Extracting and Using Relative Duration Information in Pure Qualitative Simulation

Tolga Könik and A. C. Cem Say

Artificial Intelligence Laboratory
Department of Computer Engineering, Boğaziçi University
Bebek 80815, İstanbul, Turkey
konik@iname.com, say@boun.edu.tr

Abstract

We show that qualitative simulation algorithms can make better use of their input to deduce significant amounts of information about the relative lengths of the time intervals in their output behavior predictions. Simple techniques employing concepts like symmetry and periodicity, and comparison of the circumstances during multiple traversals of the same interval can enable the reasoner to build a list of facts representing the deduced information about relative durations. These facts are used by a new filter, which eliminates proposed spurious behaviors leading to inconsistent duration data. Surviving behaviors are annotated with richer descriptions of the qualitative properties of system variables, in addition to the extracted relative duration information.

Introduction

The prediction of spurious solutions for some qualitative differential equation systems is a major problem of qualitative simulation. Improvements in this area involve the development of methods which increase the mathematical and representational sophistication of qualitative simulators to eliminate different classes of spurious predictions (Kuipers 1994) (Say & Kuru 1993) (Say 1997b) (Say 1998). In this paper, we show that qualitative simulation algorithms can make better use of their input to deduce significant amounts of information about the relative lengths of the time intervals in their output behavior predictions. Simple techniques employing concepts like symmetry and periodicity, and comparison of the circumstances during multiple traversals of the same interval can enable the reasoner to build a list of facts representing the deduced information about relative durations. These facts are used by a new filter, which eliminates proposed spurious behaviors leading to inconsistent duration data. Surviving behaviors are annotated with richer descriptions of the qualitative properties of system variables, in addition to the extracted relative duration information.

We have implemented our technique in the framework of the "standard" qualitative simulation algorithm QSIM, details on which can be found in (Kuipers 1994).

The Idea

As an example to the sort of problem solved by our work, consider the following scenario: Two balls are thrown upward from ground level with unknown speeds at time t_0. We are interested in enumerating all (and only) the physically possible orderings of the time-points in which the balls reach the highest points of their trajectories or hit the ground. We simulate the simple QSIM model in Table 1. The simulator is set to stop extending a prediction when either ball hits the ground, that is, at time-points where H_1 or H_2 has the value <0, ↓>.

Name	Explanation	Constraint
A	upward gravitational acceleration	(constant A $g_0 < 0$)
V_1	upward velocity of the first ball	(d/dt V_1 A)
V_2	upward velocity of the second ball	(d/dt V_2 A)
H_1	height of the first ball	(d/dt H_1 V_1)
H_2	height of the second ball	(d/dt H_2 V_2)

TABLE 1. The Two-Ball System

The QSIM algorithm predicts 13 distinct behaviors in this simulation. Table 2 depicts one of these predictions. It is easy to see that this is a spurious prediction, since it describes a behavior in which it takes the balls the same time to reach their maximum heights, but then the first ball overtakes the second ball in the next half of what is clearly a symmetric trajectory. There are five other similarly inconsistent predictions in this QSIM output.

time	A	V_1	V_2	H_1	H_2
t_0	g_0, ⊙	$(0,\infty)$, ↓	$(0,\infty)$, ↓	0, ↑	0, ↑
(t_0, t_1)	g_0, ⊙	$(0,\infty)$, ↓	$(0,\infty)$, ↓	$(0,\infty)$, ↑	$(0,\infty)$, ↑
t_1	g_0, ⊙	0, ↓	0, ↓	h_1^*, ⊙	h_2^*, ⊙
(t_1, t_2)	g_0, ⊙	$(-\infty,0)$, ↓	$(-\infty,0)$, ↓	$(0, h_1^*)$, ↓	$(0, h_2^*)$, ↓
$t_2 < \infty$	g_0, ⊙	$(-\infty,0)$, ↓	$(-\infty,0)$, ↓	0, ↓	$(0, h_2^*)$, ↓

TABLE 2. A Spurious Prediction for the Two-Ball System

What modifications have we made to avoid this error? In this example, one can deduce that the heights are symmetric functions of time around the point t_1 by examining the constraint model and the qualitative state at

t_1. We have incorporated a routine, which checks the current workspace to discover such symmetry information about variables after the creation of each time-point state by the simulator. These symmetry data can be used later to derive relative length information about the time intervals in the computed behavior. For instance, during the creation of the state labelled t_2 in Table 2, the symmetry property of H_1 can be exploited to deduce that the time intervals (t_0, t_1) and (t_1, t_2) should be of equal length. A similar reasoning about H_2 indicates that (t_0, t_1) is longer than (t_1, t_2). The *relative duration facts* about intervals obtained in this manner are accumulated in a global data structure associated with each behavior. Each candidate time-point state has to pass our new *duration consistency filter*, which is satisfied only if no inconsistency can be found in the set of relative duration facts implied by the partial behavior that would be constructed by the addition of this candidate state. In the example of Table 2, the state t_2 would not pass this filter because of the two inconsistent assertions about $|t_0, t_1|$ and $|t_1, t_2|$, and so that spurious behavior would not be predicted.

In the following sections, we describe how to augment the qualitative simulation algorithm so that it notices and uses several different mathematical properties (including symmetry) of the computed behavior prefixes to eliminate a class of spurious predictions containing such durational inconsistencies and to present relative length information about the time intervals in the predicted behaviors.

Symmetric Functions

Symmetry is an important qualitative property. In the next section, we describe how the input model can be used to deduce the existence of symmetric functions in a partial behavior. This section is an introduction to the terminology and mathematics that will be employed during that procedure.

Definition 1. If a function $f(t)$ has, for a given point t_i in its domain $[a,b]$, the property that

$$f(t_i - s) = f(t_i + s),$$

$$\lim_{\mu \to s^+} f(t_i - \mu) = \lim_{\mu \to s^+} f(t_i + \mu), \text{ and}$$

$$\lim_{\mu \to s^-} f(t_i - \mu) = \lim_{\mu \to s^-} f(t_i + \mu)$$

for all s such that $t_i - s \in (a,b)$ and $t_i + s \in (a,b)$, then f is said to be *even symmetric around* t_i, denoted $even(f, t_i)$.

The positive legal range for s described above, namely, $(0, min(t_i - a, b - t_i))$, is said to be the *symmetry radius around* t_i.

Definition 2. If a function $f(t)$ has, for a given point t_i in its domain $[a, b]$, the property that

$$f(t_i - s) = -f(t_i + s),$$

$$\lim_{\mu \to s^+} f(t_i - \mu) = -\lim_{\mu \to s^+} f(t_i + \mu), \text{ and}$$

$$\lim_{\mu \to s^-} f(t_i - \mu) = -\lim_{\mu \to s^-} f(t_i + \mu)$$

for all s such that $t_i - s \in (a, b)$ and $t_i + s \in (a, b)$, then f is said to be *odd symmetric around* t_i, denoted $odd(f, t_i)$.

If a function f is (even or odd) symmetric around t_i, t_i is said to be f's *symmetry point*. In the remainder of this section, all appearances of s are assumed to be universally quantified over the symmetry radius around the symmetry point under discussion.

Note that the function $x(t) \equiv 0$ is both even and odd symmetric everywhere in its domain.

The following theorems establish the correctness of a set of rules used by the symmetry recognition procedure incorporated to QSIM. (Könik & Say 1998)

Theorem 1. If $f(t)$ is continuous on the domain $[a,b]$, then

(i) $even(f, t_i) \leftrightarrow f(t_i - s) = f(t_i + s)$
(ii) $odd(f, t_i) \leftrightarrow f(t_i - s) = -f(t_i + s)$

Theorem 2. Given $y(t) = f(x(t))$,

(i) $even(x, t_i) \rightarrow even(y, t_i)$
(ii) $odd(x, t_i) \wedge odd(f, 0) \rightarrow odd(y, t_i)$

Theorem 3. $x(t) = k$, where k is a nonzero constant, is even symmetric at every point.

Theorem 4. Given $x(t) = y(t) + z(t)$,

(i) $even(y, t_i) \wedge even(z, t_i) \rightarrow even(x, t_i)$
(ii) $even(x, t_i) \wedge even(z, t_i) \rightarrow even(y, t_i)$
(iii) $even(x, t_i) \wedge even(y, t_i) \rightarrow even(z, t_i)$
(iv) $odd(y, t_i) \wedge odd(z, t_i) \rightarrow odd(x, t_i)$
(v) $odd(x, t_i) \wedge odd(z, t_i) \rightarrow odd(y, t_i)$
(vi) $odd(x, t_i) \wedge odd(y, t_i) \rightarrow odd(z, t_i)$

Theorem 5. Given $x(t) = y(t) . z(t)$,

(i) $even(y, t_i) \wedge even(z, t_i) \rightarrow even(x, t_i)$
(ii) $even(x, t_i) \wedge even(z, t_i) \rightarrow even(y, t_i)$
(iii) $even(x, t_i) \wedge even(y, t_i) \rightarrow even(z, t_i)$
(iv) $odd(y, t_i) \wedge odd(z, t_i) \rightarrow even(x, t_i)$
(v) $odd(x, t_i) \wedge odd(z, t_i) \rightarrow even(y, t_i)$
(vi) $odd(x, t_i) \wedge odd(y, t_i) \rightarrow even(z, t_i)$
(vii) $even(x, t_i) \wedge odd(y, t_i) \rightarrow odd(z, t_i)$
(viii) $even(x, t_i) \wedge odd(z, t_i) \rightarrow odd(y, t_i)$
(ix) $even(y, t_i) \wedge odd(x, t_i) \rightarrow odd(z, t_i)$
(x) $even(y, t_i) \wedge odd(z, t_i) \rightarrow odd(x, t_i)$
(xi) $even(z, t_i) \wedge odd(x, t_i) \rightarrow odd(y, t_i)$
(xii) $even(z, t_i) \wedge odd(y, t_i) \rightarrow odd(z, t_i)$

Theorem 6. Given $y(t) = f(x(t))$, where $f \in M^+ \cup M^-$,

(i) $even(y, t_i) \leftrightarrow even(x, t_i)$

(ii) If $odd(f, 0)$ $(f(-x) = -f(x))$ then $odd(y, t_i) \leftrightarrow odd(x, t_i)$

Theorem 7. Given $x = \frac{dy}{dt}$,

(i) $even(y, t_i) \leftrightarrow odd(x, t_i)$

(ii) $odd(y, t_i) \leftrightarrow even(x, t_i) \wedge y(t_i)=0$

How can symmetry information be exploited for comparing durations? Note that the definition of a function x being even symmetric around t_i entails that

$$x(t_i - s) = k \ \leftrightarrow \ x(t_i + s) = k,$$

which, when translated to the QSIM representation, means the following: If we "see" x to be at a landmark k at a time-point t_a before t_i, then x is "destined" to reach k again at some point t_c after t_i (unless the simulation terminates for another reason.) Furthermore, we can conclude that $|t_a, t_i|=|t_i, t_c|$, and, of course, $|t_a, t_i| < |t_i, t_b|$ for any t_b in which x has not yet reached k.

For example, assume that x, as illustrated in Figure 1, has been discovered to be even symmetric at time-point t_6, and the list of landmarks crossed by x in $[t_0, t_6)$ is $\{x_a, 0, x_b, 0\}$. "x_c" is a new landmark discovered at the symmetry point t_6. In the continuation of this behavior, it is certain that x will cross the landmarks listed above in the reverse order; namely, $\{0, x_b, 0, x_a\}$. Whenever x arrives at a landmark in this new list, we will be sure that exactly the same amount of time has elapsed from t_i as it took x to reach the symmetry point from the corresponding appearance of that landmark before the symmetry point. (Note that no new landmarks can be created after the symmetry point until all landmarks in that list have been crossed.)

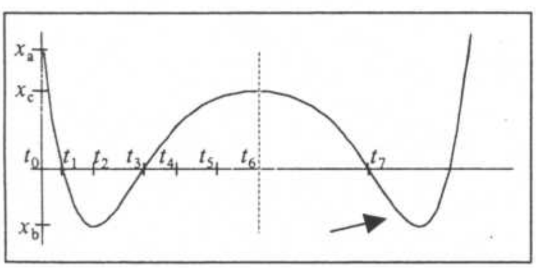

FIGURE 1. An Even Symmetric Variable

For odd symmetric functions, zero crossings contribute relative duration data. To see this, we consider the definition of odd symmetry around t_i, that is,

$$f(t_i - s) = -f(t_i + s)$$

This entails

$$f(t_i - s) = 0 \ \leftrightarrow \ f(t_i + s) = 0.$$

Qualitative directions of odd symmetric variables are useful too. Since the derivative of an odd symmetric variable f will be even symmetric around the symmetry point t_i, it must be the case that

$$f'(t_i - s) = 0 \ \leftrightarrow \ f'(t_i + s) = 0,$$

which means that the qualitative direction of x becoming steady s units before t_i forces a "mirror-event" where x stops again s units after t_i.

The next section illustrates the algorithm for extracting the relative duration facts in more detail.

Recognizing & Using Symmetries in QSIM

The theorems in the previous section describe the ways in which symmetry information about functions can be propagated through a model. The only way of obtaining symmetry information from "scratch," as it were, is provided by Theorem 3. In our modifications which enable QSIM to recognize symmetric variables, the results of Theorems 3-7 are used as rules which add new symmetry data whenever they are able to "fire" in a given state.

We will describe the working of the symmetry recognition procedure in terms of our introductory example about the two-ball system. Before the start of simulation, a preprocessor checks the constraint model to see if the rule of Theorem 3 can be applied to deduce any symmetry information about the variables. At this stage, the only constant function in the model, A is found to be even symmetric (everywhere) by an application of that rule. No such information about the other variables can be deduced at this point. This single item of symmetry information is placed into the *symmetry list*, a structure that will be inherited by all behaviors, which are continuations of this state.

An examination of the rules of the previous section shows that new firings are possible only in time-points where a variable has the value zero. Since zero-crossing leads to a new time-point state in the qualitative simulation setup, we can make maximum use of the symmetry derivation rules if we run them just for each completed time-point state. Our modified algorithm therefore submits each time-point state to the set of symmetry rules, and any new symmetry information obtained as a result is added to the symmetry list associated with the current behavior.

In our example, the state t_1 causes the reasoning steps described in Table 3 to be performed.

Trigger	Fired rule	Conclusion
A is even everywhere and $V_1(t_1) = 0$	7.ii	V_1 is odd around t_1
A is even everywhere and $V_2(t_1) = 0$	7.ii	V_2 is odd around t_1
V_1 is odd around t_1	7.i	H_1 is even around t_1
V_2 is odd around t_1	7.i	H_2 is even around t_1

TABLE 3. Derivation of New Symmetries from the State at t_1

Further simulation of this model does not lead to the discovery of any new symmetry information.

Each candidate time-point state is examined by our algorithm to see if it contributes any new relative duration facts due to previously discovered symmetries. For this purpose, we make use of the fact that the behavior of a symmetric variable up to the symmetry point determines a prefix of that variable's future behavior, as explained in the previous section.

Our algorithm uses the reasoning described in that section to assert new relative duration facts. Each symmetric variable past its symmetry point can contribute one such fact at each time-point. For the even symmetric variable of Figure 1, assume that we are considering a candidate state for t_8, after a partial behavior in which x has been simulated to move up to the interval marked by the arrow in the figure. The algorithm first prepares a list of <landmark, time-point> tuples crossed by x from the beginning of the simulation up to the currently considered time-point. This list, $\{<x_a, t_0>, <0, t_1>, <x_b, t_2>, <0, t_3>, <x_c, t_6>, <0, t_7>\}$, is split through the symmetry point into two lists representing the landmarks crossed before and after the symmetry point, respectively. In our example, the "before" list is $\{<x_a, t_0>, <0, t_1>, <x_b, t_2>, <0, t_3>\}$, and the "after" list is $\{<0, t_7>\}$. We then "subtract" the "after" list from the "before" list (cancelling "mirror-image" landmark appearances from both lists) to obtain the "reverse expectation list" $\{<x_a, t_0>, <0, t_1>, <x_b, t_2>\}$. This means that the "expected landmark" to be crossed by x is x_b, and (t_6, t_8) will be deduced to be of the same length as (t_2, t_6) if $x(t_8)$ is indeed x_b. If, on the other hand, t_8 is created as a result of another variable reaching a landmark and x is still $(x_b, 0)$ at that time-point, the fact "$|t_2, t_6| > |t_6, t_8|$" will be asserted.

Odd symmetric functions, which contribute useful duration information when they cross zero and/or "stop," as explained in the previous section, are treated using a variant of the procedure described above.

Symmetries of "non-analytic" functions, which stay at the same landmark value for a finite time interval during their behavior, are handled in a somewhat more sophisticated way by the duration fact extraction algorithm.

Returning to our two-balls example, the duration fact extraction procedure works as follows when it is called during the creation of state t_2 of Table 2: Variable H_1 is known to be even symmetric around t_1, and its "before" list indicates that it is supposed to reach zero exactly $|t_1-t_0|$ time units after t_1. The proposed magnitude of zero for H_1 causes the assertion of $|t_0, t_1| = |t_1, t_2|$ to the relative duration fact list. A similar reasoning about H_2 adds $|t_0, t_1| > |t_1, t_2|$ to the same data structure.

Other Ways of Comparing Durations

Periodicity

The QSIM algorithm already has a cycle detection feature which lets it decide that a branch of the state tree corresponds to a periodic behavior and therefore need not be expanded any more. Every further traversal of the cycle will be of the length $|t_a, t_b|$, where t_a and t_b are the time-points in which the two instances of the same state that lead to the detection of the cycle appear for the first and second times, respectively.

Some sets of constraints are known to model systems with periodic behaviors, the most famous example being the spring-mass model (Kuipers 1994) of Table 4.

Assume that the three constraints in Table 4 appear in a bigger model containing several other constraints and variables. It is clear that the three variables X, V, and A now form three "clocks" with the same period. Barring the case where all three have the value <0, ⊙> at t_0, the subsystem comprising them will oscillate throughout the behavior of the overall system, "ticking" at time-points where either V or both X and A reach their critical points. This property can be exploited for our purposes. A preprocessor would scan the constraint model for known patterns to see if any embedded clock subsystems can be identified. If such a clock were found, its variables would be noted for future use. During the global filtering of each time-point state, the current behavior prefix would be examined to see if one of the noted variables has "ticked," contributing a new relative duration fact to be used by the duration consistency constraint.

Name	Explanation	
X	displacement of mass from equilibrium	
V	velocity of mass	(d/dt X V)
A	acceleration of mass	(d/dt V A) ((M⁻ X A) (0 0))

TABLE 4. A Periodic Subsystem Model

Multiple Traversals of the Same Interval

Yet another opportunity for comparing durations arises in the following setup: Assume that the system contains four variables x_1, x_2, v_1, and v_2, such that $v_1=dx_1/dt$ and $v_2=dx_2/dt$. Two durations $|t_{1b}, t_{1e}|$ and $|t_{2b}, t_{2e}|$ can be compared if the "distance" covered by x_1 during (t_{1b}, t_{1e}) can be compared with the distance covered by x_2 during (t_{2b}, t_{2e}), and, the average magnitude of v_1 during (t_{1b}, t_{1e}) can be compared with the average magnitude of v_2 during (t_{2b}, t_{2e}).

The basic reasoning process involved here is the one behind intuitive statements such as "It takes longer to traverse a longer path with a lower speed." We will now formalize this approach. Let us start with the following definitions:

$|\Delta x_1| = |x_1(t_{1e}) - x_1(t_{1b})|$: distance travelled by x_1 in (t_{1b}, t_{1e})

$|\Delta x_2| = |x_2(t_{2e}) - x_2(t_{2b})|$: distance travelled by x_2 in (t_{2b}, t_{2e})

$\Delta t_1 = |t_{1e} - t_{1b}|$: length of the time interval (t_{1b}, t_{1e})

$\Delta t_2 = |t_{2e} - t_{2b}|$: length of the time interval (t_{2b}, t_{2e})

$|\bar{v}_1| = |\Delta x_1| / \Delta t_1$: average speed of x_1 in (t_{1b}, t_{1e})

$|\bar{v}_2| = |\Delta x_2| / \Delta t_2$: average speed of x_2 in (t_{2b}, t_{2e})

To compare these quantities, we make the following definitions.

$$\Delta|\Delta x| = |\Delta x_2| - |\Delta x_1| , \quad \Delta\Delta t = \Delta t_2 - \Delta t_1 , \quad \Delta|\bar{v}| = |\bar{v}_2| - |\bar{v}_1|$$

We now derive the comparison formula.

$$\Delta|\Delta x| = |\Delta x_2| - |\Delta x_1| = |\bar{v}_2| . \Delta t_2 - |\bar{v}_1| . \Delta t_1 =$$
$$|\bar{v}_2| . \Delta t_2 - |\bar{v}_1| . \Delta t_2 + |\bar{v}_1| . \Delta t_2 - |\bar{v}_1| . \Delta t_1 =$$
$$(|\bar{v}_2| - |\bar{v}_1|). \Delta t_2 + |\bar{v}_1| (\Delta t_2 - \Delta t_1) = \Delta|\bar{v}| . \Delta t_2 + |\bar{v}_1| . \Delta\Delta t$$

Since we are interested only in the signs of these quantities,

$[\Delta|\Delta x|] = [\Delta|\bar{v}|] . [\Delta t_2] + [|\bar{v}_1|] . [\Delta\Delta t]$, and, since $[\Delta t_2] = [+]$,
$[\Delta|\Delta x|] = [\Delta|\bar{v}|] + [|\bar{v}_1|] . [\Delta\Delta t]$, yielding

$$[\Delta\Delta t] = [\Delta|\Delta x|] - [\Delta|\bar{v}|] \quad \text{if } |\bar{v}_1| > 0. \qquad (1)$$

Note that we can now check the correctness of the statement "It takes longer to traverse a longer path with a lower speed" by seeing whether it satisfies Equation (1): The assignment of signs results in [+] = [+] – [–],which is indeed correct. ($|\bar{v}_1| > 0$ in this case, since the sentence implies that $|\bar{v}_2|$ is less than $|\bar{v}_1|$.)

Applying Equation (1) for duration fact extraction in QSIM is possible when $[\Delta|\Delta x|]$, $[\Delta|\bar{v}|]$, and $|\bar{v}_1|$ can be unambiguously computed from the information at hand, which is feasible in certain restricted cases:

$[\Delta|\Delta x|]$ can be evaluated when x_1 and x_2 are the same variable, say x, (which forces v_1 and v_2 to be a single "velocity" variable as well,) and the landmark interval spanned by x in one of (t_{1b}, t_{1e}) and (t_{2b}, t_{2e}) is a subset of the other one. So our technique boils down to comparing two traversals of the same interval by the same variable.

Comparison of the average speeds is performed via ordinal comparisons on upper and lower bounds. For example, if we know that the velocity is positive in both (t_{1b}, t_{1e}) and (t_{2b}, t_{2e}) (meaning $|\bar{v}_1| > 0$,) and the minimum value attained by it during (t_{1b}, t_{1e}) is greater than its maximum value during (t_{2b}, t_{2e}), we can conclude that $|\bar{v}_1| > |\bar{v}_2|$, and hence $\Delta|\bar{v}| = [-]$.

In certain (rather unlikely) circumstances, it is possible to compare landmark intervals of separate variables in pure QSIM; see (Say 1997a) for a discussion of these issues.

The Duration Consistency Constraint

The duration consistency constraint operates on the relative duration fact lists accumulated as a result of the application of the methods explained in the previous sections. Each such fact can be in one of two forms: "$|t_a, t_b| = |t_c, t_d|$", or "$|t_a, t_b| > |t_c, t_d|$". The consistency-checking problem at hand is transformed to a problem of the determination of the satisfiability of linear inequalities as follows: Time-points appearing in the relative duration facts are sorted to a linear list. Each minimal interval in this list is given a name. The relative duration facts are rewritten in terms of these interval names. Inequalities asserting that each interval length is greater than zero are incorporated to this set of linear inequalities.

After this transformation is complete, a consistency analyser based on (Clarke and Zhao 1992) is run on the obtained constraint set. If an inconsistency is discovered, the filter routine fails, and the candidate state is eliminated.

In our two-ball example, the relative duration facts available during the preparation of t_2 are, once again, $|t_0, t_1| = |t_1, t_2|$ and $|t_0, t_1| > |t_1, t_2|$. The interval names are I_1, representing $|t_0, t_1|$, and I_2, representing $|t_1, t_2|$. The system of inequalities $I_1 = I_2$, $I_1 > I_2$, $I_1 > 0$, $I_2 > 0$ is easily found to be inconsistent, and Table 2 is eliminated from the output.

Richer Behavior Descriptions

Our modified simulator annotates the output predictions with the additional information about variables and intervals that it extracts during the computation of each behavior. Table 5-6 illustrates this for one of the seven surviving predictions for the two-ball system.

time	A	V_1	V_2	H_1	H_2
t_0	g_0, ⊙	$(0,\infty)$, ↓	$(0,\infty)$, ↓	0, ↑	0, ↑
(t_0, t_1)	g_0, ⊙	$(0,\infty)$, ↓	$(0,\infty)$, ↓	$(0,\infty)$, ↑	$(0,\infty)$, ↑
t_1	g_0, ⊙	0, ↓	$(0,\infty)$, ↓	h1*, ⊙	$(0,\infty)$, ↑
(t_1, t_2)	g_0, ⊙	$(-\infty,0)$, ↓	$(0,\infty)$, ↓	(0,h1*), ↓	$(0,\infty)$, ↑
t_2	g_0, ⊙	$(-\infty,0)$, ↓	0, ↓	(0, h1*), ↓	h2*, ⊙
(t_2, t_3)	g_0, ⊙	$(-\infty,0)$, ↓	$(-\infty,0)$, ↓	(0, h1*), ↓	(0, h2*), ↓
$t_3 < \infty$	g_0, ⊙	$(-\infty,0)$, ↓	$(-\infty,0)$, ↓	0, ↓	(0, h2*), ↓

TABLE 5. A Surviving Prediction for the Two-Ball System

Variable	Symmetry Type	Symmetry Point	Comparisons
A	even	everywhere	
H_1	even	t_1	$|t_0, t_1| > |t_1, t_2|$ $|t_0, t_1| = |t_1, t_3|$
V_1	odd	t_1	
H_2	even	t_2	$|t_0, t_2| > |t_2, t_3|$
V_2	odd	t_2	

TABLE 6. Additional Information for Prediction of Table 5

Related Work

Relative duration fact extraction was first implemented by Çivi (1992), who presents a postprocessor which annotates QSIM outputs with deduced temporal interval comparisons. Çivi's work does not deal with spurious behaviors noticeable due to these items of information.

Weld's *differential qualitative* (DQ) *analysis* (1988) technique involves conceptually comparing two behaviors of the same variable for purposes of perturbation analysis. When comparing multiple traversals of the same interval, we make use of the same simple mathematical foundations, albeit for a different purpose.

Some of the simulations improved by the duration consistency constraint involve *occurrence branching*, in which multiple branches are added to the behavior tree to represent different possible time-orderings of two "unrelated" variables reaching their respective landmarks. "History"-based reasoners like Williams' TCP (Williams 1986) were designed with the purpose of eliminating this phenomenon. There has been some work (Tokuda 1996) to modify the QSIM framework in this direction. Our approach would be useful in cases where the distinctions created by the "global state"-based branching mechanisms are relevant from the user's point of view, and incorrect predictions in this format need to be minimised.

Hybrid qualitative-quantitative reasoners (Kuipers and Berleant 1990) enable the association of numerical values with the time-points in the qualitative simulation output, rendering the comparison of interval lengths trivial. Our work shows that such comparisons are possible and useful in pure qualitative simulation as well.

Conclusion

We have presented methods of eliminating a class of spurious predictions from the output of qualitative simulators. Predictions of this class are identified by inconsistencies in the sets of conclusions, which can be drawn about the relative lengths of the time intervals that they contain. Duration comparisons of this nature can be soundly based on several mathematical properties of the simulated functions, including symmetry and periodicity. The symmetry recognition and analysis procedure, as well as the duration consistency constraint itself, have been implemented and tested in our PROLOG version of QSIM.

Just as multiple traversals of the same landmark interval leads to conclusions about temporal length comparisons, relative duration information can be used for inferences about the relative "distances" among various landmark pairs in the same quantity space. This can, in turn, lead to the detection and elimination of a class of spurious behaviors containing inconsistencies involving landmark distances. We plan to extend our research in that direction, so that qualitative simulators with even greater predictive performance can be built.

Acknowledgments

We thank Özer Yalçın for his technical contribution in the early stages of this research. This work was supported by the Boğaziçi University Research Fund. (Grant no: 97HA101)

References

Clarke, E., and Zhao, X. 1992. Analytica: A Theorem Prover for Mathematica. Technical Report CMU-CS-92-117. School of Computer Science. Carnegie Mellon University.

Çivi, H. 1992. Duration Analysis in QSIM and Extension of QSIM to Discrete Time Systems. M.S. Thesis, Dept. of Computer Eng., Boğaziçi Univ., İstanbul, Turkey.

Könik, T., and Say A. C. C. 1998. Extracting and Using Relative Duration Information in Pure Qualitative Simulation, Technical Report, FBE–CMPE–02/98–02, Dept. of Computer Eng., Boğaziçi Univ., İstanbul, Turkey.

Kuipers, B. J. 1994. Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. Cambridge, MA: The MIT Press.

Kuipers, B. J., and Berleant, D. 1990. A Smooth Integration of Incomplete Quantitative Knowledge into Qualitative Simulation, Technical Report, AI TR 90-122, Artificial Intell. Lab., Univ. of Texas at Austin.

Say, A. C. C. 1997a. Numbers Representable in Pure QSIM. In *Proc. 11th Int. Workshop on Qualitative Reasoning*, 337-344, Cortona, Italy.

Say, A. C. C. 1997b. Improved Reasoning About Infinity in Qualitative Simulation. In *Proc. 12th Int. Symposium on Computer and Information Sciences*, 36-43. Antalya, Turkey.

Say, A. C. C. 1998. L'Hôpital's Filter for QSIM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(1):1-8

Say, A. C. C., and Kuru, S. 1993. Improved Filtering for the QSIM Algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 15:967-971.

Tokuda, L. 1996. Managing Occurrence Branching in Qualitative Simulation. In *Proc. 13th National Conference on Artificial Intelligence(AAAI-96)*, AAAI/MIT Press.

Weld, S. D. 1988. Comparative analysis. *Artificial Intelligence* 36:333-373.

Williams, B. C. 1986. Doing Time: Putting Qualitative Reasoning on Firmer Ground. In *Proc. 5th National Conference on Artificial Intelligence*, 105-112. San Mateo: CA. Morgan Kaufmann.