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

The function will be built using nodes, which are combinations of compounds. E.g. {ab}, {c}, {abc}, etc (the final function will look something like Fa = ab V c).

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

Q: How do we create our nodes?

```
1 {node_regulator(1..MN,C) : maximum_node_number(MN), observation_potential_regulator(E,T,C)} :- unique_positive_observation(E,T).
```

A: We use positive observations (observations where the compound is active) which are unique.

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).

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 replicate observation at time step 2, that same node can replicate the observation at time step n (because they have the same inputs).

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.

```
1 {node_regulator(1..MN,C) : maximum_node_number(MN), observation_potential_regulator(E,T,C)} :- unique_positive_observation(E,T).
```

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).

Q: How can we use them to generate nodes?

A: For every positive observation, we can look at the previous time step to identify potential regulators, i.e. compounds that may have contributed to a's activation. Our nodes will be combinations of these compounds.

In more detail, 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) could be: {b}, {c}, {bc} (for this example, we are considering activators only). 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, only {b}, and {bc} are valid nodes.

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

```
1 {node_regulator(1..MN,C) : maximum_node_number(MN), observation_potential_regulator(E,T,C)} :- unique_positive_observation(E,T).
```

Q: How are we upholding these conditions?

A:

```
:- timestep_positive_node(E,T-1,N), curated_observation(E,T,compound,0), T > 0.
```

If a negative observation exists in some experiment E and timestep T, we can have no positive nodes in timestep T-1 (ensures negative observations are never violated)

```
:- unique_positive_observation(E,T), not node_positive_observation_coverage(_,E,T).
```

If a unique observation exists in some experiment E and timestep T, we must have some positive node that covers that observation, i.e. is positive in timestep T-1 (ensures positive observations are never violated)

```
1 {node_regulator(1..MN,C) : maximum_node_number(MN), observation_potential_regulator(E,T,C)} :- unique_positive_observation(E,T).
```

Q: What is the maximum number of nodes we will need?

A: At most, we will need one node per unique positive observation, i.e.

```
maximum_node_number(POSITIVE_OBSERVATIONS_COUNT) :- positive_observations,
 POSITIVE_OBSERVATIONS_COUNT = #count{E,T : unique_positive_observation(E,T)}.
```

Thus, the generator creates our nodes in such a way that, for every unique positive observation E,T, some compound C, which is a potential regulator of that observation, will be placed in at least one of the MN (maximum) nodes.

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 our observations have the following possible nodes:

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

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

obsv(n-1,a,1): {a}, {b}

Then, in this scenario, the final function could be one of these two:

- Fa = a V bc
- Fa = a V b

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