A PARALLEL MODEL OF (SEQUENTIAL) PROBLEM SOLVING

Mary S. Riley and Paul Smolensky

Institute for Cognitive Science
University of California, San Diego
La Jolla, California 92093

Nature of Rules and Their Interaction

This paper is concerned with the nature of the rules involved in solving problems and the interaction between those rules. We describe a parallel model designed to solve a class of relatively simple problems from elementary physics and discuss its implications for models of problem solving in general. We show how one of the most salient features of problem solving, sequentiality, can *emerge naturally* within a parallel model that has no explicit knowledge of how to sequence analysis.

Consider the problem shown in Figure 1. The task is to determine the qualitative effects of increasing the resistance of R_2 on other circuit values, assuming the applied voltage and resistance of R_1 remain unchanged.

A common approach to modelling the process of solving problems like these is to assume that knowledge is organized as a production system, similar to that shown in Table 1 (see Riley, 1984, for a review). Here the model's rules for making inferences are in the form of condition-action pairs, or *productions*. The condition specifies the particular elements and relations that must be present in the data base in order for the condition to be true. When the production system is solving a problem, the conditions of the various productions are tested in order until one of them is true; the action of that production is then performed. The action generally makes some change in the data base which in turn means the condition of a different production will be true, causing another action to be performed.

Since production systems are universal computers, they can be programmed to display any behavior (Newell, 1981). However certain kinds of behavior can be achieved with other styles of computation in more economical, elegant, extendible and natural ways. Features that are intrinsic to, or naturally incorported within, a pure production system approach are:

1) *Sequentiality:* each action taken utilizes the knowledge contained in precisely one rule.
2) *Directionality:* the knowledge encoded in each rule has a distinct directionality from input (condition) to output (action).
3) *Exact matching:* each rule acts only when a perfect match to its condition occurs.
4) *Determinism:* performance will be identical on all solutions of a given problem.

Within the production system approach it is difficult to naturally avoid certain difficulties:

1) *Lack of robustness under degradation of rules* (either removal of correct rules or addition of incorrect ones).

This research was supported by a grant from the System Development Foundation and by contract N00014-79-C-0323, NR667-437 with the Personnel and Training Research Programs of the Office of Naval Research.

2) *Lack of robustness under ill-formed problems* that contain inconsistent or insufficient given information.
3) *Lack of variability* in routes to correct answers or in correctness of answers; a problem for modelling human behavior.
4) *Need for explicit conflict resolution rules* that determine which rule will apply when several have true conditions.

The parallel distributed processing approach represented by our model naturally avoids these difficulties, but has its own problems, as we shall see.

The Model

Our model has been constructed within the framework of harmony theory (Smolensky, 1983, 1984). Rules are represented as a collection of nodes in a network, as shown in Figure 2. A typical rule is $< I\ down, V_1\ down, R_1\ same >$; this rule states that the combination of changes "voltage across R_1 goes down, current goes down, R_1 stays the same" is a consistent set (Ohm's Law). In fact, the rules consist precisely of all allowed combinations of qualitative changes in circuit variables that are consistent with each circuit law. There are 65 such instances. [1]

Unlike condition-action rules, there is no directionality associated with the variables in the harmonium rules.

In a particular problem, only some of the instances represented by harmonium rules are relevant. To represent this, each rule node has an *activation value* that can be either 1 (active) or 0 (inactive).

In addition to rule nodes, the harmonium model contains nodes for representing the problem in terms of qualitative changes in circuit variables. Some nodes have values given by the problem ($R_2\ up$, $R_1\ same$, $V_{total}\ same$). The model's answer is represented by values assigned to the remaining nodes.

As shown in Figure 2, there is a connection between an individual circuit variable node and each rule involving it; this connection is labelled by the appropriate value for that variable according to that rule.

The goal of processing is to find a set of rule nodes to activate and a set of values for circuit variables that are consistent with those rules. Search toward this goal is guided by a measure of the consistency between a set of activated rules and a set of circuit variables: this measure is called the *harmony function*. The state of highest harmony should be the correct answer to the problem.

Processing is probabilistic and constructed so that at any moment, *the higher the harmony of a state, the more probable it is*. The spread in this probability distribution is determined by a system parameter called the *computational temperature T*. Initially, all rules are inactive, the circuit variables given by the problem are assigned their values, and the remaining circuit variables are assigned random values. The temperature is set to a high value, and the stochastic search begins. Rules are activated and deactivated, circuit variable values are changed (except the given ones), and states are visited in accordance with their harmony. As the search continues, this temperature is lowered, and the system displays less and less randomness, focussing in on the states of highest harmony. After a while, the temperature becomes very low, and the search is effectively stopped: an answer has been selected.

---

1. Thirteen each for: Kirchoff's Law, the equation $R_{total} = R_1 + R_2$, and three versions of Ohm's Law (one each for R_1, R_2, and R_{total}).

Sequentiality of deduction seems to be completely lacking from the harmonium model, although it is a very salient feature of human problem solving. Just the same, in creating this model we expected it to display an emergent seriality. If a single circuit variable is monitored during the search, it will fluctuate randomly at first, and eventually "lock in" to a value that is very resistant to change. The temperature at which this occurs is the "freezing temperature" for that variable. We expected that different variables would have different freezing temperatures, depending on the problem situation; the one with highest freezing temperature would settle first, which would in turn determine the value selected for the variable with the next lower freezing temperature, and so on.

In addition to T, harmonium models have a second global parameter, κ; it is the sole parameter in the definition of the harmony function. When κ is near one, only rules that match the current guesses for circuit values *exactly* can become active without lowering the harmony of the state; for low values of κ, approximate matches are sufficient. Initially, κ is small, approximate matching is encouraged, and many rules become activated; as the computation proceeds, κ approaches one and the set of active rules shrinks toward the five that exactly match the answer.

As the node for each circuit variable freezes into a value, it does so under the influence of all the active rule nodes connected to it. Unlike a production system, matching for rules need not be exact, and several rules can act at the same time.

The harmony function we used, as well as the schedule for lowering T and raising κ, are shown in Figure 3. A trial consisted of 400 iterations of 100 node updates each; since there are 79 nodes in the model, this corresponds to slightly over 500 updates of each node.

The stochasticity of the model produces variability in the behavior. In a run of 30 trials, the correct answer was produced 28 times. When the 30 values the system assigned to the circuit variables for each of the 400 iterations are averaged, Figure 4 results. In this graph, *up* is represented by 1 and *down* by -1. Initially, the values for all variables fluctuate around zero; eventually, each goes towards the correct value. The time at which the four decisions are made are indicated in the last portion of this figure, in which the region between .5 and $-.5$ has been removed. The sequence of assignments is R_{total}, I_{total}, V_1, V_2; *the sequence of "inferences" that emerges naturally from the parallel processing is exactly the same as the sequence produced in a production system model.*

The harmonium model displays both types of robustness that is difficult to achieve naturally with production systems. Since individual inferences are made under the simultaneous influence of several rules, they are less vulnerable to degradation of a single rule. When inconsistent information is given in a problem, the harmonium model finds the most consistent (highest harmony) answer possible. When insufficient information is given, the system finds one of the correct answers, and finds different answers on different trials. Such a robust tendency to form coherent interpretations of inputs is important both for modelling human cognition and for building intelligent machines.

Extensions

While the parallel distributed processing approach has certain advantages over the production system approach, it also has grave disadvantages. The most serious is the difficulty of performing symbol manipulation. Without variable binding mechanisms, types and tokens, it is difficult to imagine how to develop a general model capable of analyzing a variety of circuits; our model is specialized to a single circuit, and even so we must replicate the rules encoding valid instances of Ohm's Law three times (once for each relevant piece of the circuit).

It may be psychologically plausible to postulate a small collection of networks like our harmonium model (or perhaps one integrated, larger network) incorporating knowledge about similarly simple circuits (e.g. a circuit with two resistors wired in parallel). These could conceivably serve as prototypes that would be invoked to deal with pieces of or schematic versions of larger circuits. However some powerful mechanism would still have to coordinate the parallel analyses of circuit fragments.

It is tempting to use a production system for this coordination, combining the strengths of the two approaches. Such a hybrid model might well be able to analyse complex circuits, but would display the production system weaknesses (lack of robustness, and so forth) in those aspects of the analysis that were relegated to the production system.

One interpretation of such a hybrid model is that the production system component is actually just a complex parallel processing network *viewed at a higher level of description*; the hybrid is of descriptive levels, there are not two independent processes. It is a major goal of ours to see if parallel models are capable of exhibiting emergent production-like behavior; the emergent seriality of the present harmonium model is an example of just such behavior.

Discussion

The harmonium model has *implicit* knowledge of circuit laws that enable it to model naturally the nonsymbolic, intuitive component of problem solving that is difficult to model naturally with production systems and is particularly salient in expertise. At the same time it lacks the *explicit* knowledge of symbolic laws that most experts possess. Thus to model expert problem solving in general, it seems necessary to imbed the harmonium model within a hybrid parallel/production system model. We are however investigating whether the symbolic component of expert's processing can be preempted with conditions of very short response times, making such experimental conditions appropriate for testing the pure harmonium model. We are also considering unschooled electronics experts to see to what extent they are free of conscious rule application in their solution of simple circuit problems.

Much work remains to be done in analyzing the variation in the model's performance, and assessing the dependence of performance on the schedules for T and κ and the representation of the circuit. Indeed it is the development of more powerful representations within the parallel distributed processing paradigm that is the primary goal of harmony theory; by trying to enrich the knowledge of our harmonium model to incorporate more "symbol-like" explicit knowledge of circuit laws, we hope to gain more insight into how symbol manipulation might emerge from parallel distributed processing.

References

A. Newell (1981). Physical symbol systems. *Perspectives on Cognitive Science*, D. A. Norman (ed.), 37-85. Norwood, NJ: Ablex.

M. S. Riley (1984, expected date). Structural understanding in problem solving. Doctoral dissertation. Pittsburgh, PA: LRDC, University of Pittsburgh.

P. Smolensky (1983). Schema selection and stochastic inference in modular environments. *Proceedings of the National Conference on Artificial Intelligence*. Washington, DC.

P. Smolensky (1984). The mathematical role of self-consistency in parallel computation. Submitted to the Sixth Annual Conference of the Cognitive Science Society. Boulder, CO.

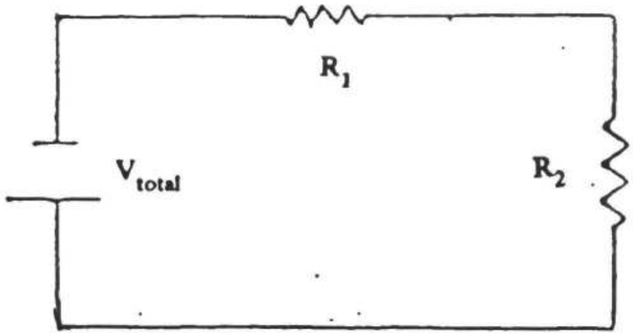


Figure 1. A series circuit with two resistors, R_1 and R_s. What are the effects of an increase in the resistance of R_2, assuming that E_{total} and the resistance of R_1 have remain the same?

Table 1

A Simple Production System for Solving the Problem in Figure 1.

Productions

Condition	Action
P1. $<V_x \textit{ same}, R_x \textit{ up}>$	$<I \textit{ down}>$
P2. $<R_x \textit{ up}, R_y \textit{ same}>$	$<R_{total} \textit{ up}>$
P3. $<V_x \textit{ down}, V_{total} \textit{ same}>$	$<V_y \textit{ up}>$
P4. $<R_x \textit{ same}, I \textit{ down}>$	$<V_x \textit{ down}>$

Problem Solution

Problem Representation	Matched Production	
Cycle	Condition	Action
1. R_2 *up*, R_1 *same*, V_{total} *same*	P2. $<R_2 \textit{ up}, R_1 \textit{ same}>$	$<R_{total} \textit{ up}>$
2. R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*	P1. $<V_{total} \textit{ same}, R_{total} \textit{ up}>$	$<I \textit{ down}>$
3. R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*, I_{total} *down*	P4. $<R_1 \textit{ same}, I \textit{ down}>$	$<V_1 \textit{ down}>$
4. R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*, I_{total} *down*, V_1 *down*	P3. $<V_1 \textit{ down}, V_{total} \textit{ same}>$	$<V_2 \textit{ up}>$
5. R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*, I_{total} *down*, V_1 *down*, V_2 *up*		

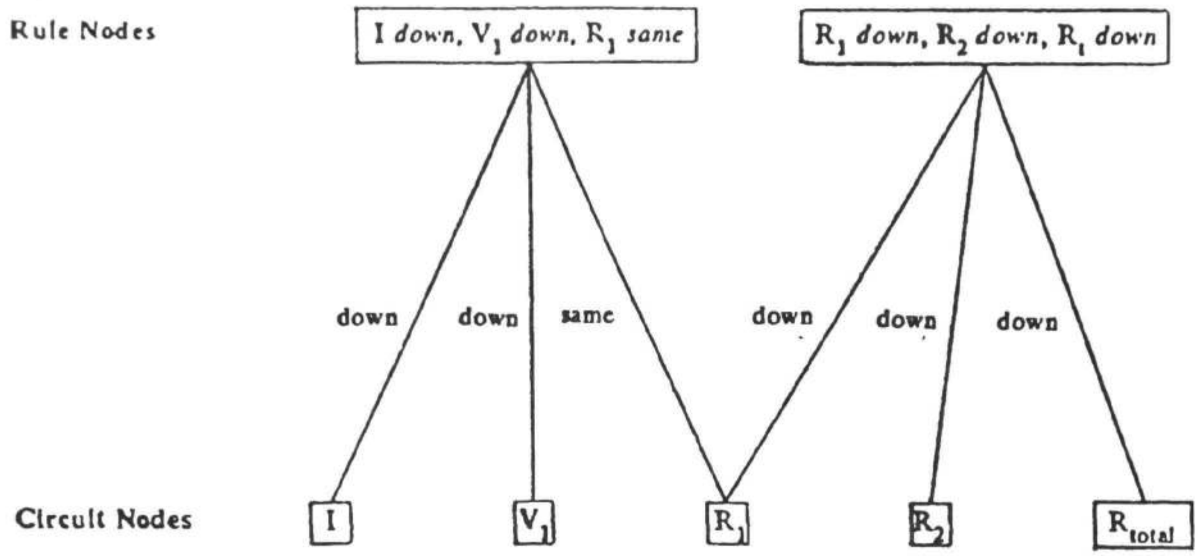
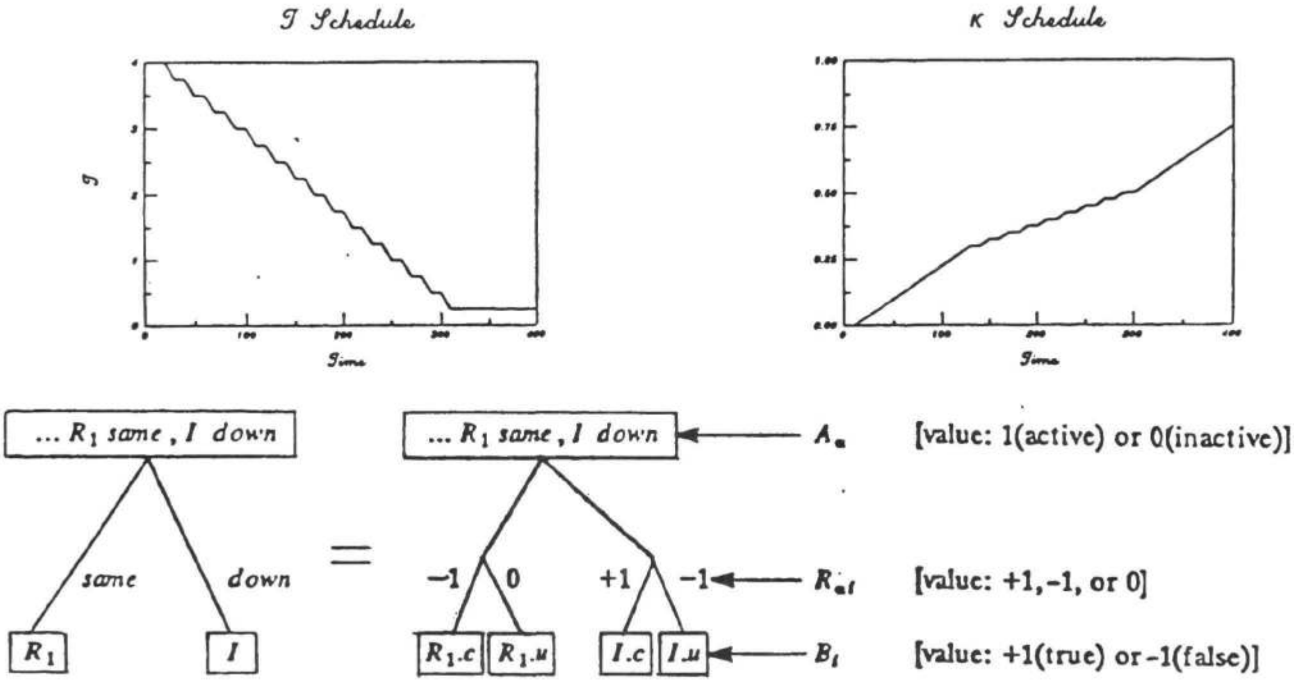


Figure 2. A portion of the harmonium model's network.



I:	*up*	*down*	*same*	
I.c:	+1	+1	−1	["*I* changed"]
I.u:	+1	−1	± 1	["*I* went up"]

Harmony function:

$$H = \sum_{a} A_a \sum_{l} \left(R_{al} B_l - \kappa \left| R_{al} \right| \right)$$

Figure 3. Schedules for T and κ, representation of *up*, *down*, *same*, and harmony function used in the harmonium model.

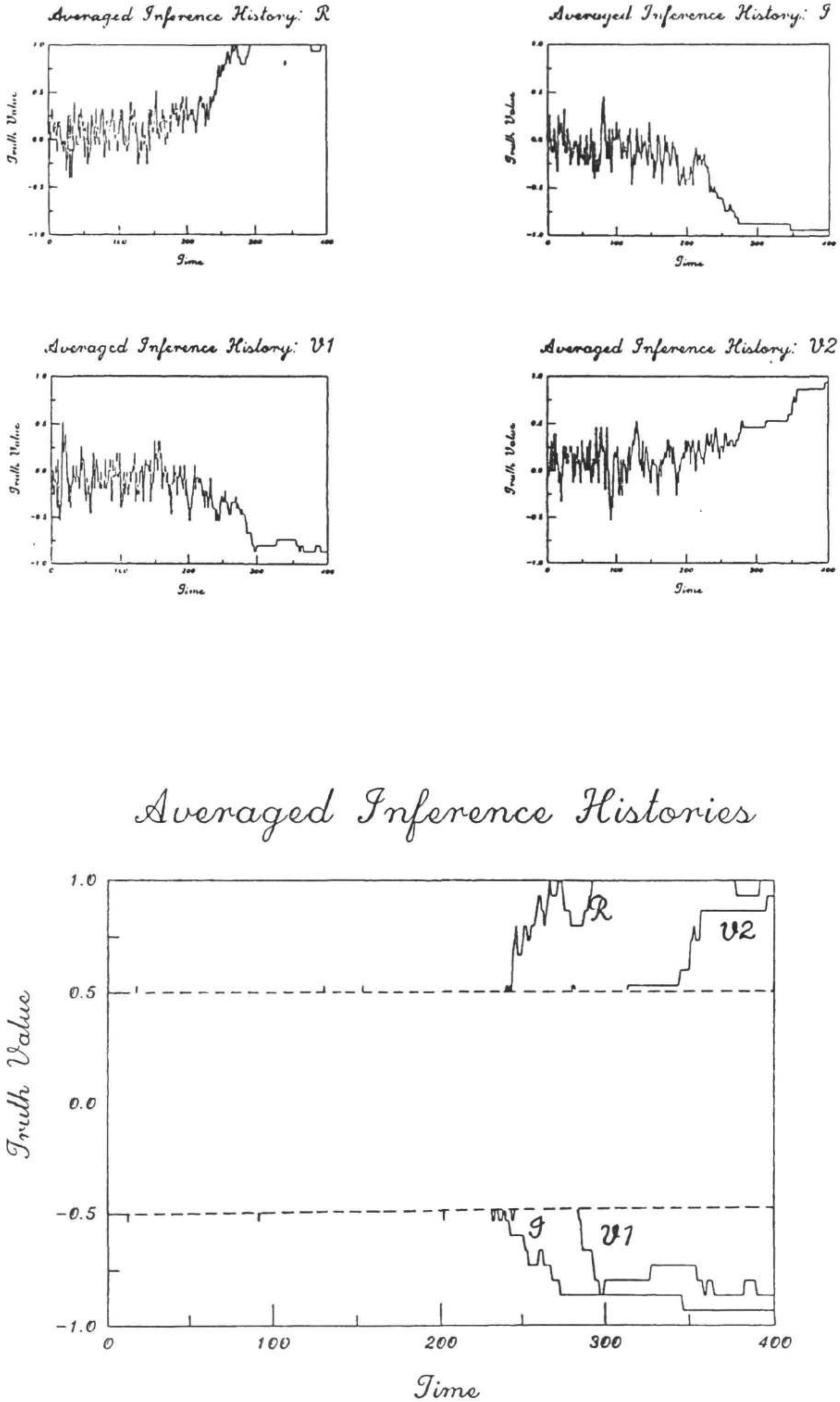


Figure 4. Emergent sequentiality: the decisions about the direction of change of the circuit variables "freeze in" in the order $R = R_{total}$, $I = I_{total}$, V_1, V_2 (R and I are quite close).