From: AAAI-90 Proceedings. Copyright ©1990, AAAI (www.aaai.org). All rights reserved.

Dynamic Across-Time Measurement Interpretation

Dennis DeCoste *
Qualitative Reasoning Group
Beckman Institute, University of Illinois
405 North Mathews Avenue, Urbana, Illinois 61801
email: decoste@cs.uiuc.edu

Abstract

Incrementally maintaining a qualitative understanding of physical system behavior based on observations is crucial to real-time process monitoring, diagnosis, and control. This paper describes the DATMI theory for dynamically maintaining a *pinterp-space*, a concise representation of the local and global interpretations consistent with observations over time. Each interpretation signifies an alternative path of states in a qualitative envisionment. DATMI can use domain-specific knowledge about state and transition probabilities to maintain the best working interpretation. By maintaining the space of alternative interpretations as well, DATMI avoids the need for extensive backtracking to handle incomplete or faulty data.

1 Introduction

Efficient incremental interpretation of observations is essential for real-time process monitoring, diagnosis, and control. A consistent interpretation of what is happening, along with alternatives, should continually be available. Since data can be incomplete and faulty, interpretations must be efficiently revised as new data arrives or inconsistencies are discovered. For example, the original hypothesis during the Three Mile Island incident was that the reactor core had too much water. The correct failure hypothesis was triggered once it was noticed that the symptoms remained even after a lot of water was removed. Capturing this kind of reasoning is clearly important for engineering-oriented applications of qualitative physics.

This paper describes the DATMI theory and algorithm for efficient dynamic across-time measurement interpretation. DATMI is an extension of the ATMI theory of measurement interpretation [7]. Like ATMI, we assume the input includes a *total envisionment* representing the qualitative states and transitions which characterize all the possible behaviors of the system being observed. This envisionment serves as a finite-state transition graph for "parsing" the observations. Each interpretation identifies the behavior as a path through the envisionment. However, ATMI did not allow incremental updating of its interpretation and made no provision for handling faulty data. DATMI overcomes these limitations by using a concise representation of the space of possible interpretations, called the *pinterp-space*. A vocabulary of dependency relations allows DATMI to dynamically adjust to changing data and to provide efficient hypothesis revision. Furthermore, DATMI can exploit two kinds of domain-specific information, when available: probabilities specified for states and transitions and duration bounds specified for states and paths. DATMI has been fully implemented and successfully tested on a variety of envisionments and data sets.

Section 2 describes DATMI's theoretical foundation and Section 3 explains its basic algorithm. Section 4 presents an example of DATMI handling faulty data. Section 5 summarizes how additional domain-specific knowledge is used. Finally, Section 6 notes some limitations and discusses related and future work.

2 DATMI Theory

As in ATMI, we assume that data consists of *measurements* of numeric variables (such as temperature) and *observations* of symbolic properties (such as whether a switch is ON or OFF and whether the temperature of one object is greater than, less than, or equal to that of another). We also assume that the measurements have been smoothed to reflect their qualitative trends. Discarding data at the fringes of each qualitative change, to account for the temporal imprecision of smoothings, is allowed since DATMI handles incomplete data.

DATMI conservatively converts measurements at time points into observations over time intervals, using probabilities and disjunctive values to express uncertainty in the sensor readings. This requires: (1) converting numerical values into qualitative ones by comparing them with limit points and (2) aggregating measurements with identical qualitative values into observations. Each property which signifies the ordering between a variable and a limit point has an associated *conversion table*. Each table maps ranges of distance from the limit

*This research was supported by the Office of Naval Research under Contract No. N00014-85-K-0225.

point into qualitative values having discrete probability assignments. Figure 1 gives an example. DATMI provides no means for determining these tables; it assumes that domain-specific information provides them, perhaps based on *a priori* sensor precisions and reliabilities. In lieu of probabilistic assignments, DATMI assumes each disjunctive value is equally probable; Section 5.1 explains how these discrete probabilities are used.

For comparison between variable A and limit point B:

$(A-B)/\lvert B\rvert \in [-\infty, -0.2)$	$\Rightarrow (A < B$, with probability p=1.0)
$(A-B)/\lvert B\rvert \in [-0.2, -0.01)$	$\Rightarrow (A < B$, p=0.6) $\wedge$ ($A = B$, p=0.4)
$(A-B)/\lvert B\rvert \in [-0.01, 0.01]$	$\Rightarrow (A = B$, p=1.0)
$(A-B)/\lvert B\rvert \in (0.01, 0.2]$	$\Rightarrow (A > B$, p=0.6) $\wedge$ ($A = B$, p=0.4)
$(A-B)/\lvert B\rvert \in (0.2, \infty]$	$\Rightarrow (A > B$, p=1.0)

(If $B = 0$ then $(A-B)/|B|$ is replaced by $(A-B)$)

Figure 1: A simple example conversion table

As the overview in Figure 2 shows, the observations are concisely represented as properties of *global segments*. As in ATMI, each segment represents the interval of maximal temporal extent over which all of its properties are identical. The interval of each segment *meets* [1] the interval of each of its two neighboring segments. Thus, the history of segments is temporally totally-ordered. *Gap-fill* segments represent intervals over which no observations are available. Since DATMI maintains the segments as observations are gathered, it splits and merges segments appropriately.

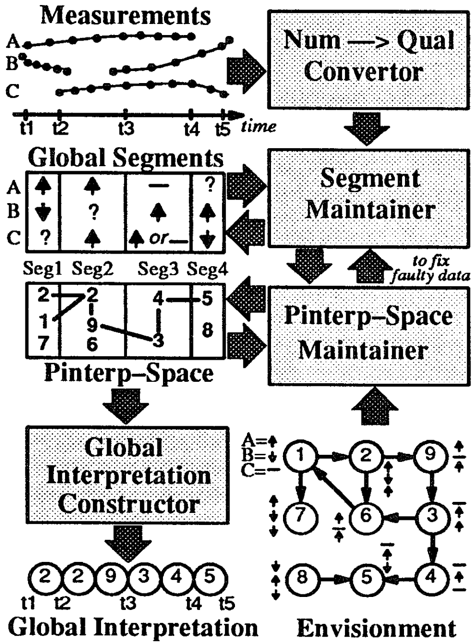


Figure 2: DATMI overview

ATMI introduced the notion of a *pinterp*: an envisionment state which can possibly occur during a particular segment. Let $P(G, S)$ be the pinterp which indicates whether state S can occur during global segment G. $P(G, S)$ is considered COMPATIBLE exactly when all the properties describing S are compatible with all the properties of G. $P(G, S)$ is also considered ACTIVE exactly when all constraints allow S to occur during G.

Clearly, an ACTIVE pinterp must be COMPATIBLE, to satisfy the constraints of its own segment. An ACTIVE pinterp must also satisfy the constraints of the other segments and the envisionment transitions. In particular, a pinterp is ACTIVE exactly when it is COMPATIBLE and there is a *transition consistency relation* between it and each neighboring segment. Alternatively, a pinterp is INACTIVE exactly when it is COMPATIBLE but not ACTIVE. Thus, each pinterp is either INCOMPATIBLE, INACTIVE, or ACTIVE. Figure 3 illustrates DATMI's five types of transition consistency relations. A relation between a pinterp $P(G, S)$ and a neighboring segment N indicates a path of ACTIVE pinterps which starts at $P(G, S)$ and reaches an ACTIVE pinterp of N. Each relation signifies a consistent way to be in S during G and also be in a state during N.

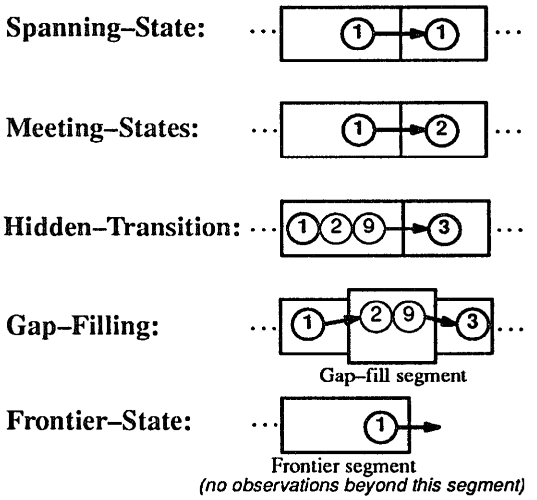


Figure 3: Types of transition consistency relations These paths through the envisionment of Figure 2 illustrate the five types of ways a pinterp can lead to a pinterp of the next observed segment.

Identifying hidden-transition and gap-filling paths allows DATMI to interpret even very incomplete data. For simplicity, DATMI considers only acyclic paths of pinterps for these two relations. Thus, it will misinterpret behaviors where the system returns to the same state during a single segment. However, unless duration constraints invalidate all acyclic relation paths, this simplification will not prevent DATMI from finding some interpretation which is at least simple, if not best.

Each global interpretation is a chain of relation paths across all segments. For example, the global interpretation given in Figure 2 is a global chain of some of the relation paths indicated in its pinterp-space. We do not show the INCOMPATIBLE pinterps of the pinterp-space; so, the INACTIVE ones are exactly those for which no relation paths are shown.

3 The DATMI algorithm

DATMI dynamically maintains the pinterp-space by keeping track of the status of each pinterp (INCOMPATIBLE, INACTIVE, or ACTIVE) as segment properties are asserted and retracted. It determines which pinterps are COMPATIBLE the same way as ATMI did; it uses a lookup-table precomputed from the envisionment that indicates every state compatible with a given property. It determines whether a COMPATIBLE pinterp is ACTIVE by using graph search through the envisionment to find the *best* relation path between the pinterp and each neighboring segment. As shown in this section, breadth-first graph search suffices to find the *simplest* paths. Section 5.1 explains how least-cost graph search finds the *most-probable* paths. In both cases, the best global paths are built from the best local (relation) paths.

After determining a relation path between a pinterp and a neighboring segment, DATMI caches that path as a *dependency path* for that pinterp. Each ACTIVE pinterp is assigned exactly two dependency paths: a *b-dependency path* reaching a pinterp of the backward neighboring segment and a *f-dependency path* reaching a pinterp of the forward neighboring segment. Since global chains of relation paths indicate global interpretations, so do global chains of b-dependency paths or f-dependency paths.

By finding dependency paths using breadth-first graph search which expands each state at most once, the interpretation indicated by a chain of b-dependency paths is the simplest (i.e. shortest) one. This search through the envisionment starts at the pinterp $P(G,S)$ whose dependency path is sought and finds the simplest path (if any) to each ACTIVE pinterp of the neighboring segment by expanding only from $P(G,S)$ and ACTIVE pinterps of G. To ensure finding the simplest b-dependency chains, DATMI records the cost (i.e. number of transitions) of the chain of b-dependency paths leading up to each pinterp $P(G,S)$ when its b-dependency path is found. This allows b-dependency path search for each pinterp $P(F,f)$ of the forward neighboring segment F to sum $P(G,S)$'s recorded cost with the cost of the relation path found from $P(F,f)$ to $P(G,S)$. DATMI compares that sum against the sums for the simplest paths found to other ACTIVE pinterps of G, to decide which one offers the simplest chain of b-dependency paths to $P(F,f)$.

The above discussion assumes that each gap-fill segment is treated as an ordinary segment whose pinterps all happen to be COMPATIBLE. For efficiency, however, DATMI does not actually maintain pinterps for gap-fill segments. It uses a lookup-table precomputed from the envisionment that indicates the best path through the envisionment between any two particular states. This table, along with the recorded b-dependency chain costs, suggests the best b-dependency path across the gap-fill segment which connects a particular pinterp of the forward neighboring segment with an ACTIVE pinterp of the backward neighboring segment. (This is why the gap-fill path in Figure 3 involves three segments.) If desired, one can determine if a particular pinterp of a gap-fill segment is ACTIVE – by checking this table to see if paths exist from it to ACTIVE pinterps of the neighboring segments.

As explained below, dependency paths play one other key role: they indicate which pinterps must find new relation paths when a pinterp of a neighboring segment ceases to be ACTIVE. In that sense, they are analogous to TMS justifications [5]. However, for our task, caching alternative dependency paths would not lead to the efficiencies that one might expect. As explained in [4], if the alternative path is short, it is more efficient to find it as needed; if it is long, it is not likely to be best or even consistent (i.e. contain only ACTIVE pinterps) when the current best path becomes inconsistent.

Asserting properties for a segment can make some pinterps of that segment cease to be ACTIVE because they are no longer COMPATIBLE. These changes can cause other pinterps to become INACTIVE if they depend on those newly INCOMPATIBLE pinterps. So, a replacement dependency path must be sought for each pinterp which has a newly INCOMPATIBLE pinterp in its dependency path. A pinterp which fails to find replacements becomes INACTIVE, which then affects the pinterps depending on it. Loss of activation is thus globally propagated in two segment-wise sweeps out from the segment with new properties. To ensure that the recorded cost of b-dependency chains (as mentioned above) is propagated correctly, the backward sweep is performed before the forward one.

When asserting the initial properties for a new frontier segment or what was a gap-fill segment, initial relation paths from the COMPATIBLE pinterps of that segment to the neighboring segments must be sought. All of these COMPATIBLE pinterps are first assumed to be ACTIVE, to allow them to be expanded during breadth-first searches to find these initial dependency paths. Each one is considered INACTIVE only if dependency paths cannot be found for it. All ACTIVE pinterps of both neighboring segments then seek (initial) dependency paths to that segment, propagating loss of activation as before.

As explained in the next section, pinterps can also become ACTIVE when dubious properties are retracted from a segment. For this to happen, some pinterps of that segment must first become newly COMPATIBLE with the reduced set of properties. The newly COMPATIBLE pinterps which find dependency paths (i.e. become ACTIVE) can allow other (INACTIVE) pinterps to become ACTIVE as well. Furthermore, newly ACTIVE pinterps can allow some ACTIVE pinterps to obtain better b-dependency paths. An inefficient technique would be to start from the first segment and re-compute dependency paths from scratch. However, by allowing intermediate steps to temporarily create dependency paths involving INACTIVE pinterps, efficient incremental propagation of activation is achieved (see [4]).

3.1 Handling Faulty Data

DATMI's conservative conversion from measurements to disjunctive qualitative values cannot always avoid the effects of faulty data. Inconsistencies due to faulty data can only be prevented if the conversion tables never assert any qualitative value to have zero probability. However, in that case, each path through the envisionment would be a consistent global interpretation. So, DATMI provides a means for modifying segment properties to recover from faulty data which do sneak past the conservative conversions.

DATMI detects inconsistencies between the envisionment and the observations as soon as all of some segment's pinterps cease to be ACTIVE. That segment is called an *inconsistent segment.* Since DATMI assumes that the envisionment itself is sound and complete, it attempts to recover from inconsistency by generating and testing sets of property changes to remove hypothesized faulty data. Associated with each hypothesis are: 1) a set of property changes, 2) its plausibility, and 3) the conditions under which it remains valid.

Because the faulty data might not be in an inconsistent segment's own properties, all segment properties are suspect. To avoid considering the power set of possible property changes, DATMI currently limits itself to an especially common subset of these. In particular, it considers forgetting properties that could arise from non-intermittent *sensor failures.* An example of such a failure is when a flow-rate seems constant because its sensor has gotten stuck.

Each DATMI sensor failure hypothesis suggests forgetting (i.e. retracting) all recent segment properties having the most-recent value for one type of property. This assumes that each sensor only contributes to one type of property. So, in the above example, it would forget all the recent observations of that flow-rate being constant, back until it was last observed to be changing. For simplicity, DATMI considers the plausibility of a sensor failure hypothesis to be the *a priori* probability that the particular most-recent value is due to that sensor failure. Determining these probabilities is outside of our theory; we currently use arbitrary values reflecting commonsense intuitions. Finally, the hypothesis holds as long as that type of property is not observed to have a different value at later times, since the hypothesized failure is non-intermittent.

DATMI generates these hypotheses in order of plausibility and then tests each by propagating any pinterp activations due to the reduced number of segment properties, as explained earlier. A useful aspect of forgetting properties, without asserting any replacements, is that no new inconsistencies can be introduced during this test. If all segments are not consistent after propagating activation, the forgotten properties are reasserted. By resorting to such reassertions, to avoid excessive removal of constraints on the interpretation space, DATMI cannot recover from an inconsistency caused by multiple sensor failures.

The generate-and-test process continues until there are no inconsistent segments or each hypothesis has been tried. If an inconsistent segment still remains, the pinterp-space is partitioned by that segment and each part is interpreted separately. In any case, if the conditions of a successful hypothesis are later violated, it must be retracted by reasserting the forgotten properties. Inconsistencies will then be redetected and rehandled unless some later successful hypothesis (that fixed some later arising inconsistencies) happened to fortuitously fix those inconsistencies as well. In the case of our sensor failure hypotheses, such fortuitous fixes can occur if the earlier hypothesis recovered solely from the early inconsistencies arising from a sensor failure.

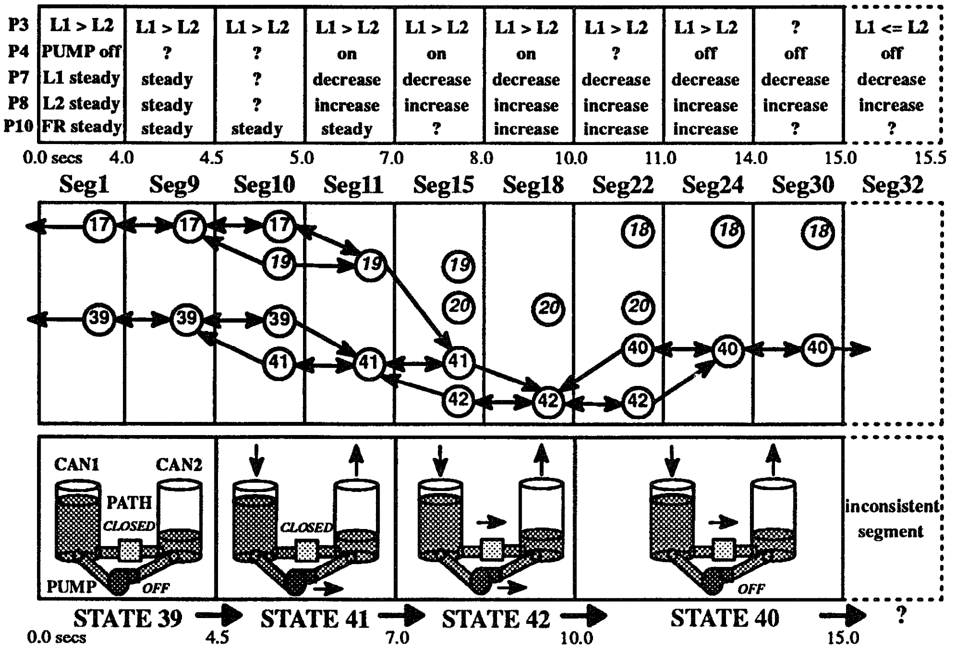
4 Example

We now highlight an implemented example of DATMI handling incomplete and faulty data. This example uses a pump-cycle system of two containers of water connected by a valved path and a pump, as illustrated in Figure 4. The QPE [8] envisionment (not shown here due to space limitations) consists of 42 states and 61 transitions. This envisionment differentiates states by ten types of properties, of which the following five are observed: P3 (the comparison between L1 and L2) P4 (the ON/OFF status of PUMP), P7 (the direction of change in L1), P8 (the direction of change in L2), and P10 (the direction of change in FR). Partial, perturbed results of a numeric simulation provided the incomplete, faulty measurements for determining those five types of properties.

As Figure 4 shows, DATMI interprets the measurements for the first 15.0 seconds with no problem. Note that properties are not asserted for every segment, as indicated by the "?" values. This is because the conservative conversions did not specify exact times for changes between qualitative values, since they are indeterminate from the sampled data. The arrows in the pinterp-space show the backward and forward dependency paths. The best working global interpretation for the first 15.0 seconds is found by following the chain of b-dependency paths starting at state 40 in Seg30 and ending at state 39 in Seg1.

However, no COMPATIBLE pinterps exist for the properties observed for Seg32. Thus, when those properties are asserted, Seg32 becomes inconsistent because it has no ACTIVE pinterps. To fix this inconsistency, DATMI considers five sensor failure hypotheses, one for each of the five types of observed properties.

DATMI first tries forgetting properties for P10; for this example, each hypothesis is considered equally plausible. Since P10's most-recent observed value is INCREASE in Seg24 and its last different value is STEADY in Seg11, the sensor failure hypothesis is to forget all properties for P10 after Seg11. DATMI quickly finds that forgetting P10 from Seg18, Seg22, and Seg24 fails to fix the pinterp-space. Indeed, in this case forgetting P10 could not possibly help since the pinterps of Seg32 are



	Seg1	Seg9	Seg10	Seg11	Seg15	Seg18	Seg22	Seg24	Seg30	Seg32
	0.0 secs – 4.0	4.0 – 4.5	4.5 – 5.0	5.0 – 7.0	7.0 – 8.0	8.0 – 10.0	10.0 – 11.0	11.0 – 14.0	14.0 – 15.0	15.0 – 15.5
P3	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	?	L1 <= L2
P4	PUMP off	?	?	on	on	on	?	off	off	off
P7	L1 steady	steady	?	decrease	decrease	decrease	decrease	decrease	decrease	decrease
P8	L2 steady	steady	?	increase	increase	increase	increase	increase	increase	increase
P10	FR steady	steady	steady	steady	?	increase	increase	increase	?	?

Figure 4: The pinterp-space when inconsistency first detected
L1=(FLUID-LEVEL CAN1), L2=(FLUID-LEVEL CAN2), and FR=(FLOW-RATE PATH).

all INCOMPATIBLE, which requires changing the properties of Seg32 itself. However, to illustrate the general case, we do not treat this case specially for this example. Even hypotheses which modify only properties of consistent segments are generally useful since they may allow INACTIVE pinterps of an inconsistent segment to become ACTIVE. In any case, DATMI then tests hypotheses to forget P8 (after Seg9) and forget P7 (after Seg9), also with no success.

DATMI succeeds by forgetting P4 after Seg18, as shown in Figure 5. This reflects the hypothesis that the pump indicator failed sometime after Seg18 and continues to indicate that the pump is OFF, perhaps because the indicator light burnt out. The OFF values for P4 are shown in parenthesis since they are now forgotten and do not constrain the pinterps. While propagating activation due to these forgotten properties, the dependency paths are updated appropriately, yielding a new simplest global interpretation where the pump is never OFF after Seg9. Of course, the pinterp-space also allows interpretations where the pump is OFF after Seg18. However, the pump must become ON again in Seg32 because its pinterps are INCOMPATIBLE otherwise. So, those alternative interpretations involve more state changes than DATMI's (simplest) working interpretation does.

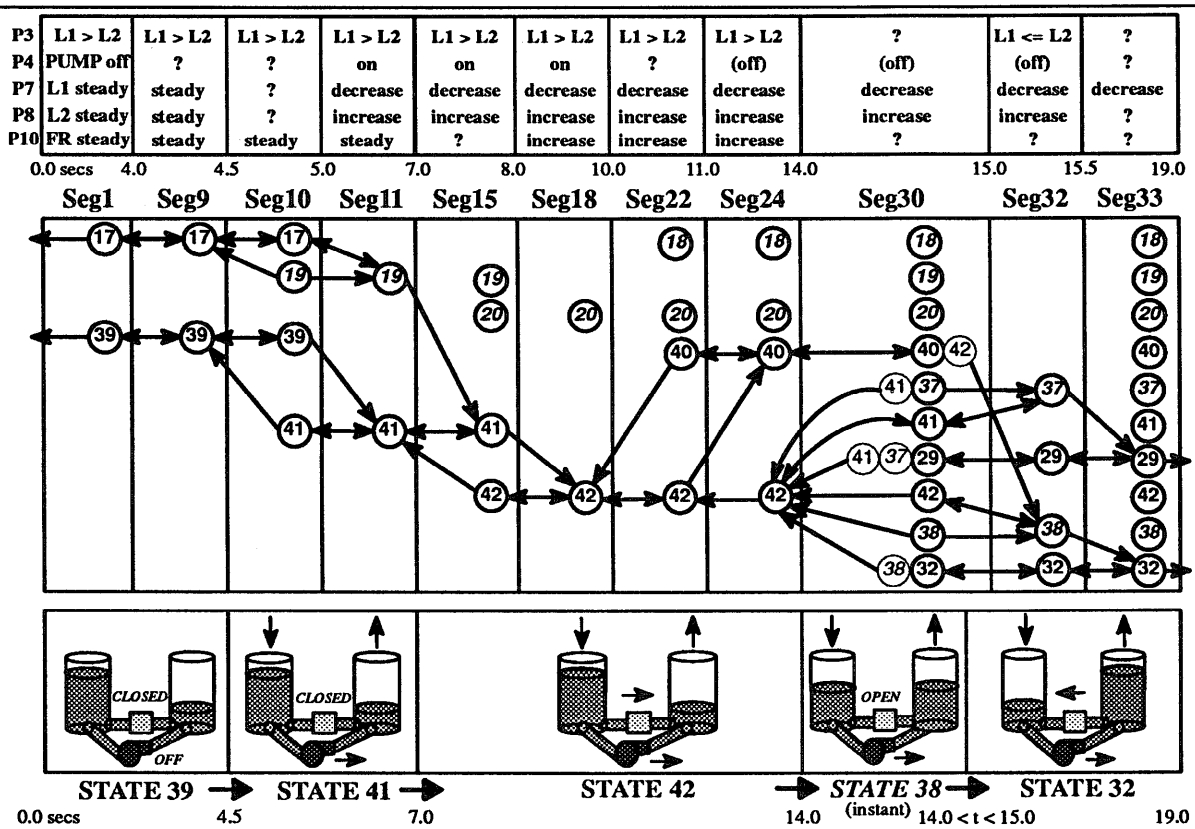
5 Using Additional Knowledge

5.1 Probabilities

DATMI can use numeric probabilities associated with the envisionment states and transitions to maintain the most-probable working global interpretation, instead of the simplest one. This requires that those probabilities have been estimated by some external means, such as the stochastic analysis technique of [6]. DATMI composes these probabilities using Bayes' chain rule by assuming independent events [12]. Thus, the *a priori* probability of a chain of b-dependency paths is considered the product of the *a priori* probability of the earliest state and the conditional probabilities of all the transitions in that chain.

This composition is valid assuming that each transition is truly independent of which states occurred earlier in the chain, which is typically the case for paths through sufficiently detailed envisionments. It overestimates the probability of spanning-state dependency paths relative to other types, since other types involve more transitions during the same period of time. Since spanning-state paths provide simpler interpretations, this is usually acceptable.

To reflect uncertainty in the measurements, the segment property probabilities from the conversion tables (e.g. Figure 1) can also be included in this composition. DATMI assumes that a segment's properties are independent, which is most reasonable when the observations are never redundant. The total property probability for a particular pinterp is then the product of the probabilities of the segment property values with which its state is compatible. Thus, DATMI multiplies the *a priori* probability of a chain of b-dependency paths by the product of the property probabilities for each pinterp in that chain to get a more accurate probability for that chain. Furthermore, to accurately reflect the probability of a b-dependency chain that is conditional on



	Seg1	Seg9	Seg10	Seg11	Seg15	Seg18	Seg22	Seg24	Seg30	Seg32	Seg33
P3	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	L1 > L2	?	L1 <= L2	?
P4	PUMP off	?	?	on	on	on	?	(off)	(off)	(off)	?
P7	L1 steady	steady	?	decrease	decrease	decrease	decrease	decrease	decrease	decrease	decrease
P8	L2 steady	steady	?	increase	increase	increase	increase	increase	increase	increase	?
P10	FR steady	steady	steady	steady	?	increase	increase	increase	?	?	?

0.0 secs 4.0 4.5 5.0 7.0 8.0 10.0 11.0 14.0 15.0 15.5 19.0
17 39 19 41 42 20 18 40 37 29 38 32
CLOSED
OFF
OPEN
STATE 39
STATE 41
STATE 42
STATE 38
STATE 32
0.0 secs 4.5 7.0 14.0 (instant) 14.0 < t < 15.0 19.0

Figure 5: The pinterp-space when hypothesized faulty data retracted

the set of observations, one must redistribute the *a priori* probability associated with pinterps which are not ACTIVE to the ACTIVE ones, as described in [4].

Standard exhaustive graph search for least-cost paths between all pairs of nodes [11] suffices for finding the most-probable b-dependency paths. DATMI records the composed probability of the chain of b-dependency paths leading up to each pinterp when its b-dependency path is found. It finds the most-probable paths from each COMPATIBLE pinterp of a segment G to each ACTIVE pinterp of the backward neighboring segment B using least-cost path search over all those pinterps, with the following cost assignments. The cost of each pinterp of B is the inverse of its (recorded) composed probability. The cost of each pinterp of G is the inverse of its property probability. The cost of each transition is the inverse of its conditional probability. The total cost of a path is thus the product of the costs it entails.

Actually, DATMI avoids the cubic time cost of exhaustive least-cost path search by performing efficient best-first search for each pinterp, resorting to least-cost search only when some cubic function of states have been examined without success, as detailed in [4].

5.2 Durations

Space limitations do not permit an explanation of how DATMI uses duration bounds to prune candidate b-dependency paths during best-first search. However, it is important to realize that such bounds can be incorporated into the DATMI framework to avoid interpretations which are inconsistent with those constraints. One simple example is indicated by the b-dependency paths in Figure 5. The b-dependency path for state 19 in segment Seg11 includes state 17 in segment Seg10 instead of state 19 in Seg10. State 19 cannot span from Seg10 to Seg11 because it is an instantaneous state.

6 Discussion

DATMI incrementally maintains a concise interpretation space, which allows it to quickly detect faulty data and then efficiently recover by doubting observations. It provides a framework for integrating domain-specific knowledge, such as probability and duration estimates, with the causal constraints given by qualitative simulation. Its use of dependency paths makes maintenance efficient and provides the best working global interpretation at all times. Furthermore, it indicates which states can consistently occur during each segment, making it suitable for monitoring tasks and providing strong focus for finding alternative global interpretations. DATMI has been fully implemented and tested on several examples from both QPE and FROB [9] domains, suggesting that it is applicable to any system of qualitative physics.

To avoid intractable temporal reasoning during interpretation, DATMI accepts two key limitations. First, global interpretations are not always most appropriately represented by DATMI's linear sequences of states. For example, partial orderings provide more general explanations, but DATMI's use of global segmentation and local dependency paths precludes them. Second, DATMI requires an appropriate envisionment as input. By referring to an available envisionment, DATMI's pinterp-space maintenance requires time at most cubic in the number of envisionment states. This bound is based on the cubic worst-case time cost for finding the most-probable b-dependency paths; DATMI's overall cubic worst-case complexity is analyzed in [4].

Although envisioning itself can be exponential in the number of system variables, we suspect that it is more efficient to cleanly separate envisioning from interpreting for tasks where a large fraction of the envisionment states are likely to occur or where observations are very sparse. This intuition is based on the efficient techniques developed for total envisioning, such as those described in [8]. To handle the other cases, we are currently developing incremental envisioning techniques which could provide DATMI with previously unavailable states and transitions when suitable interpretations cannot be found using a working partial envisionment.

By not accepting DATMI's two key limitations, related approachs allow other gains while sacrificing the ability to interpret incomplete, faulty data as efficiently and robustly as DATMI. For example, Q2 [10] can exploit more of the quantitative information in the measurements as it generates consistent histories. However, to ensure that it can always offer some interpretation, Q2 must be able to follow every branch during history generation, which can be exponential in the number of states. Although the number of paths through an envisionment is also exponential in the number of states, DATMI never needs to consider more than a cubic number of them (during dependency path search) because of the factorization of the problem provided by global segmentation.

GDE [3] provides an alternative means for handling inconsistencies between the measurements and the model. It is not directly suited for our problem because its focus is on determining faults in the system itself, not the observations. Although it acknowledges sensor failure rates, it does not attempt to reason about the nature of such failures, as DATMI does with sensor failure hypotheses. Also, GDE does not reason over time. The consequences of using TCP [13] with GDE to allow across-time reasoning, which deKleer and Williams suggest as future work, are not clear. Although TCP's *concise histories* could represent partially-ordered interpretations, that approach would suffer from overhead that DATMI's globally-segmented pinterp-space avoids.

The DATMI framework suggests future research in several directions. Continued progress in qualitative modelling, along with incremental envisioning, is needed. Multiple faults might be handled by not always retracting a hypothesis when it fails to recover from all the inconsistencies by itself. Also, more formal, general techniques for generating fault-recovery hypotheses are needed, perhaps based on *knowledge groups* as in [2]. Finally, the data selection problem (i.e. using the most informative data first) might be addressed by preferring observations at times nearest the segments having the most ACTIVE pinterps.

Acknowledgements

Thanks to Ken Forbus, John Collins, Brian Falkenhainer, and Janice Skorstad for useful comments.

References

[1] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832–843, November 1983.

[2] B. Chandrasekaran and W. F. Punch III. Data validation during diagnosis, a step beyond traditional sensor validation. In *Proceedings of AAAI-87*, pages 778–782, July 1987.

[3] Johan de Kleer and Brian Williams. Reasoning about multiple faults. In *Proceedings of AAAI-86*, pages 132–139, August 1986.

[4] Dennis DeCoste. *Dynamic Across-Time Measurement Interpretation: Maintaining Qualitative Understandings of Physical System Behavior.* Master's thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, October 1989. (Technical Report UIUCDCS-R-90-1572, February 1990).

[5] Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[6] Jon Doyle and Elisha P. Sacks. Stochastic analysis of qualitative dynamics. In *Proceedings of IJCAI-89*, pages 1187–1192, August 1989.

[7] Kenneth D. Forbus. Interpreting measurements of physical systems. In *Proceedings of AAAI-86*, pages 113–117, August 1986.

[8] Kenneth D. Forbus. The qualitative process engine. In Daniel S. Weld and Johan de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 220–235, Morgan Kaufmann, 1990.

[9] Kenneth D. Forbus. *A Study of Qualitative and Geometric Knowledge in Reasoning about Motion.* Technical Report TR-615, AI Lab, MIT, 1981.

[10] Benjamin Kuipers and Daniel Berleant. Using incomplete quantitative knowledge in qualitative reasoning. In *Proceedings of AAAI-88*, pages 324–329, August 1988.

[11] Kurt Mehlhorn. *Graph Algorithms and NP-Completeness.* Volume 2 of *Data Structures and Algorithms*, Springer-Verlag, 1984.

[12] Judea Pearl. *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[13] Brian C. Williams. Doing time: putting qualitative reasoning on firmer ground. In *Proceedings of AAAI-86*, pages 105–112, August 1986.