Repair example

For a model with compounds a,b,c, suppose a has an inconsistent function.

To render it consistent, we will be using the results obtained from a single experiment performed in the real world system, presented in the observations below:

timestep 0	timestep 1	timestep 2	timestep 3		timestep n-1	timestep n
obsv(0,a,0).	obsv(1,a,1).	obsv(2,a,1).	obsv(3,a,0).	…	obsv(n-1,a,1).	obsv(n,a,1).
obsv(0,b,1).	obsv(1,b,1).	obsv(2,b,0).	obsv(3,b,0).		obsv(n-1,b,1).	obsv(n,b,0).
obsv(0,c,1).	obsv(1,c,0).	obsv(2,c,1).	obsv(3,c,1).		obsv(n-1,c,0).	obsv(n,c,1).

timestep 0	timestep 1	timestep 2	timestep 3		timestep n-1	timestep n
obsv(0,a,0).	obsv(1,a,1).	obsv(2,a,1).	obsv(3,a,0).	…	obsv(n-1,a,1).	obsv(n,a,1).
obsv(0,b,1).	obsv(1,b,1).	obsv(2,b,0).	obsv(3,b,0).		obsv(n-1,b,1).	obsv(n,b,0).
obsv(0,c,1).	obsv(1,c,0).	obsv(2,c,1).	obsv(3,c,1).		obsv(n-1,c,0).	obsv(n,c,1).

Our function must be able to replicate the observations, i.e. produce 0s when 0s are observed, and 1s when 1s are observed.

How are we ensuring that the 0s are being produced by our function?

- We look at the negative observations, and ensure that no node is ever positive when they happen

How are we ensuring that the 1s are being produced by our function?

- We look at the positive observations, and ensure that at least one node is positive when they happen

timestep 0	timestep 1	timestep 2	timestep 3		timestep n-1	timestep n
obsv(0,a,0).	<u>obsv(1,a,1).</u>	obsv(2,a,1).	<u>obsv(3,a,0).</u>	…	obsv(n-1,a,1).	obsv(n,a,1).
obsv(0,b,1).	obsv(1,b,1).	obsv(2,b,0).	obsv(3,b,0).		obsv(n-1,b,1).	obsv(n,b,0).
obsv(0,c,1).	obsv(1,c,0).	obsv(2,c,1).	obsv(3,c,1).		obsv(n-1,c,0).	obsv(n,c,1).

How do we create our nodes? Depends on the chosen activators/inhibitors. Let's suppose that compounds a,b and c are all activators.

The nodes will be all the combinations of those activators that can replicate the activations of a, while also replicating the observations where a is inactive.

E.g., from timestep 0 to 1, a becomes active. Because both b and c are active in the timestep before, the nodes for obsv(1,a,1) would be: {b}, {c}, {bc}. But, not all of these nodes are able to respect the constraint that a's negative observations must also be upheld:

- At time step 3, we know that a should be inactive. But, if we consider {c} to be one of the function nodes, that observation would be violated. This is because, at time step 2, c is active, therefore causing that node {c} be active, and consequently that a be active in the next time step.

So, out of the initial three nodes, only two of them may be chosen to justify the positive observation in time step 1.

A similar process takes place for every positive observation, ensuring that our nodes respect all observations.

timestep 0	timestep 1	timestep 2	timestep 3		timestep n-1	timestep n
obsv(0,a,0).	obsv(1,a,1).	obsv(2,a,1).	obsv(3,a,0).	…	obsv(n-1,a,1).	obsv(n,a,1).
obsv(0,b,1).	obsv(1,b,1).	obsv(2,b,0).	obsv(3,b,0).		obsv(n-1,b,1).	obsv(n,b,0).
obsv(0,c,1).	obsv(1,c,0).	obsv(2,c,1).	obsv(3,c,1).		obsv(n-1,c,0).	obsv(n,c,1).

But, some observations are ‘duplicated’.

For example, if we look at timesteps 2 and n, both observations describe the same model state. Since they also come from observations that describe the same state (1 and n-1), if we can find a node to cover observation at time step 2, that same node can cover the observation at time step n (because they have the same inputs).

This helps us define an upper bound for the possible nodes. Instead of considering a maximum of 4 nodes (one for each positive observation), we can consider only 3.

Note: while timestep n is a duplicate of timestep 2, we do not know if n-1 is a duplicate of 1, since n-1’s previous state is not known.

timestep 0	timestep 1	timestep 2	timestep 3		timestep n-1	timestep n
obsv(0,a,0).	obsv(1,a,1).	obsv(2,a,1).	obsv(3,a,0).	…	obsv(n-1,a,1).	obsv(n,a,1).
obsv(0,b,1).	obsv(1,b,1).	obsv(2,b,0).	obsv(3,b,0).		obsv(n-1,b,1).	obsv(n,b,0).
obsv(0,c,1).	obsv(1,c,0).	obsv(2,c,1).	obsv(3,c,1).		obsv(n-1,c,0).	obsv(n,c,1).

So what would be the chosen function for this example? (considering that, when they are present as regulators in a node, a,b,c will be activators)

Our possible nodes would be:

obsv(1,a,1): {b}, {bc}

obsv(2,a,1): {b}, {ab}

obsv(n-1,a,1) - cannot determine, but let's consider: {b}

Since we will be minimizing the number of nodes, the final function would be Fa = b

Note: if we consider the hypothetical scenario where observation n-1 has {ab}, {bc} for nodes, then our possible nodes would be:

obsv(1,a,1): {b}, {bc}

obsv(2,a,1): {b}, {ab}

obsv(n-1,a,1) - not present, but let's consider: {ab}, {bc}

In this scenario, the final function could either be:

- Fa = b V ab
- Fa = b V bc
- Fa = ab V bc

In this case, whichever function had more regulators in common with the original inconsistent function would be chosen.