Taming Chatter With Relevant Envisionments

Tim Menzies[1], Sam Waugh[2], Simon Goss[2]

[1]Artificial Intelligence Department,
School of Computer Science and Engineering,
University of NSW, 2052, Australia.
[2]Defence Science and Technology Organisation,
Air Operations Division, GPO Box 4331, Melbourne, 3001, Australia
timm@cse.unsw.edu.au
sam.waugh@dsto.defence.gov.au
simon.goss@dsto.defence.gov.au
http://www.cse.unsw.edu.au/~timm

January 21, 1998

Abstract

Branching of behaviours is a major drawback for qualitative reasoning (QR). Intractable branching (chatter) due to irrelevant distinctions is one of the major factors hindering the application of qualitative reasoning techniques to large real-world problems. We should hence seek QR systems that avoid chatter. QCM models (Qualitative Compartmental Modelling) chatter very little and certain variants of QCM hardly chatter at all. We describe the special features of QCM which reduce chatter; i.e. relevant envisionments, explicit temporal linking, and domain data measurement policies.

1 Introduction

Is chatter (intractable branching of behaviours) a fundamental property of qualitative reasoners (QR)? Or is it some side-effect of current QR strategies? Clancy and Kuipers observe that... *Intractable branching due to irrelevant distinctions is one of the major factors hindering the application of qualitative reasoning techniques to large real-world problems* [5]. The problem of chatter is widely discussed and is yet to be resolved. Meta-knowledge can be used to partially tame chatter. For example, the Waltz filtering of the QSIM QR system [14] rules-out a transition of the first derivative of a variable from increasing to decreasing without first going through a zero state (more complex examples are offered in [15]). In practice however, chatter is often observed [11] and must be somehow handled by the program calling the qualitative simulator (e.g. impose sub-divisions on the model to contain the effect of chatter [5]).

In contrast to other QR systems, we observe that models in our QCM system (Qualitative Compartmental Modelling [23]) chatter very little and certain variants of QCM hardly chatter at all. This is a surprising observation since our reading of the literature indicates that chatter is a fundamental property of QR systems. This paper examines why chatter is rare in certain QCM variants. It will be shown that chatter can be reduced via certain implementation options; i.e. explicit temporal linking; data-rich or data-poor domain measurement policies; and relevant envisionments (these terms will be explained below). This last implementation option, relevant envisionments, is what distinguishes QCM from other QR systems such as QSIM.

We will proceed as follows. First, we introduce qualitative reasoning, the problem of chatter, and the QCM system. Next, we describe an experiment where we ran the QCM reasoner over a million times looking for variants on QCM problems where chatter is reduced. Lastly, we discuss how relevant envisionments can explain the lack-of-chatter in certain QCM variants.

2 QR: An Overview

The QR community focuses mainly on the processing of systems called qualitative differential equations (QDE) which are (i) piece-wise well-approximated by low-order linear equations or by first-order non-linear differential equations; (ii) whose numeric values are replaced by one of three qualitative states: *up*, *down*, or *steady* [12]. QSIM [14] is a theorem prover for QDEs and can be used to implement a range of QR systems (e.g. QPT [6]).

(Not all QR is based directly on QDEs. For example, Yip [31] discusses the qualitative dynamics of hamiltonians. Bratko *et. al.* [2] generate a rule-base for heart disease via a machine learning program that condenses the output from an indeterminate qualitative model of heart disease. In the bond graph approach [29], models are built out of components representing abstract energy sources, sinks, storage, and dissipater devices. However, QCM is based on influence statements extracted from QDEs. Hence, for the rest of this paper, we will assume QR means the processing of QDEs.)

A commonly observed property of QDEs is their indeterminacy [5]. Consider two competing influences on a variable. That variable may go *up*, *down*, or, if the influences cancel each other out, remain *steady*. One *branch* of the reasoning must be forked for each possibility. This forking process may recur downstream of the branch; i.e. the branch may generate sub-branches and sub-sub-branches and so on.

If a QR is simulating a known system, then observations of that system can constrain branching. For example, if the QR system tries to branch on *wind speed at time 5*, and it is known that wind speed at time 5 did not decrease, then we can cull the *down* branch. In general, less data means more branching. If measurements are available for all variables for all time points, then a QR system would become deterministic since only one *branch* could grow. Note that it would be naive to attempt to tame chatter by insisting that QR executes only when all variables are measured. The costs of making measurements in many domains can prohibit extensive observations. For example, the Smythe'89 compartmental model of glucose regulation [27] is a typical model from the domain of experimental neuroendocrinology. (Compartmental models utilise the principal of conservation of mass and assume that the sum of flows of substance in and out of a compartment must equal zero. Flows are typically modelled using

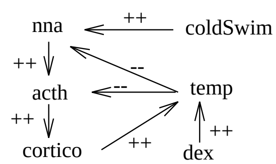

Figure 1: The Smythe'87 model. Adapted in [18] from [26].

a time-dependent exponential function since the rate of flow is often proportional to the amount of stuff in the compartment [17]). The Smythe'89 model contains 27 compartments linked by 82 flows. The model is a summary model generated for six other journal articles. On average, only 5.2 of the compartments are measured in any one experiment.

3 QCM: An Abductive Qualitative Reasoner

This section describes QCM, a qualitative compartmental modeling system. The next section describes experiments with QCM variants for handling temporal reasoning.

QCM is a generalisation of QMOD: Feldman and Compton's work [8, 9] on QR. A QCM theory contains a set of *influence statements* extracted either from known equations about the domain or from intuitions offered by experts. For example, the Smythe'87 model of human stress responses [26] studied the relations between serum adrenocorticotropin (*acth*), serum corticosterone (*cortico*), and neuro-noradrenergic activity (*nna*). The model is shown in Figure 1. *Nna* was measured as the ratio of noradrenaline to its post-cursor, 3,4-dihydroxphenylethethyleneglycol. This theory was studied via two treatments: (1) *dex* i.e. an injection of dexamethasone at 100 mg per kg; (2) *coldSwim* i.e. a two minute swim in a bath of ice cold water. A *temp* variable was used to denote that *dex* has the same effects as *cortico*. In QCM, one would represent the Smythe'87 influences on *nna* as follows:

```
direct(coldSwim,nna).
inverse(temp,nna).
```

These influence statements may be generated from equations. For example, *A=3B*2C-5D* would generate influences from right-hand-side variables to the left-hand-side-variables (and also for all valid rearrangements of the equation such as *D=(3B*2C-A)/5*). Direct influences would be generated from B to A and from C to A. Inverse influences would be generated from D to A.

QCM executes its models by first compiling them down to a directed graph. For example, the above *nna* influences would generate the following directed graph (shown in Prolog format: *e(Id, Var1, State1, Var2, State2)*):

```
e( 1, coldSwim,  up,   nna,    up).
e( 2, coldSwim,  down, nna,    down).
e( 3, temp,      up,   nna,    down).
e( 4, temp,      down, nna,    up).
e( 5, coldSwim,  up,   and004, up).
e( 6, temp,      up,   and004, up).
e( 7, coldSwim,  down, and005, up).
e( 8, temp,      down, and005, up).
```

```
e( 9, and004,    up,   nna,    steady).
e(10, and005,    up,   nna,    steady).
```

Edges 1, 2, 3, and 4 were generated via expanding the direct and inverse influences. The direct connection between *coldSwim* and *temp* means that *nna* being *up* or *down* could be explained by *coldSwim* being *up* or *down* respectively. The inverse connection between *temp* and *nna* means that *nna* being *up* or *down* could be explained by *temp* being *down* or *up* respectively. Edges 5, 6, 7, 8, 9, and 10 were generated via the QCM steady rule: i.e. competing upstream influences can cancel out. There are two upstream influences to *nna*. In the case of (e.g.) both *coldSwim* and *temp* going up, then we could explain *nna=steady* via the conjunction of upstream influences (connected using the conjunction *and004*). For full details on the QCM-compiler expansion process, see [23].

Once the model has been converted to a graph, it is executed via abduction. Roughly speaking, abduction is a search over some existing theory for consistent subsets that are relevant for achieving some goal. When contradictions can occur, abduction must create *worlds*: maximal consistent relevant subsets. If multiple such worlds can be generated, then a BEST assessment operator selects the preferred world(s). For more details on abduction, see [4, 13, 21]. The QCM abductive engine executes across a the and-or graph generated above to find consistent pathways (ordered sets of edges) between output goals back to known inputs to the system. Pathways that cross unmeasured variables must make assumptions. Worlds are generated by collecting maximal subset of these pathways with compatible assumptions. Abductive world generation is a synonym for QR branching. Incompatible state assignments to a single variable (e.g. *up* and *down*) can be stored in different worlds. Chatter in QCM would result in an explosion in the number of generated worlds.

QCM is much simpler QR language than other approaches: e.g. time averaging [16], consolidation [3], first-order logic qualitative modelling [2], non-linear QR [31] or the other QDE processors such as QSIM [14]. Despite being NP-hard [4, 22, 25], QCM is a practical tool for validating many real-world theories such as certain fielded expert systems and scientific theories from neuroendocrinology [20]. Given a library of known behaviour (expressed as input, output pairs), a qualitative theory can be faulted via QCM if it cannot create worlds containing all the outputs connected back to the inputs. In this manner, QCM has found faults in theories of neuroendocrinology published in international refereed journals [8, 9, 20, 23]. Interestingly, the faults were found using data taken from the papers that proposed those theories. Also, the faults had never been detected before, even by the reviewers of those journals. Further, when experts reviewed the detected faults, they found them exciting and insightful to their domain [28].

4 QCM Temporal Variants

In standard QR, time landmarks are created by the reasoner when it realises that the sign of some variable has changed state. In QCM, time is taken explicitly from the domain where the model is executed. QCM, variables are renamed once for each time step in the simulation; e.g. *population* could be renamed to *population1, population2 ... populationT* where T is some time point. (In this article, we will not defend the QCM approach to time except to say that some

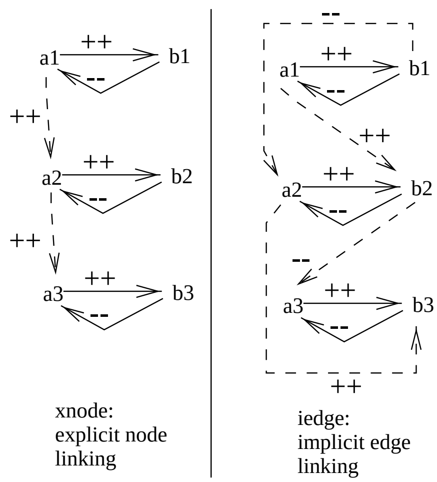

Figure 2: *Direct(A,B)* and *inverse(B,A)* renamed over 3 time intervals using different time linking policies. Dashed lines indicate time traversal edges.

classes of QCM models do not chatter.) Once the renamings are done, then QCM connects variables at time i to variables at time $i+1$ via some temporal linking policy. Consider the theory *direct(A,B)* and *inverse(B,A)*. If we execute this theory over three time steps, we could search one of the spaces illustrated in Figure 2. In the *explicit node linking* language (or XNODE), we only cross time on the nodes explicitly denoted as time nodes by the user (in this example, A). In the *implicit edge linking* language (or IEDGE), we cross time on all edges. Elsewhere, we have shown that both XNODE and IEDGE are adequate temporal linking policies for reproducing the known correct behaviour of real-world models [30]. Many other such temporal linking policies can be defined (see [19, 30]), but for the purposes of this discussion, two will suffice.

The key observation of this paper is that these temporal linking policies exhibit different amounts of chatter. To show these, we need to examine the behaviour of these linking policies over a wide range of models. To do this, we took a quantitative real-world model of a fisheries system using the equations from [1] (pages 135-141). Second, we built a QCM form of the fisheries model as shown in Figure 3. Note that this fisheries model is ambiguous concerning how to handle time. We must add in a temporal causal interpretation (e.g. XNODE or IEDGE) in order to handle the feedback loops. When using XNODE, we must somehow assign our explicit time traversal nodes. One candidate are the *first derivative variables* which show time rate of change. Example first derivative variables in the fisheries models are *fish population change* and *change in boat numbers*.

Third, we exercised this qualitative theory in a variety of ways using five problem generators. Generator1 and generator2 mutated the model by corrupting edges or adding edges respectively. Generator1 flipped between 0 to 17 edges edges in fisheries (inverse to direct, or visa versa), chosen at random. Generator2 added in 0, 5, 10, 15, 20, 25 or 30 new edges at random (checking all the time that the added edges did not exist already in the theory). Once the model was mutated, it was then copied over T time steps and connected via one of the

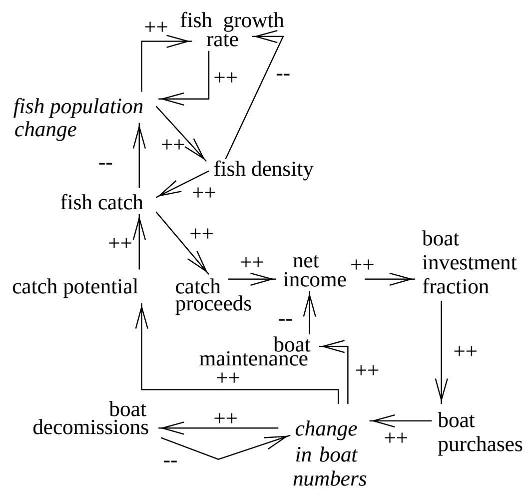

Figure 3: The fisheries model. Adapted from [1] (pp135-141). Variables in italics are the *first derivative variables* used in the XNODE study.

XNODE or IEDGE temporal linking policies.

Generators 3,4, and 5 built input and output data sets for the mutated model. inputs were always observations found in the first copy of the model. Outputs were always observations not found in this first copy. Generator3 ran the quantitative fisheries model. Generator4 created random numeric observations. Generator 3 and 4 produce 105 qualitative data sets (input and output sets comprising *up*s and *down*s) each for testing these models (there are 105 comparisons between the 15 runs of the model used in this study).

Generator5 simulated a poorly measured domain. U percent of the output from generator3 and generator4 are discarded. Generator5 produces 10 variants for U at 0, 10, 20, 30, 40, 50, 60, 70, 80, and 90 percent unmeasured.

5 Results

This section shows the number of worlds generated via a combination of the above problem generators. The generators were combined in an arbitrary manner to ensure the creation of a wide range of QR problems.

In the first experiment, generators 1,3 and 5 were combined to run QCM over variants of the fisheries model with different numbers of corrupted edges. Between 0 to 17 edges were corrupted using generator1 20 times to create 360 new models. These were further mutated 10 times each when generator5 was called (i.e. resulting in 3600 models). These were exercised for both XNODE and IEDGE using the qualitative 105 data sets created by generator3; i.e. 3600*105*2=756,000 runs. The results are shown in Figure 4.

In the second experiment, generators 2,4 and 5 were combined to run IEDGE and XNODE over the fisheries model. Between 0 to 30 edges were added to the fisheries model using generator2 20 times to create 140 new models. These were further mutated 10 times each when generator5 was called (i.e. resulting in

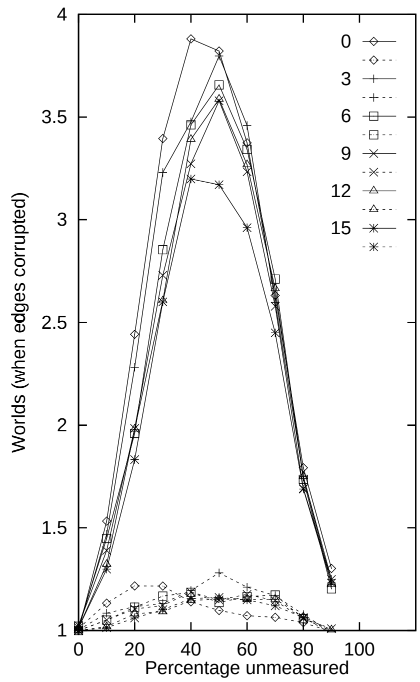

Figure 4: Corrupting 0 to 15 edges with IEDGE (solid lines); XNODE (dashed lines).

1400 models). These were exercised for both XNODE and IEDGE using the qualitative 105 data sets created by generator3; i.e. 1400*105*2=294,000 runs. The results are shown in Figure 5.

Two effects are clear. Firstly, over a wide range of models, the implicit linking policy generates far more worlds/branches than the explicit linking policy (compare the solid lines of IEDGE with the dashed lines of XNODE in both sets of results). There is nothing surprising in this result: implicit linking policies offer more connections of variables across time. The greater the number of time edges, the less restricted the temporal search space and the greater likelihood of chatter from conflicting assumptions.

The second effect is the remarkable similarity in the shape all the output graphs. Over a wide range of models, the same *hump shape* can be seen. When nothing is unmeasured, only one world is created. As more and more is unmeasured, the number of worlds created increases. This is consistent with the argument above that less data means more worlds. Then a previously unreported effect in the QR literature is seen. As the percent unmeasured increases even further, the number of created worlds drops back to nearly one; i.e. *even less data means even less worlds.* This appears to be a robust effect since it is seen in different linking policies (XNODE, IEDGE) over a range of different problems (corrupting existing edges or adding new edges).

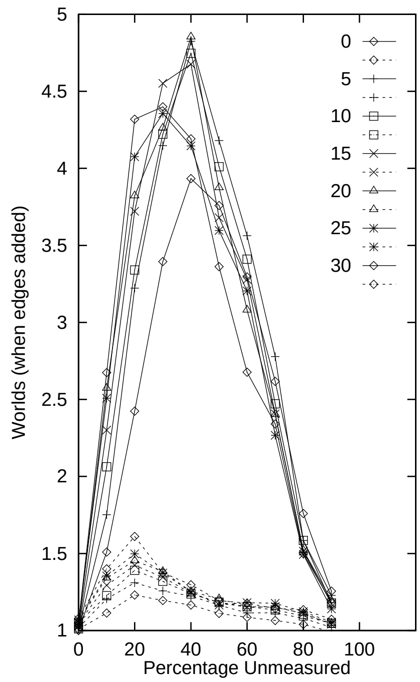

Figure 5: Adding 0 to 15 edges with IEDGE (solid lines); XNODE (dashed lines).

6 Discussion

This section argues that we can explain the *less data means less worlds* effects via the relevant envisionment policy of the abductive procedure called by QCM.

The behaviours generated by a qualitative reasoning system are called the *envisionments* of that system. Total envisionments are those behaviours which are possible, given some fixed collection of objects in some configuration. Extension generation in default logic [24] systems or the ATMS [7] produce total envisionments. For example, consider the influences around *nna* in the Smythe'87 model. Suppose our inputs were *(coldSwim=up, dex=up)* and our outputs were *nna=up*. Total envisionment would build three worlds for *nna* going *up*, *down*, and remaining *steady*.

A reasonable restriction on the total envisionments are the attainable envisionments; i.e. all behaviours possible from some given initial state [10]. QSIM using attainable envisionments. However, note that attainable envisionments, even with its restricted search space, would still include three worlds.

In essence, total and attainable envisionments are asking what follows from certain state assignments. Relevant envisionments answers another question: *Given some behaviour of interest can these behaviours be reached given certain state assignments?* To answer this question with total or attainable envisionments, one must compute the total or attainable envisionments, then search them for the required behaviour. This approach runs the risk of generating many behaviours that are irrelevant to the process of finding what percentage of known behaviours can be explained by a hypothetical model. Relevant envisionment moves the goal of the simulation into the inference procedure. Relevant envisionments only generate the behaviour possible from some given initial state (the inputs) and which can lead to some desired final state (the outputs). Continuing with the *nna* example, *nna=down* and *nna=steady* would not be found in a relevant envisionment since those assumptions do not generate a proof of *nna=up*.

To perform relevant envisionments, QCM restricts the search to the downstream transitive closure of the inputs and the upstream transitive closure of the outputs. For more details, see [23].

These three envisionment policies result in different percentages of the search space being explored: total explores more than attainable which explores more than relevant. Consequently, these envisionment policies make differing amounts of assumptions about state variables: total assumes more than attainable which assumes more than relevant.

We can now offer an explanation for the hump shape. Initially, as the percent unmeasured decreases, the *less data means more worlds* effect dominants. However, after some point (around 50 percent unmeasured), another effect dominants. Branching is a function of the number of conflicting assumptions made by the qualitative reasoner. As the percentage of unmeasured variables increases, the size of the input and output sets decreases. In total envisionments, this has no effect on the number of assumptions made since total envisionments offers assumptions for all variables. However, attainable envisionments make less assumptions while relevant envisionments make even less. Hence, for low-assumption envisionment policies (e.g. relevant envisionments), branching is reduced when the amount of data from the domain is reduced. We conjecture that since the QSIM literature is full of debates on taming chatter, that attainable envisionments still makes excessive state variable assumptions.

7 Conclusion

A general tool for reducing chatter is relevant envisionments. Relevant envisionments produce fewer assumptions and hence a reduced chance of conflicting assumptions. To use relevant envisionments, the knowledge engineer must identify some small subset of variables that are of critical importance to the decision making process which prompted the creation of the model. For example, if you are using QR to test if some disaster conditions could occur, then model that disaster condition. Identify a conjunction of state assignments which can create the disaster. Add these to the output set. Use a relevant envisionment algorithm to see if this conjunction is possible.

Apart from relevant envisionments, other modelling tools which reduce chatter in a QCM framework are:

- *Don't just model some physical situation, measure it as well.* Somehow find constraining values for either most of the data or a small portion of the data. However, avoid measuring half the data since this seems to generate the most worlds. We suspect this advice is general to any relevant envisionment QR system. To test this, we would need to repeat the above study on (e.g.) a QSIM system modified for relevant envisionments.

- *Use XNODE.* This advice implies adding first derivative variables (e.g. *fish* *population change*) to QCM models. Jumps between time steps will only occur on these first derivative variables. However, when using XNODE, use the first derivative variables sparingly since all first derivative variables increases the number of time edges. Time edges can create chatter since the greater the number of time edges, the less restricted the temporal search space and the greater likelihood of chatter from conflicting assumptions.

In summary, chatter is a fundamental property of qualitative reasoning. However it can be significantly reduced via the appropriate selection of strategies for QR systems.

References

[1] H. Bossel. *Modeling and Simulations.* A.K. Peters Ltd, 1994. ISBN 1-56881-033-4.

[2] I. Bratko, I. Mozetic, and N. Lavrac. *KARDIO: a Study in Deep and Qualitative Knowledge for Expert Systems.* MIT Press, 1989.

[3] T. Bylander. A Critique of Qualitative Simulation from a Consolidation Viewpoint. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(2):252–263, March/April 1988.

[4] T. Bylander, D. Allemang, M.C. M.C. Tanner, and J.R. Josephson. The Computational Complexity of Abduction. *Artificial Intelligence*, 49:25–60, 1991.

[5] D.J. Clancy and B.K. Kuipers. Model Decomposition and Simulation: A component based qualitative simulation algorithm. In *AAAI-97*, 1997.

[6] J. Crawford, A. Farquhar, and B. Kuipers. QPC: A Compiler from Physical Models into Qualitative Differential Equations. In B. Faltings and P. Struss, editors, *Recent Advances in Qualitative Physics.* The MIT Press, 1992.

[7] J. DeKleer. An Assumption-Based TMS. *Artificial Intelligence*, 28:163–196, 1986.

[8] B. Feldman, P. Compton, and G. Smythe. Hypothesis Testing: an Appropriate Task for Knowledge-Based Systems. In *4th AAAI-Sponsored Knowledge Acquisition for Knowledge-based Systems Workshop Banff, Canada*, 1989.

[9] B. Feldman, P. Compton, and G. Smythe. Towards Hypothesis Testing: JUSTIN, Prototype System Using Justification in Context. In *Proceedings of the Joint Australian Conference on Artificial Intelligence, AI '89*, pages 319–331, 1989.

[10] K. Forbus. Interpretating Measurements of Physical Systems. In *AAAI '86*, pages 113–117, 1986.

[11] P. Fouche and B. Kuipers. An Assessment of Current Qualitative Simulation Techniques. In B. Faltings and P. Struss, editors, *Recent Advances in Qualitative Physics*, pages 263–278. The MIT Press, 1992.

[12] Y. Iwasaki. Qualitative Physics. In P.R. Cohen A. Barr and E.A. Feigenbaum, editors, *The Handbook of Artificial Intelligence*, volume 4, pages 323–413. Addison Wesley, 1989.

[13] A.C. Kabas and P. Mancrella. Generalized Stable Models: A Semantics for Abduction. In *ECAI-90*, 1990.

[14] B. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29:229–338, 1986.

[15] B. J. Kuipers, C. Chiu, D. T. Dalle Molle, and D. R. Throop. Higher-order derivative constraints in qualitative simulation. *Artificial Intelligence*, 51:343–379, 1991.

[16] R. Levins and C.J. Puccia. *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging.* Harvard University Press, Cambridge, Mass., 1985.

[17] J.E.A. McIntosh and R.P. McIntosh. *Mathematical Modeling and Computers in Endocrinology.* Springer-Verlag, 1980.

[18] T. Menzies, A. Mahidadia, and P. Compton. Using Causality as a Generic Knowledge Representation, or Why and How Centralised Knowledge Servers Can Use Causality. In *Proceedings of the 7th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1992.

[19] Tim Menzies, Sam Waugh, Simon Goss, and Robert F. Cohen. Evaluating a Temporal Causal Ontology. Submitted to FOIS '97, 1997. Available from `http://www.cse.unsw.EDU.AU/~timm/pub/docs/97fois`.

[20] T.J. Menzies. On the Practicality of Abductive Validation. In *ECAI '96*, 1996. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96abvalid.ps.gz`.

[21] T.J. Menzies. Applications of Abduction: Knowledge Level Modeling. *International Journal of Human Computer Studies*, 45:305–355, September, 1996. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96abkl1.ps.gz`.

[22] T.J. Menzies, R.F. Cohen, Waugh S, and S. Goss. Applications of Abduction: Testing Very Long Qualitative Simulations, 1997. Submitted to IEEE Transactions of Data and Knowledge Engineering. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/97iedge.ps.gz`.

[23] T.J. Menzies and P. Compton. Applications of Abduction: Hypothesis Testing of Neuroendocrinological Qualitative Compartmental Models. *Artificial Intelligence in Medicine*, 10:145–175, 1997. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96aim.ps.gz`.

[24] R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[25] B. Selman and H.J. Levesque. Abductive and Default Reasoning: a Computational Core. In *AAAI '90*, pages 343–348, 1990.

[26] G.A. Smythe. Hypothalamic noradrenergic activation of stress-induced adrenocorticotropin (ACTH) release: Effects of acute and chronic dexamethasone pre-treatment in the rat. *Exp. Clin. Endocrinol. (Life Sci. Adv.)*, pages 141–144, 6 1987.

[27] G.A. Smythe. Brain-hypothalmus, Pituitary and the Endocrine Pancreas. *The Endocrine Pancreas*, 1989.

[28] G.A. Smythe. Concerning errors in the Smythe '87 model., 1992. Personal communication.

[29] J.L. Top and J.M. Akkermans. Computational and Physical Causality. In *IJCAI '91*, 1991.

[30] S. Waugh, T.J. Menzies, and S. Goss. Evaluating a Qualitative Reasoner. In Abdul Sattar, editor, *Advanced Topics in Artificial Intelligence: 10th Australian Joint Conference on AI*. Springer-Verlag, 1997.

[31] K.M. Yip. Understanding Complex Dynamics by Visual and Symbolic Reasoning. *Artificial Intelligence*, 51:179–221, 1991.