In: Proceedings of Qualitative Reasoning (QR2019)

Checking Qualitative Reasoning with the Conflict Resolution Method

John T Maxwell III[1] , **Johan de Kleer**[1] and **Matt Klenk**[1]
[1]Palo Alto Research Center
{maxwell, dekleer, klenk}@parc.com

Abstract

One way of implementing qualitative reasoning is to use a qualitative algebra. Unfortunately, the qualitative operator for addition is ambiguous. Therefore, qualitative analysis of constraints can be consistent but there does not exist a quantitative solution. We show that you can solve this problem by using the Conflict Resolution Method to detect such situations. We describe how to incorporate CRM into qualitative simulation.

1 Introduction

While qualitative simulation will always include *spurious states* [Struss, 1988], reducing the number of spurious states and transitions is an active research problem for integrating qualitative reasoning into larger systems. The goal of our work is to integrate qualitative reasoning into the design process. In our previous work, we have shown how to do perform envisionment from Modelica models used in the many design contexts [Klenk *et al.*, 2014] and how this technique fits into a design toolchain [Lattmann *et al.*, 2014]. In our current work on design exploration [Maxwell *et al.*, 2019], qualitative reasoning eliminates component topologies that cannot meet the design requirements. Spurious states are qualitative states for which there are no quantitative solutions. Spurious states impact design because there exist topologies of components that qualitative simulation indicates meet the requirements, but for which there are no quantitative realizations.

In this short paper, we adapt a recent advance in linear system solving, the *conflict resolution method* [Korovin *et al.*, 2009], for application in qualitative reasoning. In the following sections, we describe the problem, how we adapt the method for qualitative reasoning, and present examples of its execution with a limited analysis.

2 Problem: Spurious States

Qualitative simulation of component-based models uses a qualitative algebra to specify constraints between variables and their derivatives. A qualitative algebra defines operations over *qualitative* values instead of *quantitative* values. For instance, Table 1 shows qualitative multiplication results[1].

*	Q-	Q0	Q+
Q-	Q+	Q0	Q-
Q0	Q0	Q0	Q0
Q+	Q-	Q0	Q+

Table 1: Qualitative multiplication

Similarly, Table 2 shows qualitative addition results.

+	Q-	Q0	Q+
Q-	Q-	Q-	?
Q0	Q-	Q0	Q+
Q+	?	Q0+	Q+

Table 2: Qualitative addition

Notice that adding Q+ to Q- is undefined in Table 2, since the result can be any value (e.g., Q-, Q0, or Q+). Qualitative analysis typically propagates all three values through the constraints. But this ambiguity leads to spurious states.

To see how problems can arise, consider the constraints:

$$\begin{aligned} x + y = z \\ x + w = z \end{aligned} \tag{1}$$

Suppose that x and z are both Q+, and we want to assign values to y and w. If we look at the constraints individually in light of Table 2, then we see that y can have any value and that w can have any value. For example, y is Q+ and w is Q-. There are 9 possible value combinations for y and w. However, if we use variable substitution, we get $y = w$. So, we have a inconsistency since Q+ cannot be equal to Q-. With this constraint, only 3 of the 9 possible value combinations for y and w are valid.

3 Previous Approaches for Adding Constraints

One thread of qualitative reasoning research focuses on identifying extra constraints (e.g., [de Kleer and Bobrow, 1984],

[1]For the purpose of this paper, we will only use three qualitative values, although more are possible.

[Kuipers *et al.*, 1991]). As indicated in the example from the previous section, we can create a new constraint through substitution and use the following set of constraints.

$$\begin{aligned} x + y &= z \\ x + w &= z \\ y &= w \end{aligned} \tag{2}$$

The second constraint is now redundant. In our previous work, we discussed how the equations identified by quantitative system modeling approaches are insufficient for qualitative reasoning [Klenk *et al.*, 2014]. One way to solve this problem is to apply continuity and compatibility conditions (e.g., Kirchoff's voltage and current laws) repeatedly to deduce new constraints to add to the system. A common solution is to add one constraint for each planar window in a circuit.

To see how this can still lead to spurious states, consider the circuit defined by (3), which is shown as a planar circuit in Figure 1.

$$\begin{aligned} t_1 + v_{12} &= t_2 \\ t_1 + v_{13} &= t_3 \\ t_3 + v_{32} &= t_2 \\ t_3 + v_{34} &= t_4 \\ t_4 + v_{41} &= t_1 \\ t_4 + v_{42} &= t_2 \end{aligned} \tag{3}$$

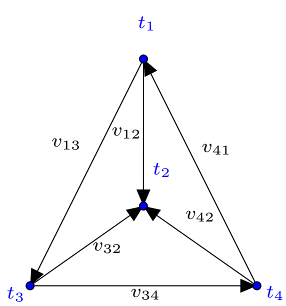

Figure 1: Planar circuit for (3). Each edge represents a component the voltage across specified using with subscripts for the positive and negative connections (v_{13} is the voltage between terminals 1 and 3), and voltages with respect to ground are specified at each terminal (t_1 is the voltage at the terminal 1).

This circuit has three windows in it. The KVL constraints for these windows are shown in (4).

$$\begin{aligned} v_{13} + v_{32} - v_{12} &= 0 \\ v_{34} + v_{42} - v_{32} &= 0 \\ v_{41} + v_{12} - v_{42} &= 0 \end{aligned} \tag{4}$$

Now consider the value assignments in (5).

$$\begin{aligned} t_1 = t_2 = t_3 = t_4 &= Q+ \\ v_{13} = v_{34} = v_{41} &= Q- \\ v_{12} = v_{42} = v_{32} &= Q+ \end{aligned} \tag{5}$$

If we check these value assignments against the constraints we have, there are no inconsistencies (6). All of the assignments are valid according to qualitative addition.

$$\begin{aligned} t_1(Q+) + v_{12}(Q+) &= t_2(Q+) \\ t_1(Q+) + v_{13}(Q-) &= t_3(Q+) \\ t_3(Q+) + v_{32}(Q+) &= t_2(Q+) \\ t_3(Q+) + v_{34}(Q-) &= t_4(Q+) \\ t_4(Q+) + v_{41}(Q-) &= t_1(Q+) \\ t_4(Q+) + v_{42}(Q+) &= t_2(Q+) \\ v_{13}(Q-) + v_{32}(Q+) &= v_{12}(Q+) \\ v_{34}(Q-) + v_{42}(Q+) &= v_{32}(Q+) \\ v_{41}(Q-) + v_{12}(Q+) &= v_{42}(Q+) \end{aligned} \tag{6}$$

However, there is a spurious assignment. If we create a KVL constraint for the outer loop of the planar circuit in Figure 1 we get $v_{13} + v_{34} + v_{41} = 0$. If we check the assignments against this constraint, we get an inconsistency (7).

$$v_{13}(Q-) + v_{34}(Q-) + v_{41}(Q-) = 0 \tag{7}$$

Adding every KVL constraint for a topology is in the worst case, exponential in the size of the topology. This motivates constructing only the additional constraints that eliminate spurious situations.

Another approach to reducing spurious solutions is to integrate qualitative and quantitative algebraic reasoning [Williams, 1990; Williams, 1991] to eliminate inconsistencies. However, this approach requires a new complex algebraic simplifier and is not designed for use in an envisioner.

4 Conflict Resolution Method

The *Conflict Resolution Method* (CRM) [Korovin *et al.*, 2009] is a new method for solving systems of linear inequalities that is an order of magnitude faster than the Fourier-Motzkin method [Williams, 1986]. This result makes it promising for adaption to envisionment. Our envisioner follows from QSIM [Kuipers, 1994] by generating new successor qualitative states from an initial state. We employ CRM as a final filter on each newly created qualitative state. If CRM identifies a conflict in the set of linear inequalities that represents the state, then it will not be added as a successor state.

The CRM method works by making a tentative assignment of values to variables, checking for inconsistencies, introducing new constraints based on the inconsistencies found, and iterating until all of the assignments are consistent or until a trivial inconsistency is found (e.g. $0 > 0$). Therefore, our adaptation requires that the model be specified as quantitative equations with parameters specified qualitatively.

To use the Conflict Resolution Method, we first assume that a qualitative reasoner has already checked that a given assignment of qualitative values to variables is consistent with the given set of constraints. Next, we convert the qualitative values into inequalities, so that $v = Q+$ becomes $v > 0$, $v = Q-$ becomes $v < 0$, and $v = Q0$ becomes $v = 0$. We then extract a subset of the original constraints that represent all of the additions. Finally, we apply the Conflict Resolution Method to these inequalities along with the addition constraints. If the Conflict Resolution Method detects that the system of constraints reduces to a trivial inconsistency, then the assignment is invalid. Otherwise, the assignment is valid.

4.1 Example

Let's see how this works with the constraints given by (2), repeated here as (8):

$$\begin{aligned} x + y = z \\ x + w = z \end{aligned} \tag{8}$$

Given a qualitative state where x is Q+, y is Q+, z is Q+, and w is Q-. The qualitative reasoner doesn't detect an inconsistency in the qualitative additions involved. Now we convert the qualitative values to inequalities and add them to the constraints that we already have (9):

$$\begin{aligned} x + y = z \\ x + w = z \\ w < 0 \\ x > 0 \\ y > 0 \\ z > 0 \end{aligned} \tag{9}$$

The Conflict Resolution Method starts by ordering the variables lexigraphically. Consider the order w, x, y, and z. It then associates each constraint with the variable in it that has the highest order. Thus, we associate $w < 0$ with w, $x > 0$ with x, $y > 0$ with y, and $x + y = z$, $x + w = z$, and $z > 0$ with z. It then starts with the lowest variable and tentatively assigns a quantitative value that is consistent with its constraints. The lowest variable is w, and assigning $w = -1$ is consistent with $w < 0$. There are no inconsistencies, therefore goes to the next variable. The next variable is x, and assigning $x = 1$ is consistent with $x > 0$. Since x has no inconsistencies, it goes to the next variable. The next variable is y, and assigning $y = 1$ is consistent with $y > 0$. Since y has no inconsistencies, it goes to the next variable.

At this point, we have assigned $w = -1$, $x = 1$, and $y = 1$, so $x + y = z$ reduces to $z = 2$ and $x + w = z$ reduces to $z = 0$. Notice that $z = 2$ and $z = 0$ are inconsistent. The method deals with this local inconsistency by creating a new constraint that is the combination of the constraints that produced the inconsistency. The new constraint is created by eliminating z from $x + y = z$ and $x + w = z$. This produces $x + y = x + w$, which reduces to $y = w$. Now, the method associates this new constraint with the variable in it with the highest order, which is y. It then returns to y and processes it again.

The method processes y again with $y > 0$ and $y = w$. Since we tentatively assigned $w = -1$, $y = -1$. But this is inconsistent with $y > 0$. The method deals with this local inconsistency by combining $y > 0$ and $y = w$ to produce $w > 0$. The method associates this new constraint with w and returns to w to process it again.

The method process w again with $w < 0$ and $w > 0$. This produces a local inconsistency. The method deals with this local inconsistency by combining $w < 0$ and $w > 0$ to produce $0 > 0$. Since this is a trivial inconsistency, no solution is possible and (8) is globally inconsistent.

Notice that when the method deduced $y = w$ from $x + y = z$ and $x + w = z$, neither constraint depended on a variable assignment. Both constraints are always true, no matter what values are assigned to variables. This means that we can add $y = w$ to the original constraints and use it with other assignments of values to values (e.g. $y = Q-$ and $w = Q+$). This speeds up the Conflict Resolution Method when checking other assignments to variables in later steps in the envisionment.

Notice also that we did not need a full assignment of values to variables to detect the inconsistency. If we had left out $x > 0$ and $z > 0$ from the constraints, the method would have still found the inconsistency (it temporarily assigns $x = 0$ if x had no constraints). This means that we can use the Conflict Resolution Method in cases where a system has variables that are under constrained. In this case, we can leave the variables unspecified, and the envisionment will still work.

	Qualitative States	Time (seconds)
Baseline	43,709	30
With CRM	35,069	31

Table 3: Results comparing qualitative simulation with and without conflict resolution algorithm in terms of envisionment size in qualitative states and computation time in seconds.

4.2 Applications in Design Space Exploration

While the above method will reduce the number of spurious qualitative states, it does require additional computation on each qualitative state. Therefore, we report some preliminary results as part of our work on automated design [Maxwell *et al.*, 2019]. Envisioning a set of 22 seven component designs, we measure the total evaluation time and total qualitative states. Table 3 illustrates a 20% reduction in qualitative states with a 3% increase in computation time.

5 Conclusion

We have shown that adding the Conflict Resolution Method provides a significant reduction in envisionment size while incurring minimal computational overhead. Although it takes time to run the Conflict Resolution Method, it also saves time, since some qualitative states get filtered early during envisionment reducing the number of states that must be expanded. These results are preliminary and should be followed with a more detailed analysis over a broad range of qualitative models. Furthermore, the promise of this approach is that by eliminating spurious qualitative states, we will more effectively filter topologies qualitatively. Therefore, we need to assess the performance of CRM within the context of our conceptual design system.

Finally, the success of the Conflict Resolution Method suggests another approach to qualitative reasoning, which is to avoid qualitative algebras and to reason about regular algebras qualitatively instead. We did this here by converting constraints like $v = Q+$ into inequalities like $v > 0$ and then applying a standard algorithm for reasoning over inequalities. We may be able to extend this approach to other operators in some situations.

Acknowledgements

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA) under contract number HR001118C0041. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[de Kleer and Bobrow, 1984] Johan de Kleer and Daniel G Bobrow. Qualitative reasoning with higher-order derivatives. In *AAAI*, pages 86–91, 1984.

[Klenk *et al.*, 2014] Matthew Evans Klenk, Johan De Kleer, Daniel Bobrow, and Bill Janssen. Qualitative reasoning with modelica models. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[Korovin *et al.*, 2009] Konstantin Korovin, Nestan Tsiskaridze, and Andrei Voronkov. Conflict resolution. In *Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming*, CP'09, pages 509–523, Berlin, Heidelberg, 2009. Springer-Verlag.

[Kuipers *et al.*, 1991] Benjamin J Kuipers, Charles Chiu, David T Dalle Molle, and DR Throop. Higher-order derivative constraints in qualitative simulation. *Artificial Intelligence*, 51(1-3):343–379, 1991.

[Kuipers, 1994] Benjamin Kuipers. *Qualitative reasoning: modeling and simulation with incomplete knowledge*. MIT press, 1994.

[Lattmann *et al.*, 2014] Zsolt Lattmann, Adrian Pop, Johan De Kleer, Peter Fritzson, Bill Janssen, Sandeep Neema, Ted Bapty, Xenofon Koutsoukos, Matthew Klenk, Daniel Bobrow, et al. Verification and design exploration through meta tool integration with openmodelica. In *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, number 096, pages 353–362. Linköping University Electronic Press, 2014.

[Maxwell *et al.*, 2019] John Maxwell, Johan de Kleer, and Matthew Klenk. Using modelica models for qualitative reasoning. In *Submitted to the 33rd Annual Workshop on Qualitative Reasoning (QR19)*, Macau, China, 2019.

[Struss, 1988] P. Struss. Mathematical aspects of qualitative reasoning. *International Journal of Artificial Intelligence in Engineering*, 3(3), 1988.

[Williams, 1986] H. P. Williams. Fourier's method of linear programming and its dual. *The American Mathematical Monthly*, 93(9):681–695, 1986.

[Williams, 1990] Brian C Williams. Minima a symbolic approach to qualitative algebraic reasoning. In *Readings in qualitative reasoning about physical systems*, pages 312–317. Elsevier, 1990.

[Williams, 1991] Brian C. Williams. A theory of interactions: Unifying qualitative and quantitative algebraic reasoning. *Artif. Intell.*, 51(1-3):39–94, 1991.