

Proof Theory For Contrary To Duty Obligations

Dov Gabbay

Amsterdam, Jan. 2010

Consider standard Deontic Logic *SDL*, with two *K* modalities \Box (for necessity) and \bigcirc (for obligation) and the axioms

$$\begin{aligned} \Box A \rightarrow \bigcirc A & \text{ What's permitted is possible} \\ \neg \Box \perp & \end{aligned}$$

It is known that some intuitively consistent contrary to duty scenarios cannot adequately be formalised in this system.

We offer Reactive Kripke semantics for this system and consider, relative to the reactive semantics, the notion of reactive structured CTD theories. This syntax and semantics contains the traditional ones as special cases. In this expanded logic CTD scenarios can adequately be expressed as reactive CTD theories.

We offer a three-component proof procedure for reactive CTD theories.

1. A tableaux like procedure for manipulating the theory (this is parallel to traditional rules like Modus Ponens).
2. An algorithm for inheriting CTDs from a theory onto a subtheory, (this is parallel to rules like $\bigcirc_A B \wedge \bigcirc C \rightarrow \bigcirc_A(B \wedge C)$).
3. A non-monotonic syntactic mechanism for settling conflicts between opposing inherited CTDs.

We now describe our system. A reactive structural theory for the language with \Box and \bigcirc has the form of a finite directed graph

$$\Delta = (S, \mathbb{R}, \infty, a, \Pi, \mathbb{E})$$

where $S \neq \emptyset$ is a set of nodes and $\infty, a \in S$. ∞ is the trigger point, a is the initial point (the actual world) and Π is a sequence of points $\Pi = (\infty, a, x_1, \dots, x_n, d)$ where $x_i, d \in S$ and d is the terminal evaluation point. The following holds:

1. $\mathbb{R} \subseteq (S \times S) \cup (S^2 \times S^2)$
 The elements $(x, y) \in S \times S$ are called arrows (denoted by $x \longrightarrow y$) and elements $((x, y), (u, v)) \in S^2 \times S^2$ are called double arrows denoted by $(x \longrightarrow y) \twoheadrightarrow (u \longrightarrow v)$.
2. $(\infty, a) \in \mathbb{R}$ and it is the only pair of the form (∞, x) in \mathbb{R} .
3. $(a, x_1), (x_1, x_2), \dots, (x_n, d)$ of Π are all in \mathbb{R} .
4. \mathbb{E} is a set of pairs of the form $t : A$ for $t \in S$ and A a wff in the language of \square and \bigcirc .

We imagine a hierarchical flow of obligations as moving along the path Π . \mathbb{E} tells us of the conditions under which an obligation is triggered (CTD or not). The non-monotonic mechanism gives priority among inherited CTDs, taking into account the history of each along Π . This language is richer than dyadic attempts which represents CTDs as $A \rightarrow \bigcirc_A B$ because when we have a conflict (e.g. $\bigcirc \neg B$ and $\bigcirc_A B$), we can resolve it by looking at the history of violations and not just comparing A and B in the modal system.

The example of Figure 1 illustrates our method. Each English item of data is a theory $\Delta_n = (S_n, \mathbb{R}_n, \infty, a, \Pi, \mathbb{E}_n)$. In Figure 1 we list the English items of data together with \mathbb{R}_n and \mathbb{E}_n . S_n is all the elements mentioned in \mathbb{R}_n .

We perform logical operation between $\Delta_1, \Delta_2, \Delta_4$ and Δ_5 and then apply Δ_3 to the result and we get the following theory Δ which we now draw. (Intuitively we got a dog and built a fence and now we have many conflicting obligations).

A conflict arises at node e_2 and the non-monotonic part of the logic is applied to it.

n	English Statement	Property of \mathbb{R}_n	Property of \mathbb{E}_n
1	There should be no fence	$(\infty, a) \in \mathbb{R}_1$ $(t_0, t_2) \in \mathbb{R}_1$ $((\infty, a), (t_0, t_2)) \in \mathbb{R}_1$	$t_2 : \text{fence} \in \mathbb{E}_1$
2	There should be no dog	$((\infty, a) \in \mathbb{R}_2$ $(r_0, r_2) \in \mathbb{R}_2$ $((\infty, a), (r_0, r_2)) \in \mathbb{R}_2$	$r_2 : \text{dog} \in \mathbb{E}_2$
3	If there is a fence it should be taken down	$(u, v_0) \in \mathbb{R}_3$ $(v_0, v_2) \in \mathbb{R}_3$ $((u, v_0), (v_0, v_2)) \in \mathbb{R}_3$ $(\infty, a) \in \mathbb{R}_3$	$v_0 : \text{fence} \in \mathbb{E}_3$ $v_2 : \text{fence} \in \mathbb{E}_3$
4	If there is a dog there should be a fence	$(s, s_0) \in \mathbb{R}_4$ $(s_0, s_1) \in \mathbb{R}_4$ $(\infty, a) \in \mathbb{R}_4$	$s_0 : \text{dog} \in \mathbb{E}_4$ $s_1 : \neg \text{fence} \in \mathbb{E}_4$
5	There is a dog	$(\infty, a) \in \mathbb{R}_5$ $(a, x_1), \dots, (x_k, d) \in \mathbb{R}_5$	$d : \text{dog} \in \mathbb{E}_5$ $\Pi_5 = (\infty, a, x_1, \dots, x_k, d)$

Figure 1: Dogs and fences

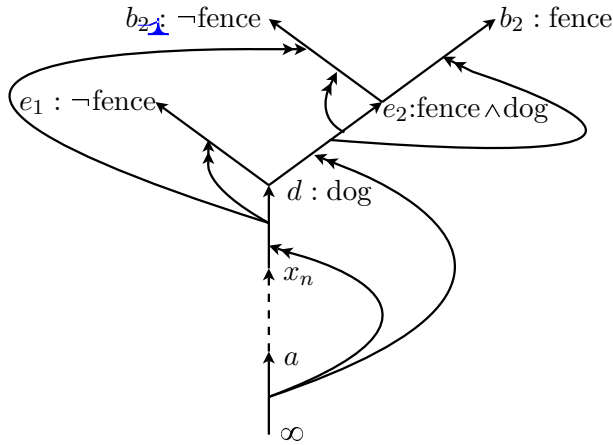


Figure 2: $\Pi = (\infty, a, \dots, x_n, d, e_2)$