monet-3

Scaling Up in Size and Knowledge

- **The Good News**:

  QSIM predicts all possible behaviors consistent with given qualitative and semi-quantitative knowledge.

- **The Bad News**:

  QSIM output can be large, even infinite.
  The problem is real, not spurious, behaviors.

- **The Good News**:

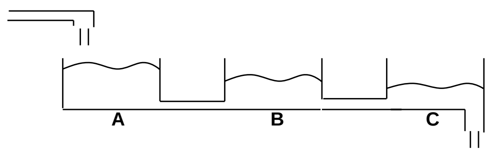
  There are solutions.

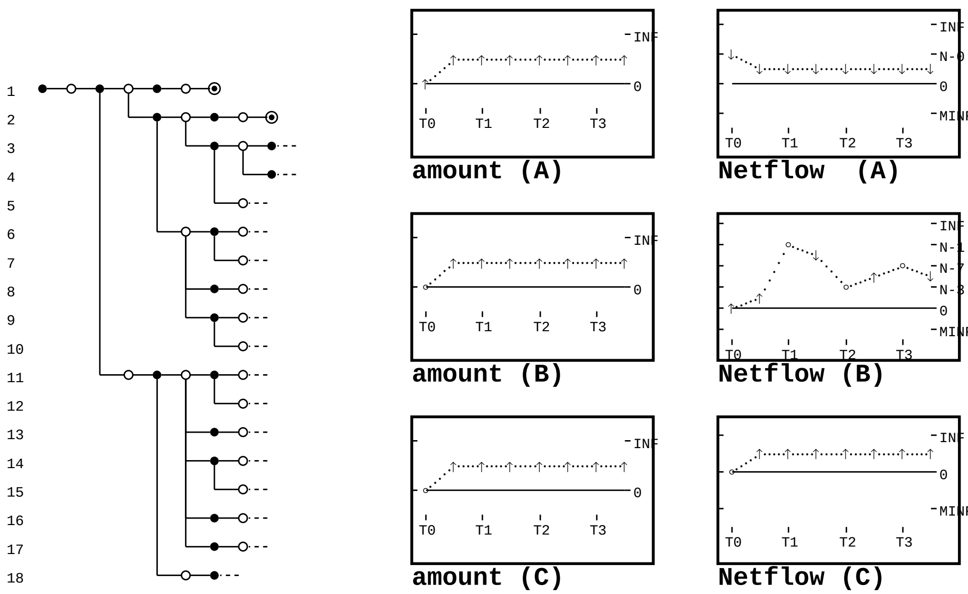
Four Solutions to Intractability

- **(1) Chatter Abstraction**: detect and abstract a region of unconstrained change to a single qualitative state.
- **(2) Model Decomposition**: use both state-based and history-based simulation to ignore irrelevant relationships.
- **(3) Temporal Logic Model-Checking**: use a theorem-prover to query the behavior tree.
- **(4) Temporal Constraints**: guide the simulator's attention to specified portions of the state space.

(1) The Problem of Chatter

Chatter occurs when a variable’s direction of change is unconstrained, except by continuity.

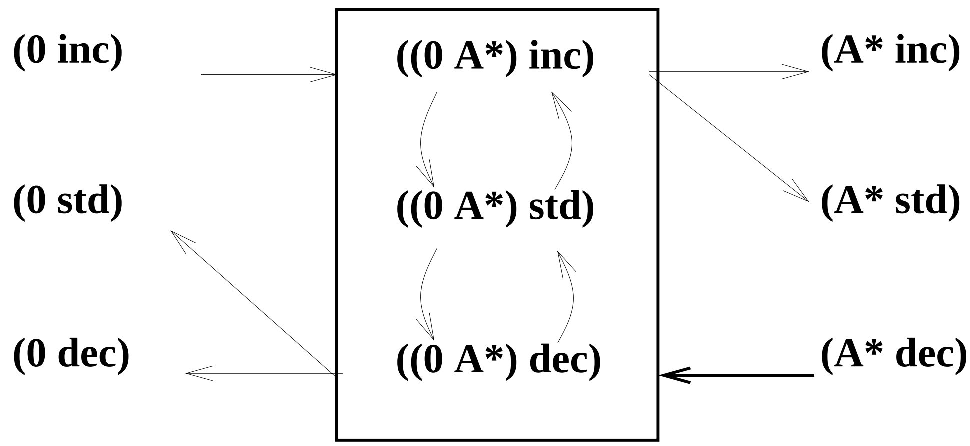




Chatter in one variable can propagate to others.

The Chatter Box

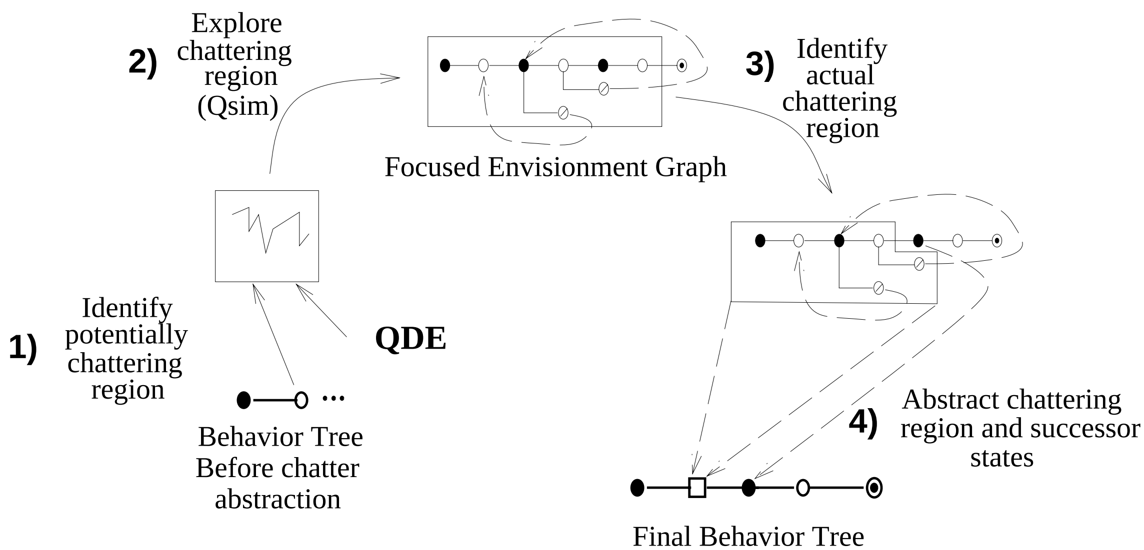
Qualitative behaviors are trajectories through state space.
Chatter is a property of a region of the state space.



Sometimes knowledge of higher-order derivatives can help.
Often not.

Chatter Box Abstraction

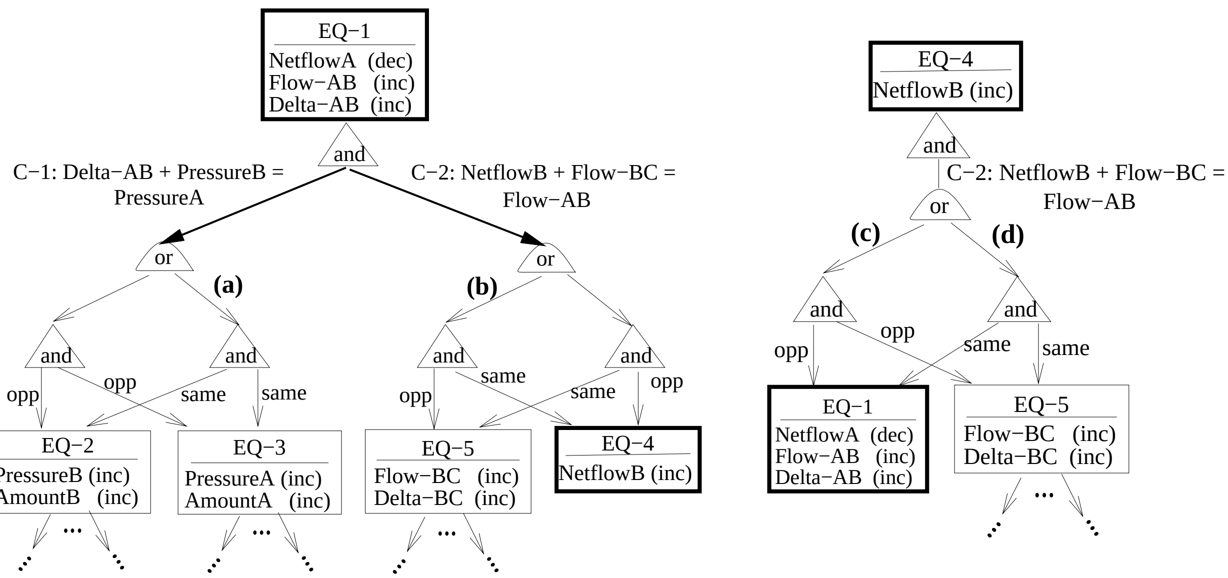
- Detect entry into a chatter box.
- Identify chattering variables and boundary values.
- Do focused envisionment to detect exits from chatter box.
- Replace envisionment with a single abstract state.



But: envisionment is still exponential in number of chattering variables.

Dynamic Chatter Abstraction

- Detect entry into chatter box.
- Create *chatter dependency graph*:



- Evaluate status of classes of chatter-equivalent variables.
- Simulate with abstracted qdirs. (Unique values indicate exit from chatter box.)

Analysis is complex, but the algorithm is efficient.

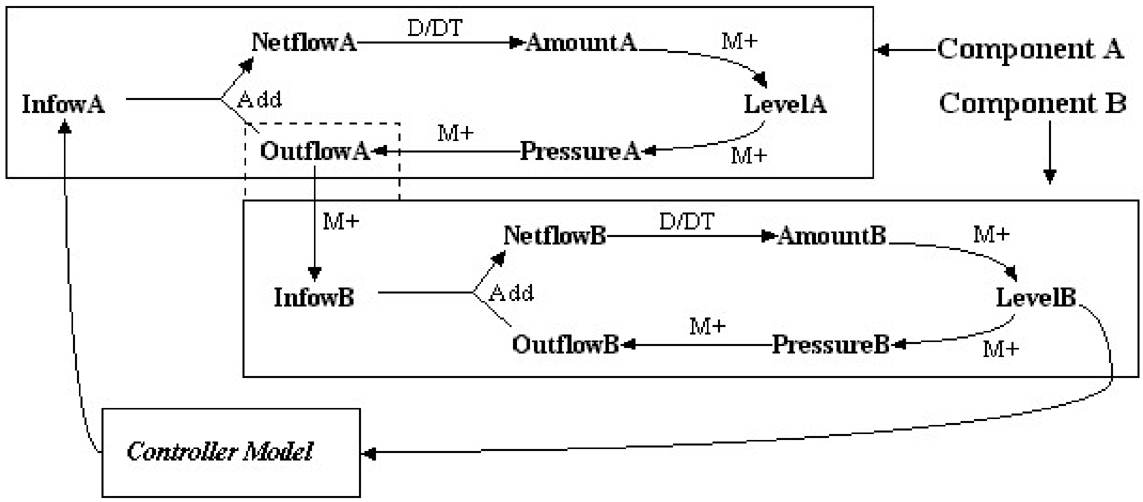
(2) Model Decomposition

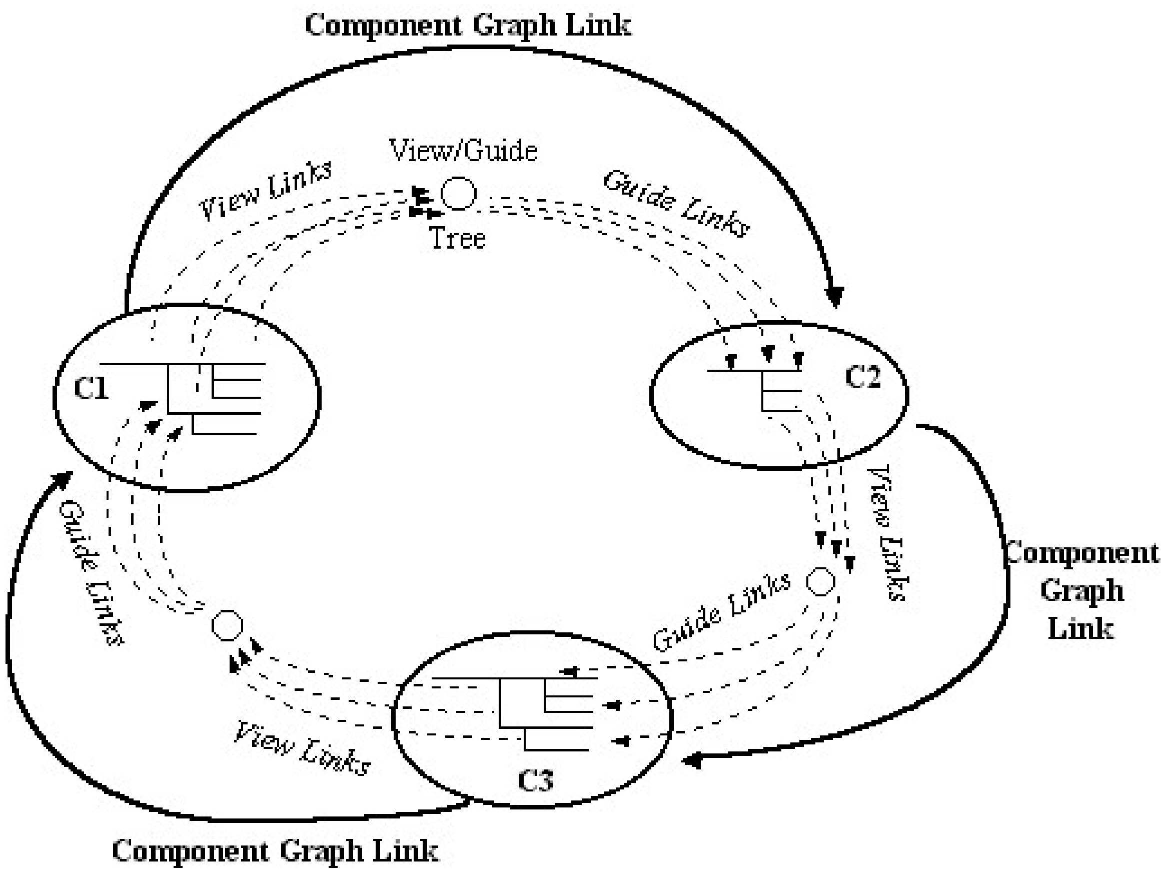
- **The Problem**:
  - Basic QSIM uses a global state representation.
  - Unrelated changes must be temporally ordered.
  - Branch on all possible orders.
- **The Solution**:
  - Decompose complex model into weakly-interacting components. [Simon, 1969]
  - Combine component behaviors into model behavior.

QSIM = Temporally-Extended CSP

- Compute *all* behaviors of each component.
  - State-based simulation within components.
  - Abstract values for unknown boundary variables.
  - Guided simulation for known boundary variable behavior.
- Each component behavior must belong to *some* global behavior.
  - History-based analysis between components.
  - Causal dependency among components controls simulation order: sequential or concurrent.
- Record dependencies among component behaviors.

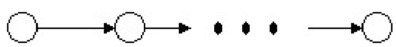
Example: Controlled Two-Tank Cascade



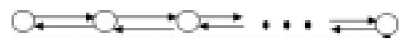


Efficiency Gains on N-Tank Systems

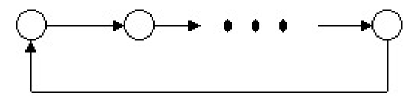
Different causal topologies:



Cascade

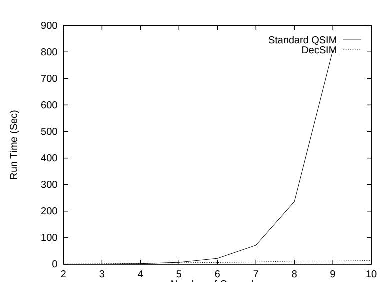


Chain

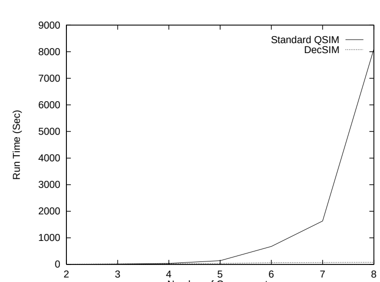


Loop

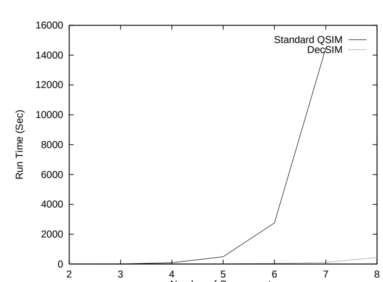
Number of Comp's	Cascade		Chain		Loop	
	QSIM	**DecSIM**	**QSIM**	**DecSIM**	**QSIM**	**DecSIM**
2	0.204	0.815	3.075	6.79	0.757	5.587
3	0.621	1.6	10.94	19.903	16.149	8.147
4	2.2	3.12	37.55	25.984	89.418	12.67
5	7.09	5.49	139.3	36.712	493.88	23.28
6	21.92	6.32	676	62.405	2758	48.73
7	71.59	8.39	1633	70	14474	116.1
8	236	11.67	8101	77	nc	442.4
9	806	11.75	nt	nt	nt	nt
10	nc	14.05	nt	nt	nt	nt



Cascade



Chain



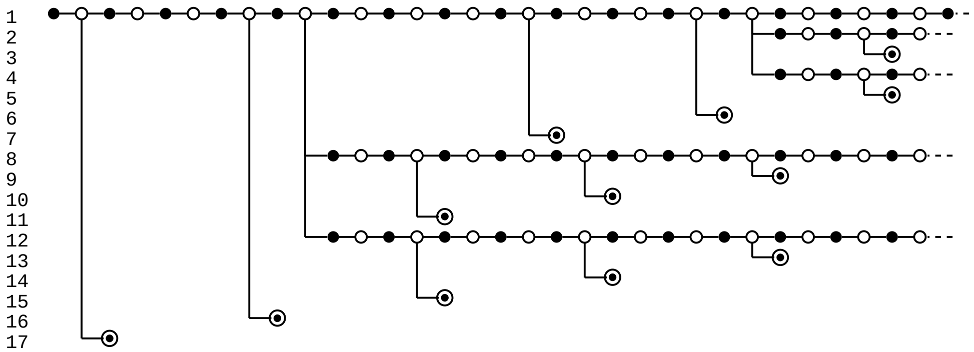
Loop

(3) Temporal Logic Model-Checking

- Temporal logic expresses what we want to know about the behaviors.
- The QSIM behavior tree can be viewed as a model for statements in a branching-time temporal logic.
- Model-checking determines whether a statement is true of the behavior tree.
  - Sound and complete.
- QSIM behavior tree predicts behaviors of dynamical systems.
  - Guaranteed coverage, but possible spurious behaviors. (Filtering is sound but incomplete.)

A universal statement can thus be proved by qualitative simulation.

Temporal Logic and Behavior Trees



modal	temporal	logical	QSIM
necessarily	*until*	*and*	*qval*
possibly	*next*	*or*	*status*
	eventually	*implies*	*funcall*
	always	*not*	

```
Simulating KJA PI controller.
Behavior tree rooted at S-0,
     with 1 initial states and 17 behaviors.

Some behaviors don't terminate...
Checking:  (EVENTUALLY (STATUS QUIESCENT)).
Validity = (NIL NIL T NIL T T T NIL T T T NIL T T T T T).

...but all that terminate have zero error.
Checking:  (NECESSARILY (ALWAYS (IMPLIES (STATUS QUIESCENT)
                                         (QVAL E (0 STD))))).
Validity = T.

Every fixed point is stable.
Checking:  (NECESSARILY (ALWAYS (IMPLIES (STATUS QUIESCENT)
                                         (STATUS STABLE)))).
Validity = T.
```

Technical Issues

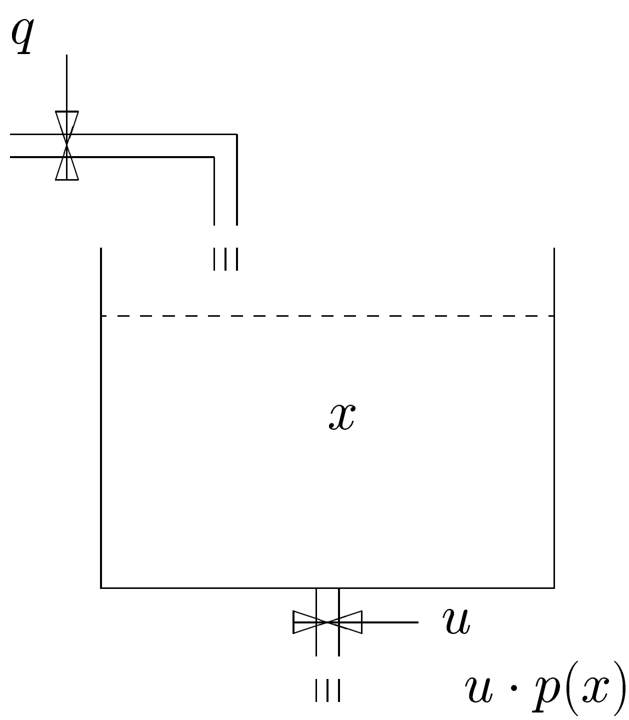
Matching assumptions between QSIM and Model-Checking.

- A behavior tree is *closed* when every behavior terminates with a quiescent state, a region transition, or a cycle.
- The QSIM Guaranteed Coverage theorem applies only to closed behavior trees.
- For effective model-checking, cycles in the behavior tree output by QSIM must be unwound one extra time.

The Main Theorem

- **Main Theorem**:
  - If Φ is a universal state formula in EBTL and M is a closed tree and (TL M Φ) returns T, then Φ' is true of every real function consistent with the QDE.
- **Lemma**: the QSIM Guaranteed Coverage Theorem
  - If QSIM returns a closed tree then QSIM predicts every real function consistent with the QDE input.
- **Caveats**:
  - If M is not closed, some real behaviors may not be predicted (yet).
  - If Φ is not universal, the model of Φ' could be a spurious behavior.

Level Control of the Water Tank

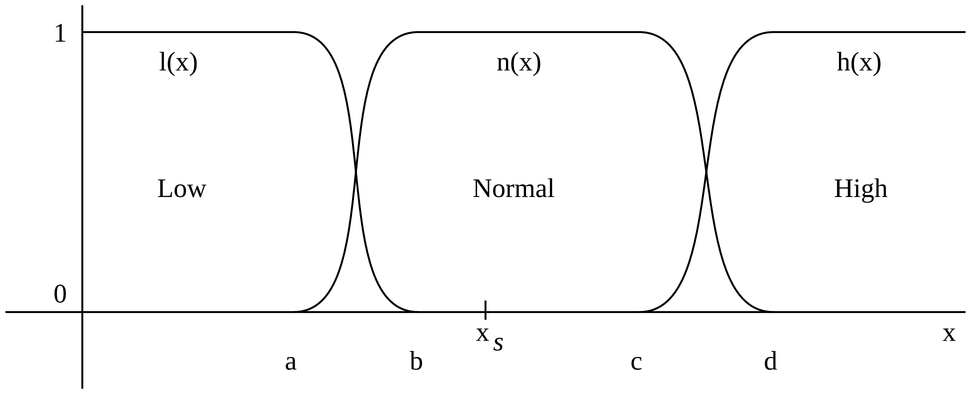


$$\dot{x} = f(x,u) = q - u \cdot p(x).$$

$$\begin{aligned} x &= \text{amount in tank} \\ q &= \text{inflow into tank} \\ u &= \text{drain area} \\ p(x) &= \text{influence of pressure at drain} \end{aligned}$$

A Heterogeneous Controller

The operating regions and their appropriateness measures:



The local control laws:

$$
\begin{aligned}
x \in Low \;&\Rightarrow\; u_l(x) = 0 \\
x \in Normal \;&\Rightarrow\; u_n(x) = k(x - x_s) + u_s \\
x \in High \;&\Rightarrow\; u_h(x) = u_{max}
\end{aligned}
$$

The global control law:

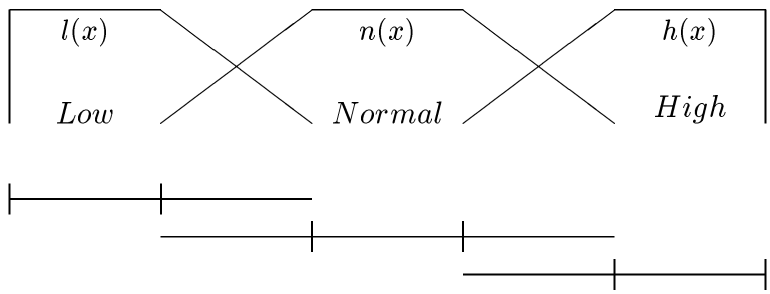
$$u(x) = l(x)u_l(x) + n(x)u_n(x) + h(x)u_h(x).$$

The discrete abstraction:

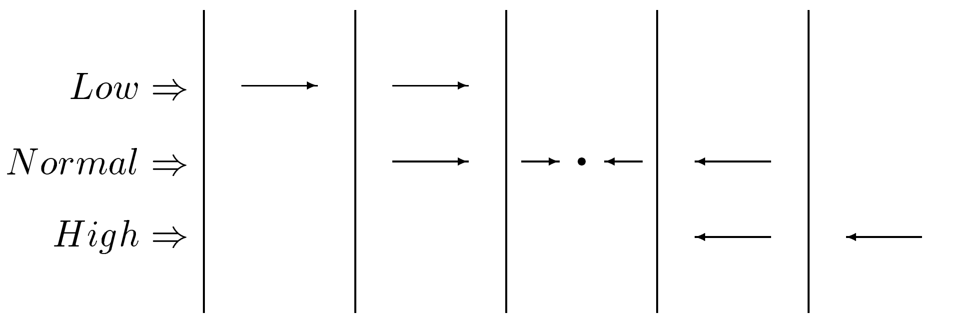
$$\boxed{\text{Low}} \longrightarrow \boxed{\boxed{\text{Normal}}} \longleftarrow \boxed{\text{High}}.$$

Qualitative Combination of Behaviors

- Overlapping operating regions for the local laws.



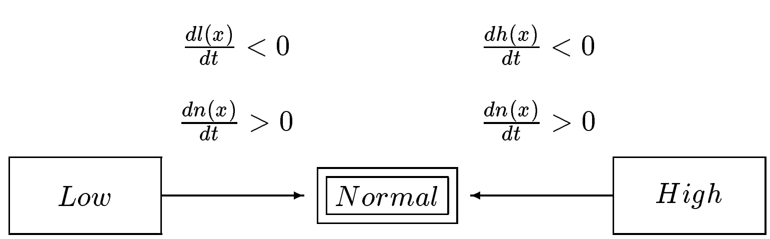
- Require qualitative agreement where laws overlap.



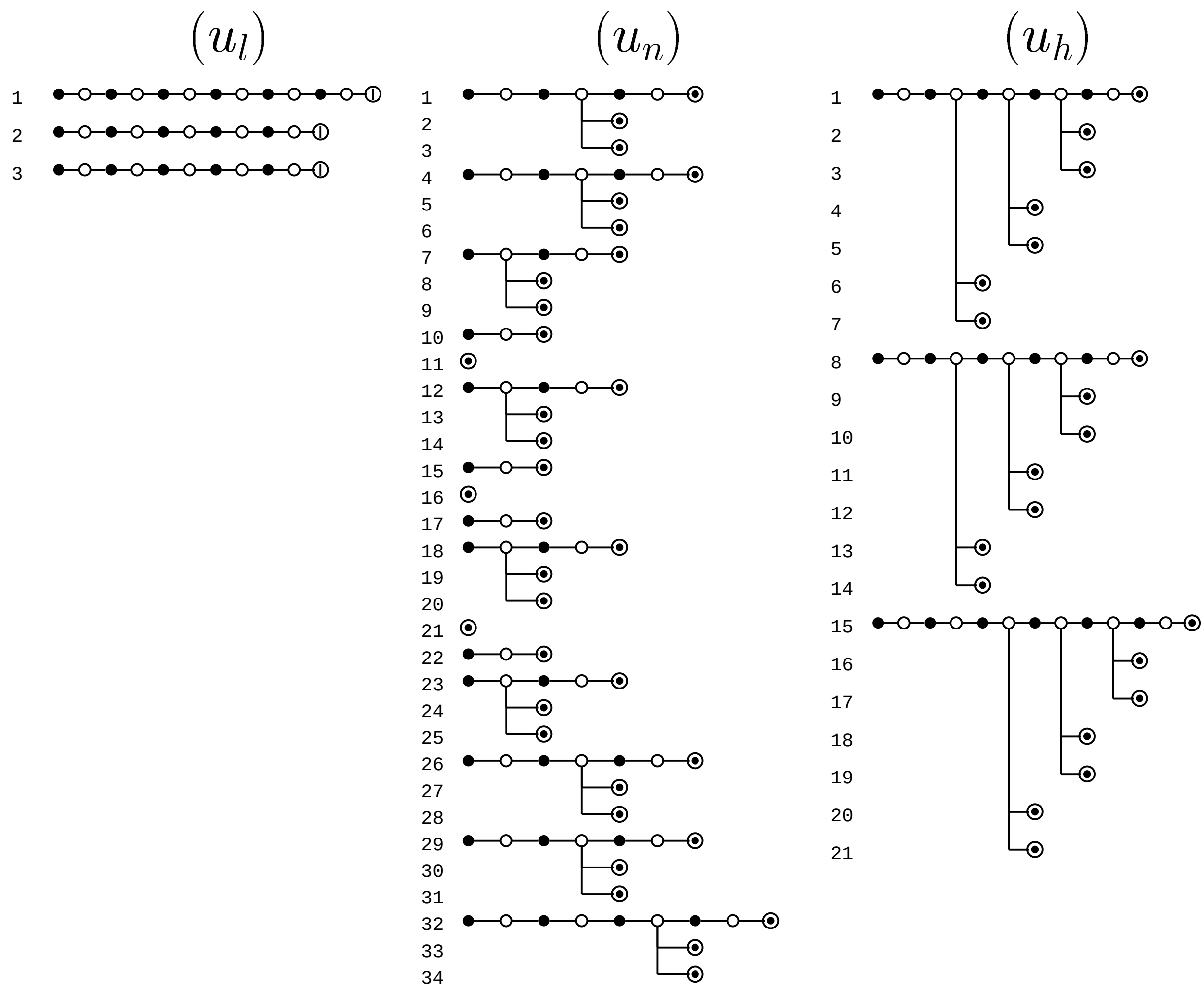
- Guarantee monotonic behavior in overlap regions.

$$
\begin{aligned}
Low &\Rightarrow q > 0 \\
Normal &\Rightarrow q_b < q < q_c \\
High &\Rightarrow q < u_{max} \cdot p(c)
\end{aligned}
$$

- Abstract the control law to a finite transition diagram.



Behavior Trees for Local Control Laws



Validity of Qualitative Properties

```
Simulating controller U_l.
Behavior tree rooted at S-0, with 3 initial states and 3 behaviors.
Checking UPWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((NIL B) NIL
                                                       (QVAL X (NIL INC))))
Validity at S-0 = T.
Checking DESTINATION:  (NECESSARILY (EVENTUALLY (QVAL X ((B C) NIL)))).
Validity at S-0 = T.

Simulating controller U_n.
Behavior tree rooted at S-40, with 16 initial states and 34 behaviors.
Checking UPWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((NIL B) NIL
                                                       (QVAL X (NIL INC))))
Validity at S-40 = T.
Checking DOWNWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((C NIL) NI
                                                         (QVAL X (NIL DEC))
Validity at S-40 = T.
Checking DESTINATION:  (NECESSARILY (EVENTUALLY (QVAL X ((B C) NIL)))).
Validity at S-40 = T.
Checking STABILITY:  (NECESSARILY (EVENTUALLY (AND (QVAL X ((B C) STD))
                                                   (STATUS QUIESCENT)
                                                   (STATUS STABLE)))).
Validity at S-40 = T.

Simulating controller U_h.
Behavior tree rooted at S-167, with 3 initial states and 21 behaviors.
Checking DOWNWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((C NIL) NI
                                                         (QVAL X (NIL DEC))
Validity at S-167 = T.
Checking DESTINATION:  (NECESSARILY (EVENTUALLY (QVAL X ((B C) NIL)))).
Validity at S-167 = T.
```

(4) Temporal Constraints: TeQSIM

Temporal logic lets the modeler use knowledge not expressible in the QDE or SQDE.

- Trajectory constraints describe intended behaviors.
  - Time-varying exogenous variables.
  - Events and discontinuous changes.
  - Semi-quantitative bounds on behaviors.
- Interleave QSIM with the temporal logic model-checker. Accept only behaviors consistent with TL constraints.
- Focus attention on subset of behavior space.

TeQSIM Examples

- Specify exogenous input.

```
(event step-up :time (2 3))
(event step-down :time (17 24))
(disc-change (event step-up)
             ((inflow (normal high)
                      :range (1500 1800))
(disc-change (event step-down)
             ((inflow normal)))
```

- Focus on overflow scenario.

```
(event open)
(disc-change (event open)
             ((valve (normal max)
                     :range (4 nil))))
(before (qvalue level (top nil)) (event o
(eventually (qvalue level (top nil)))
```

Derive temporal bounds on (event open) to prevent overflow.

More Information:
http://www.cs.utexas.edu/users/qr

Qualitative Reasoning

- Daniel J. Clancy and Benjamin J. Kuipers. 1998. Qualitative simulation as a temporally-extended constraint satisfaction problem. *AAAI-98*.

- Daniel J. Clancy and Benjamin Kuipers. 1997. Model decomposition and simulation: a component based qualitative simulation algorithm. *AAAI-97*.

- Daniel J. Clancy and Benjamin Kuipers. 1997. Static and dynamic abstraction solves the problem of chatter in qualitative simulation. *AAAI-97*.

- Giorgio Brajnik and Daniel J. Clancy. 1996. Trajectory constraints in qualitative simulation. *AAAI-96*.

- Benjamin Shults and Benjamin Kuipers. 1997. Proving properties of continuous systems: qualitative simulation and temporal logic. *Artificial Intelligence Journal* **92**: 91–129.

- Benjamin J. Kuipers and Karl J. Åström. 1994. The composition and validation of heterogeneous control laws. *Automatica* **30**(2), February 1994.