LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TM-274

QUALITATIVE SIMULATION OF MECHANISMS

Benjamin Kuipers

April 1985

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

Qualitative Simulation of Mechanisms[1]

Benjamin Kuipers
Clinical Decision Making Group,
MIT Laboratory for Computer Science.

March 18, 1985

Abstract:

Qualitative simulation is a key inference process in qualitative causal reasoning. However, the precise meaning of the different proposals and their relation with differential equations is often unclear. In this paper, we present a precise definition of qualitative structure and behavior descriptions as abstractions of differential equations and continuously differentiable functions. We present a new algorithm for qualitative simulation that generalizes the best features of existing algorithms, and allows direct comparisons among alternate approaches. Starting with a structural description abstracted from a differential equation, we prove that the QSIM algorithm is guaranteed to produce a qualitative behavior corresponding to any solution to the original equation. We also show that any qualitative simulation algorithm, because of its local point of view, will sometimes produce spurious qualitative behaviors: ones which do not correspond to any mechanism satisfying the structural description. These observations suggest specific types of care that must be taken in designing applications of qualitative causal reasoning systems, and in constructing and validating a knowledge base of mechanism descriptions.

Keywords: qualitative simulation, qualitative mathematics, naive physics.

---

[1]This research was supported in part by the National Library of Medicine through NIH Grants LM 03603, LM 04125, and R01 LM 04374, and by the National Science Foundation through grants MCS-8303640 and DCR-8417934.

Contents

1 Introduction

An expert system is often a "shallow model" of its application domain, in the sense that conclusions are drawn directly from observable features of the presented situation. Researchers have long felt that genuinely expert performance must also rest on knowledge of "deep models," in which an underlying mechanism, whose state variables may be not be directly observable, accounts for the observable facts [Gentner and Stevens, 1983].

One major line of research toward the representation of deep models is the study of qualitative causal models [de Kleer, 1977, 1979; de Kleer and Bobrow, 1984; de Kleer and Brown, 1983, 1984; Forbus, 1981, 1982, 1983, 1984; Hayes, 1979; Kuipers, 1982, 1984; Kuipers and Kassirer, 1983, 1984; Williams, 1984a, 1984b]. Research on qualitative causal models differs from more general work on deep models in focussing on qualitative descriptions of the deep mechanism, capable of representing incomplete knowledge of the structure and behavior of the mechanism. Symbolic manipulation of qualitative descriptions also appears to be a plausible model of human expertise [Kuipers and Kassirer, 1983, 1984].

Qualitative causal reasoning consists of a number of different operations, ranging from the initial formulation of the problem, to prediction of possible behaviors, to explanations of observations. These operations vary considerably from one problem domain to another, ranging from engineered electronic devices with a carefully designed correspondence between physical component and functional behavior [deKleer, 1977; Williams, 1984b], to commonsense predictions of the behavior of physical objects under the influence of active processes [Forbus, 1982, 1984], to the behavior of physiological mechanisms whose relation to the physical organ is only partially understood [Kuipers and Kassirer, 1983, 1984]. Forbus (1982, 1984), in particular, has developed a sophisticated Qualitative Process Theory, which proposes that commonsense reasoning about mechanisms is based on determining the *processes* that are active in a given physical situation at a particular time. The processes then yield the state variables and constraints, from which the behavior of the mechanism is derived.

A central inference within all of these approaches is qualitative simulation: derivation of a description of the behavior of a mechanism from

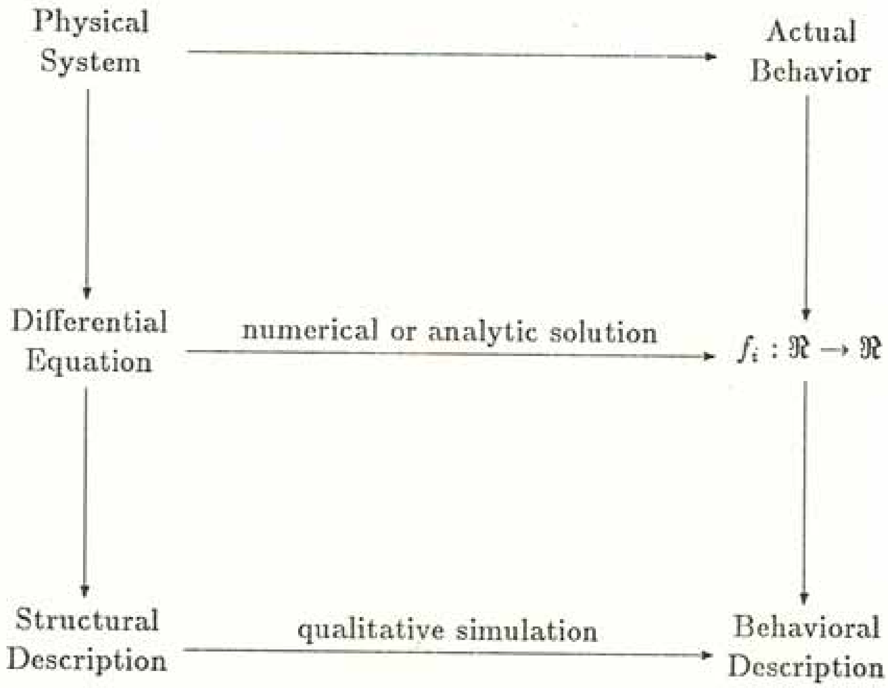

Figure 1: Qualitative simulation and differential equations are both abstractions of actual behavior.

a qualitative description of its structure. Differential equations provide a useful analogy (figure 1). A differential equation describes a physical system in terms of a set of state variables and constraints. The solution to the equation may be a function representing the behavior of the system over time. The qualitative structural description is a further abstraction of the same system, and qualitative simulation is intended to yield a corresponding abstraction of its behavior. This paper formalizes and investigates that relationship.

A theory and algorithm for qualitative reasoning must address several issues, which provides a framework for comparing the proposals of different researchers, and the contribution of this paper.

- how quantities are described qualitatively,

- how state transitions are selected,
- whether quantities correspond to standard mathematical analysis,
- whether qualitative simulation produces all and only valid behaviors.

All qualitative simulation systems describe quantities in terms of their ordinal relations with a small set of landmark values. De Kleer, Bobrow, and Brown [de Kleer and Brown, 1984; de Kleer and Bobrow, 1984] and Williams [1984a, 1984b] take the only landmark to be zero, and thus define three *qualitative values*, $\{+, 0, -\}$, and define addition and multiplication as operations over the qualitative values. Forbus (1982, 1984) and Kuipers (1984) define a *quantity space* as a partially ordered set of landmark values, so that a quantity is described in terms of its ordinal relations with the landmarks. The Kuipers (1984) approach is different from the others in allowing new landmarks to be discovered during the qualitative simulation, and used to define new qualitative distinctions. The QSIM algorithm presented here describes quantities in terms of a *linearly ordered* set of landmarks, but still allowing new landmarks to be discovered and inserted. We demonstrate below that without discovering and using new landmark values, important qualitative distinctions can be missed, such as the distinction between increasing, decreasing, and stable oscillation.

Different qualitative simulation systems take different positions on whether quantities should be an abstraction of the *standard* mathematical notion of real numbers — in which case $\Re$ is described as an alternating sequence of points and open intervals — or whether a non-standard model should be used, allowing two points to be infinitesimally separated. Forbus (1982, 1984) and de Kleer [de Kleer and Brown, 1984] adopt a non-standard model in which "mythical time" separates qualitative states that correspond to the same physical point in time. Such mythical time points appear to be required when a propagation step must run more than once to generate a state corresponding to the next physical state. De Kleer and Bobrow (1984) adopt the standard model for quantities, but appear less committed to alternating points and intervals in the time domain. Kuipers (1984) and Williams (1984a, 1984b) follow the standard model. As Williams' work and this paper demonstrate, the standard model makes it possible to state and prove useful theorems about the validity of the predictions made by qualitative simulation.

All qualitative simulation systems produce the set of possible behaviors by generating, then filtering, possible transitions between pairs of qualitative state descriptions. Most systems simulate forward, by generating all possible successors of the current state; de Kleer [de Kleer and Brown, 1984; de Kleer and Bobrow, 1984] generates all possible qualitative states, then determines the valid transitions among them. De Kleer's approach can only succeed if there is a fixed set of qualitative values, so that the set of possible states can be generated in advance. In both cases, the filtering criteria are local: they depend on the quantities in the two state descriptions, and on the structural constraints in the system.

An important class of filtering criteria are *transition ordering* rules [de Kleer and Bobrow, 1984; Kuipers, 1984; Williams, 1984a]. For example, if $A + B = C$ with $A, B, C > 0$, and B and C are approaching zero, then B must reach zero before C. A large number of these rules can be formulated, corresponding to different signs, directions of approach, and combinations of quantities approaching limits. In designing a system, it is difficult to be sure that all possible such rules have been captured; in implementing it, it is difficult to check that they have been written correctly. As described in Appendix B, all of the transition ordering rules can be recognized as special cases of a simple test of valid relationships between the current values of a set of quantities and a set of corresponding values. These tests, applying to the ADD, $MULT$, M^+, and M^- constraints, capture all single-constraint transition-ordering criteria of this type, can be implemented efficiently, and most importantly, can be straight-forwardly proven correct.

All qualitative simulation systems predict multiple possible behaviors given certain structural descriptions and initial conditions. Researchers in this area (myself included) have hoped to prove that the predicted behaviors include all and only the possible behaviors of real mechanisms satisfying the given description. Half of this is correct: we prove below that qualitative simulation cannot miss any actual behavior. However, because of the local nature of its decision criteria, qualitative simulation *can* predict behaviors that are not possible for any real mechanism satisfying the given description, and we construct a counterexample. We discuss the implications of these results for the construction of a qualitative causal reasoning system.

Qualitative simulation systems vary widely in speed [de Kleer, personal communication; Forbus, personal communication]. In order to be useful as part of an expert problem-solver, a qualitative simulation system must be efficient. The QSIM algorithm is very fast. Furthermore, experiments with semantic variants (e.g. the $\{+, 0, -\}$ semantics) can be made easily by changing the entries in Table 3. It has been implemented in Lisp on the Symbolics 3600, and all examples in this paper have been run, as well as numerous others in elementary physics and in nephrology.

1.1 Overview

This section provides an overview of qualitative simulation and the QSIM algorithm. The concepts presented here are defined more formally below.

Qualitative simulation of a system starts with a description of the known structure of the system, and an initial state, and produces a tree consisting of the possible future states of the system. The possible behaviors of the system are the paths from the root of this tree to its leaves. Table 1 gives an informal description of the structure and unbranching behavior a simple system.

The **structural description** consists of a set of symbols representing the **physical parameters** of the system (continuously differentiable real-valued functions), and a set of **constraints** on how those parameters may be related to each other. The constraints are two- or three-place relations on physical parameters. Some specify familiar mathematical relationships: $DERIV(vel, acc)$, $ADD(net, out, in)$,
$MULT(mass, acc, force)$, $MINUS(fwd, rev)$. Others assert qualitatively that there is a functional relationship between two physical parameters, but only specify that the relationship is monotonically increasing or decreasing: $M^+(price, power)$ and $M^-(mph, mpg)$. The constraints are designed to permit a large class of differential equations to be mapped straight-forwardly into structural descriptions.

Each physical parameter is a continuously differentiable real-valued function of time. Its value at any given point in time is specified qual-

Table 1: Informal structure and behavior of the "Ball" system.
A ball is thrown upward and falls to the ground under constant gravity. Continuous quantities are described in qualitative terms, with no more commitment to numerical detail than absolutely necessary.

Structure

- There is a constant downward acceleration from gravity.
- Acceleration is the derivative of velocity.
- Velocity is the derivative of height.
- Initially, we have zero height and positive upward velocity.

Behavior

- The ball is initially at zero height and positive upward velocity.
- The ball is at positive height and rising; velocity is positive but decreasing.
- The ball's velocity becomes zero while the ball is at some positive height.
- The ball is still at positive but decreasing height, while velocity is negative and decreasing away from zero.
- The ball returns to zero height while at a negative velocity.
- Unless we have included an arbitrary range restriction, height and velocity both become negative and decreasing away from zero, forever.

Qualitative simulation determines the essentially different regions of the system's behavior. It need not be given the acceleration due to gravity nor the initial upward velocity. It does not determine the height to which the ball rises, nor its velocity when it returns to the ground. It does guarantee that the ball rises, then falls back to the ground with non-zero velocity.

itatively, in terms of its relationship with a totally ordered set of **landmark values**. The landmark values may be either numerical (e.g. zero) or symbolic; their ordinal relationships are their essential properties. As the qualitative simulation proceeds, it can discover and add new landmark values to the sequence. The **qualitative state** of a parameter consists of its ordinal relations with the landmark values and its direction of change.

Time, similarly, is represented as a totally ordered set of symbolic **distinguished time-points**. The current time is either at or between distinguished time-points. All of the time-points are generated as a result of the qualitative simulation process.

At a distinguished time-point, if several physical parameters linked by a single constraint are equal to landmark values, they are said to have **corresponding values** which can be discovered and used by the qualitative simulation. The special case of a monotonic function constraint with corresponding values $(0,0)$ is sufficiently common that it is signified by the constraints M_0^+ and M_0^-.

A set of constraints on the physical parameters of the system is only valid in some **operating region**, defined by the legal ranges of values that some parameters may take on. The **legal range** of a parameter is a closed interval whose endpoints are landmark values of that parameter. These endpoints may be associated with transitions to other operating regions where a different set of constraints apply. The operating regions are designed as an interface to Forbus' (1982, 1984) concept of processes, but that topic is beyond the scope of this paper.

The **initial state** of the system is defined by the operating region and a set of qualitative values for the physical parameters. The qualitative simulation proceeds by determining all of the possible changes in qualitative value permitted to each parameter, then filtering the combinations by applying progressively broader constraints. If more than one qualitative change is possible, the current state has multiple successors, and the simulation produces a tree.

Two qualitative states in the same operating region are **identical** if all parameters are equal to the same landmark values, and all the directions of change are the same. If one of the successors to a given state is identical to a direct predecessor, a cyclic behavior can be created.

Table 2 presents the formal structure and behavior descriptions of the Ball system, for comparison with Table 1. In the next sections we define the formal notation necessary to state the QSIM algorithm and prove its validity.

Table 2: Formal structure and behavior of the Ball system.
The Ball system is formally described in terms of three physical parameters, Y, V, and A.

Structure

$$DERIV(Y,V)$$
$$DERIV(V,A)$$
$$Y(t_0) = 0$$
$$V(t_0) = V^* > 0$$
$$\forall t.A(t) = g < 0.$$

Behavior

	Y	V	A
t_0	$\langle 0, inc\rangle$	$\langle V^*, dec\rangle$	$\langle g, std\rangle$
(t_0, t_1)	$\langle (0,\infty), inc\rangle$	$\langle (0, V^*), dec\rangle$	$\langle g, std\rangle$
t_1	$\langle Y_{max}, std\rangle$	$\langle 0, dec\rangle$	$\langle g, std\rangle$
(t_1, t_2)	$\langle (0, Y_{max}), dec\rangle$	$\langle (-\infty, 0), dec\rangle$	$\langle g, std\rangle$
t_2	$\langle 0, dec\rangle$	$\langle (-\infty, 0), dec\rangle$	$\langle g, std\rangle$
(t_2, ∞)	$\langle (-\infty, 0), dec\rangle$	$\langle (-\infty, 0), dec\rangle$	$\langle g, std\rangle$

QSIM also produces "qualitative graphs" to represent the behavioral description.

```
INF                                              INF

Y1                        =                      Y1
                   ^            v
0          ----^------------------v-----         0
                                       v
MINF                                             MINF
           Y
```

2 Qualitative Behavior

In the following sections, we present a more rigorous definition of qualitative simulation, leading up to a definition of the QSIM algorithm and the proof of several theorems characterizing its strengths and limitations. The validity proofs for several steps of the algorithm are contained in the appendices.

A physical system is characterized by a number of real-valued parameters, which vary continuously over time. We consider each physical parameter to be a function $f : [a,b] \to \Re^*$, where $\Re^* = [-\infty, \infty]$, the extended real number line. The domain and range of a function f are both closed intervals in the extended reals, $\Re^*$. We use $\Re^*$ instead of $\Re$, treating ∞ as a genuine landmark value, because it is useful (though not essential) to have the invariant that t and $f(t)$ are always bounded by explicitly stated landmark values in the domain and range of f. The function $f : [0,\infty] \to \Re^*$ is defined to be continuous at ∞ exactly if $\lim_{t\to\infty} f(t)$ exists. For example, both e^{-t} and e^t are continuously differentiable on $[0,\infty]$, but $\sin t$ is not.

2.1 Behavior of a Single Function

We will define the qualitative behavior description first for a single, continuously differentiable function $f : [a,b] \to \Re^*$.

Definition 1 *For* $[a,b] \subseteq \Re^*$, *define* $f : [a,b] \to \Re^*$ *to be a* **reasonable function** *if*

- f *is continuous on* $[a,b]$,
- f *is continuously differentiable on* (a,b),
- f *has only finitely many critical points on* (a,b),
- $\lim_{t\searrow a} f'(t)$ *and* $\lim_{t\nearrow b} f'(t)$ *exist in* $\Re^*$. *Define* $f'(a)$ *and* $f'(b)$ *to be equal to these limits.*

The restriction to finitely many critical points excludes examples like $f(t) = t^2 * \sin 1/t$ that are continuously differentiable, but whose behavior changes infinitely quickly around $t = 0$. Without the fourth restriction, $f'(t)$ can still behave pathologically around the endpoints of the interval, even without crossing zero.

Periodic behavior like an oscillating spring can be accomodated with only finitely many critical points by explicitly detecting the repeated state and creating a cyclical behavior description. Then the domain of f need only contain one period.

Definition 2 *Every reasonable function $f : [a, b] \rightarrow \Re^*$ has associated with it a finite set of* **landmark values**, *including, but not limited to, 0, $f(a)$, $f(b)$, and the value of $f(t)$ at each of its critical points.*

Definition 3 *Where f is a reasonable function, $t \in [a, b]$ is a* **distinguished time-point** *of f if t is a boundary element of the set $\{t \in [a, b] \mid f(t) = x$, where x is a landmark value of $f\}$.*

That is, the distinguished time-points are those points where f passes its landmarks, even if its derivative is not zero at the time. The restriction to boundary elements handles the case where f becomes constant over an interval: only the endpoints of the interval are distinguished time-points. De Kleer and Bobrow (1984) eliminate this case by assuming that parameters have derivatives of all orders, in which case any function which is constant over an interval is constant everywhere.

All functions mentioned below should be presumed reasonable unless specified otherwise. A reasonable function $f : [a, b] \rightarrow \Re^*$ has the finite set of distinguished time-points:

$$a = t_0 < t_1 < \cdots < t_n = b,$$

and the finite set of landmark values:

$$l_1 < l_2 < \cdots < l_k.$$

We can now define the qualitative state of f at t in terms of its ordinal relations with its landmarks, and its direction of change.

We reluctantly contribute to the proliferation of notations for qualitative description of continuous functions. The advantages of the notation used here are that it (1) naturally allows for an arbitrary and changing set of landmark values, (2) uses a single term for the qualitative description of a function's magnitude and derivative, and (3) emphasizes that the qualitative description of the derivative is of low and fixed resolution, while qualitative description of magnitude is of higher and possibly changing resolution.

Definition 4 *Let $l_1 < \cdots < l_k$ be the landmark values of $f : [a,b] \to \Re^*$. For any $t \in [a,b]$, $QS(f,t)$, the* **qualitative state of f at t**, *is a pair $\langle qval, qdir \rangle$, defined as follows:*

1.

$$qval = \begin{cases} l_j & \text{if } f(t) = l_j, \text{ a landmark value} \\ (l_j, l_{j+1}) & \text{if } f(t) \in (l_j, l_{j+1}) \end{cases}$$

2.

$$qdir = \begin{cases} inc & \text{if } f'(t) > 0 \\ std & \text{if } f'(t) = 0 \\ dec & \text{if } f'(t) < 0. \end{cases}$$

Proposition 1 *Where $a = t_0 < \cdots < t_n = b$ are the distinguished time-points of f, consider $s, t \in (a,b)$ such that $t_i < s < t < t_{i+1}$ for some i. Then $QS(f,s) = QS(f,t)$.*

Proof: By the Intermediate Value Theorem, since f is continuously differentiable, $f(t)$ cannot pass a landmark value, and $f'(t)$ cannot change signs between adjacent distinguished time-points. []

This justifies our basic intuition that the qualitative state of the function is constant over intervals between landmarks. Hence, we may make the following definitions.

Definition 5 *For adjacent distinguished time-points* t_i *and* t_{i+1}, *define* $QS(f, t_i, t_{i+1})$, *the* **qualitative state of** f **on** (t_i, t_{i+1}), *to be* $QS(f, t)$ *for any* $t \in (t_i, t_{i+1})$.

Definition 6 *The* **qualitative behavior** *of* f *on* $[a, b]$ *is the sequence of qualitative states of* f*:*

$$QS(f, t_0), QS(f, t_0, t_1), QS(f, t_1) \ldots QS(f, t_{n-1}, t_n), QS(f, t_n)$$

alternating between qualitative states at distinguished time-points, and qualitative states on intervals between distinguished time-points.

2.2 Systems of Functions

Definition 7 *A* **system** *is a set* $F = \{f_1 \ldots f_m\}$ *of reasonable functions* $f_i : [a, b] \to \Re^*$, *each with its own set of landmarks and distinguished time-points. The* **distinguished time-points of a system** F *are the union of the distinguished time-points of the individual functions* $f_i \in F$. *The* **qualitative state** *of a system* F *of* m *functions is the* m*-tuple of individual qualitative states:*

$$QS(F, t_i) = [QS(f_1, t_i), \ldots QS(f_m, t_i)]$$

$$QS(F, t_i, t_{i+1}) = [QS(f_1, t_i, t_{i+1}), \ldots QS(f_m, t_i, t_{i+1})]$$

If t_i *and/or* t_{i+1} *are not distinguished time-points of a particular* f_j, *then* t_i *and the interval* (t_i, t_{i+1}) *must be between two distinguished time-points of* f_j, *say* t_k *and* t_{k+1}. *Then* $QS(f_j, t_i)$ *and* $QS(f_j, t_i, t_{i+1})$ *are defined to be the same as the containing* $QS(f_j, t_k, t_{k+1})$. *The* **qualitative behavior of** F *is the sequence of qualitative states of* F*:*

$$QS(F, t_0), QS(F, t_0, t_1), QS(F, t_1), \ldots QS(F, t_n).$$

These definitions give us a precise semantics for the qualitative description of continuous functions, and clarifies the concept of the "next state." Every state has a qualitative description $QS(F, t)$, but that description changes only at discrete distinguished time-points, and remains constant on the open intervals between them. Thus the "next state" of a mechanism is more properly called the next distinct qualitative state description of the mechanism.

2.3 Qualitative State Transitions

Since a reasonable function f is continuously differentiable, the Intermediate Value Theorem and the Mean Value Theorem restrict the way it can change from one qualitative state to the next. There are two types of qualitative state transitions: **P-transitions**, moving from a point to an interval, and **I-transitions**, moving from an interval to a point.

Definition 8 *Where t_i is a distinguished time-point, a* **P-transition** *of f is a pair of adjacent qualitative states of f,*

$$QS(f, t_i) \Rightarrow QS(f, t_i, t_{i+1})$$

whose first state is the qualitative state at a distinguished time-point. An **I-transition** *is a pair of adjacent qualitative states of f,*

$$QS(f, t_{i-1}, t_i) \Rightarrow QS(f, t_i)$$

whose first state is the qualitative state on the interval between distinguished time-points.

Table 3 specifies the set of possible transitions that can take place in the qualitative behavior of a single function. The validity of this table is proved by the propositions in Appendix A.

Table 3: The possible transitions

A reasonable function $f : [a, b] \rightarrow \Re^*$ is restricted to the following set of possible transitions from one qualitative state to the next. The contents of this table are justified by Propositions 3, 4, 6, and 7 in Appendix A.

P-Transitions

Name	$QS(f, t_i)$	$\Rightarrow$	$QS(f, t_i, t_{i+1})$
$P1$	$\langle l_j, std\rangle$		$\langle l_j, std\rangle$
$P2$	$\langle l_j, std\rangle$		$\langle (l_j, l_{j+1}), inc\rangle$
$P3$	$\langle l_j, std\rangle$		$\langle (l_{j-1}, l_j), dec\rangle$
$P4$	$\langle l_j, inc\rangle$		$\langle (l_j, l_{j+1}), inc\rangle$
$P5$	$\langle (l_j, l_{j+1}), inc\rangle$		$\langle (l_j, l_{j+1}), inc\rangle$
$P6$	$\langle l_j, dec\rangle$		$\langle (l_{j-1}, l_j), dec\rangle$
$P7$	$\langle (l_j, l_{j+1}), dec\rangle$		$\langle (l_j, l_{j+1}), dec\rangle$

I-Transitions

Name	$QS(f, t_i, t_{i+1})$	$\Rightarrow$	$QS(f, t_{i+1})$
$I1$	$\langle l_j, std\rangle$		$\langle l_j, std\rangle$
$I2$	$\langle (l_j, l_{j+1}), inc\rangle$		$\langle l_{j+1}, std\rangle$
$I3$	$\langle (l_j, l_{j+1}), inc\rangle$		$\langle l_{j+1}, inc\rangle$
$I4$	$\langle (l_j, l_{j+1}), inc\rangle$		$\langle (l_j, l_{j+1}), inc\rangle$
$I5$	$\langle (l_j, l_{j+1}), dec\rangle$		$\langle l_j, std\rangle$
$I6$	$\langle (l_j, l_{j+1}), dec\rangle$		$\langle l_j, dec\rangle$
$I7$	$\langle (l_j, l_{j+1}), dec\rangle$		$\langle (l_j, l_{j+1}), dec\rangle$
$I8$	$\langle (l_j, l_{j+1}), inc\rangle$		$\langle l^*, std\rangle$
$I9$	$\langle (l_j, l_{j+1}), dec\rangle$		$\langle l^*, std\rangle$

In cases $I8$ and $I9$, f becomes *std* at l^*, a new landmark value such that $l_j < l^* < l_{j+1}$. In these cases, a previously unknown landmark value is discovered because other constraints force $f'(t)$ to become zero.

3 Qualitative Structure

The qualitative structural description consists of a set of parameters representing the state variables of the mechanism. Simulation assigns a qualitative behavior to each parameter. Constraints holding between two or three parameters in the structural description serve to limit the possible combinations of qualitative behavior. The constraint notation used here has the advantage, like de Kleer's confluences, of having a clear correspondence with differential equations by making explicit all the functions and operators in the equation.

3.1 Arithmetic Constraints

Constraints corresponding to the basic arithmetic and differential operators are fundamental to a structural description.

Definition 9 *$ADD(f,g,h)$ is a three-place predicate on reasonable functions $f,g,h : [a,b] \rightarrow \Re^*$ which holds iff $f(t)+g(t) = h(t)$ for every $t \in [a,b]$.*

Definition 10 *$MULT(f,g,h)$ is a three-place predicate on reasonable functions $f,g,h : [a,b] \rightarrow \Re^*$ which holds iff $f(t)*g(t) = h(t)$ for every $t \in [a,b]$.*

Definition 11 *$MINUS(f,g)$ is a two-place predicate on reasonable functions $f,g : [a,b] \rightarrow \Re^*$ which holds iff $f(t) = -g(t)$ for every $t \in [a,b]$.*

Since addition and multiplication are commutative,

$$\begin{gathered} ADD(f,g,h) \Leftrightarrow ADD(g,f,h), \\ MULT(f,g,h) \Leftrightarrow MULT(g,f,h), \\ MINUS(f,g) \Leftrightarrow MINUS(g,f). \end{gathered}$$

Definition 12 *$DERIV(f,g)$ is a two-place predicate on reasonable functions $f,g : [a,b] \rightarrow \Re^*$ which holds iff $f(t) = g'(t)$ for every $t \in [a,b]$.*

3.2 Qualitative Function Constraints

The qualitative structure description of a mechanism might state that one physical parameter is a function of another, without specifying the function completely. Rather, it can be described qualitatively in terms of regions of monotonic increase or decrease, and landmark values passed through.

The most common and important cases are functional relationships that are strictly monotonic everywhere. The monotonic function constraint M^+ applies in the situation when the function is strictly monotonically increasing, and M^- when it is decreasing.

Definition 13 *M^+ is a two-place predicate on reasonable functions $f, g : [a,b] \rightarrow \Re^*$. $M^+(f,g)$ is true iff $f(t) = H(g(t))$ for all $t \in [a,b]$, where H is a function with domain $g([a,b])$ and range $f([a,b])$, differentiable and with $H'(x) > 0$ for all x in the interior of the domain. M^- is defined similarly, except that $H'(x) < 0$.*

The restrictions on H are motivated by two requirements. First, the critical points of f and g must match across a $M^+(f,g)$ constraint. Second, it must be possible to break a function such as $\sin x$ "at the joints" into regions of monotonic increase and decrease, so $H'(x) = 0$ must be allowed at the boundary of the domain.

Clearly, $M^+(f,g) \Leftrightarrow M^+(g,f)$, and $M^-(f,g) \Leftrightarrow M^-(g,f)$.

Note that $M^+(f,g)$ does not imply that f and g are monotonic functions on $[a,b]$. For example, $M^+(2\sin t, \sin t)$ holds on $[0, 2\pi]$, where $H(x) = 2x$. As a notational variant of $M^+(f,g)$, we may write $f = M^+(g)$ or $g = M^+(f)$.

Proposition 2 *Consider two continuously differentiable functions $f, g : [a,b] \rightarrow \Re^*$, where $M^+(f,g)$. Then for all $t \in (a,b)$,*

$$\begin{aligned} f'(t) > 0 \quad &iff \quad g'(t) > 0 \\ f'(t) = 0 \quad &iff \quad g'(t) = 0 \\ f'(t) < 0 \quad &iff \quad g'(t) < 0 \end{aligned}$$

Proof: $M^+(f,g)$ means that $f(t) = H(g(t))$, so $f'(t) = H'(g(t)) * g'(t)$. Since $H'(x) > 0$, $g'(t) = 0$ if and only if $f'(t) = 0$. The two strict inequalities follow from the monotonicity of H. []

Thus, the sets of distinguished time-points may not correspond precisely across f and g, but their critical points, and hence their regions of constant directions-of-change, are identical.

A qualitative functional relationship need not be strictly monotonic if it can be divided into sections that are alternately increasing or decreasing monotonic, with critical points at the joints between sections. An example may be clearer than the definition. Suppose that $x = \cos\theta$ for $x \in [-1, 1]$ and $\theta \in [0, 2\pi]$. We may say that $FC(\theta, x, descrip)$, where

$$descrip = \begin{cases} (0,1) \\ ((0,\pi),(-1,1),M^-) \\ (\pi,-1) \\ ((\pi,2\pi),(-1,1),M^+) \\ (2\pi,1) \end{cases}$$

That is, if $\theta = 0$, $x = 1$; when $\theta \in (0,\pi)$, then $x \in (-1,1)$ and $M^-(\theta, x)$; and so on.

Definition 14 *A* **qualitative functional constraint** *is a three-place predicate $FC(f, g, descrip)$, where f and g are reasonable functions $f, g : [a,b] \to \Re$, and descrip is a list of descriptors. The predicate is true iff f and g satisfy the qualitative description descrip. A descriptor is either a pair of corresponding landmark values for f and g, or a triple consisting of two intervals defined by landmark values for f and g, and the symbol M^+ or M^- indicating the relationship between f and g when their values lie in the intervals.*

At the joints between monotonic sections, the restrictions on permissible combinations of directions of change are weaker. In particular, it is possible for one parameter to have direction of change *std* while the other is *inc* or *dec*. Otherwise, for qualitative simulation, these more complex functional constraints can be treated exactly like the monotonic constraints they imply at the current time-point. Only the strict monotonic constraints M^+ and M^- are used in the remainder of this paper.

3.3 Qualitative Structure Description

The definitions of these constraints now allow us to define qualitative structural descriptions as an abstraction of differential equations. If a mechanism can be described by a certain differential equation, there is a corresponding, but strictly weaker, qualitative structural description for it. That is, any behavior which satisfies the differential equation must necessarily satisfy the qualitative structural description.

Theorem 1 *Let*

$$F[u(t), u'(t), \ldots, u^{(n)}] = 0 \tag{1}$$

be an ordinary differential equation of order n, to be satisfied by a function $u : [a,b] \rightarrow \Re$*, where F is defined only in terms of the arithmetic operations addition, multiplication, and negation, along with functions of continuous and strictly non-zero derivative. Then a set of parameters and constraints can be defined, corresponding with equation (1), such that any reasonable function* $u : \Re \rightarrow \Re$ *which satisfies equation (1) also satisfies the set of constraints.*

Proof. Define parameters $f_1, \ldots f_m$ corresponding to the arguments $u(t), u'(t), \ldots u^{(n)}$ to F, and to each subexpression appearing in equation (1). Equation (1) can then be transformed into a system of simpler equations, derived from the subexpressions of F, each involving a single function or operator applied to parameters as arguments, and with the result assigned to another parameter. For example, if one subexpression of F is $(exp_1 + exp_2)$, and the parameters corresponding to exp_1, exp_2, and the entire subexpression are f_i, f_j, and f_k, respectively, then the subexpression corresponds to the equation $f_i + f_j = f_k$. This, in turn, can be transformed to the constraint $ADD(f_i, f_j, f_k)$. If f_i corresponds to a subexpression of the form $g(exp)$, where f_j corresponds to exp, and g' is strictly positive, then the constraint generated is $M^+(f_i, f_j)$. Similarly, if f_1 and f_2 correspond to the arguments $u(t)$ and $u'(t)$, the constraint generated is $DERIV(f_1, f_2)$.

Suppose some function $u : [a,b] \rightarrow \Re$ satisfies equation (1). Other real-valued functions corresponding to each parameter f_i are defined by the subexpression equations of F. The ADD, MULT, MINUS, and DERIV constraints are equivalent to the corresponding equations, by definition, and so are satisfied by u and the functions derived from it. The M^+ and M^- constraints are strictly weaker than the specific function appearing in F, so if u and its derived functions satisfy equation (1), they must satisfy the M^+ and M^- constraints as well. []

4 Qualitative Simulation

This section describes the QSIM qualitative simulation algorithm, and refers to the proofs of the various steps, appearing in the appendices.

4.1 Input and Output

The qualitative simulation algorithm is given the following structural description of a mechanism.

1. A set $\{f_1 \ldots f_m\}$ of symbols representing the functions in the system.

2. A set of constraints applied to the function symbols: $M^+(f,g)$, $M^-(f,g)$, $ADD(f,g,h)$, $MULT(f,g,h)$, $MINUS(f,g)$ or $DERIV(f,g)$. Each constraint may have associated corresponding values for its functions.

3. Each function is associated with a totally ordered set of symbols representing landmark values. Each function has at least the basic set of landmarks $\{-\infty, 0, \infty\}$.

4. Each function may have upper and lower range limits, which are landmark values beyond which the current set of constraints no longer apply. A range limit may be associated with a new operating region which has its own constraints and range limits.

5. An initial time-point symbol, t_0, and qualitative values for each of the f_i at t_0.

The result of the qualitative simulation is one or more qualitative behavior descriptions for the function symbols given. Each qualitative behavior description consists of the following:

1. A sequence $\{t_0 \ldots t_n\}$ of symbols representing the distinguished time-points of the system's behavior.

2. For each function f_i, a totally ordered set of landmark values, possibly extending the originally given set.

3. For each function, at each distinguished time-point or interval between adjacent time-points, a qualitative state description expressed in terms of the landmark values of that function.

4.2 The Algorithm QSIM.

The qualitative simulation algorithm, QSIM, repeatedly takes an active state and generates all possible successor states, filtering out states that violate some consistency criterion. Because it may not be able to determine the next state uniquely, QSIM builds a tree of states representing the possible behaviors of the mechanism.

Place the initial state on the list ACTIVE, of states whose successors need to be determined. Repeat the following steps until ACTIVE becomes empty or a resource limit is exceeded.

1. Select a qualitative state from ACTIVE.

2. For each function in the structural description, determine (from Table 1) the set of transitions possible given the current qualitative state.

3. For each constraint, generate the set of tuples of transitions and filter for consistency with that constraint.

4. Perform pair-wise consistency filtering on the sets of tuples associated with the constraints in the system, applying the consistency criterion that adjacent constraints must agree on the transition assigned to the shared parameter.

5. Generate all possible global interpretations from the remaining tuples. If there are none, mark the behavior as inconsistent. Create new qualitative states resulting from each interpretation, and make them successors of the current state.

6. Apply global filtering rules to the new qualitative states, and place any remaining states on ACTIVE.

After an example, the individual steps of the algorithm are discussed in detail.

4.3 Example from the Ball System

This section illustrates one cycle of the QSIM algorithm by showing how the third state ($t = t_1$) of the Ball system (Table 2) is derived from its predecessor.

We start with an active state whose description is:

$$QS(A, t_0, t_1) = \langle g, std \rangle$$
$$QS(V, t_0, t_1) = \langle (0, \infty), dec \rangle$$
$$QS(Y, t_0, t_1) = \langle (0, \infty), inc \rangle$$

First, we determine the possible transitions for the individual functions:

A

$$I1 \quad \langle g, std \rangle \Rightarrow \langle g, std \rangle$$

V

$$I5 \quad \langle (0, \infty), dec \rangle \Rightarrow \langle 0, std \rangle$$
$$I6 \quad \langle (0, \infty), dec \rangle \Rightarrow \langle 0, dec \rangle$$
$$I7 \quad \langle (0, \infty), dec \rangle \Rightarrow \langle (0, \infty), dec \rangle$$
$$I9 \quad \langle (0, \infty), dec \rangle \Rightarrow \langle L^*, std \rangle$$

Y

$$I4 \quad \langle (0, \infty), inc \rangle \Rightarrow \langle (0, \infty), inc \rangle$$
$$I8 \quad \langle (0, \infty), inc \rangle \Rightarrow \langle L^*, std \rangle$$

Next, each constraint forms a set of transition tuples. Those marked with c below are eliminated by constraint consistency filtering. Then those marked with w are eliminated by pairwise consistency filtering.

$DERIV(Y,V)$		$DERIV(V,A)$	
$(I4, I5)$	c	$(I5, I1)$	c
$(I4, I6)$	c	$(I6, I1)$	
$(I4, I7)$		$(I7, I1)$	
$(I4, I9)$	w	$(I9, I1)$	c
$(I8, I5)$	w		
$(I8, I6)$			
$(I8, I7)$	c		
$(I8, I9)$	c		

These tuples can be formed into the following two global interpretations:

Y	V	A
$I4$	$I7$	$I1$
$I8$	$I6$	$I1$

The first of these interpretations yields a qualitative state description identical to the preceding state, so it is considered redundant. The only remaining possibility then becomes the unique successor:

$$QS(A, t_1) = \langle g, std \rangle$$
$$QS(V, t_1) = \langle 0, dec \rangle$$
$$QS(Y, t_1) = \langle Y_{max}, std \rangle.$$

The following sections explain the steps of the QSIM algorithm in detail and discuss the proofs of their validity.

4.4 Function Consistency

The possible transitions that a single parameter can take from one qualitative state to the next are given in Table 3. In Step 2, the current state of each function is used to retrieve the set of applicable transition patterns from Table 3. Constraints between neighboring functions are not considered until Step 3. Transitions are also checked against invariant assertions at this stage, to eliminate impossible transitions for functions that are (e.g.) always finite or never negative.

For any particular qualitative state, Table 3 provides at most 4 possible transitions. Thus, if there are n functions in the system, the possible next states are to be found within a product space of at most 4^n points. At this stage, however, we do not explicitly generate this product space, so we need create at most $4n$ individual transitions.

Appendix A presents the proofs that justify the possible transitions given in Table 3. It also discusses the handling of divergence to ∞ and asymptotic approach to limiting values.

4.5 Constraint Consistency

Step 3 of the QSIM algorithm aggregates the individual transitions into 2-tuples and 3-tuples corresponding to the arguments of individual constraints. These tuples can then be checked for consistency according to two criteria local to individual constraints. (See Appendix B.)

- The tuple of directions of change must be consistent with the constraint in the state resulting from the transition.
- The result of the transition-tuple can be compared with corresponding values of the arguments to that constraint.

Definition 15 *Landmark values p and q are* **corresponding values** *of f and g if there is some $t \in [a, b]$ such that $f(t) = p$ and $g(t) = q$.*

$M_0^+(f, g)$ and $M_0^-(f, g)$ are abbreviations for $M^+(f, g)$ and $M^-(f, g)$, respectively, with corresponding values $(0, 0)$.

Definition 16 *Suppose $QS(f, t_i, t_{i+1}) = \langle (l_k, l_{k+1}), inc \rangle$. Then l_{k+1} is the* **limit of f during** (t_i, t_{i+1}). *If $f(t_{i+1}) = l_{k+1}$, we say that f has moved to its limit. Otherwise, $f(t_{i+1}) < l_{k+1}$, and we say that f has moved toward, but not reached, its limit. Similarly if f is decreasing during $QS(f, t_i, t_{i+1})$.*

Informally, if f is between landmark values but moving toward a limit, it may or may not reach that limit by the next distinguished time-point. If several functions are moving toward limits, constraints between functions limit the space of possibilities. For example, if $M^+(f, g)$ is true, and f and g are moving toward corresponding limit values, then either both will reach their limits, or neither will. Similarly, if $f + g = h$, and two functions are moving toward corresponding limits, while one is bounded away from the third corresponding value, the possible transitions can be filtered.

These constraint-based consistency criteria generalize the *Transition Ordering* rules of Williams (1984a). The propositions in Appendix B define and justify the comparison of the proposed transition-tuple with a particular set of known corresponding values.

4.6 Pairwise Consistency Filtering

Two constraints are *adjacent* if they share an argument. At this point, each constraint has an associated set of transition tuples, consistent with that individual constraint. A tuple is a proposed assignment of transitions to the functions in that constraint. To be pairwise consistent, tuples on adjacent constraints must assign the same transition to the function they share. For certain tuples, there may be no opposite number to make such a consistent pair. If so, that tuple may be deleted.

Waltz (1975) developed this local consistency filtering algorithm to converge quickly on a small set of possible labelings for a graph representing the edges, vertices, and regions of a visual scene. (The algorithm is also described in detail in Winston (1984).) A key step in the development of the QSIM algorithm was the observation that if *transitions*, rather than *qualitative states* are taken as the analog of edge labels, the Waltz algorithm could be applied directly.

Filtering on transitions rather than states simplifies several steps of the algorithm. The possibility of creating new landmarks can be considered without actually creating landmarks that might have to be retracted. The pairwise and global consistency filtering can match atomic transition names rather than much more expensive structure-matching on the predicted next state. Finally, some of the global filters (Section 4.8) depend on the sequence of transitions leading up to a proposed state, and would be more difficult to express in terms of state descriptions.

The Waltz algorithm visits each constraint in the mechanism description, looking at all the adjacent constraints and the function joining the pair. It applies the following rule to each transition tuple associated with the constraint it is visiting.

if: that tuple assigns a transition to the function which is not assigned by any tuple associated with the other constraint,

then: delete that tuple.

The algorithm then visits each constraint adjacent to a constraint at which a tuple was deleted, and terminates when no more filtering is possible. This process is important to the efficiency of the QSIM algorithm, since deleting a single tuple eliminates an entire region of the cross-product space of global interpretations.

4.7 Generating Global Interpretations

A global interpretation is an assignment of a transition to each function in the system. The result of Waltz filtering is a reduced set of tuples associated with each constraint. Not all combinations of these tuples are possible global interpretations. Suppose, for example, that we have the following constraints and associated transition tuples:

$M^+(f,g)$	$M^+(g,h)$
$(I2, I2)$	$(I2, I2)$
$(I3, I3)$	$(I3, I3)$

Clearly, although no further local consistency filtering is possible, there are only two possible assignments of transitions to (f, g, h), namely $(I2, I2, I2)$ and $(I3, I3, I3)$. This pruning takes place as the global interpretations are created.

Global interpretations are built one at a time, by a depth-first traversal of the space of assignments of tuples to constraints. An attempted interpretation fails if the next tuple cannot be assigned without conflicting with transitions assigned to functions by previous tuples. In case all possible next states are eliminated, the current state must be the endpoint of the domain.

A global interpretation is then used to construct a new qualitative state description, which is added to the tree of state descriptions as a successor to the current state. At this point, if all functions in a constraint are equal to landmark values, the constraint records them as a set of corresponding values.

4.8 Global Filters

The completed qualitative state descriptions are mathematically plausible successors to the current state. There are, however, several global filters that can be applied before a new state is added to ACTIVE.

The mathematically valid filters applied at this stage are the following.

- (No Change.) Delete the new state if all transitions are in the set $\{I1, I4, I7\}$, because the new state description would be identical to its immediate predecessor, which therefore already captures its qualitative behavior. In other words, *something* must reach a limit point for an I-transition to take place.

- (Cycle.) If the new state is identical to one of its predecessors (all functions have identical *landmark* values, and all directions of change are the same), then mark the behavior as cyclic, install a pointer to the identical predecessor, and do not add the new state to ACTIVE.

- (Divergence.) If any function takes on the value ∞ or $-\infty$, the current time-point must be the endpoint of the domain, so the new state does not go onto ACTIVE.

The first filter does not reduce the number of behaviors described, but only eliminates a redundant description. The second detects when all the consequences of a particular state have already been determined, and need not be explored anew. The third determines when a state must be at the endpoint of the domain, and thus can have no successors.

We refer to the qualitative simulation algorithm described here as the *pure* QSIM algorithm. For a particular application, additional heuristic filters may be added.[2]

4.9 Complexity

The process of formalizing qualitative simulation led to the improved

[2] Some possible heuristics include:

- (Quiescence.) If all functions have derivative zero, conclude that the system is quiescent, the new time-point is the endpoint of the domain (possibly $t = \infty$), and do not place the new state on ACTIVE.

- (No Divergence.) In physical systems, eliminate transitions in which any state goes to ∞ or $-\infty$. A more accurate description of the system would include an operating region change corresponding to some component breaking.

QSIM algorithm, which turned out to be 30 to 60 times faster than its predecessor ENV (Kuipers, 1982, 1984) on a variety of examples ranging from 3 parameters and 2 constraints (the Ball) up to 16 parameters and 14 constraints (the Starling equilibrium [Kuipers and Kassirer, 1983, 1984]) We can estimate the algorithmic complexity of QSIM as follows. Suppose there are n parameters in the system, m constraints, and the longest behavior has length t. (t is then log of the total number of qualitative states.) Since a constraint can have no more than three parameters, $n = o(m)$.

- A set of possible state transitions is assigned to each parameter from a fixed-length table, and no more than 4 transitions can be assigned to any parameter. This defines a search space of 4^n state transitions, but only $4n$ transitions need actually be created, requiring $o(n)$ time.

- A constraint can have no more than $4^3 = 64$ transition tuples. Filtering a tuple against the direction-of-change tables (Appendix A) takes constant time, but the number of corresponding values grows linearly (though slowly) with the length t of the behavior. Thus constraint filtering requires $o(mt)$ time.

- Waltz filtering visits each constraint at least once, but beyond that visits only neighbors of constraints where it was able to delete a tuple. Thus, the number of constraints visited is proportional to the total number of tuples, which is linear in the number of constraints. Each visit takes bounded time. Thus, Waltz filtering takes $o(m)$ time.

- Generating the global interpretations explicitly constructs the remaining parts of the product space. Typically, the remaining space is small, but unfortunately there are pathological cases which yield 2^n possible successor states.

- The most expensive of the global filters is the check for previous identical states, which requires $o(nt)$ time.

A pathological case where the number of global interpretations generated is exponential in the number of functions in the system can be easily constructed. Consider a system with three parameters f, g, and h, and two constraints, $DERIV(f,g)$ and $DERIV(g,h)$, in a state where f, g, and h are all positive and increasing. Then the possible tuples are:

$DERIV(f,g)$	$DERIV(g,h)$
$(I3, I3)$	$(I3, I3)$
$(I3, I4)$	$(I3, I4)$
$(I4, I3)$	$(I4, I3)$
$(I4, I4)$	$(I4, I4)$

Neither local consistency filtering nor the formation of global interpretations eliminate any of the possible assignments, so for n parameters linked by a chain of $DERIV$ constraints, there are 2^n interpretations.

f	g	h
$I3$	$I3$	$I3$
$I3$	$I3$	$I4$
$I3$	$I4$	$I3$
$I3$	$I4$	$I4$
$I4$	$I3$	$I3$
$I4$	$I3$	$I4$
$I4$	$I4$	$I3$
$I4$	$I4$	$I4$

In practice, creation of the global interpretations significantly reduces the number of compatible assignments. The algorithm need not halt, however, and can continue forever producing longer and longer (but always finite) behaviors, each of which satisfies the qualitative constraints.

Although the QSIM algorithm is exponential in the worst case, in practice generating the successors of a given state appears to be approximately $o(mt)$. The Spring example (3 parameters, 3 constraints) takes about 0.4 seconds, and the Starling mechanism (16 parameters, 14 constraints) [Kuipers and Kassirer, 1983, 1984] takes about 1.0 second on the Symbolics 3600.

5 Questions and Answers

Now that we have defined the QSIM algorithm, with a clear structure and mathematically accessible properties, we can examine it to answer some of our questions about the utility of qualitative simulation as a reasoning method. We can also compare different approaches to qualitative simulation by changing the table of permissible transitions.

5.1 Should Simulation Create Landmarks?

The most important semantic difference between QSIM and other approaches to qualitative simulation is that QSIM can create new landmark values during the simulation, while the other algorithms require all landmarks to be specified when the structure is defined. In this section, we show that the inability to create new landmark values makes it impossible to express certain important qualitative distinctions, such as that between increasing, decreasing, and stable oscillation.

The fixed landmark assumption is particularly deeply embedded in de Kleer's approach, which depends on arithmetic operators defined over a fixed set of *qualitative values*, $\{+, 0, -\}$. A change in landmarks would change the qualitative values, and thus require the operators to be redefined. Such a redefinition is not always possible.

The structure of QSIM makes it possible to experiment with $\{+, 0, -\}$ semantics for qualitative simulation simply by replacing Table 3 with an alternate table of legal transitions (Table 4).

Table 5 shows the behavior of the Spring system under the $\{+, 0, -\}$ semantics. This behavior can be considered a cycle only if two functions are allowed to match between landmark values. That is, only if we may conclude from this table that $V(t_4) = V(t_0)$. A match between the states t_4 and t_0 in this behavior suppresses the distinction between increasing, stable, or decreasing amplitude (see Table 6). De Kleer and Bobrow (1984) present an example of a spring with frictional damping, whose actual behavior is a decreasing oscillation. The behavioral description they present is cyclic, and similar to that given in Table 5 above, with the addition of terms for the frictional force. Their description accurately captures the repetitive series

Table 4: Possible transitions under $\{+, 0, -\}$ semantics

P-Transitions

Name	$QS(f, t_i)$	$\Rightarrow$	$QS(f, t_i, t_{i+1})$
$P1$	$\langle l_j, std \rangle$		$\langle l_j, std \rangle$
$P2$	$\langle l_j, std \rangle$		$\langle (l_j, l_{j+1}), inc \rangle$
$P3$	$\langle l_j, std \rangle$		$\langle (l_{j-1}, l_j), dec \rangle$
$P4$	$\langle l_j, inc \rangle$		$\langle (l_j, l_{j+1}), inc \rangle$
$P5$	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle (l_j, l_{j+1}), inc \rangle$
$P6$	$\langle l_j, dec \rangle$		$\langle (l_{j-1}, l_j), dec \rangle$
$P7$	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle (l_j, l_{j+1}), dec \rangle$
$Q8$	$\langle (l_j, l_{j+1}), std \rangle$		$\langle (l_j, l_{j+1}), std \rangle$
$Q9$	$\langle (l_j, l_{j+1}), std \rangle$		$\langle (l_j, l_{j+1}), inc \rangle$
$Q10$	$\langle (l_{j-1}, l_j), std \rangle$		$\langle (l_{j-1}, l_j), dec \rangle$

I-Transitions

Name	$QS(f, t_i, t_{i+1})$	$\Rightarrow$	$QS(f, t_{i+1})$
$I1$	$\langle l_j, std \rangle$		$\langle l_j, std \rangle$
$I2$	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle l_{j+1}, std \rangle$
$I3$	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle l_{j+1}, inc \rangle$
$I4$	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle (l_j, l_{j+1}), inc \rangle$
$I5$	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle l_j, std \rangle$
$I6$	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle l_j, dec \rangle$
$I7$	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle (l_j, l_{j+1}), dec \rangle$
$J8$	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle (l_j, l_{j+1}), std \rangle$
$J9$	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle (l_j, l_{j+1}), std \rangle$
$J10$	$\langle (l_j, l_{j+1}), std \rangle$		$\langle (l_j, l_{j+1}), std \rangle$

The landmarks are fixed as $\{-\infty, 0, \infty\}$. The transitions that create new landmarks (I8 and I9 from Table 3) are eliminated, and new transitions are added (with Q and J names) to permit direction of change *std* between landmarks.

Table 5: The Spring behavior with $\{+, 0, -\}$ semantics
De Kleer's three qualitative values, +, 0, and −, are given here in the QSIM notation: $(0, \infty)$, 0, and $(-\infty, 0)$.

	X	V	A
t_0	$\langle 0, inc\rangle$	$\langle (0,\infty), std\rangle$	$\langle 0, dec\rangle$
	$\langle (0,\infty), inc\rangle$	$\langle (0,\infty), dec\rangle$	$\langle (-\infty,0), dec\rangle$
t_1	$\langle (0,\infty), std\rangle$	$\langle 0, dec\rangle$	$\langle (-\infty,0), std\rangle$
	$\langle (0,\infty), dec\rangle$	$\langle (-\infty,0), dec\rangle$	$\langle (-\infty,0), inc\rangle$
t_2	$\langle 0, dec\rangle$	$\langle (-\infty,0), std\rangle$	$\langle 0, inc\rangle$
	$\langle (-\infty,0), dec\rangle$	$\langle (-\infty,0), inc\rangle$	$\langle (0,\infty), inc\rangle$
t_3	$\langle (-\infty,0), std\rangle$	$\langle 0, inc\rangle$	$\langle (0,\infty), std\rangle$
	$\langle (-\infty,0), inc\rangle$	$\langle (0,\infty), inc\rangle$	$\langle (0,\infty), dec\rangle$
t_4	$\langle 0, inc\rangle$	$\langle (0,\infty), std\rangle$	$\langle 0, dec\rangle$

of increase and decrease in the different parameters, but since it does not express a distinction between increasing, decreasing and steady amplitude, it cannot even ask which qualitative behavior is correct.

The heart of the problem is the inability to create new landmarks. Without representing the initial value (or subsequent critical values) of a parameter in a way that permits ordinal comparison, it is not possible to ask whether the next repetition of a cycle leaves that parameter increased, decreased, or stable. If, in addition, states can be matched between landmark values, three very distinct types of behavior can be collapsed into a single, apparently cyclic, behavior.

In order to express important qualitative distinctions between possible behaviors, it appears important to be able to identify a new critical point of a changing parameter (e.g. a turning point or new equilibrium value), and represent it so that the subsequent simulation treats it as a first-class landmark value.

5.2 Is the Real Behavior Found?

In this section, we show that all actual behaviors of a mechanism are predicted by its qualitative simulation. We take as our "gold standard" the solutions to the ordinary differential equation describing the mechanism.

We say that a real-valued function *satisfies* a given qualitative behavior description if the qualitative description of the function matches the given qualitative behavior. We then prove that any solution to a differential equation satisfies some qualitative behavior produced by the corresponding structural description. The proof is straight-forward, since the bulk of the work has already been done in validating the individual steps of the QSIM algorithm. The algorithm generates a space including all possible behaviors of a given qualitative description, and then discards only behaviors which are internally inconsistent. Thus, the remaining behaviors necessarily include all of the actual behaviors of the mechanism.

Definition 17 *Suppose we have a reasonable function* $u : [a, b] \rightarrow \Re$ *and a qualitative behavior description of the function-symbol* f,

$$QS(f, t_0), QS(f, t_0, t_1) \ldots QS(f, t_{n-1}, t_n), QS(f, t_n)$$

with distinguished time-points $\{t_0, \ldots t_n\}$ *and landmarks* $\{l_1, \ldots l_k\}$. *We say that* u **satisfies** *the behavior description if there is an order-preserving mapping* m *of* $\{t_0, \ldots t_n\}$ *into* $[a, b]$ *with* $m(t_0) = a$ *and* $m(t_n) = b$, *and an order-preserving mapping of* $\{l_1, \ldots l_k\}$ *into* $\Re$, *such that, for all distinguished time-points* t_i, $QS(u, m(t_i))$ *matches* $QS(f, t_i)$ *and* $QS(u, m(t_i), m(t_{i+1}))$ *matches* $QS(f, t_i, t_{i+1})$.

Theorem 2 *Let*

$$F[u(t), u'(t), \ldots, u^{(n)}] = 0 \tag{2}$$

be an ordinary differential equation of order n, *and let* $\{u(t_0) = y_0, u'(t_0) = y_1, \ldots u^{(n)}(t_0) = y_n\}$ *be the initial conditions on the solution to (2). Suppose that equation (2) and its initial conditions are satisfied by a reasonable function* $u : [a, b] \rightarrow \Re$. *Let* C *be the set of functions and constraints derived from (2) by the methods of Theorem 1, and let* $QS(F, t_0)$ *be the qualitative state description derived from the given set of initial conditions. Let* T *be* *the tree of qualitative state descriptions derived from C and $QS(F, t_0)$ by the pure QSIM algorithm. Then the function u and the subexpression functions derived from it satisfy some behavioral description in T.*

Proof. QSIM works by progressively restricting the region of a space of qualitative behaviors that it is considering. By showing that any actual solution u is initially in the space, and that no filtering operation can eliminate a genuine solution, we conclude that u and its derived functions must satisfy some behavior in T.

The function u satisfies the initial state description $QS(F, t_0)$ because it is a qualitative abstraction of the initial conditions to equation (2). Step 2 in QSIM generates all possible qualitative state transitions for the functions in C from a given qualitative state, using Table 3 which is justified by Propositions 3, 4, 6, and 7. Thus, any change in qualitative state of the system must be included in the possibilities generated. Step 3 of QSIM filters out combinations of transitions whose result is a state which fails to satisfy individual constraints. Inconsistent sets of directions of change are detected by comparison with tables in Appendix A. The proper implications of sets of corresponding values are checked against Propositions 10,11, 12, and 18. The pairwise consistency filtering of Step 4 simply eliminates from consideration transition tuples which are inconsistent with all neighboring tuples, and thus could not contribute to a global interpretation. Step 5, similarly, eliminates combinations of tuples which do not make consistent assignments of state transitions to particular functions. Finally, the global filters included in the pure QSIM algorithm are discussed in section 4.8 and shown not to eliminate possible behaviors of the system. Thus, at each stage of the simulation, all possible successors to the current qualitative state lie in the space generated, and no genuinely possible successor is eliminated. Since a reasonable function u has a qualitative behavior of finite length, if the tree of states is generated in breadth-first order, it must be generated by the QSIM algorithm within finite time. []

5.3 Are All the Behaviors Real?

In this section, we show that the QSIM algorithm, and local qualitative simulation algorithms in general, cannot be guaranteed against producing spurious behaviors: behaviors which are not actual behaviors for *any* physical system satisfying the structural description.

A weak qualitative structure description provides few constraints, and thus is consistent with many possible behaviors. However, it is also consistent with many possible *mechanisms*, and we might hope that each the qualitative behavior always corresponds to some mechanism that satisfies the structural description. Although this is often the case, and has been conjectured to be universally true, there are cases where spurious behaviors are generated. Thus, if several behaviors are generated, some of them may not be possible behaviors of the mechanism.

One of the attractive applications of qualitative simulation is to predict possible future states, particularly to warn of surprising or disastrous events. Theorem 2 above guarantees that there can be no false negatives: every actual behavior is predicted. However, if a valid description of the mechanism can produce invalid predictions (false positives), its usefulness is limited. As we discuss below, the problem is not fatal, but requires substantial care in the construction and use of a problem-solver.

Theorem 3 *Let C be a set of function-symbols and qualitative constraints, and let $QS(F,t_0)$ be the initial qualitative state description. Let T be the tree of qualitative state descriptions derived from C and $QS(F,t_0)$ by the pure QSIM algorithm. For some C and $QS(F,t_0)$ there are behaviors in T which do not correspond to any solution $u : [a,b] \rightarrow \Re$ to any differential equation and initial condition corresponding to C and $QS(F,t_0)$.*

Proof. Consider a mass on a spring, oscillating on a frictionless surface. The qualitative structural description of this system is

$$\begin{array}{l} DERIV(X,V) \\ DERIV(V,A) \\ M_0^-(A,X), \end{array} \tag{3}$$

which might also be written in the form of a second-order differential equation:

$$\frac{d^2X}{dt^2} = -M_0^+(X). \tag{4}$$

With initial state $X(t_0) = 0$, $V(t_0) = V_{init}$, $A(t_0) = 0$, this system is periodic for any function $A = -M_0^+(X)$, because if we define total energy as

$$TE(x, v) = \int_0^x M_0^+(y)dy + \frac{1}{2}v^2,$$

then equation (4) implies that $\frac{d}{dt}TE = 0$.

The local inference methods of QSIM are not able to determine, at the end of one cycle, whether the oscillation of the system is periodic, or increases or decreases in magnitude. It can, however, branch to express all three behaviors.

Table 6 shows the behavioral description produced by qualitative simulation of the Spring system. The simulation proceeds without branching through the cycle, predicting and creating the new landmarks X_{max}, X_{min}, V_{min}, A_{min}, and A_{max} until X, V, and A are approaching 0, V_{init} and 0, respectively. X and A must reach their limits together, but the simulation branches according to whether V reaches its limit at the same time (t_4), later (t'_4), or earlier (t''_4). In the first case, the state at t_4 matches the state at t_0, so the behavior is stable and periodic. In the second, the oscillation is decreasing with a new critical point $V^-_{max} < V_{init}$. And in the third case, motion continues past V_{init} to a different new critical point $V^+_{max} > V_{init}$. Furthermore, having taken this branch, there is no way to represent the decision as a permanent selection of divergence, convergence, or stable oscillation. The same choice recurs at approaches to other landmarks.

Only the stable periodic behavior is an actual behavior possible for this structural description, but the local inference methods of QSIM cannot prove this fact. Thus, there are behaviors produced by the qualitative simulation algorithm which do not correspond to the behavior of any system satisfying the qualitative structure description. []

The problem also occurs with the algorithms of de Kleer and Forbus, even without creating new landmarks. If we add a single structural landmark corresponding to the initial velocity, by defining a translated variable

Table 6: The Spring Simulation

In $QS(F, t_3, t_4)$, we have $V \to V_{init}$ while $X \to 0$ and $A \to 0$. No local rule can determine whether V reaches its limit before, after, or at the same time as X and A reach theirs. The simulation branches three ways, even though only one behavior is valid.

The first behavior is recognized as a cycle, since the state at t_4 matches the state at t_0. The last two cases are not cycles, since neither t'_4 nor t''_5 matches t_0, so the simulation continues onward from that point.

	X	V	A
t_0	$\langle 0, inc\rangle$	$\langle V_{init}, std\rangle$	$\langle 0, dec\rangle$
	$\langle (0,\infty), inc\rangle$	$\langle (0, V_{init}), dec\rangle$	$\langle (-\infty, 0), dec\rangle$
t_1	$\langle X_{max}, std\rangle$	$\langle 0, dec\rangle$	$\langle A_{min}, std\rangle)$
	$\langle (0, X_{max}), dec\rangle$	$\langle (-\infty, 0), dec\rangle$	$\langle (A_{min}, 0), inc\rangle$
t_2	$\langle 0, dec\rangle$	$\langle V_{min}, std\rangle$	$\langle 0, inc\rangle$
	$\langle (-\infty, 0), dec\rangle$	$\langle (V_{min}, 0), inc\rangle$	$\langle (0, \infty), inc\rangle$
t_3	$\langle X_{min}, std\rangle$	$\langle 0, inc\rangle$	$\langle A_{max}, std\rangle$
	$\langle (X_{min}, 0), inc\rangle$	$\langle (0, V_{init}), inc\rangle$	$\langle (0, A_{max}), dec\rangle$
t_4	$\langle 0, inc\rangle$	$\langle V_{init}, std\rangle$	$\langle 0, dec\rangle$
t'_4	$\langle 0, inc\rangle$	$\langle V^-_{max}, std\rangle$	$\langle 0, dec\rangle$
t''_4	$\langle (X_{min}, 0), inc\rangle$	$\langle V_{init}, inc\rangle$	$\langle (0, A_{max}), dec\rangle$
	$\langle (X_{min}, 0), inc\rangle$	$\langle (V_{init}, V^+_{max}), inc\rangle$	$\langle (0, A_{max}), dec\rangle$
t''_5	$\langle 0, inc\rangle$	$\langle V^+_{max}, std\rangle$	$\langle 0, dec\rangle$

$W(t) = V(t) - V_{init}$ [deKleer & Brown, 1984], we obtain precisely the same three-way branching behavior. Without the translated variable, the de Kleer approach does not express the distinction between increasing, steady, and decreasing amplitudes.

The fundamental problem is that simulation, qualitative or quantitative, is inherently local: the successors to the current state are computed given only the information in the current state. Like QSIM, the Forbus (1982, 1984) and Williams (1984a, 1984b) algorithms are genuine simulations, examining individual states to determine their successors. The de Kleer and Brown (1984) algorithm is effectively a simulation; even though the states are not necessarily generated in chronological order, the existence of a state transition is a local decision, based only on the two states involved. Given a descriptive framework consisting of the functions and constraints describing the mechanism, and the states to be linked, there is simply not enough information available to eliminate all spurious behaviors.

These observations yield some important warnings about the proper use of qualitative descriptions of mechanisms, and the result of their simulation.

- The structural description must be shown to be consistent, preferably by demonstrating that it is an abstraction of an accurate quantitative description, to guarantee (Theorem 2) that the qualitative simulation will include a genuine behavior.

- If the qualitative simulation of a consistent structure yields a single behavior, then that behavioral description must represent the actual behavior of the mechanism.

- If qualitative simulation yields several possible behaviors, further analysis is required before concluding that they represent possible futures.

Qualitative simulation is a useful tool for causal reasoning about the behavior of mechanisms, and QSIM is a particularly complete, efficient implementation of it. However, like all tools, it has important limitations that any user should be familiar with. The formal analysis we have used in this paper is valuable both for the design of the QSIM algorithm and for determining the strengths and limitations of qualitative simulation in general.

5.4 What Next?

Two directions for further research appear promising for more accurate qualitative predictions of behavior. First, the *dynamical systems* approach to qualitative analysis of differential equations (e.g. Arnold, 1973) has greater expressive and inferential power than local qualitative simulation methods. By describing the behaviors of the Spring as trajectories through phase space rather than temporal sequences of qualitative states, it is possible to take a single branch between increasing, decreasing and stable oscillation, rather than repeating the choice at each move toward limits. The theory of dynamical systems also includes global classification theorems delimiting the possible qualitatively distinct behaviors. Further study is needed to determine how practical problems can be stated and solved, and how the solutions can be applied.

Second, if one structural description of a mechanism has spurious behaviors, a different description might not. By changing the problem to take into account the conservation of total energy, an expanded view of the spring mechanism allows QSIM to determine that there is a single, periodic behavior. A physicist can look at equation (4) and recognize or derive the fact that it represents an energy conserving system, and therefore that the behavior must be periodic. Part of this knowledge is the ability to recognize the physical system described by a structure, and to know that there is a better structural description for it; one which adds parameters and constraints (e.g. energy) that illuminate the actual behavior.

This approach takes us outside the realm of qualitative simulation, and into the realm of problem formulation. Chi, Feltovich, and Glaser (1982) have shown that an important distinction between novice and expert problem-solving in physics is that the experts are able to select a description of the physical situation that yields the answer directly; novices search a space of alternate models. Thus, expert causal reasoning uses domain-specific knowledge to select the correct formulation of a problem, leading to its direct solution.

6 References

V. I. Arnold. (Translated and edited by R. A. Silverman.) 1973. *Ordinary Differential Equations.* Cambridge, MA: MIT Press.

M. T. H. Chi, P. J. Feltovich, and R. Glaser. 1982. Categorization and representation of physics problems by experts and novices. *Cognitive Science 5*: 121-152.

J. de Kleer. 1977. Multiple representations of knowledge in a mechanics problem-solver. Proceedings of the Fifth International Joint Conference on Artificial Intelligence.

J. de Kleer. 1978. Causal and teleological reasoning in circuit recognition. MIT Artificial Intelligence Laboratory TR-529.

J. de Kleer. 1979. The origin and resolution of ambiguities in causal arguments. Proceedings of the Sixth International Joint Conference on Artificial Intelligence. Tokyo, Japan.

J. de Kleer and D. G. Bobrow. 1984. Qualitative reasoning with higher-order derivatives. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-84).* Los Altos, CA: William Kaufman.

J. de Kleer and J. S. Brown. 1983. The origin, form, and logic of qualitative physical laws. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* Los Altos, CA: William Kaufman.

J. de Kleer and J. S. Brown. 1984. A qualitative physics based on confluences. *Artificial Intelligence*, v. 24, n. 1.

K. D. Forbus. 1981. Qualitative reasoning about physical processes. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence.* Los Altos, CA: William Kaufman.

K. D. Forbus. 1982. Qualitative process theory. Cambridge, MA: MIT Artificial Intelligence Laboratory Memo 664.

K. D. Forbus. 1983. Measurement interpretation in qualitative process theory. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* Los Altos, CA: William Kaufman.

K. D. Forbus. 1984. Qualitative process theory. Ph.D. thesis, MIT Department of Electrical Engineering and Computer Science.

D. Gentner and A. Stevens, (Eds.). 1983. *Mental Models.* Hillsdale, NJ: Erlbaum.

P. J. Hayes. 1979. The naive physics manifesto. In D. Michie (Ed.), *Expert Systems in the Micro-electronic Age.* Edinburgh: Edinburgh University Press, 1979.

B. J. Kuipers. 1982. Getting the envisionment right. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-82).* Pittsburgh, Pennsylvania, August 1982.

B. J. Kuipers. 1984. Commonsense reasoning about causality: deriving behavior from structure. *Artificial Intelligence*, v. 24, n. 1. (Preprinted as Tufts University Cognitive Science Working Paper No. 18, 1982.)

B. J. Kuipers and J. P. Kassirer. 1983. How to discover a knowledge representation for causal reasoning by studying an expert physician. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* Palo Alto: William Kaufman.

B. J. Kuipers and J. P. Kassirer. 1984. Causal reasoning in medicine: analysis of a protocol. *Cognitive Science* 8: 363-385.

D. Waltz. 1975. Understanding line drawings of scenes with shadows. In P. H. Winston (Ed.), *The Psychology of Computer Vision.* New York: McGraw-Hill, 1975.

B. Williams. 1984a. The use of continuity in a qualitative physics. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-84).* Los Altos, CA: William Kaufman.

B. Williams. 1984b. Qualitative analysis of MOS circuits. *Artificial Intelligence*, v. 24, n. 1.

P. H. Winston. 1984. *Artificial Intelligence, 2nd ed.* Reading, MA: Addison-Wesley.

6.1 Acknowledgements

The environment of the MIT LCS Clinical Decision Making Group provides valuable support and stimulation for this research. Christopher Eliot is responsible for the implementation of the QSIM algorithm with a very comfortable interface, and for a number of valuable technical contributions. I am also grateful for helpful discussions and observations on qualitative simulation in general, QSIM in particular, and drafts of this paper by Daniel Bobrow, Johan de Kleer, Ken Forbus, Walter Hamscher, Ramesh Patil, Elisha Sacks, Mike Wellman, and Brian Williams.

A The Qualitative State Transitions

This appendix applies the Intermediate-Value and Mean-Value Theorems to prove the validity of the transition rules in Table 3 on page 18 that restrict the possible qualitative behaviors of a single function.

Let $f : [a, b] \to \Re$ be a reasonable function with distinguished time-points

$$a = t_0 < \cdots < t_n = b,$$

and landmark values

$$l_1 < \cdots < l_k.$$

Definition: Where t_i is a distinguished time-point, a **P-transition** of f is a pair of adjacent qualitative states of f,

$$QS(f, t_i) \Rightarrow QS(f, t_i, t_{i+1})$$

whose first state is the qualitative state at a distinguished time-point. An **I-transition** is a pair of adjacent qualitative states of f,

$$QS(f, t_{i-1}, t_i) \Rightarrow QS(f, t_i)$$

whose first state is the qualitative state on the interval between distinguished time-points.

Proposition 3 *Let $QS(f, t_i)$ and $QS(f, t_i, t_{i+1})$ be adjacent qualitative states of f. Then there is some landmark value l_j such that $f(t_i) = l_j$, and the only possible P-transitions of f are given by the table below:*

	$QS(f, t_i)$	$\Rightarrow$	$QS(f, t_i, t_{i+1})$
1.	$\langle l_j, std\rangle$		$\langle l_j, std\rangle$
2.	$\langle l_j, std\rangle$		$\langle (l_j, l_{j+1}), inc\rangle$
3.	$\langle l_j, std\rangle$		$\langle (l_{j-1}, l_j), dec\rangle$
4.	$\langle l_j, inc\rangle$		$\langle (l_j, l_{j+1}), inc\rangle$
5.	$\langle l_j, dec\rangle$		$\langle (l_{j-1}, l_j), dec\rangle$

Proof: If t_i is a distinguished time-point, then by definition there must be a landmark value l_j such that $f(t_i) = l_j$. In cases 1-3, there are reasonable functions f with $QS(f, t_i) = \langle l_j, std \rangle$, and with direction of change *std*, *inc*, or *dec* in $QS(f, t_i, t_{i+1})$, so no subsequent direction of change can be excluded. In these cases, by the Mean Value Theorem, $f(t)$ must be equal to, greater than, or less than $f(t_i) = l_j$, respectively, on the interval (t_i, t_{i+1}). By Proposition 1 and the Intermediate Value Theorem, $f(t)$ must be within (l_j, l_{j+1}) if it is increasing, or (l_{j-1}, l_j) if it is decreasing. In case 4, if the direction of change is *inc*, then $f'(t_i) > 0$. Since the derivative is continuous, there is an interval around $t = t_i$ in which $f'(t) > 0$. By Proposition 1, since there are points within (t_i, t_{i+1}) where the direction of change is *inc*, it must be *inc* throughout (t_i, t_{i+1}), so $f(t)$ must be within (l_j, l_{j+1}). Case 5 is similar. []

The three P-transitions from the state $\langle l_j, std \rangle$ handle the case where a higher order derivative, which is not explicitly represented by QSIM, determines the direction of motion. In such a situation, all three transitions are generated, and other constraints filter out the impossible cases. De Kleer and Bobrow (1984) determine and use higher-order derivatives explicitly to make that decision, at least for linear equations. Their approach determines the order of the structural description, and thus knows how many higher-order derivatives to compute. The advantages of the current approach are that it places much weaker conditions on the differentiability of the parameters, and that it is not restricted to linear equations.

Proposition 4 *Let $QS(f, t_i, t_{i+1})$ and $QS(f, t_{i+1})$ be adjacent qualitative states of f. Then there are landmark values l_j and l_{j+1} such that the only possible I-transitions are given by the table below:*

	$QS(f, t_i, t_{i+1})$	$\Rightarrow$	$QS(f, t_{i+1})$
1.	$\langle l_j, std \rangle$		$\langle l_j, std \rangle$
2.	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle l_{j+1}, std \rangle$
3.	$\langle (l_j, l_{j+1}), inc \rangle$		$\langle l_{j+1}, inc \rangle$
4.	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle l_j, std \rangle$
5.	$\langle (l_j, l_{j+1}), dec \rangle$		$\langle l_j, dec \rangle$

Proof: Similar to the proof of Proposition 3. Cases 1 - 5 here correspond to cases 1, 3, 5, 2, and 4, respectively, in Proposition 3. []

When considering f in the context of a system F, we need transitions appropriate to a time-point t which is distinguished for the system but not for the individual function. In other words, t might reach a distinguished time-point before $f(t)$ reaches its next landmark value.

Proposition 5 *Suppose that an additional time-point $t^* \in [a,b]$ is introduced into the set of distinguished time-points of f:*

$$a = t_0 < \cdots < t_k < t^* < t_{k+1} < \cdots < t_n = b.$$

Then

$$QS(f, t_k, t^*) = QS(f, t^*) = QS(f, t^*, t_{k+1}) = QS(f, t_k, t_{k+1}).$$

Proof: Since, for all $t \in (t_k, t_{k+1})$, $QS(f,t) = QS(f, t_k, t_{k+1})$ by definition, we conclude that $QS(f, t_k, t_{k+1})$ is identical to each of $QS(f, t_k, t^*)$, $QS(f, t^*)$, and $QS(f, t^*, t_{k+1})$. []

Proposition 6 *Let $f : [a,b] \rightarrow \Re$ be a reasonable function, and let $a = t_0 < \cdots < t_n = b$ be a set of time-points including all the distinguished time-points of f, but possibly additional points in $[a,b]$. Then the possible transitions of f consist of those listed in Propositions 3 and 4, plus the following I-transitions and P-transitions:*

$$\begin{array}{ccc}
QS(f, t_i, t_{i+1}) & \Rightarrow & QS(f, t_{i+1}) \\
\\
\langle (l_j, l_{j+1}), inc \rangle & & \langle (l_j, l_{j+1}), inc \rangle \\
\langle (l_j, l_{j+1}), dec \rangle & & \langle (l_j, l_{j+1}), dec \rangle \\
\\
QS(f, t_i) & \Rightarrow & QS(f, t_i, t_{i+1}) \\
\\
\langle (l_j, l_{j+1}), inc \rangle & & \langle (l_j, l_{j+1}), inc \rangle \\
\langle (l_j, l_{j+1}), dec \rangle & & \langle (l_j, l_{j+1}), dec \rangle
\end{array}$$

Proof: Consider an I-transition beginning with $QS(f, t_i, t_{i+1})$. If t_{i+1} is a distinguished time-point of f, the transition is specified by Proposition 4. If t_{i+1} is not a distinguished time-point of f, Proposition 5 shows that the qualitative state remains constant across the transition, providing the above alternatives as the only additional possibilities. If the transition is a P-transition beginning with $QS(f, t_i)$ the proof is similar, depending on whether t_i is a distinguished time-point of f. []

A.1 Discovering a New Landmark

Suppose that $QS(f, t_i, t_{i+1}) = \langle (l_j, l_{j+1}), inc \rangle$. Since there is no landmark in (l_j, l_{j+1}), we can exclude the transition to

$$QS(f, t_{i+1}) = \langle (l_j, l_{j+1}), std \rangle$$

because that would imply that $f'(t_{i+1}) = 0$, making $f(t_{i+1})$ a landmark value, by definition. However, if $l_1 < \ldots < l_k$ is only a partial set of the landmarks of f, then the above transition is possible, but only when $f(t_{i+1}) = l^*$, a landmark value of f such that $l_j < l^* < l_{j+1}$.

In this case, the following partial behavior is possible:

qualitative state	known landmarks
$QS(f, t_i, t_{i+1}) = \langle (l_j, l_{j+1}), inc \rangle$	$l_j < l_{j+1}$
$QS(f, t_{i+1}) = \langle l^*, std \rangle$	$l_j < l^* < l_{j+1}$

$QS(f, t_i, t_{i+1})$ is now seen to be syntactically incorrect, given subsequent information acquired about the true set of landmarks. It should be revised to be $QS(f, t_i, t_{i+1}) = \langle (l_j, l^*), inc \rangle$. Furthermore, it is possible for f to move across l^* several times before encountering the critical point that reveals its existence as a landmark. However, the modifications needed to correct the behavioral description are straight-forward and locally computable. In the Ball system example discussed above, the maximum height of the ball is such a new landmark, discovered when $V(t)$, and therefore $Y'(t)$, become zero. We summarize this discussion in:

Proposition 7 *Suppose that $l_1 < \ldots < l_k$ are all the known landmarks of a reasonable function f, which may have other landmarks as yet unknown.*

Then, in addition to the transitions listed in Propositions 3, 4, and 6 the following I-transitions are possible:

$QS(f, t_i, t_{i+1})$	$\Rightarrow$	$QS(f, t_{i+1})$
$\langle (l_j, l_{j+1}), inc \rangle$		$\langle l^*, std \rangle$
$\langle (l_j, l_{j+1}), dec \rangle$		$\langle l^*, std \rangle$

In case one of these transitions is followed, the set of landmark values in $QS(f, t_{i+1})$ is augmented by $l_j < l^* < l_{j+1}$. Note that the total ordering on the set of landmarks is preserved.

Table 3 on page 18 collects and names the transitions permitted by Propositions 3, 4, 6, and 7, for use in the QSIM algorithm.

A.2 Infinity and Asymptotic Approach

I-transitions express the possible consequences of a changing parameter reaching a limiting landmark value. But what if a parameter approaches its limit asymptotically? By allowing both domain and range to include $+\infty$ and $-\infty$ as endpoints, we can express asymptotic approach as reaching the limit point at $t = \infty$. The same method allows us to treat divergence to infinite values as a possible behavior. Thus, every time-interval has an endpoint, but some distinguished time-points (e.g. $t = \infty$, or t such that $f(t) = \infty$) may have no successor states. There are two constraints on these types of behavior.

First, at $t = \infty$, every function in the system must be equal to some landmark value and must, if that landmark is finite, have derivative zero (i.e. direction of change *std*). Recall that oscillatory systems are handled with a finite domain and repeated states, rather than with an infinite domain.

Proposition 8 *Let $f : [a, \infty] \rightarrow \Re$ be a reasonable function. If the limit of $f(t)$ as $t \rightarrow \infty$ is finite, then* $\lim_{t \rightarrow \infty} f'(t) = 0$.

Proof. If $\lim_{t \rightarrow \infty} f'(t) > 0$, then for some interval (c, ∞), $f'(t)$ must be bounded away from zero. In this interval, $f(t) = f(c) + f'(t^*) * (t - c)$ for some $t^* \in (c, \infty)$, by the Mean Value Theorem. Thus, $\lim_{t \to \infty} f(t) = \infty$. Similarly in case $\lim_{t \to \infty} f'(t) < 0$, so the limit must be zero. []

Second, if $f(t) = \infty$, then $t = b$, the right-hand endpoint of the domain, since a function cannot be continuously differentiable across ∞. If $b < \infty$, then the direction of change must be *inc*.

Proposition 9 *Let $f : [a, b] \to \Re^*$ be a reasonable function such that $\lim_{t \to b} f(t) = \infty$, where b is finite. Then $\lim_{t \to b} f'(t) = \infty$.*

Proof. Suppose that $\lim_{t \to b} f'(t)$ has a finite limit $M > 0$. Then for some interval $(b - \delta, b)$, $f'(t) \in (M - \epsilon, M + \epsilon)$, which implies that

$$f(b - \delta) + \delta * (M - \epsilon) < f(b) < f(b - \delta) + \delta * (M + \epsilon)$$

which contradicts $f(b) = \infty$. []

Using these propositions, we can test whether a distinguished time-point can match $t = \infty$, and test moves to ∞ for consistency. With these observations, the extended reals $[-\infty, \infty]$ can be treated like any other closed interval, and asymptotic approach is handled.

B Constraint Consistency

This appendix specifies the rules by which each type of constraint tests a tuple of qualitative state transitions for consistency. There are separate tests for consistency of the qualitative magnitudes and the directions of change.

B.1 Qualitative Magnitude Consistency

This appendix defines and justifies the evaluation of M^+, M^-, ADD, or $MULT$ constraints when applied to particular qualitative values.

The magnitude of a quantity is described qualitatively in terms of its ordinal relations with a set of landmark values. The validity of a particular application of an M^+, M^-, ADD, or $MULT$ predicate is tested using not only the signs of the arguments, but also their relations with other sets of corresponding values. For example, if we know that $ADD(p,q,r)$ is true, then (p,q,r) is a set of corresponding values for this ADD constraint, and if $p' < p$ and $r' > r$, we can determine that $ADD(p',q,r')$ must be false. In the QSIM algorithm, these predicates are evaluated in order to test the validity of a possible tuple of transitions at a particular constraint.

The criteria below generalize the *Transition Ordering* rules of Williams (1984a, 1984b), and show how possible transition-tuples are compared with known sets of corresponding values. The qualitative state before the transition is presumed to be consistent. Each criterion asks whether the ordinal relations between the current values and the corresponding values will remain consistent with the constraint after the proposed transition.

B.1.1 Monotonic Function Constraints

If two function f and g, related by $M^+(f,g)$, are approaching corresponding limits, we know that either both reach their limits together, or neither does.

Proposition 10 *Suppose $M^+(f,g)$, with corresponding values (p,q), and*

$$QS(f,t_1,t_2) = \langle (p,p'), dec\rangle$$
$$QS(g,t_1,t_2) = \langle (q,q'), dec\rangle$$

Then one of the following two possibilities must be true at t_2*:*

	f	g
(1)	$f(t_2) = p$	$g(t_2) = q$
(2)	$f(t_2) > p$	$g(t_2) > q$.

Proof: Since $M^+(f,g)$ is true, there is a strictly monotonic function H such that $f(t) = H(g(t))$ for all $t \in [a,b]$. In particular, since p and q are corresponding values, $p = H(q)$. Thus, if $g(t_2) = q$, we know that $f(t_2) = p$, and conversely by the symmetry of $M^+(f,g)$. []

If f and g are approaching limits, but only one of the limit points belongs to a corresponding value pair, then only the *other* limit point is possibly reachable in the next state.

Proposition 11 *Suppose* $M^+(f,g)$*, with corresponding values* (p,q)*, and*

$$QS(f,t_1,t_2) = \langle (p,p'), dec\rangle$$
$$QS(g,t_1,t_2) = \langle (q'',q'), dec\rangle$$

where $q'' \neq q$*. Then one of the following two possibilities must be true at* t_2*:*

	f	g
(1)	$f(t_2) > p$	$g(t_2) = q''$
(2)	$f(t_2) > p$	$g(t_2) > q''$.

Proof: Since (p,q) is a corresponding value pair, there must be some $t^* \in [a,b]$ such that $f(t^*) = p$ and $g(t^*) = q$. Notice that it is not possible for $q' \leq q$, because then $g(t) < g(t^*)$ while $f(t) > f(t^*)$ for $t \in (t_1,t_2)$, which contradicts $M^+(f,g)$. Thus $q < q'' < q'$.

f cannot reach p without g simultaneously reaching q, as shown in the previous proposition. Thus the only possibilities are that g reaches q'', or that neither reaches its limit. []

By symmetry, it is clear that analogous propositions hold whether the constraint is M^+ or M^-, or whether the corresponding limits are approached from above, below, or one from each side.

B.1.2 Addition Constraint

We can use exactly the same technique to prove similar consistency criteria for $ADD(f,g,h)$ and $MULT(f,g,h)$, in cases where three, two or only one of the limit points belongs to a known set of corresponding values. The complexity of determining, implementing, and verifying all such tests is formidable. Fortunately, there is a general relationship that captures all possible such criteria.

Proposition 12 *Let p, q, and r be corresponding values of f, g, and h, where $ADD(f,g,h)$. Then, for any $t \in [a,b]$, the following holds:*

$$(f(t)-p)+(g(t)-q)=(h(t)-r). \tag{5}$$

Proof. $f(t)+g(t)=h(t)$ and $p+q=r$. []

The signs of the three terms in equation (5) can be determined directly from the ordinal relations among current values and landmarks, and checked for consistency with the ADD relation by table lookup. The qualitative state of the three terms in equation (5) can only change if at least one of f, g, and h reaches p, q, and r as a limit. A proposed tuple is rejected if the resulting state would fail to satisfy equation (5).

The following propositions demonstrate the effect of this filter on cases where the limits of f, g, and h share three, two, or only one value with a particular correspondence.

Proposition 13 *Suppose $ADD(f,g,h)$, with corresponding values $p+q=r$, and*

$$QS(f,t_1,t_2)=\langle(p,p'),dec\rangle$$
$$QS(g,t_1,t_2)=\langle(q,q'),dec\rangle$$
$$QS(h,t_1,t_2)=\langle(r,r'),dec\rangle$$

Then exactly one of the following four possibilities must be true at t_2:

	f	g	h
(1)	$f(t_2)=p$	$g(t_2)=q$	$h(t_2)=r$
(2)	$f(t_2)=p$	$g(t_2)>q$	$h(t_2)>r$
(3)	$f(t_2)>p$	$g(t_2)=q$	$h(t_2)>r$
(4)	$f(t_2)>p$	$g(t_2)>q$	$h(t_2)>r$

Proof. For $t \in [t_1, t_2]$, $f(t) - p \geq 0$, $g(t) - q \geq 0$, and $h(t) - r \geq 0$. The three terms of equation (5) must have compatible signs, and cannot change discontinuously from their state in (t_1, t_2), so cases 1-4 above are the only possibilities. []

In case we are not so fortunate as to have f, g, and h approaching corresponding limits, we may have two of the functions approaching corresponding limits, and know where the corresponding value of the third function is with respect to its limit. This allows us further to constrain the set of possible next states.

Proposition 14 *Suppose $ADD(f,g,h)$, with corresponding values $p+q=r$, and*

$$\begin{aligned} QS(f,t_1,t_2) &= \langle (p,p'), dec \rangle \\ QS(g,t_1,t_2) &= \langle (q,q'), dec \rangle \\ QS(h,t_1,t_2) &= \langle (r'',r'), dec \rangle \end{aligned}$$

where $r'' \neq r$. Then it is not possible to have both $f(t_2) = p$ and $g(t_2) = q$.

Proof. Consider equation (5). If $r \geq r'$, then the term $h(t) - r$ is negative for $t \in (t_1, t_2)$, while the other two terms are positive, which is a contradiction. Thus $r < r''$.

All terms of equation (5) are positive on (t_1, t_2), and $h(t) - r$ must be strictly positive at $t = t_2$, so at most one of the other terms can be zero at t_2. []

Proposition 15 *Suppose $ADD(f,g,h)$, with corresponding values $p+q=r$, and*

$$\begin{aligned} QS(f,t_1,t_2) &= \langle (p,p'), dec \rangle \\ QS(g,t_1,t_2) &= \langle (q'',q'), dec \rangle \\ QS(h,t_1,t_2) &= \langle (r,r'), dec \rangle \end{aligned}$$

where $q'' \neq q$. Then it is not possible to have both $f(t_2) = p$ and $h(t_2) = r$. If $q < q''$, it is impossible to have $h(t_2) = r$. If $q > q''$, it is impossible for $f(t_2) = p$.

Proof. If $q < q''$, the middle term of equation (5) is strictly positive on $[t_1, t_2]$, so only the first term can possibly become zero at t_2, so only

f and g can possibly reach their limits. If $q > q''$, then the second term of equation (5) is strictly negative on $(t_1, t_2]$, so only the third term can possibly become zero at t_2, so only g and h can possibly reach their limits. $\square$

The same technique can be used in case only one of a set of corresponding values appears in the current set of limits.

Proposition 16 *Suppose $ADD(f, g, h)$, with corresponding values $p + q = r$, and*

$$QS(f, t_1, t_2) = \langle (p, p'), dec \rangle$$
$$QS(g, t_1, t_2) = \langle (q'', q'), dec \rangle$$
$$QS(h, t_1, t_2) = \langle (r'', r'), dec \rangle$$

where $q'' \neq q$ and $r'' \neq r$. Then if $q'' > q$ and $r'' < r$, or if $q'' < q$ and $r'' > r$, it is impossible for $f(t_2) = p$.

Proof. Examination of equation (5) shows that these cases would result in the first term being zero, while the other two have opposite signs, which is impossible. $\square$

Proposition 17 *Suppose $ADD(f, g, h)$, with corresponding values $p + q = r$, and*

$$QS(f, t_1, t_2) = \langle (p'', p'), dec \rangle$$
$$QS(g, t_1, t_2) = \langle (q'', q'), dec \rangle$$
$$QS(h, t_1, t_2) = \langle (r, r'), dec \rangle$$

where $p'' \neq p$ and $q'' \neq q$. Then if $p'' > p$ and $q'' > q$, or if $p'' < p$ and $q'' < q$, it is impossible for $h(t_2) = r$.

Proof. Examination of equation (5) shows that these cases would result in the last term being zero, while the other two have the same signs, which is impossible. $\square$

By symmetry, similar propositions hold in the cases where f, g, and h are approaching their limits from various combinations of directions, not only when all are decreasing. Fortunately, equation (5) makes it unnecessary to implement checks based directly on Propositions 13 - 17.

B.1.3 Multiplication Constraint

If the three functions in a MULT constraint are approaching related limits, we can constrain the possible results, similarly to what we did with ADD constraints in the previous section. A separate consistency test checks for legal combinations of signs $(+, 0, -)$ at a multiplication constraint.

Proposition 18 *Let p, q, and r be non-zero corresponding values of the functions f, g, and h, respectively, where $MULT(f, g, h)$. Then, for any $t \in [a, b]$, the following holds:*

$$\left(\frac{f(t)}{p}\right) * \left(\frac{g(t)}{q}\right) = \left(\frac{h(t)}{r}\right). \tag{6}$$

Proof. $f(t) * g(t) = h(t)$ and $p * q = r$. []

As with the addition constraint, QSIM uses equation (6) directly to test the consistency of various combinations of f, g, and h reaching their limit values, in comparison with a corresponding set of values, $p * q = r$. When $f(t)$ and p have the same sign, we can retrieve their ordinal relations to classify the term $f(t)/p$ as greater than, less than, or equal to 1. With respect to this classification, the legal combinations of A, B, and C, where $MULT(A, B, C)$ are given by the following table:

C		B		
		< 1	$= 1$	> 1
	< 1	< 1	< 1	*any*
A	$= 1$	< 1	$= 1$	> 1
	> 1	*any*	> 1	> 1

Note that it is not necessary for the value 1 to be a landmark value of any of the functions involved. The table is a guide to the implementation of a consistency test for $MULT(f, g, h)$, rather than representing an inference that QSIM makes explicitly. The consistency test applies when the ordinal relation between $f(t)$ and p changes as f reaches its limit.

Propositions can be proved to demonstrate the degree of filtering possible with different sets of corresponding values, similar to Propositions 13 through 17 for addition, but they are omitted here.

B.2 Direction-of-Change Consistency

An important difference between QSIM and the algorithms used by de Kleer and Forbus is that quantities are represented by qualitative *descriptions* rather than qualitative *values*. Thus, rather than being a partial function that sometimes fails to compute a result, ADD is a three-place relation evaluating to true or false according to whether its arguments satisfy the addition constraint. This appendix specifies the tables of acceptable directions of change for ADD and MULT. The corresponding tables for M^+ and M^- are obvious.

B.2.1 $ADD(f, g, h)$

The following table summarizes the combinations of directions of change that satisfy the $ADD(f, g, h)$ constraint.

h		*g*		
		inc	*std*	*dec*
	inc	*inc*	*inc*	*any*
f	*std*	*inc*	*std*	*dec*
	dec	*any*	*dec*	*dec*

B.2.2 $MULT(f, g, h)$

The combinations of directions of change that satisfy the MULT constraint depend on the signs of f, g, and h, as shown in the following tables, derived from the identity $h' = f'g + fg'$.

1. If $f > 0$, $g > 0$, $h > 0$,

h		*g*		
		inc	*std*	*dec*
	inc	*inc*	*inc*	*any*
f	*std*	*inc*	*std*	*dec*
	dec	*any*	*dec*	*dec*

2. If $f < 0$, $g < 0$, $h > 0$,

h		g		
		inc	*std*	*dec*
	inc	*dec*	*dec*	*any*
f	*std*	*dec*	*std*	*inc*
	dec	*any*	*inc*	*inc*

3. If $f > 0$, $g < 0$, $h < 0$,

h		g		
		inc	*std*	*dec*
	inc	*any*	*dec*	*dec*
f	*std*	*inc*	*std*	*dec*
	dec	*inc*	*inc*	*any*

In case $f < 0$ and $g > 0$, the table is transposed.

4. If $f > 0$, $g = 0$, $h = 0$,

h		g		
		inc	*std*	*dec*
	inc	*inc*	*std*	*dec*
f	*std*	*inc*	*std*	*dec*
	dec	*inc*	*std*	*dec*

In case $f < 0$, the table remains the same, but with the signs reversed. If $f = 0$ and $g \neq 0$, the table is transposed.

5. If $f = 0$, $g = 0$, $h = 0$,

h		*g*		
		inc	*std*	*dec*
	inc	*std*	*std*	*std*
f	*std*	*std*	*std*	*std*
	dec	*std*	*std*	*std*

C The QSIM Program and its Output

This appendix provides the structure-definition, the initialization, and a complete trace of the output of QSIM on the Spring mechanism.

```
(define-structure spring
  (functions a v x)
  (landmarks (v (minf 0 v* inf)))
  (constraints (d//dt v a) (d//dt x v) (m0- a x))
  (invariants
    (a ((minf inf) nil))
    (v ((minf inf) nil))
    (x ((minf inf) nil))))

(defun initialize-spring ()
  (make-initialization spring (generate-time-point)
                       '((x (0 inc))
                         (v (v* std))
                         (a (0 dec)))))
```

The "paragraphs" of the following trace output each predict the successors of a given state. The sections of a paragraph describe:

- The initial qualitative state description;
- The number of qualitative state transitions assigned to each individual function;
- The decrease in number of transition tuples as each constraint applies its filters;
- The effect of Waltz filtering when consideration of an adjacent constraint decreases the number of tuples;
- The effect of global filters;

- The assignment(s) of transition rules that will create the successor state(s).

```
Completing S1.
Completed S1 in 1 iterations.

Predicting successors of S1 in region SPRING.
  0 = X [INC]
  V* = V [STD]
  0 = A [DEC]
Function A has 1 transitions.
Function V has 3 transitions.
Function X has 1 transitions.
Constraint M-(A X) has (1) -> (1) tuples.
Constraint D//DT(X V) has (3) -> (3) tuples.
Constraint D//DT(V A) has (3) -> (1) tuples.
Waltz filter of D//DT(X V):  (3) -> (2).
Waltz filter of D//DT(X V):  (2) -> (1).
Global interpretations:  (1) -> (1).
Predicting:
   V: P3 (L1 STD) => ((L0 L1) DEC)
   X: P4 (L1 INC) => ((L1 L2) INC)
   A: P6 (L1 DEC) => ((L0 L1) DEC)
 => 1 successors.

Predicting successors of S2 in region SPRING.
  0 < V[DEC] < V*
  0 < X[INC] < INF
  MINF < A[DEC] < 0
Function A has 2 transitions.
Function V has 4 transitions.
Function X has 2 transitions.
Constraint M-(A X) has (4) -> (2) tuples.
Constraint D//DT(X V) has (8) -> (4) tuples.
Constraint D//DT(V A) has (8) -> (4) tuples.
```

```
Waltz filter of D//DT(X V):  (4) -> (3).
Waltz filter of D//DT(X V):  (3) -> (2).
Global interpretations:  (2) -> (1).
Predicting:
   V: I6 ((M1 M2) DEC) => (M1 DEC)
   X: I8 ((M1 M2) INC) => (M* STD)
   A: I9 ((M1 M2) DEC) => (M* STD)
 => 1 successors.

Predicting successors of S3 in region SPRING.
  0 = V [DEC]
  X1 = X [STD]
  A1 = A [STD]
Function A has 3 transitions.
Function V has 1 transitions.
Function X has 3 transitions.
Constraint M-(A X) has (9) -> (3) tuples.
Constraint D//DT(X V) has (3) -> (1) tuples.
Constraint D//DT(V A) has (3) -> (3) tuples.
Waltz filter of M-(A X):  (3) -> (2).
Waltz filter of M-(A X):  (2) -> (1).
Waltz filter of D//DT(V A):  (3) -> (2).
Waltz filter of D//DT(V A):  (2) -> (1).
Global interpretations:  (1) -> (1).
Predicting:
   V: P6 (L1 DEC) => ((L0 L1) DEC)
   X: P3 (L1 STD) => ((L0 L1) DEC)
   A: P2 (L1 STD) => ((L1 L2) INC)
 => 1 successors.

Predicting successors of S4 in region SPRING.
  MINF < V[DEC] < 0
  0 < X[DEC] < X1
  A1 < A[INC] < 0
Function A has 4 transitions.
```

```
Function V has 2 transitions.
Function X has 4 transitions.
Constraint M-(A X) has (16) -> (4) tuples.
Constraint D//DT(X V) has (8) -> (4) tuples.
Constraint D//DT(V A) has (8) -> (4) tuples.
Waltz filter of M-(A X):  (4) -> (3).
Waltz filter of M-(A X):  (3) -> (2).
Waltz filter of D//DT(V A):  (4) -> (3).
Waltz filter of D//DT(V A):  (3) -> (2).
Global interpretations:  (2) -> (1).
Predicting:
   V: I9 ((M1 M2) DEC) => (M* STD)
   X: I6 ((M1 M2) DEC) => (M1 DEC)
   A: I3 ((M1 M2) INC) => (M2 INC)
 => 1 successors.

Predicting successors of S5 in region SPRING.
  V1 = V [STD]
  0 = X [DEC]
  0 = A [INC]
Function A has 1 transitions.
Function V has 3 transitions.
Function X has 1 transitions.
Constraint M-(A X) has (1) -> (1) tuples.
Constraint D//DT(X V) has (3) -> (3) tuples.
Constraint D//DT(V A) has (3) -> (1) tuples.
Waltz filter of D//DT(X V):  (3) -> (2).
Waltz filter of D//DT(X V):  (2) -> (1).
Global interpretations:  (1) -> (1).
Predicting:
   V: P2 (L1 STD) => ((L1 L2) INC)
   X: P6 (L1 DEC) => ((L0 L1) DEC)
   A: P4 (L1 INC) => ((L1 L2) INC)
 => 1 successors.
```

```
Predicting successors of S6 in region SPRING.
  V1 < V[INC] < 0
  MINF < X[DEC] < 0
  0 < A[INC] < INF
Function A has 2 transitions.
Function V has 4 transitions.
Function X has 2 transitions.
Constraint M-(A X) has (4) -> (2) tuples.
Constraint D//DT(X V) has (8) -> (4) tuples.
Constraint D//DT(V A) has (8) -> (4) tuples.
Waltz filter of D//DT(X V):  (4) -> (3).
Waltz filter of D//DT(X V):  (3) -> (2).
Global interpretations:  (2) -> (1).
Predicting:
   V: I3 ((M1 M2) INC) => (M2 INC)
   X: I9 ((M1 M2) DEC) => (M* STD)
   A: I8 ((M1 M2) INC) => (M* STD)
 => 1 successors.

Predicting successors of S7 in region SPRING.
  0 = V [INC]
  X2 = X [STD]
  A2 = A [STD]
Function A has 3 transitions.
Function V has 1 transitions.
Function X has 3 transitions.
Constraint M-(A X) has (9) -> (3) tuples.
Constraint D//DT(X V) has (3) -> (1) tuples.
Constraint D//DT(V A) has (3) -> (3) tuples.
Waltz filter of M-(A X):  (3) -> (2).
Waltz filter of M-(A X):  (2) -> (1).
Waltz filter of D//DT(V A):  (3) -> (2).
Waltz filter of D//DT(V A):  (2) -> (1).
Global interpretations:  (1) -> (1).
Predicting:
```

```
   V: P4 (L1 INC) => ((L1 L2) INC)
   X: P2 (L1 STD) => ((L1 L2) INC)
   A: P3 (L1 STD) => ((L0 L1) DEC)
 => 1 successors.

Predicting successors of S8 in region SPRING.
  0 < V[INC] < V*
  X2 < X[INC] < 0
  0 < A[DEC] < A2
Function A has 4 transitions.
Function V has 4 transitions.
Function X has 4 transitions.
Constraint M-(A X) has (16) -> (4) tuples.
Constraint D//DT(X V) has (16) -> (8) tuples.
Constraint D//DT(V A) has (16) -> (8) tuples.
Waltz filter of M-(A X):  (4) -> (3).
Waltz filter of M-(A X):  (3) -> (2).
Waltz filter of D//DT(V A):  (8) -> (6).
Waltz filter of D//DT(V A):  (6) -> (4).
Global interpretations:  (4) -> (3).
Predicting:
   V: I3 ((M1 M2) INC) => (M2 INC)
   X: I4 ((M1 M2) INC) => ((M1 M2) INC)
   A: I7 ((M1 M2) DEC) => ((M1 M2) DEC)
Predicting:
   V: I8 ((M1 M2) INC) => (M* STD)
   X: I3 ((M1 M2) INC) => (M2 INC)
   A: I6 ((M1 M2) DEC) => (M1 DEC)
Predicting:
   V: I2 ((M1 M2) INC) => (M2 STD)
   X: I3 ((M1 M2) INC) => (M2 INC)
   A: I6 ((M1 M2) DEC) => (M1 DEC)
Found a cycle from S1 to S11.
 => 3 successors.
No active states.
```

```
Runtime: 0.4 seconds.
Total number of states 10.
```

The simulation concludes here when three possible states are generated for the final state of the cycle, due to a heuristic rule that recognizes the existence of a cycle and deactivates its competitor branches. The alternate branches can be reactivated by hand for further exploration.

The following "qualitative graphs" may also be automatically generated, and provide a convenient way to inspect the output of QSIM. The graph below shows only the main, periodic, behavior of the Spring system.

```
INF                                                       INF

A2                                          =             A2
                                       ^         v
0         ----v-------------------^-------------------v   0
                    v         ^
A1                       =                                A1

MINF                                                      MINF
          Acceleration


V*            =                                       =   V*
                    v                            ^
0         --------------v-------------------^----------   0
                              v        ^
V1                                =                       V1

MINF                                                      MINF
          Velocity


INF                                                       INF

X1                       =                                X1
                    ^         v
0         ----^-------------------v-------------------^   0
                                       v         ^
X2                                          =             X2

MINF                                                      MINF
          Position
```

C.1 Availability of the QSIM Program

The QSIM program is available to researchers interested in qualitative simulation. We are making the code available in order to encourage detailed exploration and evaluation of these ideas and their possible applications beyond what is possible in a published paper. The current implementation runs in ZetaLisp on the Symbolics 3600 and the CADR, and with a reduced graphical interface on the VAX in NIL.

It is often difficult to evaluate ideas in artificial intelligence without experimenting with the strengths and limitations of a program implementing those ideas. A paper making a particular set of theoretical points cannot fully describe the capabilities and limitations of a complex algorithm, particularly as they relate to application areas of interest to particular readers.

Experiments are not often reproduced in AI because of the substantial overhead in writing and debugging a sophisticated program. Distributing the code makes it possible for the scientific community to evaluate the program and the method for cases beyond those presented in publications. As such, distributing the code functions as a form of secondary publication.

The QSIM program is made available to researchers in this spirit, in return for comments, criticism, counterexamples, and bug reports. It is not advertised or warranted as a software product, and all commercial rights to the program are retained. Please contact me if you are interested.