The Reactive Theories and Norm Change

Dov M. Gabbay\textsuperscript{123}

\textsuperscript{1}King's College, UK
\textsuperscript{2}University of Luxembourg, Luxembourg
\textsuperscript{3}Bar-Ilan University, Israel

19-01-2010
The Reactive Approach
New Ideas

- The reactive idea (2004)
- Structured theories and semantics (1990)
- Reactive structured theories (2007)
- Contrary to duties as reactive structure theories
Modal Logic and Possible Worlds

\[ a \models \Diamond A \text{ iff } \exists y (aRy \land y \models A) \]
## Modal Logic and Possible Worlds

### Static theoretic evaluation

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition on R</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Box A \rightarrow A$</td>
<td>R reflexive</td>
</tr>
<tr>
<td>$\Box A \rightarrow \Box \Box A$</td>
<td>R transitive</td>
</tr>
</tbody>
</table>
**New point of view**

\[ a \models \Diamond A \quad \text{iff} \quad \text{“run up the graph” to find a point } y \text{ where } y \models A \]

- Actual walking the graph
- No longer philosophical “possible world”, just graph and paths on graph

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition where to go</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Box A \rightarrow A)</td>
<td>check locally as well</td>
</tr>
<tr>
<td>(\Box A \rightarrow \Box \Box A)</td>
<td>Run as far as you can. (R) need not be transitive or reflexive</td>
</tr>
</tbody>
</table>
If you pass through $a \rightarrow c$, then disconnect $d \rightarrow e$

$$a \models \Diamond (d \land \Box \bot)$$

- ordinary model $= (S, R, a, h)$
- reactive model $= (S, R, a, h)$
Satisfaction in reactive model

\[ m \models A \quad \text{iff} \quad A \text{ models in the actual world of } m \]
Reactive Approach

Reactive semantics is stronger

The logic defined by this frame cannot be characterized by any class of ordinary frames
The Reactive Approach

- Abandon possible world if we want
- Evaluation in information bearing graphs
- Retain possible world view
- Reactive evaluation in reactive models
Ordinary Theory

\[ A \land \Diamond B \]

**Model 1**

**Model 2**

**Figure:** \( a \models A \land B \)

**Figure:** \( a \models A \land b \models B \)

*We can have \( a = b \)*
Structured Theory

Figure: $a \neq b$

Figure: $a \neq b; a \models A; b \models B$
\[ t_1 R t_2 \neq t_3 R t_1 \rightarrow a_i \text{ related as requested} \]
If you get to \( e \) you no longer see a world with no fence.

\( T \) requires a reactive model such that when we pass an arc leading to a point \( e \) where no fence is not possible, then double arrows block access to all non-white fence worlds.
T holds in this model if we can assign points as shown and have a path from $x$ to $a$ satisfying $T$. 
Possible Models for the fence theory (From Carmo-Jones)

1. There is a white fence, it is fixed that there will be a fence white or not

   \[ a \models fence \land white \land \square fence \]

2. There is a fence and it is possible not to have a fence

   \[ a : fence \rightarrow a' : \neg fence \]

3. There is no fence and it is possible to have a fence, and it is possible not to have a fence
Reactive Theory for Chishlom

- **Ought to help**: from a block all arcs to not help
- **If help then tell**: From any first point where $\neg \Box \neg \text{help}$ block all arcs to $\neg \text{tell}$
- **If not help then ought not tell**: From any point $\neg \Box \text{hel}$ block any arcs to $\neg \text{tell}$
- **Does not go**: Actual world $a$ points where $\Box \neg \text{help}$
Case analysis of Carano Joes-Chishlom

They describe the model in other terms

f1 The mand intends not to go and help

f2 It is potentially possible for the man to help and to tell and potentially possible for the man to help and not tell.

f3 The man has not in fact told that he is coming to halp although it is still actually possible that he does tell and actually possible he does not tell.
Suppose you have a language $L_1$ and a stronger language $L_2$ you want to express the contents of an $L_2$-theory $\Delta^{L_2}$ using the language $L_1$. It may not be possible to find $\Omega^{L_1} \equiv \Delta^{L_2}$. But under special models of $L_1$, defined by semantic conditions $S_i$ we have

$$\Delta^{L_2} \equiv \bigvee_{S_i} \Omega^{L_1}_i$$
sought to model CTDs in terms of defeasible logic. The CTD proof theory does have a nonmonotonic component. The above describes in outline our proof theory.

To summarise:
We assume we have defined the syntactic notion of a CTD theory and a notion of consistency (or the notion of 'a problem' for such a theory). Given a conjunction of CTD theories (which all have to hold), our proof system does the following:

1. First we interact them with each other to get as much data as we can.
2. Second we find for any path in any theory what is inherited. If there is a problem we have a separate CTD nonmonotonic logic based on δ-annotated paths, which can resolve the conflict. This nonmonotonic part would probably be an inheritance system.

We postpone the details to part 2. Meanwhile we conclude with a detailed example of conflicting norms.

Remark 5.5 (Conflicting Norms) We can cope more easily with conflicting norms. The modern world is full of them. Think of

1. There should be no fence
2. There should be no dog
3. If there is a dog there should be a fence
4. If there is a fence it should be demolished
5. There is a dog

Let us analyse this example using our methods of Section 5 (proof theory).

We get the following theories, corresponding to the clauses and figures of the example

To use proof theory as described in Section 5, we impose Figure 74 on Figure 76 and get Figure 77.

We now impose Figure 75 on Figure 77 and get Figure 78.

We now impose Figure 74 on Figure 78 (since there is a dog at x : dog, there should be no fence, and get Figure 79.

The impossible situation arises at node e₂. e₂ inherits two CTDs, as seen in Figure 80.

The double arrow \((x \rightarrow e₂) \rightarrow (e₂ \rightarrow b₁)\) is labelled \((∞, a, y, x, e₂)\) and it conflicts with the CTD double arrow \((x \rightarrow e₂) \rightarrow (e₂ \rightarrow b₂)\). Which one wins depends on our nonmonotonic CTD proof system.

In real life we can loop, repeatedly erecting and demolishing a fence and thus fulfill our obligations, that is assuming we insist on a dog.\(^{27}\)

\(^{27}\)We get conflicting norms even without item 4. If we insist on a dog then by item 1 there should be no fence and by item 3 there should be a fence. However item 4 makes the story more dramatic and expensive. Our honest home and dog owner cannot be faulted if he keeps on erecting and demolishing his fence. He is trying to comply with the norms. A human agent would never do that but he may fall victim to two municipal departments each insisting on their own rules and not communicating to resolve the conflict. We have heard of horror stories about an ambulance with a critically ill patient being shuttled between two hospital clinics because of conflicting rules of jurisdiction!
Figure 72: Dog 1: there should be no fence

Figure 73: Dog 2: there should be no dog
Figure 72: Dog 1: there should be no fence

Figure 73: Dog 2: there should be no dog
6 Supplementary Remarks

Remark 6.1 (Triple arrows and multiple level CTD) Previous sections used mainly double arrows to model CTDs. This example shows we can also use triple arrows for multiple CTDs.

For example:

1. It is obligatory to have no fence.
2. If there is a fence it should be white.
3. If it is not white it should be stripped of its paint by an expert and painted white.
4. If it cannot be painted (some plastics cannot take paint, I have some in my office at King's) then it should be demolished.

Figure 81 illustrates a possible model.

The beginning position is that all arrows are off except the one leading to no fence. The arrows emanating from a block the path to no fence and clear a path to fence not painted. The arrows from the agent's arrows force him to demolish.

Remark 6.2 (Expressing Ideal Worlds using Double Arrows) The additional power of the double arrows and triple arrows can be used to eliminate the ideal world function $\lambda I(t)$. In a model with connections capable of being on or off, we can characterise
6 Supplementary Remarks

Remark 6.1 (Triple arrows and multiple level CTD) Previous sections used mainly double arrows to model CTDs. This example shows we can also use triple arrows for multiple CTDs.

For example:

1. It is obligatory to have no fence.
2. If there is a fence it should be white.
3. If it is not white it should be stripped of its paint by an expert and painted white.
4. If it cannot be painted (some plastics cannot take paint, I have some in my office at King’s) then it should be demolished.

Figure 81 illustrates a possible model.

The beginning position is that all arrows are off except the one leading to no fence. The arrows emanating from a block the path to no fence and clear a path to fence not painted. The arrows from the agent’s arrows force him to demolish.

Remark 6.2 (Expressing Ideal Worlds using Double Arrows) The additional power of the double arrows and triple arrows can be used to eliminate the ideal world function $\lambda(t)$. In a model with connections capable of being on or off, we can characterise
Figure 75: Dog 4: if there is a fence it should be demolished (i.e. go to ¬fence)

Figure 76: Dog 5: there is a dog in the evaluation world
Figure 75: Dog 4: if there is a fence it should be demolished (i.e. go to $\neg$fence)

Figure 76: Dog 5: there is a dog in the evaluation world
Figure 77: Dog 6

$I(t)$ as all those worlds accessible to $t$ by an active direct connection and the non-ideal worlds as those not accessible. The minute we make our first move we can activate and deactivate connections to bring us back to whatever connections we want. So in fact the ideal worlds are recognised by the way we do our on and off switches. Figure 34 can be replaced by Figure 82 below.

Here we use arrows, double arrows and triple arrows.

The starting position is that this access (on) only to the ideal world $f^-$. The minute we move from $a$, we cancel access to it and activate access in the $\{b, w^+, w^-\}$ direction. We know $f^-$ is ideal because disconnecting access to it makes the difference.

There may be better ways to do the coding. We just want to illustrate the principle involved.

**Remark 6.3 (Solving CTD Problems using Reactive Proof Theory)** Reactive proof theory can be used directly to solve problems of CTD. The idea of reactive proof theory is that using a rule can activate or deactivate other rules. So, for example, we may have

1. $O \neg f$. There must be no fence.
2. $f \rightarrow Ow^+$ If there is a fence then it must be a white fence.
3. $\vdash w^+ \rightarrow f$. White fence is a fence
Figure 78: Dog 7
Figure 79: Dog 8

Figure 80: dog9
Figure 79: Dog 8

Figure 80: dog9
The Reactive Approach

Norm

\( \Box A \Rightarrow \Diamond \neg B \)

\[ \begin{array}{c}
\infty \\
\downarrow \\
da : A \\
\uparrow \\
a \\
\downarrow \\
\infty
\end{array} \]

\[ \begin{array}{c}
B \\
\downarrow \\
\rightarrow B
\end{array} \]

\[ \begin{array}{c}
\neg B \\
\uparrow \\
\rightarrow B
\end{array} \]
Change of Norms

Model

$\Delta_1$  \(\iff\)  Norm

$\Delta_2$  \(\iff\)  Norm on Norms

$\Delta_3$
Change of Norms

Non-Monotonic Logic
to decide between conflicting Norms

\[ a, x_1, x_2, \ldots, x_n, d \]
\[ A_0, A_1, A_2, \ldots, A_n, B \]

\[ x_2, \ldots, x_n, d \]
\[ E_2, \ldots, E_n, C \]
Change of Norms

\[ \Delta_1 \quad \Delta_1' \quad \text{impose} \quad \Delta_2 \]

Models

Norm
Change of Norms

\[ Norm \Delta \]

\[ \Pi_1 : Agent_1 \quad \ldots \quad \Pi_n : Agent_n \]

Model

Patterns for \( n \) agents interactions
Dynamic Logic

\[ \text{Model} \models \square \varphi \psi \]

iff

Apply \varphi

to Model \models \psi

1. Run \varphi and change Model

2. Impose \varphi