Generating all states in a Qualitative Reasoning project

Gongze Cao
12161381

Jonathan Mitnik
10911197

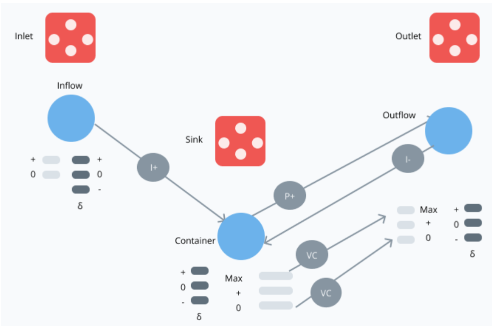

Figure 1: Causal model of the sink

[1]

1 Introduction

2 The Domain: Causal Model

The constraints as presented within the frame of this assignment consist of the the following:

- A P+ (positive proportional) relationship from Container to Outflow. This insinuates that whenever the Container's derivative, the outflow goes through a proportional change for its derivative.

- A I+ (influential positive) from Inflow to Container. Whenever the inflow is active, which is understood as having a magnitude greater than zero, the container will be influenced to increase (thus setting the derivative above zero).

- An I- relation from Outflow to Container. The existence of the outflow means that there water in the container is decreasing.

- A VC (value correspondence) from the containers max value to the outflow's max. If the container reaches the top, the outflow will also be at its strongest.

- A VC relation from the containers empty value to the outflow's empty. With no water in the container, there is no water to be flowing out.

Figure 1 captures this relationship.

3 Methodology

3.1 Assumptions and decisions

Within the frame of this assignment, a number of assumptions were made when implementing a set of rules for the generation of the state graph. The level of

Aside from the given constraints regarding the domain's relations, a number of additional constraints were set as inferred by the existing ones. The exogenous variable, the inflow, has been set to be a binary variable, which means that either the sink is on in an instant, or it is off: this is to simplify the unnecessary amount of states which may not add much to the model itself. Furthermore, the exogenous variable will be modeled based off a negative parabola. The reason a negative parabola is chosen, is twofold: a real interaction with a sink would go in the way off washing hands (on), then turning it off, and turning it on yet again after some time went by. As such, a negative parabola starts this pattern immediately, and can represent the inactivity (drainage) of a sink as well as the reactivation.

Furthermore, the representation to a real sink is taken further to the landmark states of our container. If the container, and by proxy the outflow, reaches a maximum or minimum, then the derivatives are instantly set within the same state to zero. This way, the state will instantly achieve maximum without insinuating that there is more to come, or less to come on a minimum state.

```
States 0 and 1 have a number of differences between them.

There is a difference for the container entity!
        The magnitude is not the same anymore! The difference of state 0 for container is -1 lesser compared to state 1
        The derivative is not the same! The difference of state 0 for container is 1 compared to state 1

There is a difference for the outflow entity!
        The derivative is not the same! The difference of state 0 for outflow is 1 compared to state 1
```

Figure 2: Inter-state trace

```
State 1 has 0 number of nodes pointing out, and 1 pointing in.
                - Inflow: The magnitude is 1 and derivative is 0!
                - Outflow: The magnitude is 1 and derivative is 0!
                - Container: The magnitude is 1 and derivative is 0!
```

Figure 3: Intra-state trace

3.2 Generating a state-graph

Representing the state-graph. A state-graph is initialized with an empty node. This node contains a collection of the different magnitudes and derivatives of a quantity, which is referred to as a state. By turning these values of the entities into a string, it is possible to hash a state by the string representation of all these values. This will be important when storing each state in a global variable 'visitedStates', which keeps track of new encountered variables.

Generating new states. Upon entering a state, the main goal is to apply a number of functions on the value of the entities, and to generate as many possible children as possible. This is done in three stages: first apply the current derivatives on the children. If the magnitude value of an entity's quantity is on a landmark (such as zero and maximum), this will produce a single state. However, due to the nature of intervals as explained in [1], the values could either transition to another value in the interval or to the next state in the quantity space: as such, these possible ambiguity leads to an additional child being generated to account for this chance. The second stage is to apply the current relations to the states. This can either change the state immediately (e.g. apply when no ambiguities are present), or the examined state is split into multiple states with different outcomes based on the ambiguities to resolve. The order of resolution is to start with ensuring influence relations first, due to the nature of the causal graph of this project: the container is affected first by its ambiguities, and the outflow always follows due to the Value Correspondence and proportionality relationships as stated in section 2. This will output a wide list of children, each represented as a State with its own individual hashable value. Finally, to account for the exogenous variables, each stack will have a reference to a stack which is copied to each child. This stack represents the 'lifecycle' of the exogenous pattern, and denotes which of the actions have been taken related to the pattern of the exogenous variable. For instance, because this application uses the parabola, the initial state starts with 0 and will follow a derivative incline of 1, followed by an eventual decline back to 0. For each transition, a value of the stack is popped, until the stack is empty, and the exogenous variable remains constant at 0. The ambiguity arises when in the generation this applies, and so, each state accounts for the possibility. After applying the exogenous variability, the remaining step is to check for consistency. This is done by pruning all of the states which do not follow the value correspondence relations, and by setting the derivatives for entities which have arrived at their maximum and minimum landmark, to be zero.

Exploring all states With the new list of states, a depth first search is done to continue exploring until no new states can be found eventually. For each time that a state generates a new state, a Node is created for a parent and for the children. The edges generated to connect the parent to the children.

4 Results

4.1 Trace

The trace in the algorithm can indicate a number of difference between two states, such as quantities. The built-in traces either can focus on the state itself, or between states, such as in figures 3 2 respectively.

4.2 State graph

The generated state graph resulted in approximately 37 states. Initially it seems difficult to follow the state graph. From an implementation standpoint, the intuition that accompanies the interpretation of the state graph is derived from mostly building the model. By pruning the states, the state graph represents the rules setup well and established.

It is interesting that not many clear termination or stable nodes can be found. One of the more surprising findings aside from that is how many states are generated from these three simple interactions between entities. The exogenous variable's pattern (parabola) doesn't seem to have much impact on the state graph rather than allowing for a larger number of extra states.

There are often transitions between states in which they seem to fall back and jump between each other. This often happens in intervals, where an interval might refer often to itself, and just as easily could jump between landmark states.

References

[1] Bert Bredeweg, Anders Bouwer, Jelmer Jellema, Dirk Bertels, Floris Linnebank, and Jochem Liem. Garp3 - a new workbench for qualitative reasoning and modelling. In *20th International Workshop on Qualitative Reasoning (QR-06), C. Bailey-Kellogg and B. Kuipers (eds*, pages 21–28, 2006.