 \models \alpha_1$. For example, if α_1 is the action "open the door" and α_2 is the action "turn on the light," then w_1 would be the subworld in which the door is opened but the light remains off, and w_2 would be the subworld in which the light is turned on but the door remains closed. Since both subworlds are permitted, neither $\langle\phi|\alpha_1\rangle_O$ nor $\langle\phi|\alpha_2\rangle_O$ would be true. Suppose, however, that these are the *only* permitted subworlds, at least when the initial condition ϕ is satisfied. Since both w_1 and w_2 satisfy $\alpha_1 \vee \alpha_2$, it is easy to see that the disjunctive obligation $\langle\phi|\alpha_1 \vee \alpha_2\rangle_O$ is true at r. In other words, one can be obligated to "open the door or turn on the light," without being obligated to "open the door," and without being obligated to "turn on the light."

Now consider a conjunctive obligation. In this case, the implication goes both ways:

$$\vdash_D \langle\phi|\alpha_1\rangle_O \wedge \langle\phi|\alpha_2\rangle_O \equiv \langle\phi|\alpha_1 \wedge \alpha_2\rangle_O \quad (2)$$

Proof: The proof is analogous to the proof of (1). To establish the implication from left to right, we pick $\langle v,w\rangle \in P_r$, assume that $v \models \phi$, and show that $w \models \alpha_1 \wedge \alpha_2$. To establish the implication from right to left, we show separately that $\langle v,w\rangle \in P_r$ implies $w \models \alpha_1$ and that $\langle v,w\rangle \in P_r$ implies $w \models \alpha_2$.

B. Compound Permissions

The formula for a disjunctive permission takes the following form:

$$\vdash_D \langle\phi|\alpha_1\rangle_P \wedge \langle\phi|\alpha_2\rangle_P \equiv \langle\phi|\alpha_1 \vee \alpha_2\rangle_P \quad (3)$$

Proof: To establish the implication from left to right, we assume that both $\langle\phi|\alpha_1\rangle_P$ and $\langle\phi|\alpha_2\rangle_P$ are true at r. Using Definition 4, we pick a $\langle v,w\rangle$ such that $v \models \phi$ and $w \models_o \alpha_1 \vee \alpha_2$. By the definition of strict satisfaction, this means that $w \models_o \alpha_1$ or $w \models_o \alpha_2$. But in either case, $\langle v,w\rangle \in P_r$ and thus $\langle\phi|\alpha_1 \vee \alpha_2\rangle_P$ is true. To establish the implication from right to left, we construct a similar argument for both $\langle\phi|\alpha_1\rangle_P$ and $\langle\phi|\alpha_2\rangle_P$. For example, if $v \models \phi$ and $w \models_o \alpha_1$, then $w \models_o \alpha_1 \vee \alpha_2$, and the right hand side of (3) then tells us that $\langle v,w\rangle \in P_r$.

The implication in (3) from a single disjunctive permission to a conjunction of two permissions violates the standard results of modal logic, but it corresponds closely to our ordinary usage of deontic terminology. To use our earlier example, if we are permitted to "open the door or turn on the light," then surely we are permitted to "open the door." In our ordinary deontic language, an agent is free to *choose* among the disjuncts, a fact that is captured in our semantics and revealed in (3).

Our analysis becomes somewhat more complex, however, if we consider the following plausible theorem:

$$* \vdash_D \langle\phi|\alpha_1\rangle_P \wedge \langle\phi|\alpha_2\rangle_P \supset \langle\phi|\alpha_1 \wedge \alpha_2\rangle_P \quad (4)$$

Should this result be provable in our system? It is easy to see that (4) cannot be derived from the semantic assumptions we have made so far. Consider our earlier example. Let α_1 be the action "open the door," let α_2 be the action "turn on the light," let w_1 be the subworld in which the door is opened, but nothing else occurs, and let w_2 be the subworld in which the light is turned on, but nothing else occurs. Suppose furthermore that $\langle v,w_1\rangle$ and $\langle v,w_2\rangle$ are the *only* members of P_r. If ϕ is satisfied whenever the door is initially closed and the light is initially off, then $w_1 \models_o \alpha_1$ and $w_2 \models_o \alpha_2$, and we can see from an inspection of Definition 4 that both $\langle\phi|\alpha_1\rangle_P$ and $\langle\phi|\alpha_2\rangle_P$ are true. But a subworld in which we *both* opened the door *and* turned on the light would *not* be permitted under these assumptions, and hence $\langle\phi|\alpha_1 \wedge \alpha_2\rangle_P$ cannot be true. We have thus refuted the implication in (4) from left to right, and a similar example can be constructed to refute any attempted implication from right to left. Notice, though, that this refutation depends critically on our use of the concept of strict satisfaction in Definition 4. In fact, if we replaced "$\models_o$" by "$\models$" in our definition of $\langle\phi|\alpha\rangle_P$ we could establish an even stronger result:

$$* \vdash_D \langle\phi|\alpha_1\rangle_P \supset \langle\phi|\alpha_1 \wedge \alpha_2\rangle_P \quad (5)$$

by an argument closely analogous to our proof of (3). But since (5) would hold for *any* action α_2, it would be totally unacceptable. Alternatively, and less drastically, we could constrain our definition of P_r by the following *closure* axiom:

$$\langle v,w_1\rangle \in P_r \text{ and } \langle v,w_2\rangle \in P_r \Rightarrow \langle v,w_1+w_2\rangle \in P_r$$

With this assumption, the proof of (4) would go through, but the proof of (5) would be blocked.

Which result do we want? Suppose we have been told separately that we are "permitted to open the door," and that we are "permitted to turn on the light." Would we then conclude that we are permitted *simultaneously* to open the door and to turn on the light? Maybe yes, maybe no. We would not want (4) to hold in all cases, because that would totally conflate a very useful distinction between disjunction and conjunction. But if we were given a long list of permitted actions, intended to be composable in parallel, in would be inefficient to list *all* possible combinations of these actions, and to grant permissions separately for each one. As a pragmatic alternative, we might construct a special rule of inference that would operate on "independent" permissions, applying the formula in (4) only to a designated set of expressions $\langle\phi|\alpha_i\rangle_P$. In particular, if the expressions $\langle\phi|\alpha_1\rangle_P$ and $\langle\phi|\alpha_2\rangle_P$ had originally been derived from the expression $\langle\phi|\alpha_1 \vee \alpha_2\rangle_P$ by the use of (3), then the special rule of inference would *not*

be applicable, thus preserving the distinction between a disjunctive and a conjunctive rule of permission.

C. Compound Deontic Conditions

For the sake of completeness, we include here several results on the conditional component of the deontic expressions:

$$\vdash_D \langle\phi_1|\alpha\rangle_P \wedge \langle\phi_2|\alpha\rangle_P \equiv \langle\phi_1 \vee \phi_2|\alpha\rangle_P \quad (6)$$

$$\vdash_D \langle\phi_1|\alpha\rangle_O \wedge \langle\phi_2|\alpha\rangle_O \equiv \langle\phi_1 \vee \phi_2|\alpha\rangle_O \quad (7)$$

$$\vdash_D \langle\phi_1|\alpha\rangle_P \supset \langle\phi_1 \wedge \phi_2|\alpha\rangle_P \quad (8)$$

$$\vdash_D \langle\phi_1|\alpha\rangle_O \supset \langle\phi_1 \wedge \phi_2|\alpha\rangle_O \quad (9)$$

Proof: Omitted, but analogous to prior proofs.

D. The Relation Between Permission and Obligation

Let us now consider two questions which were initially raised in Section I as part of our discussion of standard modal logic. First, does the formula $Pp \equiv \sim O \sim p$ hold for permissions and obligations? In particular, if it is false that you are forbidden to do p, does it follow that p is permitted? Second, does the formula $Op \supset Pp$ hold? In other words, is every obligatory action permitted?

The answer to the first question is negative. The following implication holds in only one direction in our system:

$$\vdash_D \langle\phi|\alpha\rangle_P \supset \sim \langle\phi|\alpha\rangle_F \quad (10)$$

Proof: Immediate, from Definitions 4 and 5. For a counter example to the reverse implication, we pick any α which is strictly satisfied by two subworlds w_1 and w_2, and we put $\langle v,w_1\rangle \in P_r$ and $\langle v,w_2\rangle \notin P_r$. Then the right side of (10) is true, while the left side is false.

The answer to the second question is also negative, if it is translated literally into our notation. Since obligations cumulate by set theoretic intersection, each obligation narrowing the scope of the remaining permitted acts, there is no reason to expect that *all* the courses of action within a single rule of obligation will be permitted by other rules of obligation. A weaker version of this formula is more plausible: namely, that if α is obligatory there exists *some* permitted course of action which satisfies α. It turns out that this weaker formula is true in our system only if P_r satisfies an additional assumption: $(\forall v)(\exists w)[\langle v,w\rangle \in P_r]$, which asserts that P_r has no "dead ends." Under this assumption, we can prove the following:

$$\vdash_D \langle\phi|\alpha\rangle_O \supset \sim \langle\phi|\alpha\rangle_F \quad (11)$$

Proof: We will assume $\langle\phi|\alpha\rangle_F$ and derive a contradiction. By Definition 5, if we pick a v such that $v \vDash \phi$, then for all w $w \vDash \alpha \Rightarrow \langle v,w\rangle \notin P_r$. Because of the assumption that P_r has no "dead ends," there exists a w' such that $\langle v,w'\rangle \in P_r$. Clearly this w' cannot satisfy α. Since $\langle\phi|\alpha\rangle_O$ is also true, however, Definition 6 tells us that $w' \vDash \alpha$, a contradiction.

Note that (10) and (11) are equivalent to the formulae cited in Section II.D above, thus allowing us to construct a general action for the modification of an arbitrary deontic expression.

E. Quantification Over Actions

Let $\alpha(x)$ be an action in L_A containing a free variable x, and let ϕ be a condition in L_A which does not contain x as a free variable. Think of the existential quantifier as a disjunctive operator, and think of the universal quantifier as a conjunctive operator. Then, by analogy to (1), (2), (3) and (4), we can write down the following variants of the Barcan formulae of standard modal logic:

$$\vdash_D (\exists:Rx)\langle\phi|\alpha(x)\rangle_O \supset \langle\phi|(\exists:Rx)\alpha(x)\rangle_O \quad (12)$$

$$\vdash_D (\forall:Rx)\langle\phi|\alpha(x)\rangle_O \equiv \langle\phi|(\forall:Rx)\alpha(x)\rangle_O \quad (13)$$

$$\vdash_D (\forall:Rx)\langle\phi|\alpha(x)\rangle_P \equiv \langle\phi|(\exists:Rx)\alpha(x)\rangle_P \quad (14)$$

$$* \vdash_D (\forall:Rx)\langle\phi|\alpha(x)\rangle_P \supset \langle\phi|(\forall:Rx)\alpha(x)\rangle_P \quad (15)$$

Proof: These results can be established by modifying the proofs of (1), (2), (3) and (4), respectively, so that they apply to the quantified expressions of Definition 3. The most interesting case is (14). Reading from right to left: "If we are permitted to transfer stock to *some* corporation x, then, for *every* corporation x, we are permitted to transfer stock to that x."

F. Rules of Inference

Most axiomatizations of modal logic include an axiom:

$$O(p \supset q) \supset (Op \supset Oq)$$

and a rule of inference:

$$\text{if } \vdash p, \text{ then } \vdash Op$$

which enable us to transfer provable formulae from nonmodal contexts into modal contexts, and to carry out deductions inside the modal operators. However, as Moore has pointed out [18], these axiomatizations are notoriously inefficient for automatic theorem proving. In our present formalization of deontic logic, of course, the modal axiom makes no sense at all, and the rule of inference applies only to trivial cases. What we need instead are a number of specialized rules of inference which would enable us to "lift" deductions from the language L_1 up to the language L_A, and then up to the language L_D, and to do so in an efficient manner.

For example, consider the action $\alpha_S(y)$ in which $\alpha(y)$ represents the "transfer of a property y" and the predicate restriction $(S\ y)$ represents either the fact that "y is a stock" or that "y is a security." Suppose $(\forall y)[(Stock\ y) \supset (Security\ y)]$ in the language L_1. Then any subworld w which strictly satisfies $\alpha_{Stock}(y)$ also strictly satisfies $\alpha_{Security}(y)$, and we can show:

$$\vdash_D \langle\phi|\alpha_{Stock}(y)\rangle_O \supset \langle\phi|\alpha_{Security}(y)\rangle_O \quad (16)$$

$$\vdash_D \langle\phi|\alpha_{Security}(y)\rangle_P \supset \langle\phi|\alpha_{Stock}(y)\rangle_P \quad (17)$$

In ordinary language: "If we are obligated to transfer a stock y, then we are obligated to transfer a security y." Conversely: "If we are permitted to transfer a security y, then we are permitted to transfer a stock y." Given our previous discussion of the intuitive meaning of permissions and obligations, these results seem correct. What is needed, however, is a more general set of rules to carry out these inferences. We have some partial results along these lines, but they are not yet complete.

IV. CONCLUSION

We have presented in this paper a formal semantics for the concepts of permission and obligation which seems to avoid the paradoxes of the standard deontic logics. The major deficiencies at the moment are two: we have not yet worked out the complete rules of inference for our

system; and we have confined our attention so far to a relatively simple language of actions Unfortunately, these two points are intimately connected The rules of inference proposed for our system are complex because they take into account the structure of the language of actions L and as we add further complexity to L_A we will certainly add complexity to the rules of inference, too Nevertheless, we believe that the semantics of the deontic language itself, the language L_D is basically correct, and robust, and that it will remain in its present form as the language L_A evolves Perhaps this is even a fact of our cognitive lives that the concepts of permission and obligation are relatively simple, and the complexity arises instead from our concept of action

Acknowledgments: The author wishes to thank N.S Sndharan. Ray Reiter. Alex Borgida and David Raab for their many helpful discussions The design of a computational version of the present deontic representation is the subject matter of a dissertation proposal by David Raab

REFERENCES

[I] McCarty. L.T.. and Sndharan, N.S. "The Representation of an Evolving System of Legal Concepts I Logical Templates' In Proceedings 3rd CSCSI-SCE10 Conference. Victoria. British Columbia. 1980, 304-311

[2] McCarty. LT and Sndharan. N.S "The Representation of an Evolving System of Legal Concepts II Prototypes and Deformations" In Proceedings IJCAI-81. University of British Columbia. August. 1981. 246-53

[3] DeBessonet. CG An Automated Approach to Scientific Codification' Rutgers Computer and Technology Law Journal. 9:1 (1982) 27-75

[4] Sacerdoti, ED A Structure for Plans and Behavior. Elsevier North-Holland, 1977

[5] DeMillo, R.A.. Dobkm. DP. Jones. A.K and Lipton, R.J Foundations of Secure Computation. Academic Press, 1978

[6] von Wright. GH Norm and Action. London Routledge and Kegan Paul. 1963

[7] Rescher, N The Logic of Commands. London Routledge and Kegan Paul, 1966

[8] Alchourron, C.E., and Bulygin, E Normative Systems. Springer Verlag, 1971

[9] Castaneda. H-N Thinking and Doing: The Philosophical Foundations of Institutions. Boston D. Reidel. 1975

[10] Hohfeld, W.N. "Fundamental Legal Conceptions as Applied in Judicial Reasoning: I" Vale Law Journal. 23 (1913) 16

[II] Hohfeld. W.N. "Fundamental Legal Conceptions as Applied in Judicial Reasoning: II" Yale Law Journal. 26 (1917) 710.

[12] Allen, L.E. "Formalizing Hohfeldian Analysis to Clarify the Multiple Senses of Legal Right A Powerful Lens for the Electronic Age." Southern California Law Review. 48 (1974) 428-.

[13] Kripke. S.A. "Semantic Analysis of Modal Logic I." Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik. 9 (1963) 67-96

[14] von Wright. G.H., "On the Logic of Norms and Actions." in New Studies in Deontic Logic, Hilpinen, R. ed.. D Reidel, 1981. 3-35

[15] Castaneda. H.-N.. "The Paradoxes of Deontic Logic The Simplest Solution to All of Them in One Fell Swoop.' in New Studies in Deontic Logic, Hilpinen, R, ed, D Reidel. 1981. 37-85

[16] von Wright, GH An Essay in Deontic Logic and the General Theory of Action. North-Holland, 1968

[17] McCarthy, J. "Programs with Common Sense,' in Semantic Information Processing, Minsky, M, ed, MIT Press. 1968 403-418

[18] Moore. R.C "Reasoning About Knowledge and Action, Technical report 191. SRI International, October 1980

[19] McDermott. D "A Temporal Logic for Reasoning About Processes and Plans' Cognitive Science. 6 (1982) 101-155

[20] Harel. D First Order Dynamic Logic. Spnnger-Verlag Lecture Notes in Computer Science, Vol 68, 1979

[21] Rosenschem. S.J "Plan Synthesis A Logical Perspective' In Proceedings IJCAI-81. University of British Columbia. August. 1981, 331-337

[22] Barwise. J and Perry. J 'Situations and Attitudes" Journal of Philosophy. 78 11 (1981) 668-691

[23] Kripke. S.A, "Semantical Analysis of Intuitionistic Logic I.,' in Formal Systems and Recursive Functions, Crossley, J.N and Dummett, M.A.E, eds. North-Holland, 1965, 92-130

[24] McCarthy, J "Circumscription A Form of Non-Monontonic Reasoning" Artificial Intelligence. 13 (1980) 27-39

[25] McDermott, D and Doyle, J "Non-Monotonic Logic I" Artificial Intelligence. 13 (1980) 41-72

[26] Reiter, R "A Logic for Default Reasoning" Artificial Intelligence. 13 (1980) 81-132

[27] Allen. J.F "An Interval-Based Representation of Temporal Knowledge" In Proceedings IJCAI-81. University of British Columbia. August. 1981. 221-226