Towards Qualitative Causal Model Synthesis with Answer Set Programming

Moritz Bayerkuhnlein[1] , **Bert Bredeweg**[2,3] , **Marco Kragten**[2] and **Diedrich Wolter**[1]
[1]University of Luebeck, Germany
[2]Amsterdam University of Applied Sciences, The Netherlands
[3]University of Amsterdam, The Netherlands
{moritz.bayerkuhnlein, diedrich.wolter}@uni-luebeck.de, {m.kragten,b.bredeweg}@hva.nl

Abstract

In this demonstration, we present an Answer Set Programming (ASP) encoding for Qualitative Causal models. In particular, we show how this encoding can be used to reason about qualitative simulations, where answer sets represent states or state transitions of a given qualitative model. Furthermore, the same encoding can be used to synthesize a qualitative model from a set of provided states.

1 Introduction

Qualitative simulation uses symbolic models to reason about the possible behaviours of dynamic systems without requiring precise numerical information. Two prominent approaches are Kuipers' QSIM [1986], which represents dynamics through qualitative differential equation constraints (QDEs), and Qualitative Causal Models [Bredeweg *et al.*, 2009], which build on Qualitative Process Theory [Forbus, 1984] and explain change through explicit causal dependencies between quantities.

Such models have been applied to monitoring and diagnosis [Das *et al.*, 2025], as well as to education and conceptual modelling [Bredeweg and Forbus, 2003; Spitz *et al.*, 2021]. Beyond prediction, they make explicit the causal assumptions underlying a simulation, rendering the resulting behaviours interpretable and allowing the model itself to serve as an explanation of the system dynamics.

However, constructing such models by hand is laborious and error-prone. This has motivated a long-standing line of work on qualitative system identification and qualitative model learning, including several contributions presented at the QR Workshop [Bratko *et al.*, 1991; Ramachandran *et al.*, 1994; Coghill *et al.*, 2008]. More recently, approaches involving LLMs have been used to support model formulation, but the resulting intermediate models still require formal validation and inspection [Suzuki and Yoshioka, 2024; Forbus, 2025].

The problem also connects to a broader tradition in AI concerned with synthesizing symbolic representations from data. Inductive Logic Programming (ILP) [Muggleton, 1991] is a prominent example, where the goal is to learn a logic program that explains examples relative to given background knowledge. Answer Set Programming (ASP) is a logic programming paradigm in Knowledge Representation and Reasoning, combining concise declarative problem specifications with efficient solving technology for hard combinatorial problems [Gebser *et al.*, 2011].

We present an ASP encoding for the qualitative simulation of qualitative causal models, following the semantics of Garp3 [Bredeweg *et al.*, 2009]. The encoding serves as a declarative interpreter for Garp-style models, making their simulation semantics explicit in ASP. Beyond simulation, this representation provides a basis for abductive reasoning over model candidates in qualitative model learning [Pang and Coghill, 2010]. In this paper we sketch this direction as ongoing work.

2 Qualitative Simulation in ASP

Declarative languages are a natural fit for qualitative simulation. Kuipers' QSIM has already been encoded as an intuitive Prolog program using a state generator and predicates representing QDE constraints [Bratko, 2012]. QSIM models are purely constraint-based and, accordingly, qualitative simulation with QSIM has been encoded as a constraint satisfaction problem in ASP by Wiley and colleagues in the form of ASPQSIM [Wiley *et al.*, 2014].

A *simulation* is a sequence of qualitative states. Each state represents the configuration of the system at a point in time or between time points. Every qualitative quantity has a magnitude from its finite ordered quantity space and a derivative, which is either *steady* (0), *increasing* (▲), or *decreasing* (▼). Since both QDE-based and qualitative causal models constrain this same qualitative state space, our encoding reuses ASPQSIM's representation of quantity assignments and state transitions.

In ASPQSIM, qualitative models are represented by facts. A qualitative quantity is declared by a fact of the form $\texttt{qvar}(q)$, and its quantity space is given by landmarks and their adjacency relation. These declarations induce qualitative magnitudes, consisting of landmark values and intervals between adjacent landmarks. Together with the possible qualitative derivatives, an assignment of a quantity V at time T is represented by atoms of the form $\texttt{holds}(T, V, Q, D)$, where Q is a qualitative magnitude and D is a qualitative derivative.

$$\{\texttt{holds}(T,V,Q,D) : \texttt{qmag}(V,Q), \texttt{qdir}(D)\} = 1 \\ \leftarrow \texttt{qvar}(V), \texttt{time}(T). \quad (1)$$

$$\bot \leftarrow \texttt{holds}(T,Q_1,M_1,D_1), \texttt{holds}(T,Q_2,M_2,D_2), \\ \texttt{qde}(T,ID,\texttt{mplus}(Q_1,Q_2)), D_1 \neq D_2. \quad (2)$$

The choice rule in (1) generates exactly one qualitative magnitude and derivative for each variable at each considered time point. The integrity constraint then eliminates candidate states that violate an enforced model constraint. In the example shown, a monotonicity constraint $\texttt{mplus}(q_1, q_2)$ excludes states in which the derivatives of q_1 and q_2 do not match.

The stable models of the resulting ASP program, also called *answer sets*, represent admissible qualitative states or collections of states. For a horizon $k = 1$, each answer set corresponds to one admissible qualitative state, so enumerating all answer sets yields the qualitative state space. For larger values of k, answer sets may represent state transitions, paths, or more general state configurations, depending on the transition constraints imposed by the encoding.

3 Encoding Qualitative Causal Models

We extend the ASPQSIM representation from Section 2 to qualitative *causal* models following the simulation semantics of Garp3. We will refer to the system as ASPGARP. While QSIM models are purely constraint-based, qualitative causal models also contain *causal dependencies* that actively justify change. In particular, *influences* and *proportionalities* explain why a quantity is increasing or decreasing. Thus, in addition to eliminating inconsistent states, the encoding must ensure that every non-steady derivative is causally justified.

For an influence $\texttt{influence}(P, q_1, q_2)$, the sign P and the qualitative value of the source quantity q_1 determine the possible effect on the target quantity q_2. The encoding derives such effects as causal justifications for derivatives of the target. An integrity constraint then excludes states in which a quantity changes without any corresponding justification. Unlike in purely constraint-based QSIM, adding a causal dependency may therefore increase the set of admissible behaviours, because it can justify changes that would otherwise remain unsupported.

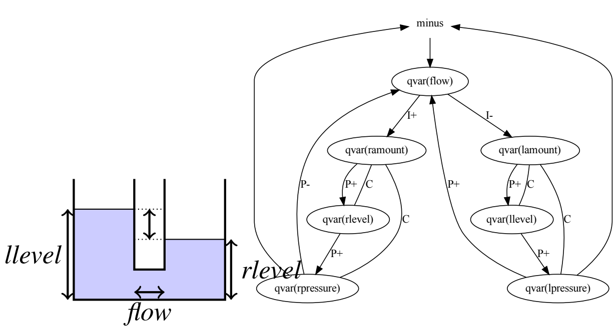

Figure 1: Communicating vessels example and corresponding causal-rule visualization.

For example see Figure 1, where a qualitative model of the *communicatign vessels* is presented using a clingraph visualisation [Hahn *et al.*, 2022], featuring influences (I), proporitionaliteis (P) and correspondeces (C).

Garp3 also reasons about qualitative relations between quantities, such as equality and order relations. We represent these relations by instantiating a qualitative constraint network over quantities and landmarks, following previous work on point-calculus reasoning in ASP [Bayerkuhnlein *et al.*, 2024]. The network contains relations explicitly present in the model, as well as relations induced by landmark orderings, correspondences, and value assignments. Each qualitative state is therefore accompanied by a constraint network that must be locally consistent.

These relational constraints are also used to enforce continuity. For example, a transition may not change a relation directly from $q_1 < q_2$ to $q_1 > q_2$ without passing through an intermediate state in which $q_1 = q_2$. The same relational representation is used to encode influence balances, where simultaneous influences may dominate, reinforce, or cancel one another depending on the qualitative relations between their sources.

Other model components, such as correspondences and algebraic constraints, are encoded in the same generate-and-test style as ASPQSIM constraints. Correspondences restrict co-occurring values of related quantities, while calculi such as $\texttt{minus}(q_1, q_2, q_3)$ eliminate states that violate the corresponding qualitative relation. When such components impose relations between quantities or landmarks, they are added to the qualitative constraint network and checked together with the remaining state description.

The encoding of ASPGARP is available in our companion repository[1].

3.1 Simulation

Given an initial state and a qualitative model simulation can reason about possible evolutions of a system. In terms of qualitative states.

Using the ASPGARP and multi-shot ASP solving we allow either the generation of traces over a finite horizon by reusing the encoding by Wiley [2014]. By sampling consistent states and checking whether ASPGARP accepts a transition between those states based on the provided model, we can construct a state-graph. Sampling only from states reachable by an initial state or satisfying some other operating region condition can be used to implement a custom state expander.

A full envisioning of the communicating vessels model from Figure 1, is shown in table 1. Where each state is represented by an answer set of ASPGARP. And Figure 2 where edges between states are represented by answer sets of ASPGARP where with the transition horizon $k = 2$ i.e. each answer set represents a pair of states with a transition relation.

3.2 Model Synthesis

Finally, we consider the use of the encoding for model synthesis. Conceptually, this inverts the simulation task. Instead of

[1] Repository: https://github.com/mob00/aspgarp

Table 1: Envisionment of the communicating vessels model. Rows show the qualitative values of quantities as magnitude–derivative tuples, where magnitudes are either landmark values or intervals between landmarks.

State	*flow*	*lamount*, *llevel*, *lpressure*	*ramount*, *rlevel*, *rpressure*
S_0	$\langle 0, 0\rangle$	$\langle (0, \mathit{max}), 0\rangle$	$\langle (0, \mathit{max}), 0\rangle$
S_1	$\langle 0, 0\rangle$	$\langle \mathit{max}, 0\rangle$	$\langle \mathit{max}, 0\rangle$
S_2	$\langle 0, 0\rangle$	$\langle 0, 0\rangle$	$\langle 0, 0\rangle$
S_3	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle (0, \mathit{max}), \blacktriangledown\rangle$	$\langle (0, \mathit{max}), \blacktriangle\rangle$
S_4	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle (0, \mathit{max}), \blacktriangledown\rangle$	$\langle 0, \blacktriangle\rangle$
S_5	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle \mathit{max}, \blacktriangledown\rangle$	$\langle 0, \blacktriangle\rangle$
S_6	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle \mathit{max}, \blacktriangledown\rangle$	$\langle (0, \mathit{max}), \blacktriangle\rangle$
S_7	$\langle (-\infty, 0), \blacktriangle\rangle$	$\langle (0, \mathit{max}), \blacktriangle\rangle$	$\langle (0, \mathit{max}), \blacktriangledown\rangle$
S_8	$\langle (-\infty, 0), \blacktriangle\rangle$	$\langle (0, \mathit{max}), \blacktriangle\rangle$	$\langle \mathit{max}, \blacktriangledown\rangle$
S_9	$\langle (-\infty, 0), \blacktriangle\rangle$	$\langle 0, \blacktriangle\rangle$	$\langle \mathit{max}, \blacktriangledown\rangle$
S_{10}	$\langle (-\infty, 0), \blacktriangle\rangle$	$\langle 0, \blacktriangle\rangle$	$\langle (0, \mathit{max}), \blacktriangledown\rangle$

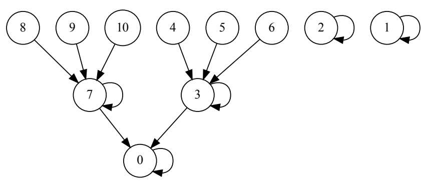

Figure 2: Clingraph visualizaton of the state-graph for Table 1

computing the admissible states of a given qualitative model, we search for a qualitative model whose simulations reproduce a given set of expected states.

We illustrate this with a small running example[2] over the quantities *growth* (g), *shade* (sh), and *size* (sz). Consider a tree growing in a garden. When its size increases, so does the shade it produces. The reference behaviour is given by the three qualitative states in Table 2. During synthesis, we assume that there are no hidden or unobservable variables. That is, all quantities used in the model are present in the reference states. We also assume that the reference specification is intended to be tight. States not included in the reference set may therefore be treated as negative examples when they are encountered during synthesis.

Table 2: Reference states for the running synthesis example.

State	*growth*	*shade*	*size*
S_0	$\langle 0, 0\rangle$	$\langle \mathit{large}, 0\rangle$	$\langle \mathit{large}, 0\rangle$
S_1	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle (\mathit{small}, \mathit{large}), \blacktriangle\rangle$	$\langle (\mathit{small}, \mathit{large}), \blacktriangle\rangle$
S_2	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle \mathit{small}, \blacktriangle\rangle$	$\langle \mathit{small}, \blacktriangle\rangle$

We formulate synthesis as a meta-level ILP task in ASP, where ASPGARP acts as an interpreter for candidate Garp models. Candidate model components are abduced from a hypothesis space. Causal dependencies, such as influences and proportionalities, are introduced to justify the observed changes, while constraint-like components, such as correspondences and calculi constraints, are introduced to exclude unintended states.

The synthesis procedure is organised in two phases. In the first phase, we search for causal dependencies that justify all non-steady derivatives occurring in the reference states. Since qualitative causal models require change to be causally explained, the synthesized influences and proportionalities must provide sufficient justification for the observed behaviour. This phase may yield several structurally different causal explanations.

[2] Online Demo: will be setup for Workshop

Table 3: Sampled negative states generated by a phase-one candidate.

State	*growth*	*shade*	*size*
C_0	$\langle (0, \infty), 0\rangle$	$\langle \mathit{large}, 0\rangle$	$\langle \mathit{small}, 0\rangle$
C_1	$\langle 0, 0\rangle$	$\langle \mathit{small}, 0\rangle$	$\langle \mathit{large}, 0\rangle$
C_2	$\langle 0, 0\rangle$	$\langle (\mathit{small}, \mathit{large}), 0\rangle$	$\langle \mathit{small}, 0\rangle$
C_3	$\langle (0, \infty), \blacktriangledown\rangle$	$\langle (\mathit{small}, \mathit{large}), \blacktriangle\rangle$	$\langle \mathit{small}, \blacktriangle\rangle$

Additional syntactic constraints are imposed to prune the model space, for example, models containing loops composed exclusively of proportionalities are infeasible.

The first synthesis phase may then return structurally different causal models such as $M_1 = \{\mathrm{I}^+(g, sh), \mathrm{P}^+(sh, sz), \mathrm{P}^-(sh, g)\}$. This model explains the increase of *shade* by a positive influence from *growth*, propagates the change from *shade* to *size*, and explains the decrease of *growth* by a negative proportionality from *shade*. Other candidates such as $M_2 = \{\mathrm{I}^+(g, sh), \mathrm{I}^+(g, sz), \mathrm{P}^-(sz, g)\}$ instead use two direct influences from *growth*.

A phase-one model may still admit additional states that are not part of the reference behaviour. In phase two, we sample some of these additional states as counterexamples. For the first model M_1, examples of such sampled states are shown in Table 3.

One model obtained after the refinement step is shown in Equation (3). In addition to causal dependencies, the model contains a correspondence between *shade* and *size*, as well as (directional) value correspondences between zero *growth* and large *size*. These components restrict the admissible state space and reject the sampled counterexamples.

$$M^* = \{\mathrm{C}^+(sh, sz), \mathrm{I}^+(g, sh), \mathrm{P}^-(sz, g), \mathrm{P}^+(sh, sz),$$
$$\mathrm{VC}(g, 0, sz, \mathit{large}), \mathrm{VC}(sz, \mathit{large}, g, 0)\}. \quad (3)$$

The resulting model admits the expected states, justifies their qualitative changes, and rejects the sampled states outside the intended behaviour. In this sense, the same ASP encoding supports both forward simulation and abductive model synthesis.

4 Conclusion

We presented an ASP encoding for qualitative causal models that supports simulation, meta-level reasoning, and model synthesis. The encoding makes Garp-style simulation semantics explicit and allows answer sets to represent qualitative states, transitions, or model candidates.

Future work will focus on scaling the approach to larger models and hypothesis spaces. More broadly, this work adds to the suite of ASP formalizations in the qualitative reasoning community.

References

[Bayerkuhnlein *et al.*, 2024] Moritz Bayerkuhnlein, Tobias Schwartz, and Diedrich Wolter. From resolving inconsistencies in qualitative constraints networks to identifying robust solutions: A universal encoding in asp. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 3–16. Springer, 2024.

[Bratko *et al.*, 1991] Ivan Bratko, Stephen Muggleton, and Alen Varšek. Learning qualitative models of dynamic systems. In *Machine Learning Proceedings 1991*, pages 385–388. Elsevier, 1991.

[Bratko, 2012] Ivan Bratko. *Prolog Programming for Artificial Intelligence*. Addison-Wesley, 4 edition, 2012.

[Bredeweg and Forbus, 2003] Bert Bredeweg and Kenne th D Forbus. Qualitative modeling in education. *AI magazine*, 24(4):35–35, 2003.

[Bredeweg *et al.*, 2009] Bert Bredeweg, Floris Linnebank, Anders Bouwer, and Jochem Liem. GARP3—workbench for qualitative modelling and simulation. 4(5-6):263–281, 2009.

[Coghill *et al.*, 2008] George M. Coghill, Ashwin Srinivasan, and Ross D. King. Qualitative system identification from imperfect data. *Journal of Artificial Intelligence Research*, 32:825–877, 2008.

[Das *et al.*, 2025] Ankita Das, Roxane Koitz-Hristov, and Franz Wotawa. Using qualitative simulation models for monitoring and diagnosis. In Marcos Quiñones-Grueiro, Gautam Biswas, and Ingo Pill, editors, *36th International Conference on Principles of Diagnosis and Resilient Systems, DX 2025, Nashville, TN, USA, September 22-24, 2025*, volume 136 of *OASIcs*, pages 4:1–4:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[Forbus, 1984] Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24(1–3):85–168, 1984.

[Forbus, 2025] Kenneth D Forbus. Exploring hybrid model formulation strategies for qualitative reasoning. *Proceedings of QR2025, Montreal, Canada*, 2025.

[Gebser *et al.*, 2011] Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Schneider. Potassco: The potsdam answer set solving collection. *Ai Communications*, 24(2):107–124, 2011.

[Hahn *et al.*, 2022] Susana Hahn, Orkunt Sabuncu, Torsten Schaub, and Tobias Stolzmann. Clingraph: Asp-based visualization. In *International conference on logic programming and nonmonotonic reasoning*, pages 401–414. Springer, 2022.

[Kuipers, 1986] Benjamin Kuipers. Qualitative simulation. *AI*, 20:289–338, 1986.

[Muggleton, 1991] Stephen Muggleton. Inductive logic programming. *New generation computing*, 8(4):295–318, 1991.

[Pang and Coghill, 2010] Wei Pang and George M. Coghill. Learning qualitative differential equation models: A survey of algorithms and applications. *The Knowledge Engineering Review*, 25(1):69–107, 2010.

[Ramachandran *et al.*, 1994] Sowmya Ramachandran, Raymond J Mooney, and Benjamin J Kuipers. Learning qualitative models for systems with multiple operating regions. In *Proceedings of the 8th International Workshop on Qualitative Reasoning about Physical Systems*, pages 212–223, 1994.

[Spitz *et al.*, 2021] Loek Spitz, Marco Kragten, and Bert Bredeweg. Learning domain knowledge and systems thinking using qualitative representations in secondary education (grade 8-9). In *Proceedings of the 34th International Workshop on Qualitative Reasoning*, 2021.

[Suzuki and Yoshioka, 2024] S Suzuki and M Yoshioka. Preliminary experiments of qualitative reasoning model construction using large language model. *Proceedings of QR2024, Santiago de Compostela, Spain*, 2024.

[Wiley *et al.*, 2014] Timothy Wiley, Claude Sammut, and Ivan Bratko. Qualitative simulation with answer set programming. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18–22 August 2014, Prague, Czech Republic*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 915–920. IOS Press, 2014.