Master WIP

Kjetil Midtgarden Golid

February 2, 2016

1 Introduction

1.1 Paradoxes

A theory in propositional logic is semantically *inconsistent* if it has no model, i.e., there exists no variable assignment making all formulae in the theory true. Consider the following example:

$$x \wedge \neg x \tag{1}$$

While a sentence like $\neg a \rightarrow b$ can be satisfied by, for instance, letting both a and b be true, no such assignment can be made for the statement above. The statement is therefore inconsistent.

A paradox is usually informally defined as something along the lines of *"a statement that can be neither true nor false"*. We can immediately note one thing from this intuitive definition: Since no paradoxes can be true, all paradoxes are, by definition, inconsistent. It does, however, seem like a stretch to say that all inconsistent theories are paradoxes. Just consider the inconsistent statement, $x \wedge \neg x$ again: this statement simply seem false, and thus not paradoxical.

A different view is that a paradox is a *dialetheia*, a sentence that is *both* true and false[1]. We will however not spend much time exploring these philosophical differences, being this is not a philosophical paper and it won't change much for our definitions.

The liar sentence is probably the most famous example of a paradox:

$$\text{"\textit{This sentence is false}".} \tag{2}$$

If the statement is true, then the statement is false, but if the statement is false, then the statement is true. It can thus neither be true nor false. Note how the liar sentence is a statement about other statements (in this case itself). In order to study these kinds of meta-statements, we need a way to reference other statements within a statement. In propositional logic, we can do this by giving statements "names" in the form of adding fresh variables with equivalences to their correponding statements[1]. Consider the left statements below, together with their corresponding named statements on the right.

$$a \qquad\qquad x_1 \leftrightarrow a \tag{3}$$

$$a \wedge \neg a \qquad\qquad x_2 \leftrightarrow a \wedge \neg a \tag{4}$$

$$a \vee \neg a \qquad\qquad x_3 \leftrightarrow a \vee \neg a \tag{5}$$

[1] bad wording?

By performing this naming-operation on these statements, one is obviously changing their truth value. Even though we have one consistent, one inconsistent and one tautological statement on the left, all the statements become consistent after they have been named. This is because we in all the cases above can find a truth value for x_i that matches the one of the corresponding statement. This will not be the case for paradoxes, so our new named statements will be consistent if and only if they are not paradoxical.

Consider the liar sentence. It can be written as a named statement in the following way:

$$x \leftrightarrow \neg x \tag{6}$$

This statement is obviously inconsistent, making it a paradox by our newly acquired definition. The study of *discourses* takes this formalization a step further.

1.2 Discourses

A propositional theory is in **graph normal form (GNF)** if all its formulae is on the following form:

$$x \leftrightarrow \bigwedge_{i \in I_x} \neg y_i \tag{7}$$

such that every variable occurs exactly once on the left of $\leftrightarrow$ across all the formulae in the theory.

There is a simple translation from conjunctive normal from to graph normal form (shown in the end of this section), showing that any propositional theory can be written in GNF. By interpreting the variable on the left as the name of the statement on the right, like shown earlier, one can start using GNF to model meta-statements.

We now formally define a **discourse** to be a theory in GNF, and a **paradox** to simply be an inconsistent discourse.

We will later show a very handy correspondence between these discourse theories and certain graphs. The correspondence lets us not only decide the satisfiability of a discourse theory (i.e. whether or not it is paradoxical) by looking at certain properties of the corresponding graph. The properties in the graph also provides us with the satisfying models, if they exist. In order to express this logic/graph correspondence, we first need to establish some graph terminology.

1.3 Graphs, Kernels and Solutions

A directed graph is a pair $\mathbf{G} = \langle G, N \rangle$ where G is a set of vertices while $N \subseteq G \times G$ is a binary relation representing the edges in $\mathbf{G}$. We use the notation $N(x)$ to denote the set of all vertices that are targeted by edges originating in x (successors of x). Similarly, $N^-(x)$ denotes the set of all vertices with edges targeting x (predecessors of x. We define these two predicates formally as follows:

$$N(x) := \{y \mid (x, y) \in N\} \tag{8}$$
$$N^-(x) := \{y \mid (y, x) \in N\} \tag{9}$$

We extend these relations to sets of vertices in the following way:

$$N(X) = \bigcup_{x \in X} N(x) \tag{10}$$

$$N^{-}(X) = \bigcup_{x \in X} N(x) \tag{11}$$

A kernel is a set of vertices $K \subseteq G$ such that:

$$G \setminus K = N^{-}(K) \tag{12}$$

The above equivalence can ble split up into two inclusions to be more easily understood:

$G \setminus K \subseteq N^{-}(K)$, saying that each vertex outside the kernel, has to have an edge into the kernel (K is dominating).

$N^{-}(K) \subseteq G \setminus K$, saying that each edge targeting a vertex within the kernel has to come from outside, thus no two vertices in the kernel are connected by an edge (K is independent).

We get the correspondence between satisfying models of a discourse theory and kernels in a graph through an alternative, equivalent kernel definition called a *solution*. This equivalence of kernels and solutions was shown by Roy Cook in [].

Given a directed graph $\mathbf{G} = \langle G, N \rangle$, an assignment $\alpha \in 2^{G}$ is a function mapping every vertex in the graph to either 0 or 1. A solution is an assignment α such that for all $x \in G$:

$$\alpha(x) = 1 \iff \alpha(N(x)) = \{0\} \tag{13}$$

In simple words, this means that for any node x, if x is assigned to 1, then all its successors has to be assigned to 0, and if x is assigned to 0, then there has to exist a node assigned to 1 among its successors. A consequence of this definition is that all sink nodes (nodes with no outgoing edges) in the graph have to be assigned to 1, since it vacuously does not point to any node assigned to 1. We use the notation $sol(\mathbf{G})$ to denote the set of all solutions of the graph $\mathbf{G}$.

1.4 Discourse Theories and Digraphs

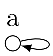
As mentioned earlier, there is a close connection between (1) models of a discourse theory, (2) kernels of a graph and (3) solutions of a graph. While Roy Cook has given us the equality of (2) and (3), we will now look at two functions connecting (1) and (2). We get the following definitions from []

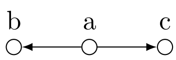
$\mathcal{T}$: translating a digraph $\mathbf{G}$ into a corresponding theory $\mathcal{T}(\mathbf{G})$ such that $sol(\mathbf{G}) = mod(\mathcal{T}(G))$.

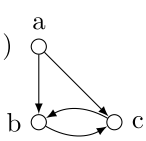
$\mathcal{G}$: translating a theory T into a corresponding digraph $\mathcal{G}(T)$ such that $mod(T) = sol(\mathcal{G}(T))$.

Given any digraph $\mathbf{G}$ we get the theory $\mathcal{T}(\mathbf{G})$ by taking, for each $x \in G$, the formula $x \leftrightarrow \bigwedge_{y \in N(x)} \neg y$ where $\bigwedge \emptyset = 1$.

Example 1.

$(\mathbf{G_1})$ 

$(\mathbf{G_2})$ 

$(\mathbf{G_3})$ 

Using the graphs from above, we get the following theories using $\mathcal{T}$:

$$\mathcal{T}(\mathbf{G_1}) = \{a \leftrightarrow \neg a\} \tag{14}$$
$$\mathcal{T}(\mathbf{G_2}) = \{a \leftrightarrow (\neg b \wedge \neg c), b, c\} \tag{15}$$
$$\mathcal{T}(\mathbf{G_3}) = \{a \leftrightarrow (\neg b \wedge \neg c), b \leftrightarrow \neg c, c \leftrightarrow \neg b\} \tag{16}$$

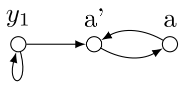
We will not be proving that $sol(\mathbf{G}) = mod(\mathcal{T}(G))$, but observe that $\mathbf{G_1}$ has no solution, just like its corresponding theory $\mathcal{T}(\mathbf{G_1})$ has no satisfying models. $\mathbf{G_2}$ has one solution, where one assigns $a = 0, b = 1, c = 1$. This assignment also works as a satisfying model for $\mathcal{T}(\mathbf{G_2})$. In $\mathbf{G_3}$, we get two solutions, both with a assigned to 0, but with 0 and 1 distributed on b and c. These are also the only two models satisfying $\mathcal{T}(\mathbf{G_3})$. It is generally true that $\mathcal{T}$ gives us our requested correspondence.

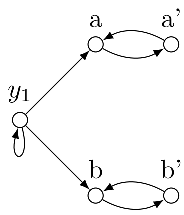
Conversely, given any discourse theory T (in fact, this will work given any PL theory, since we can translate CNF to GNF), we can derive the corresponding graph $\mathcal{G}(T)$ in the following way: All variables in the theory are vertices, and for each formula $x \leftrightarrow \bigwedge_{i \in I_x} y_i$ make a directed edge $\langle x, y_i \rangle$ for each $i \in I_x$.

Example 2.

$$T_1 = \neg a \qquad \iff (a \leftrightarrow \neg a'), (a' \leftrightarrow \neg a), (y_1 \leftrightarrow (\neg a' \wedge \neg y_1)) \tag{17}$$
$$T_2 = a \vee b \qquad \iff (a \leftrightarrow \neg a'), (a' \leftrightarrow \neg a), (b \leftrightarrow \neg b'), (b' \leftrightarrow \neg b), (y_1 \leftrightarrow (\neg a \wedge \neg b \wedge \neg y_1)) \tag{18}$$

Using $\mathcal{G}$ on the above theories – translated to GNF – gives us the following graphs:

$(\mathcal{G}(T_1))$


$(\mathcal{G}(T_1))$


Again, will we not be proving the correspondence, but notice T_1 has one model satisfying it, where $a = 0$.

1.5 Proofs

1.5.1 Translating CNF to GNF

Since any PL theory can be expressed in CNF, showing that any theory P in CNF can be translated to a theory R in GNF such that $P \leftrightarrow R$ gives us that any PL theory can be expressed in GNF.

Given any CNF theory P, start with an empty theory R and for each formula in P, follow the steps below to acquire its corresponding GNF formulae.

Step 1: For each literal $\neg x_i$ in the formula, introduce a fresh variable x_i' and add the following two GNF formulae to R: $x_i' \leftrightarrow \neg x_i, x_i \leftrightarrow \neg x_i'$, (unless this has already been done while translating an earlier formula in the theory).

Step 2: In each clause, replace every negative literal $\neg x_i$ with its corresponding x_i' fom step 1. Every clause does now contain all positive literals. For every clause $(x_1 \vee x_2 \vee \cdots \vee x_n)$, create a fresh variable y and add the following GNF formula to R:

$$y \leftrightarrow (\neg x_1 \wedge \neg x_2 \wedge \cdots \wedge \neg x_n \wedge \neg y)$$

The combined set of all these acquired formulae will make up the the corresponding theory R. We have only added proper GNF formulae and all variables appear to the left in exactly one clause, so R will indeed be a GNF theory.

Example 3.

$$\text{CNF:} \quad (a \vee b) \tag{19}$$
$$\text{GNF:} \quad (a \leftrightarrow \neg a'), (a' \leftrightarrow \neg a), (b \leftrightarrow \neg b'), (b' \leftrightarrow \neg b), (y_1 \leftrightarrow (\neg a \wedge \neg b \wedge \neg y_1)) \tag{20}$$
$$\tag{21}$$
$$\text{CNF:} \quad (\neg a) \tag{22}$$
$$\text{GNF:} \quad (a \leftrightarrow \neg a'), (a' \leftrightarrow \neg a), (y_1 \leftrightarrow (\neg a' \wedge \neg y_1)) \tag{23}$$
$$\tag{24}$$
$$\text{CNF:} \quad (a \vee b \vee \neg c) \wedge (\neg d \vee e) \wedge f \tag{25}$$
$$\text{GNF:} \quad c' \leftrightarrow \neg c, d' \leftrightarrow \neg d, y_1 \leftrightarrow (\neg a \wedge \neg b \wedge \neg c'), y_2 \leftrightarrow (\neg d' \wedge \neg e), y_3 \leftrightarrow \neg f \tag{26}$$

References

[1] G. Priest and F. Berto. (2013). Dialetheism. E. N. Zalta, Ed., [Online]. Available: `http://plato.stanford.edu/archives/sum2013/entries/dialetheism/`.