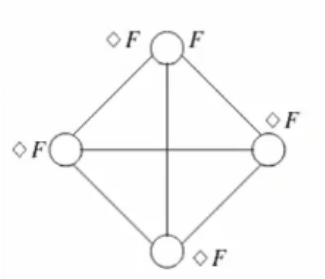
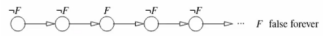
Planning and Reasoning

Gallotta Roberto

Thursday 10th December, 2020

1 Proof of $\Diamond F \Rightarrow \Box\Diamond F$





In S5, this is always true since $\Diamond F$ is true in all worlds. This proof can be extended to more complex models but still holds because whichever world we choose, $\Diamond F$ is true in all linked ones.

For TL this isn't always true: in this counterexample we have a valid TL model. $\Diamond F$ is true in the initial state since F is true in the thrid state. However, $\Diamond F$ is false in the fourth state. Then, $\Diamond F \Rightarrow \Box\Diamond F$ is false since $\Box\Diamond F$ is false in the inital state.

Obviously we can have formulas that are true for both S5 and TL always, formulas that are false in S5 and true in TL etc... since the validity of the formulas depends on the graph.

2 Conditions on relations

We define as accessibility relation $\langle s, t\rangle \in R$ as sRt, which simply means there is a directed edge from s to t. This can be:

- reflexive: For every s there exists sRs
- symmetric: For every sRt there exists tRs
- serial: for every node there is a successor, so for every s there exsists a t such that sRt
- transitive: if there is sRt and tRu, then there must be sRu

Some formulas are always true if the relation is of one or more type:

- $\Box F \Rightarrow F$ if the relation is reflexive
- $F \Rightarrow \Box\Diamond F$ if the relation is symmetric
- $\Box F \Rightarrow \Diamond F$ if the relation is serial

A formula is true in a modal logic if it's true in all models of that modal logic. A formula F is satisfiable if it is satisfied by at least one model: $\langle R, S, V\rangle, s \models F$. We can constrain satisfiability also to a particular modal logic.

3 Tableuax for modal logics

We assume no particular restrictions on the graph. We want to establish the satisfiability of a formula by checking if there exists a Kripke model such that $M, s \models F$. As opposed to tableaux for propositional logic where we have only one node (one world), here we start with a single world but we have to check also other worlds if F contains modal operators. We build the model such that it satisfies the formula, if we fail then the formula is unsatisfiable.

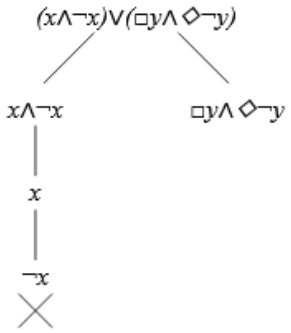
Facts:

- $\Box x \wedge \Box \neg x$ is satisfiable? The model should be such that in all successors both x and $\neg x$ are true. So, we simply have a model with no successors and just a node in which both x and $\neg x$ are true: $G = \langle N, E \rangle, N = \{s\}, E = 0$. If we don't create successors, we don't have to worry about the modal constraints too much since they're implicitly true in the initial state. **Do not create successors if not needed**. This doesn't work in S5 and TL since there is always at least the loop.
- $\Box x \wedge \Diamond \neg x$ is satisfiable? If we build a model with no successors $\Box x$ is automatically true, but $\Diamond \neg x$ needs at least one successor to be true. We then add a successor in which $\neg x$ is true, but now the first condition is false. The formula is then unsatisfiable because there will always be a contradiction. **Whenever we have $\Diamond F$, we need a successor where F is true**. **For every successor we add, any $\Box F$ formula will act on it by creating a constraint F there**.
- $\Box x \wedge \Diamond y \wedge \Diamond \neg y$ is satisfiable? We have to add two successors: one where y is true, one where $\neg y$ is true. In both these successors we make x true. **If possible, create different successors for different diamond formulas**. This is because satisfying formulas in a single successors is more complicated than satisfying each in each successor.

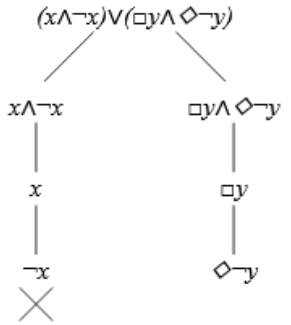
Example:

$$(x \wedge \neg x) \vee (\Box y \wedge \Diamond \neg y)$$

We want to use propositional tableau as much as possible.



Trying to satisfy the leftmost branch, we have a contradiction. This is fine since we have a disjunction.

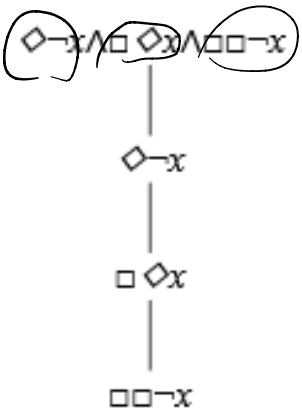


Then we want $\neg y$ true in a successor. However, $\Box y$ means that we should have y true in the same successor. This is a contradiction.

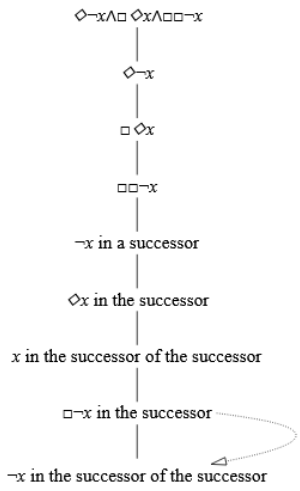
This is an easy example, more realistically we may have multiple successors. It'd be easier to assign names to this successors.

We want to formalize how we specify the relations:

$$\Diamond \neg x \wedge \Box \Diamond x \wedge \Box \Box \neg x$$



We don't have successors, so the boxes are inactive for now, but the diamond is. So, we add a successor where $\neg x$ is true. We then apply the effect of the boxes and diamonds until we reach a conclusion.



We reach a contradiction.

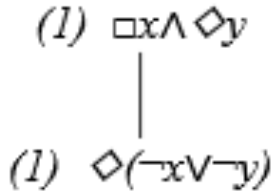
We want to keep track of where the formula holds. We can do this using integers: $(1, 5)$ is a world and is the predecessor of $(1, 5, 2)$ and $(1, 5, 2, 3)$ is the successor of $(1, 5, 2)$. This makes it very easy to create new successors. As a rule, we have for a sequence $(i_1, ..., i_m)$ in the Tableuax:

$$\frac{(i_1, ..., i_m)\Box F}{(i_1, ..., i_m, i_{m+1})F}$$

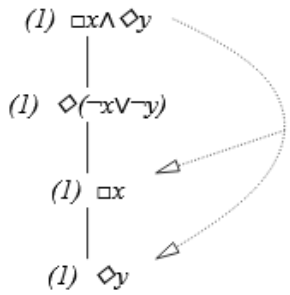
$$\frac{(i_1, ..., i_m)\Diamond F}{(i_1, ..., i_m, i_{m+1})F}$$

The closure is when F and $\neg F$ have the same sequence as label.
Example:

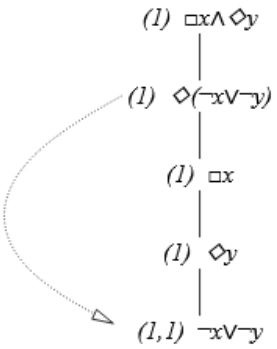
$$\{\Box x \wedge \Diamond y; \Diamond(\neg x \vee \neg y)\}$$



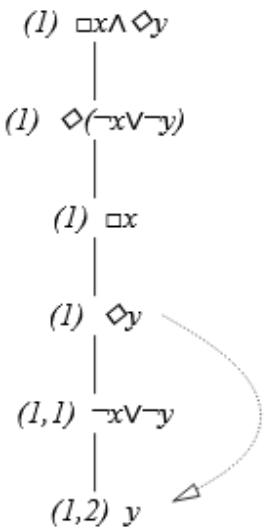
Initial states are labeled as (1). We want both formulas to be true. The first step could be to express the conjunction.



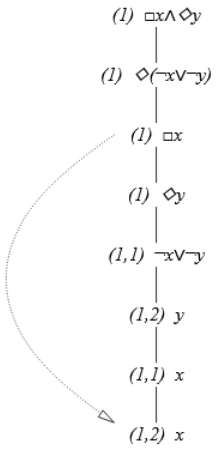
They still have label (1). Then we expand the second initial state (the $\Diamond(...)$).



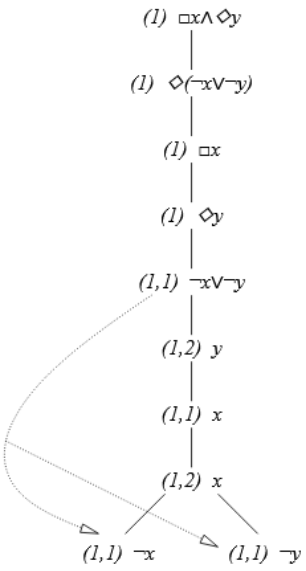
We generate the new successor $(1, 1)$. Note that in the successor there is no $\Diamond$. We then expand $\Diamond y$.



We generate the new successor $(1, 2)$ (since $(1, 1)$ already exists). Now $\Box x$ is expandable.

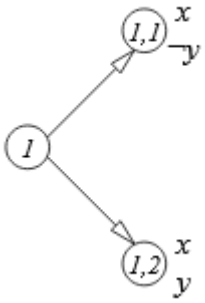


x is true in both $(1, 1)$ and $(1, 2)$. The only expandable formula now is $\neg x \vee \neg y$.



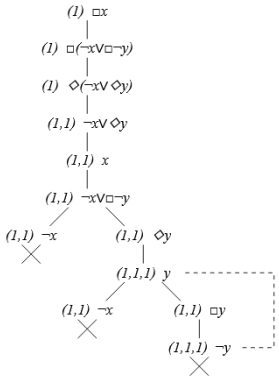
The disjunction is still applied to the world $(1,1)$. We reach a contradiction: $(1,1)$ has both x true and $\neg x$ true. We also have y true in $(1,2)$ and $\neg y$ true in $(1,1)$, but this is not a contradiction since they're different worlds, so the set is satisfiable.

The worlds are (1), $(1,1)$ and $(1,2)$. What is the model? In $(1,1)$ x and $\neg y$ are true. In $(1,2)$ x and y are true. We still don't have the values in (1), so we can choose whatever (for example, $\neg x$ and $\neg y$).



Another example:

$$\{\Box x; \Box(\neg x \vee \Box\neg y); \Diamond(\neg x \vee \Diamond y)\}$$



We first expand $\Diamond(\neg x \vee \Diamond y)$. Then we have that $\Box x$ and the existence of $(1,1)$ means that x is true in $(1,1)$. Similarly for $\Box(\neg x \vee \Box\neg y)$. Then we continue to expand until we reach a contradiction.

Another possibility instead of using integers is to use symbols $w, w', w'', ...$ to label the worlds. The behavior is similar to that of integers. The connection between worlds is represented using the pseudoformulas wRw',

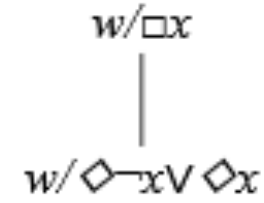
which means that w is connected to w'. The rules are:

$$\frac{w/\Box F, wRw'}{w'/F}$$

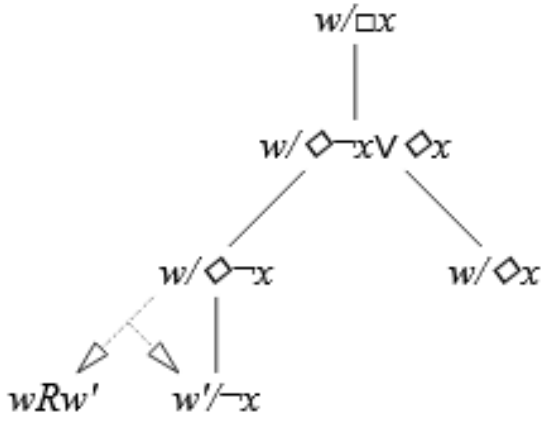
$$\frac{w/\Diamond F}{w'/F, wRw'}$$

Example:

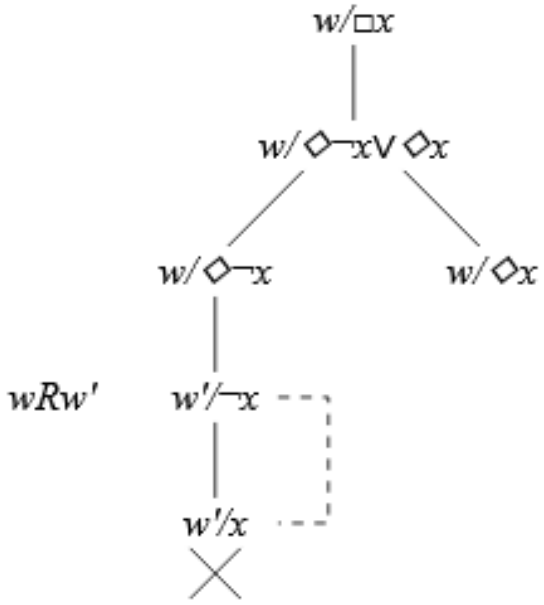
$$\{\Box x; \Diamond \neg x \vee \Diamond x\}$$



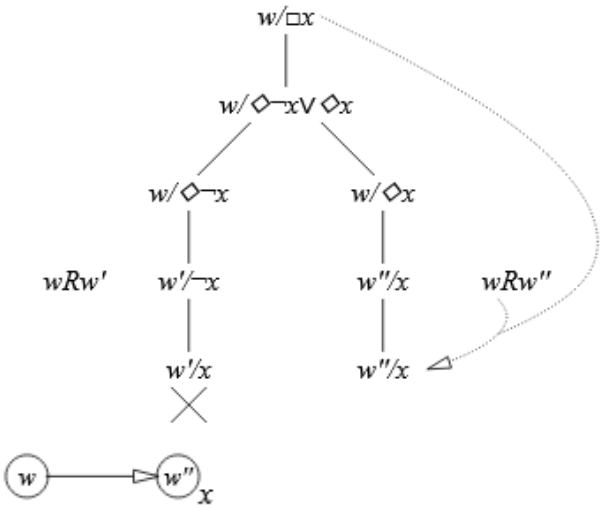
The first is inactive since the precondition also requires wRw', so it's not applicable. The second formula instead can be expanded using the usual rule for disjunction and the new worlds will have the same label. We then expand the $\Diamond$ formula.



Note that the $\Diamond$ generates both a formula and a pseudoformula (wRw') that is applied to the branch. Now we can expand the $\Box$ since we also have a pseudoformula for the precondition.



Here we have a contradiction: both x and $\neg x$ are true in the same world w', so the branch is closed. And we can expand the other $\Diamond$ formula (remember that they are always expandable).



We have no more expansions available, so the set is satisfiable. What is the model? We have to take both literals and the pseudoformulae on the sat branch: $w''/x, w''/x, wRw''$. Since in w the x is not specified, we can choose whatever we want.

Also, note that we don't include w' since it's in the closed branch only (it's irrelevant).

4 Set-based tableaux

Differently from previous Tableaux, here we consider all the formulas of the set in each node. The rules are:

- For propositional formulae:

$$\frac{A_1, ..., A_n, B \wedge C}{A_1, ..., A_n, B, C}$$

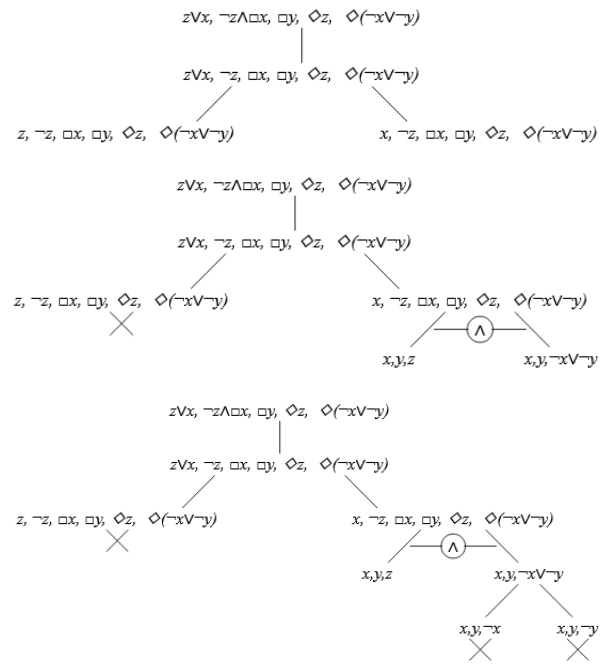
$$\frac{A_1, ..., A_n, B \vee C}{A_1, ..., A_n, B | A_1, ..., A_n, C}$$

- For modal formulae:

$$\frac{A_1, ..., A_n, \Box B_1, ..., \Box B_n, \Diamond C}{B_1, ..., B_n, C}$$

Example:

$$\{z \vee x; \neg z \wedge \Box x; \Box y; \Diamond z; \Diamond(\neg x \wedge y)\}$$



First we make the $\wedge$ explicit. Then we separate the $\vee$ in the branches. Note that in one branch we have z and in the other we have x. The branch with z results in a contradiction since there is also $\neg z$. The other branch has two $\Diamond$ formulas, so we will have to branch again.

Here the branching is an "and" branching, meaning that both branches must then be satisfied (if one closes, we can close the other). We also already applied the $\Box$. x, y, z can't be expanded further, so I expand the other formula.

Since the branches closes, the entire branch closes as well. The formula is thus unsatisfiable.