MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 811 December 1984

# HYPOTHESIZING AND REFINING CAUSAL MODELS

Richard J. Doyle

## Abstract

An important common sense competence is the ability to hypothesize causal relations. This paper presents a set of constraints which make the problem of formulating causal hypotheses about simple physical systems a tractable one. The constraints include: 1) a temporal and physical proximity requirement, 2) a set of abstract causal explanations for changes in physical systems in terms of dependences between quantities, and 3) a teleological assumption that dependences in designed physical systems are functions.

These constraints were embedded in a learning system which was tested in two domains: a sink and a toaster. The learning system successfully generated and refined naive causal models of these simple physical systems.

The causal models which emerge from the learning process support causal reasoning – explanation, prediction, and planning. Inaccurate predictions and failed plans in turn indicate deficiencies in the causal models and the need to rehypothesize. Thus learning supports reasoning which leads to further learning. The learning system makes use of standard inductive rules of inference as well as the constraints on causal hypotheses to generalize its causal models.

Finally, a simple example involving an analogy illustrates another way to repair incomplete causal models.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505.

## Acknowledgements

This work was supported and guided by many people. I would like to thank:

Patrick Winston, my thesis supervisor, for showing me how to do research by example (how else?), for putting issues and results in perspective when I had become myopic, and for having the same interests at the same time.

Boris Katz, for developing the wonderful parser which made creating the knowledge base for this work a pleasure rather than a chore, and for looking benevolently over my shoulder throughout.

David Kirsh, for helping me organize my thoughts and my words.

Peter Andreae, Ken Forbus, Mike Kashket, Rick Lathrop, Jintae Lee, Ryszard Michalski, David Tye, Dan Weld, Ken Yip, and others for being interested, for engaging in fruitful discussions, and for making helpful suggestions and honing my ideas.

Bruce Donald and Mike Erdmann, for support throughout the process of completing a thesis.

TABLE OF CONTENTS

# CHAPTER 1
# INTRODUCTION

## The Problem

"Common sense reasoning" subsumes a vast repertoire of familiar but hard to articulate skills for understanding and dealing with the world. One of the most important skills underlying common sense is the ability to recognize and describe regularities in the world in terms of *causal* relations. Causal descriptions enable us to generate useful explanations of events, recognize the consequences of our actions, reason about how to make things happen, and constrain hypotheses when expected events do not occur. Without the ability to construct causal descriptions, we would be unable to impose any order on the bewildering changes that pervade our everyday experiences; we would be unable to understand or control our environments.

Imagine waking up in the morning to find the refrigerator door ajar and the food spoiled. One can construct an explanation easily, even if one is not quite awake. People commonly turn down the volume control on the home stereo before turning the power on, anticipating and knowing how to prevent a possibly unpleasant jolt. When the lamp does not work, we will sooner or later change the bulb, check the plug, and check the fuse.

## Goals of this Work

This thesis investigates ways to construct causal descriptions of physical systems which undergo continuous changes. The learning process produces a causal model – a set of rules which make explicit the causal mechanisms underlying the behavior of the system and its parts. The particular goals of this work are:

- To present common-sense heuristics and a learning procedure which show how causal models of physical systems can be hypothesized.
- To show how a causal model can be refined by generalizing over further experience.
- To show how a learned causal model can support causal reasoning, particularly planning.
- To illustrate how representations for quantities provide a basis for qualitative reasoning which supports both learning and planning.

- To demonstrate how causal models can be extended through the use of analogy.

## Key Ideas – Constraints on Causal Hypotheses

This thesis argues for a set of constraints on causal explanations which make the problem of formulating causal hypotheses a tractable one. These constraints are:

- Temporal and physical proximity.

This constraint reflects the common sense notion of causality which states that causally connected events are contiguous in space and time.

- Causal explanation abstractions.

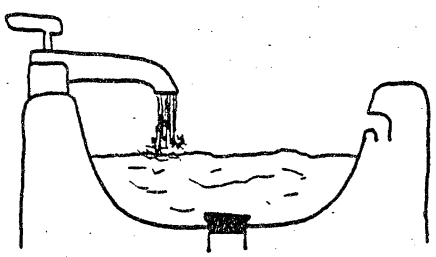
Changes and causal relations are represented perspicuously in terms of changing values of quantities and dependences between quantities. This representation language exposes constraints which reduce the number of causal explanation types to a manageable size.

- Teleological assumptions.

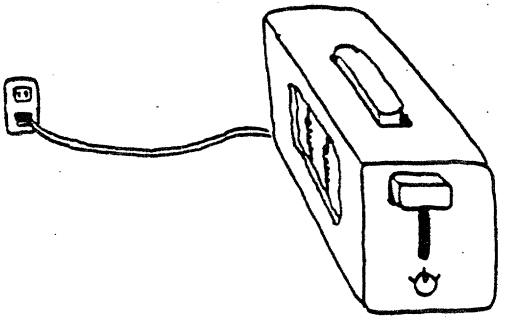
Assumptions about the nature of dependences between parameters in designed physical systems can be used to test causal hypotheses.

## The Domains

A learning system, with these embedded constraints to guide it, was tested in two experimental domains: a sink, the familiar kind of sink one finds in kitchens and other places, and a toaster.



The Kitchen Sink



The Toaster

These particular devices were chosen for several reasons. They are composed of many parts without being overwhelmingly complex. They display continuous changes which can be modelled by qualitative representations for quantities. Water rises in the sink; bread turns to toast in the toaster. Because external inputs control their behavior, planning problems can be posed. External inputs of the sink include the setting of the faucet and placement of the stopper. For the toaster, external inputs include the depressing of the lever, the placement of the bread, and the setting of the thermostat.

In addition, the sink displays an equilibrium state – water flowing in at the tap and out at the safety drain. The problem of explaining why water rising in the sink does not overflow will illustrate how abstract causal explanations can effectively constrain the set of admissible causal hypotheses so that the correct one can be generated quickly.

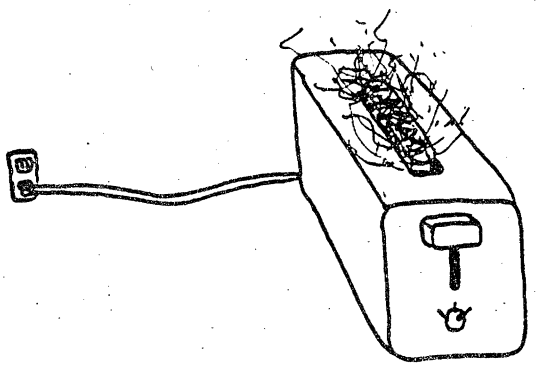
Finally, the toaster will demonstrate how the learning system can produce usable causal models of electronic devices without resorting to a wiring diagram. This thesis is concerned with naive rather than expert causal hypothesizing and reasoning.

## A Preview

Causally connected events are often temporally and spatially contiguous. This principle is easy to see in reasoning about sinks and toasters. Pulling the plug makes the water flow out immediately. The controls on the blender across the counter cannot affect the toaster. This principle is embodied in heuristics which guide the recognition of causality.

These heuristics capture a useful, but not always correct notion of causality. Causes and effects may appear to be quite separated in time when the causal chain is hidden. Furthermore, some causal relations involve interactions which occur over large distances without an apparent medium, e.g. gravity.

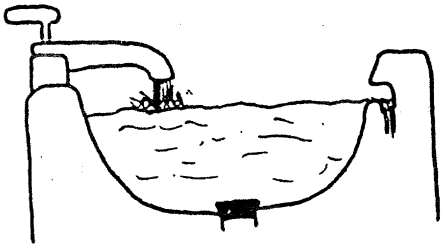
Further experience provides opportunities to generalize causal models originally constructed on the basis of a single experience. Various generalization heuristics allow conditions to be dropped, boundaries on the closed system to be better circumscribed, and dependences between parameters to be recognized. For example, two observations of the toaster for two different settings of the thermostat lead to the recognition of the correspondence between the thermostat setting and the darkness of the resulting toast.



How to Avoid Burnt Toast?

Quantities model the continuous changes which occur in physical systems and expose constraints which can be exploited. Qualitative reasoning with quantities supports both learning of and planning with causal models.

The most interesting problem posed for the learning program in the sink domain is to understand what is happening when the water reaches the level of the safety drain and stops rising. This is a passive change of behavior; no overt, external action occurs. The learning program solves this problem by making an imaginative though tightly constrained conjecture about the function of the safety drain.

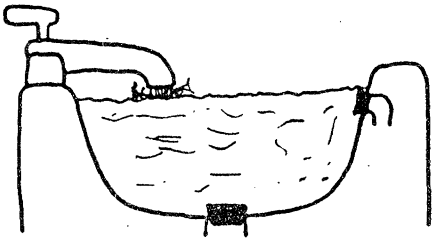


Why did the Water Stop Rising?

The learning program knows from its background knowledge about equilibrium states that either 1) no influences, or 2) balancing influences, could be the explanation for the change. At this point in the learning session, the learning system already knows that the faucet being on and the presence of the water column makes the water rise. These causes are intact, so equilibrium must be the explanation. Since only one influence is explicitly known, there must be an unknown influence of opposite direction, i.e., one which makes the water fall. These inferences lead to the identification of the safety drain as the causal culprit. Without the reasoning afforded by quantities, the learning program would have been hard put to construct the correct causal explanation in this situation.

The reasoning which the causal model supports also provides feedback about deficiencies in the model. When plans fail, this can indicate an incomplete or too-abstract model. Analogies can help when the causal model is deficient by mapping missing information from other areas of knowledge.

The most interesting of the planning problems in the sink domain is one that cannot be solved with the causal model as it exists after the initial learning session is complete. The problem is this: how to make the water rise above the safety drain. The planner, by reasoning in terms of quantities, realizes that the equilibrium state at the safety drain must be changed to a state of increase. However, it finds no way to do so. The problem is solved finally by extending the causal model through the use of an analogy. This analogy is illustrated below.



When You've Seen One Drain...

Chapter 2 presents the representation language for describing physical systems and their changes. Chapter 3 describes the constraints on causal hypotheses exploited by the learning system and the heuristics and procedures based on these constraints which are used to hypothesize and refine causal models. This chapter is the core of the thesis. Chapter 4 shows that learning has taken place by describing how a planner can use the causal models which emerge from the learning process. Chapter 5 discusses analogy as one way to extend causal models. Chapter 6 recounts the accomplishments of this thesis, discusses limitations, and suggests areas for further research.

## How This Work Fits In

Previous work in artificial intelligence has addressed acquiring descriptions of static or non-causal structures or concepts [Winston 75, Michalski and Chilausky 80, Michalski 83, Mitchell 82], representing causality [Rieger 76, Rieger and Grinberg 77], representing and reasoning about structures which undergo changes [Hayes 79, de Kleer 79, Forbus 84, Kuipers 82, de Kleer and Brown 83, Simmons 83, Weld 84], and representing continuous changes in physical systems in terms of quantities [Forbus 84].

This thesis builds on this foundation and shows how causal models of physical systems which undergo continuous changes can be hypothesized and refined. The ideas presented hopefully point the way towards further research on integrating learning and common sense reasoning systems.

# CHAPTER 2
# REPRESENTING PHYSICAL SYSTEMS AND THEIR CHANGES

The input to the learning program is English text. This text substitutes for visual perceptions and describes the structure of the sink and toaster and the changes that occur over time. The text is translated by a parser which is part of a general natural language/knowledge representation system. This chapter summarizes the knowledge representation language of this system, describes its time representation, and shows how physical systems and their changes can be represented within this framework. The entire parser/representation system is described in [Katz 80, Katz and Winston 82, Doyle and Katz 84].

## Relations and Truth-Values

The basic structure in the knowledge representation scheme is the *relation.* Relations have the following form:

<SUBJECT RELATION OBJECT>

Some examples are:

<STOPPER IN DRAIN>

<COILS PART-OF TOASTER>

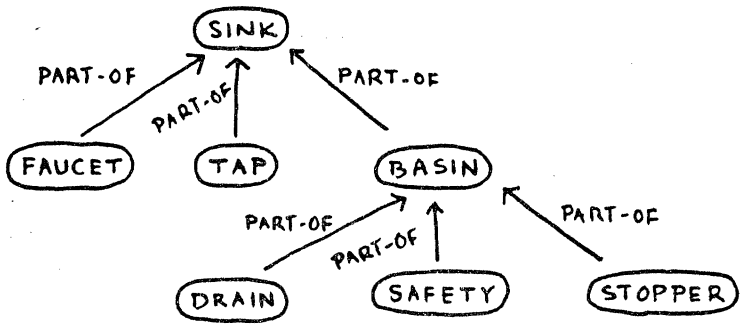
<WATER-COLUMN CONNECTED-TO TAP>

Any SUBJECT or OBJECT may itself be a <SUBJECT RELATION OBJECT> triple and OBJECTS can be omitted:

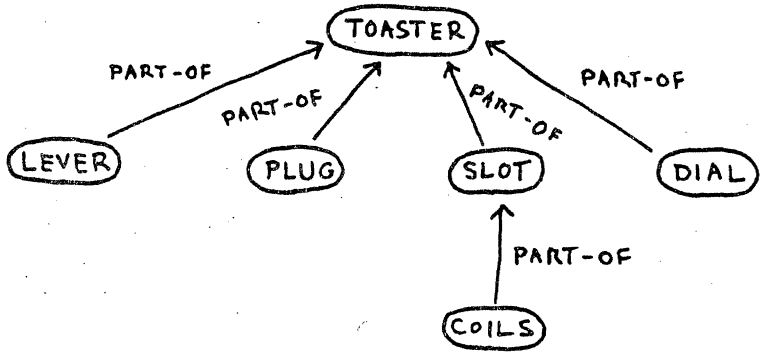
<<WATER APPEAR> IN BASIN>

Truth-values are attached to each relation, either TRUE or FALSE.

The PART-OF relation enables hierarchical structural descriptions of the sink and the toaster to be constructed.

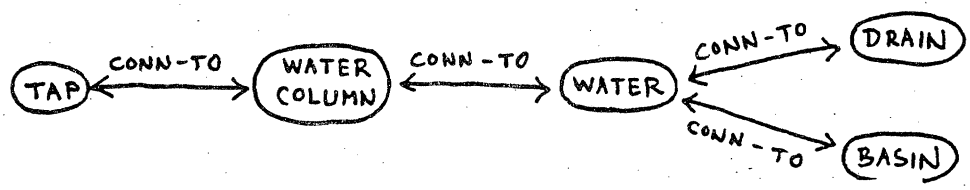



Components of the Sink




Components of the Toaster

The CONNECTED-TO relation gives stronger information about the visible topology of a physical system.




Topological Description of the Sink

## Relations Can Change

Before change can be represented, there must be a representation for *time*.

Our time representation is simpler than others that have appeared in the artificial intelligence literature, [Allen 81, McDermott 82, Vere 83, Simmons 83]. [Simmons 84] is an excellent discussion of the important issues for designing a time representation.

The basic unit of time is the *interval*, which is represented by specifying two *moments*, one being the beginning of the interval, the other being the end of the interval. Moments themselves are the primitive intervals and they meet at points. Time is divided into a sequence of moments at the finest level of resolution, and all intervals are defined on these moments. Moments are conveniently represented as integers.

As an example, the interval <3,8> starts at the beginning of the third moment and stops at the end of the eighth moment. An interval such as <5,5> is well-defined. This interval is exactly the fifth moment.

In principle, it would be useful to be able to define intervals on top of other intervals, rather than on moments only, to any number of levels. Then seconds, minutes, hours, days, etc. would be easy to represent. (See [Allen 81] for a solution). The two-level partitioning of time into moments and intervals is sufficient to support the temporal reasoning the learning program needed to do.

Another limitation in our time representation is the absence of any information about scale, i.e., about the absolute duration of any particular moment. (See [Vere 83] for a solution and [Simmons 84] for a discussion). Once again, there was no need for this kind of information to support the research at hand, so the issue was not addressed. Only information about the ordering of events was necessary, not about their relative durations.

The interaction between the parser and the time representation is clean and simple. The parser strips temporal adverbs from sentences and makes them available to the knowledge system. There is an internal clock which keeps track of the current moment, the "now". Temporal adverbs adjust the clock as follows:

- INITIALLY – sets the clock to 1.
- ALREADY – sets the clock to 0, i.e., sometime in the past.
- ALWAYS – sets the clock to $-\infty$, i.e., for all time.
- NEXT – advances the clock once.
- LATER – advances the clock twice, to create an intermediate interval during which nothing changed.

When a relation is inserted into the knowledge base, both a truth-value and an interval are assigned to it, representing when that relation has that truth-value.

The starting moment of the interval is always the current moment given by the internal clock. The stopping moment is, by default, $+\infty$. Relations are assumed to be *persistent*.

Finally, it is possible to represent how relations can change. The interval/truth-value construct is generalized to a *history*, a list of interval/truth-value pairs. If a relation is asserted again, it is not created anew, rather the history of the relation is modified according to the following rules:

- If the value has not changed; do nothing.
- If the value has changed and time has passed; close the previous interval and create a new persistent interval with the new value.

As an example, the following set of sentences,

Initially, the stopper is in the drain.
Later, the stopper is not in the drain.
Next, the stopper is still not in the drain.
Later still, the stopper is in the drain again.

would be represented in the knowledge base as:

IN-1 <STOPPER IN DRAIN>
(1–TRUE–2) (3–FALSE–5) (6–TRUE–>)

Note that the final interval is an open interval, while the others have been closed.

For the purposes of this research, it was critical to be able to represent a sequence of events, in which the learning program would look for causal relations. It was necessary to be able to control the exact temporal ordering of relations in the knowledge base. The particular interaction between the parser and the time representation described above might be called the "sequence-of-events" mode. Other interactions between the parser and the time representation are possible.

## Quantities Capture Continuous Change

Truth-value histories represent how propositional statements about the world change, but they are not well-suited to representing how properties of objects can change. In physical systems, changes often occur continuously. The representations for quantities described in this section capture continuous properties of physical

objects. They are borrowed from Ken Forbus' seminal work on representing and reasoning about physical processes [Forbus 84].

Quantities are always associated with a physical object and the possible values of a quantity correspond to well-defined states of the object. The values of quantities are symbolic, not numeric. An ordering is imposed on the set of values of every quantity, so it is possible to reason about the relative magnitudes of different values of a quantity. It is also possible to compare the relative magnitudes of values of different quantities. However, no information about absolute magnitudes is given.

A quantity has two parts, an *amount* and a *rate*. The rate is the first derivative of the amount. As an example, changes in the temperature of the coils in a toaster can be represented as:

```
QUANTITY-1      <COILS QUANTITY TEMPERATURE>
                (1-TRUE->)

AMOUNT-1        <TEMPERATURE AMOUNT>
                (1-COLD-2) (3-CHANGING-4) (5-HOT-5) (6-CHANGING-7)
                (8-COLD->)

RATE-1          <TEMPERATURE RATE>
                (1-ZERO-2) (3-POSITIVE-4) (5-ZERO-5) (6-NEGATIVE-7)
                (8-ZERO->)
```

The rate of one quantity may be the amount of another. A more powerful representation for the flight of a rock might involve another quantity called velocity whose amount is the same as the rate of the height. Higher-order derivatives can be represented as well.

The above example shows how quantities fit into the overall representational scheme. If the values placed in histories are generalized from truth-values to arbitrary values, then quantities can be represented without any additional machinery.

The set of possible values for a quantity and the ordering imposed on that set is called a *quantity space* [Forbus 84]. Usually the quantity space is a total ordering, but partial orderings are possible too. The quantity space for the amount of the coils' temperature is:

```
(COLD -> HOT)
```

The quantity space for the rate of the coils' temperature is:

(NEGATIVE -> ZERO -> POSITIVE)

Quantities extend the ability to represent change along two dimensions: first, continuous as well as discrete changes can be represented; second, the *direction* of a change can be represented, because of the ordering imposed on the set of values for a quantity. Representing directions of change can support reasoning about the next value of a quantity and equilibrium states. Quantities not only add a richer representation for change, they add reasoning power as well.

### Dependences Capture Causality

The task of the learning program is to identify causality. Causality can be represented in terms of quantities in a concise manner. A quantity that can be affected by another quantity is functionally dependent on that quantity. It is useful to define two kinds of *dependences*, the *function* and the *influence* [Forbus 84].

An example of a function is:

FUNCTION-1 <Q-1 FUNCTION Q-2> NEGATIVE

The *independent quantity* or causing quantity is Q1 and the *dependent quantity* is Q2. A dependence is usually *signed*, to indicate whether it is a direct or inverse dependence. The meaning of this function is:

- when the independent quantity's amount *increases*, the dependent quantity's amount *decreases*
- when the independent quantity's amount *decreases*, the dependent quantity's amount *increases*

The definition of a direct (positive) function is symmetrical in the obvious way. The meaning of an influence such as

INFLUENCE-1 <Q-3 INFLUENCE Q-4> POSITIVE

is slightly different:

- when the independent quantity's amount is *positive*, the dependent quantity's rate is *positive* (and the dependent quantity's amount is *increasing*)

- when the independent quantity's amount is *negative*, the dependent quantity's rate is *negative* (and the dependent quantity's amount is *decreasing*)

A negative influence is defined in the obvious way.

A function is a dependence between the amounts of two quantities or the rates of two quantities. An influence is a dependence between the amount of one quantity and the rate of another. It is possible, by chaining influences through several quantities, to represent higher-order derivatives.

It is also useful to define the *correspondence*, which is a relation (in the mathematical sense) between the values of two quantities. A correspondence represents an observation about empirical links between the values of two quantities – but it is not yet clear which quantity is independent and which is dependent.

A correspondence such as:

CORRESPONDENCE-1 <Q-5 CORRESPONDENCE Q-6> NEGATIVE

means there is a one-to-one correspondence between the values in the one quantity's quantity space and the values in the reverse of the other quantity's quantity space. A correspondence is symmetric.

## Causal Rules Capture Behavior

The representation of causality afforded by quantities facilitates the construction of causal models of physical systems by the learning program. Causal models consist of a set of *causal rules* defined on a set of physical objects and a set of quantities associated with those physical objects. The causal model makes explicit the causal relations underlying behavior. Causal rules describe causality at two levels: At the quantity level in terms of independent quantities, dependent quantities, and dependences, and constraints on the ranges of the values of the quantities. At the physical level in terms of a set of preconditions and a set of effects, both of which are relations on physical objects, and the times these relations hold. The quantity level aids the learning program in the recognition of causality, because quantities and dependences support abstract causal explanations. The physical level provides a means for describing causality in terms of objects and relations at the physical real-world level for use by a planner.

An example of a causal rule illustrates its form:

```
THE OBJECTS ARE
   THE DRAIN
   THE WATER

THE QUANTITIES ARE
   THE FLOW OF THE DRAIN
   THE HEIGHT OF THE WATER

THE DEPENDENCES ARE
   <FLOW INFLUENCE HEIGHT> NEGATIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <DRAIN CONNECTED-TO WATER> TRUE
   (T) <DRAIN CONTAIN STOPPER> FALSE
   (T) <DRAIN PART-OF BASIN> TRUE
   (T) <WATER CONNECTED-TO WATER-COLUMN> FALSE
   (T) <WATER CONNECTED-TO DRAIN> TRUE
   (T) <WATER CONNECTED-TO SAFETY> FALSE
   (T) <WATER IN BASIN> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <HEIGHT AMOUNT> BELOW-SAFETY
   <HEIGHT RATE> ZERO

THE PHYSICAL-EFFECTS ARE
   (T+2) <DRAIN CONNECTED-TO WATER> FALSE
   (T+2) <WATER CONNECTED-TO DRAIN> FALSE
   (T+2) <WATER IN BASIN> FALSE

THE QUANTITY-EFFECTS ARE
   (T) <FLOW AMOUNT> POSITIVE
   (T) <FLOW RATE> ZERO
   (T) <HEIGHT RATE> NEGATIVE

   (T+2) <HEIGHT AMOUNT> ZERO
   (T+2) <HEIGHT RATE> ZERO
```

## A Causal Rule

This causal rule describes how water flows out of a drain.

# CHAPTER 3
# PROPOSING AND GENERALIZING THE CAUSAL MODEL: LEARNING

Learning systems are often described by specifying an initial representation, a target representation, and a learning procedure. Using this framework, the construction of causal models can be described as follows:

The initial representation is a temporally ordered sequence of events describing behaviors of the physical system being investigated.

The target representation is a set of causal rules which describes the various behaviors of the physical system in terms of causal relations; each causal rule is a description of causality at two levels: the abstract level of quantities and dependences and the real-world level of physical objects and relations. The set of causal rules makes up the causal model.

The task of the learning system is to recognize causality in the sequence of events and render the identified causal relations in the form of causal rules. This chapter explains the learning procedure in full detail. This procedure was implemented in a learning system called JACK (Justifiably Assimilating Causal Knowledge).

## Identifying Causality

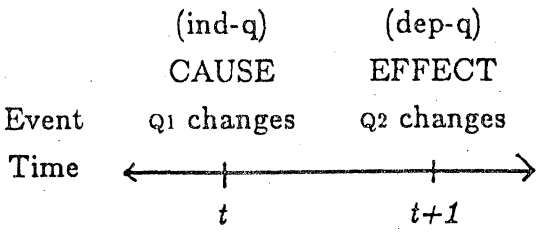
The common sense view of causality that the learning program exploits is the following: Two events which are causally connected are contiguous in space and time. This is a useful, but not always correct notion of causality, as will be discussed later.

Repetition of conjoined events is also a clue to causality but is not used by the learning program as a basis for proposing causal relations. If this heuristic were to be used alone, some kind of thresholding mechanism would be needed, which would likely be *ad hoc*. However, causal relations do have to satisfy repeatability after being proposed.

## Temporal Adjacency

A statement of the form "A causes B" almost always implies "B immediately follows A". This is not always correct, but this is the assumption which forms the basis for the temporal adjacency heuristic. This heuristic is used as follows:

The learning program looks for two changes, one immediately following the other in time. More specifically, because causal relations can be represented succinctly by dependences between quantities, the learning program looks for the pattern of one quantity's value changing immediately after another quantity's value changed. The following diagram illustrates:




Temporal Adjacency

Whenever such a pattern appears in the sequence of events, the learning program suspects causality, and the two quantities may be linked in a dependence.

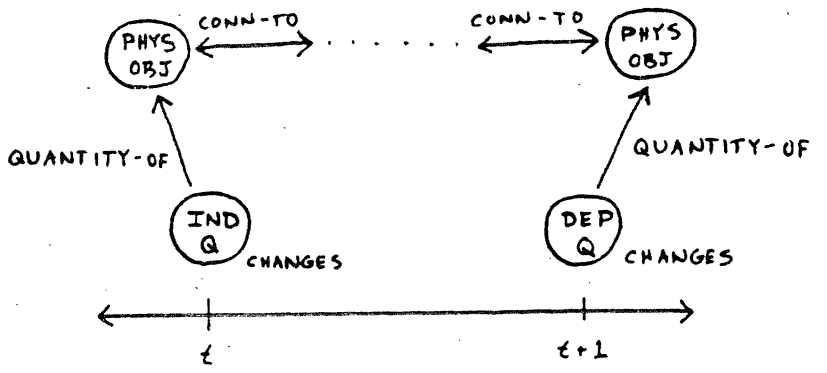
However, the temporal adjacency heuristic does not provide enough constraint by itself because coincidences are possible. Additional constraint is provided by the physical connectedness heuristic.

## Physical Connectedness

In our common sense view of causality, in order for two events to be causally connected, there must be some kind of medium between the two along which "forces" or "agents" which mediate the causality can propagate. This medium might be, for example, a mechanical, rigid connection or a fluid coupling. For the learning program to identify causality, the exact nature of the medium is not important, just whether a medium does in fact exist.

Physical connectedness is tested by determining if there is a CONNECTED-TO relation between the two objects associated with the quantities whose changes satisfied temporal adjacency. Since the CONNECTED-TO relation is transitive, the physical connectedness heuristic can be satisfied by a chain of objects.

The two heuristics – temporal adjacency and physical connectedness – are combined as illustrated in the following diagram:




Temporal Adjacency and Physical Connectedness

In summary, the identification of causality has two steps. First, two quantities are found, one changing immediately after the other. Second, it is verified that the objects which the two quantities are associated with are physically connected.

There is a theme of reasoning at two levels throughout this research. The temporal adjacency heuristic operates at the quantity level; here causality is suspected. The physical connectedness heuristic operates at the physical level; here causality is reinforced.

Coincidences can be defined and recognized. A coincidence is two events which satisfy the temporal adjacency heuristic, but do not satisfy the physical connectedness heuristic.

## Limitations and Extensions of the Heuristics

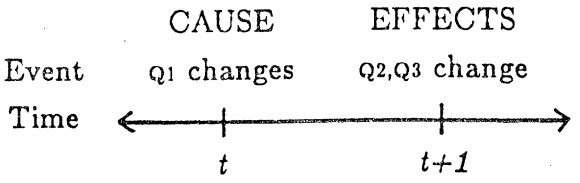
The two heuristics given above for identifying causality are general and powerful enough to be sufficient in a large number of situations. However, they can fail to identify some classes of causally connected events. This section discusses the limits of the two heuristics and some simple extensions.

Consider first the temporal adjacency heuristic.

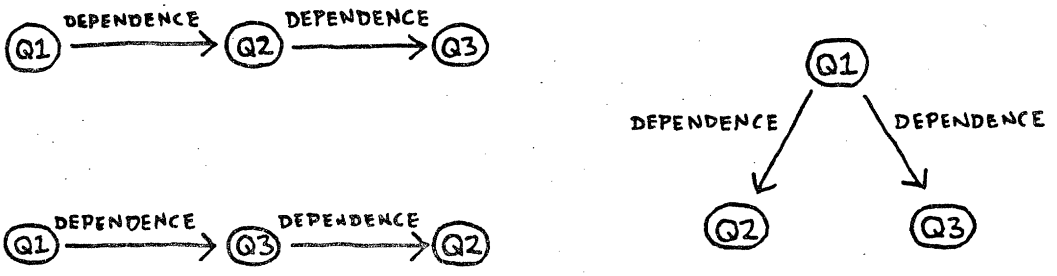
It is not strictly true that causality always implies that the cause immediately precedes the effect in time. For instance, in the case of purely mechanical, rigid connections, the cause and effect occur simultaneously. Think about pushing on one end of a rod. There is no delay before the other end of the rod starts moving.

However, the physical connectedness heuristic is already powerful enough in some cases to supply sufficient constraint to correctly identify causality when the temporal adjacency heuristic fails because of simultaneity.

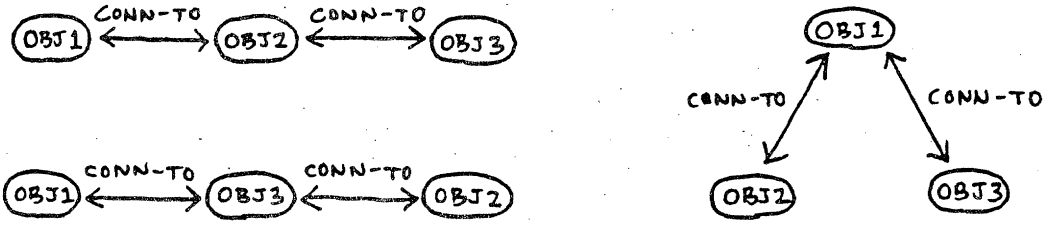
Consider the following situation:




There are three possible interpretations of causality:




The correct interpretation is the one which displays the same topology as the graph of the physical connectedness relations on the physical objects associated with the quantities.




Since the CONNECTED-TO relation is symmetric, at least one of the dependences must be given a direction independently (e.g. by the temporal adjacency heuristic or if a change in a quantity is due to an external action) to "seed" the constraint propagation. Otherwise, the independent quantities and the dependent quantities would not be distinguishable. If any cycles exist in the dependence and connectedness

graphs, they too must be individually seeded.

Another flavor of causality that is captured neither by temporal adjacency nor simultaneity is the "delayed" reaction. But delayed reactions are really just delusions; there is no real temporal discontinuity. The problem is that the structural description being used to understand the causality is too abstract, so that the causal mechanism is hidden. If enough lower-level detail were added to the model, then a causal chain would be revealed, and each causal relation in the chain would satisfy either temporal adjacency or simultaneity.

The solution then, is to have the capability for hierarchical descriptions – both structural and temporal. A hierarchical partitioning of time was discussed in the chapter on knowledge representation; a hierarchical description of structure would be a useful parallel. These descriptions could support a hierarchical description of causality so that what looked like a delayed reaction at one level, would be a continuous causal chain at a lower level. [See Allen 81, Davis et al 82, Davis 83, de Kleer and Brown 83, for work that has addressed the issue of hierarchical descriptions].

To summarize: When events in time are adjacent, the temporal adjacency heuristic is applicable. When events in time are simultaneous, the physical connectedness heuristic can sometimes disambiguate. When events in time are discontinuous, perhaps the model can be fleshed out until all events are adjacent or simultaneous.

Consider now the physical connectedness heuristic.

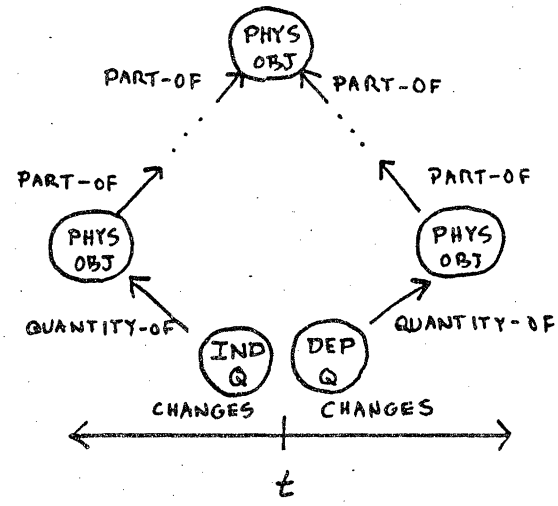
A problem with the physical connectedness heuristic is that it is incapable of handling situations which involve "forces at a distance". More accurately, it is incapable of modelling phenomena such as gravity, magnetism, heat exchange, etc. unless some kind of medium is proposed. This level of understanding can be likened to that of nineteenth-century physicists who proposed the "ether" to explain the propagation of electromagnetic radiation within the solar system.

This problem can be partially addressed by relaxing the physical connectedness requirement to physical *proximity*. However, there is a danger of removing too much constraint and it is not at all clear when two objects are near enough to possibly affect each other.

A better idea is to consider only objects which are part of the same physical system. Two objects are part of the same device if their PART-OF hierarchies join. This heuristic embodies a teleological assumption about parts of a device being

intended to interact. The structure of the physical system itself is used to draw boundaries within which to look for causal relations.

These heuristics for proposing causality effectively prune the space of admissible hypotheses. No heuristic is perfect though; these can pass incorrect hypotheses.




The Weaker Simultaneity and Same Device Heuristics

### Constraint at the Quantity Level

There are only a finite number of ways to explain changes in dependent quantities in terms of dependences and changes in independent quantities. Representing changes and causality in terms of quantities and dependences exposes constraints that collectively define a kind of syntax of causal explanation. In particular, three kinds of knowledge at the quantity level can constrain the hypothesizing of causal relations.

- Discrete vs. continuous change.

  Types of changes in quantities are linked to types of dependences.

| Type of change in independent quantity | Type of change in dependent quantity | Type of dependence |
|---|---|---|
| discrete | discrete | function |
| discrete | continuous | influence |
| continuous | continuous | function |

- Signs and directions of change.

The signs or directions of change of quantities and dependences have to be consistent.

| Direction of change or sign of independent quantity | Direction of change or sign of dependent quantity | Sign of dependence |
|---|---|---|
| + | + | + |
| + | − | − |
| − | + | − |
| − | − | + |

- Second-order causal explanations.

Define the state of a quantity to be its direction of change and the signs of the impinging contributions on that quantity. There are a finite number of ways in which the state of a quantity can change.

| Current state | Add + | Add − | Del + | Del − |
|---|---|---|---|---|
| Constant C (0,[ ]) | I | D | x | x |
| Increase I (+,[+]) | I | E | C | x |
| Decrease D (−,[−]) | E | D | x | C |
| Equilibrium E (0,[+,−]) | E | E | D | I |

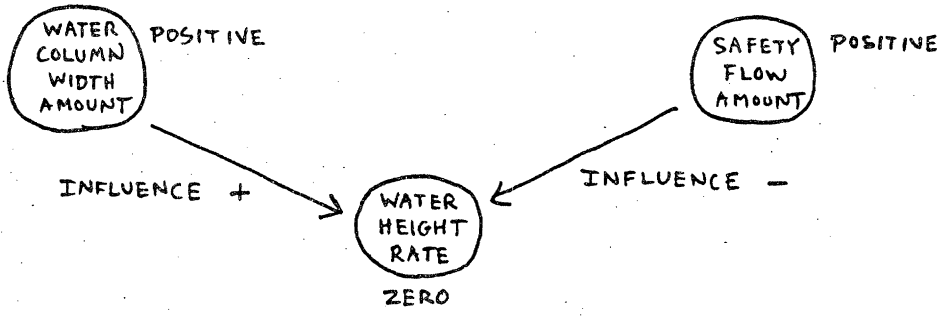
For example, a decreasing quantity can change to a steady quantity because a single negative contribution went away or because the quantity achieved an equilibrium state when a positive contribution was added.

The number of second-order causal explanations for changes in quantities is reduced considerably by a felicity condition [VanLehn 83] which excludes tradeoff situations. In general, contributions of opposite direction may resolve to a net change in either direction rather than an equilibrium state. Furthermore, equilibrium states may not be stable; they may be easily perturbed to a positive or negative tradeoff situation.

An example from the sink domain illustrates how constraint at the quantity level facilitates the identification of causality.

During the learning session, after the faucet is turned on and while the stopper is in the drain, the water rises to the level of the safety drain and stops. JACK realizes that either no influences or balancing influences could be the explanation

for the change. At this point in the learning session, the learning program already knows that the faucet being on and the presence of the water column makes the water rise. These causes are intact, so equilibrium must be the explanation. Also, the unknown contribution must be of opposite direction, i.e., it must make the water fall. All of this reasoning ultimately leads to the identification of flow at the safety drain.




An Equilibrium State

Without the reasoning available at the quantity level, particularly the representation for equilibrium states, the learning program would have failed to construct the correct causal explanation in this situation.

## Tradeoffs and Overdetermined Systems

The number of dependences in the various types of causal explanations at the quantity level outlined above is in all cases minimal. If a quantity is not steady, a single dependence explains why it is changing. If a quantity is steady, either there are no dependences, or if evidence indicates an equilibrium state, there are exactly two dependences.

In general, a changing quantity can be the result of any number of interacting dependences (except zero), all sharing the same dependent quantity, whose net effect is to move the quantity in a particular direction. Similarly, an equilibrium state can be achieved by any number of dependences greater than one, again all sharing the same dependent quantity, whose net effect is a state of balance.

However, the following qualitative checks can be performed to see if the set of dependences is at least not inconsistent with the observed change in the dependent quantity.

- For a non-steady quantity, there must be at least one dependence in the set of dependences whose contribution has the correct sign. (The sign of a dependence's contribution is the resolution of the sign of change of the independent quantity and the sign of the dependence).
- For a steady quantity, if there are no dependences, that is sufficient explanation. Otherwise, the quantity is in a state of equilibrium. In this case, there must be at least one dependence in the set of dependences in each direction.

The amount of constraint may not be sufficient when there are several dependences to resolve. For instance, if a quantity is decreasing, three negative influences and one positive influence may not be a correct explanation, because the magnitude of the single positive influence might be greater than the sum of the magnitudes of the negative influences.

When there are several dependences which satisfy the causality-proposing heuristics, the complete and correct way to verify that the net effect of the dependences is consistent with the change in the dependent quantity is to sum the contributions of all the dependences. But this would require knowledge about the absolute magnitudes of quantities and perhaps even an equation to represent the functional relationship captured by the dependence. This kind of quantitative information is not available. Therefore, tradeoff situations are not allowed.

This felicity condition [VanLehn 83] does not preclude overdetermined systems where several dependences contribute to move a quantity in the *same* direction. The learning program can construct correct causal explanations in these situations. They are the only situations in which JACK can construct causal explanations which involve more than the minimal number of dependences.

## Making Hypotheses

JACK learns by proposing causal explanations for changes in quantities. This section outlines how the causality-proposing heuristics and knowledge about quantities constrain the hypotheses which the learning program generates to explain changes.

Given a change in a (dependent) quantity, JACK tries to construct a causal explanation in terms of dependences and independent quantities with the following procedure:

Step 1.

Different kinds of changes in quantities are associated with different kinds of causal explanations in terms of dependences.

- If the quantity has stopped changing, look for no dependences or balancing dependences.
- If the quantity has begun changing, look for a new dependence or a broken equilibrium state.

Step 2.

If JACK cannot explain changes in terms of known dependences, then the causality-proposing heuristics are used to propose new dependences. This is when learning takes place. The learning program proposed new dependences by searching for a change in an independent quantity and an associated physical object which satisfy either:

- temporal adjacency or simultaneity and physical connectedness, or
- temporal adjacency or simultaneity and same device

with the change in the dependent quantity and its associated physical object.

Step 3.

The *type* of a new dependence is chosen according to the following rules:

- If the amount of the dependent quantity changed and the amount of the independent quantity changed, then the dependence is a *function.*
- If the amount of the dependent quantity changed and the rate of the independent quantity changed, then the dependence is a *influence.*
- If the rate of the dependent quantity changed and the rate of the independent quantity changed, then the dependence is a *function.*

Functions are causal relations between the amounts or rates of two quantities. Influences are causal relations between the amount of one quantity and the rate of another.

Step 4.

The *sign* of a new dependence is chosen according to the following rules:

- If the directions of change of the two quantities are of the same sign, then the dependence is *positive* (direct).
- If the directions of change of the two quantities are of opposite sign, then the dependence is *negative* (inverse).

The direction of change of any quantity is found by locating the previous value and the current value in the quantity's quantity space. If it is not possible to determine the directions of change of the two quantities, then the dependence is left unsigned.

If the changes in the two quantities satisfy simultaneity and neither is attributable to an external action, then it is not possible to determine which quantity is independent and which is dependent. In this case, a correspondence is proposed rather than a dependence.

Proposing dependences is the first step in constructing new causal rules. The dependences represent the description of causality at the quantity level. The next section explains how the description of causality at the physical level is constructed.

### Preconditions and Effects

Causality can be described concisely at the level of quantities but there are two reasons why this is an inadequate representation. First, it is too abstract – a representation of causality must also describe objects and relations at the physical, real-world level to support planning. Second, a representation of causality must include not only explicit causes, but also the enabling conditions which must hold for the causality to be realized. A good example is the operation of a gun. Pulling the trigger is the direct, overt event which causes the bullet to be fired. However, unless the safety lock is off, the gun will not fire. The release of the safety lock is a precondition which must hold before the causal relation between pulling the trigger and the bullet firing can be realized.

Similarly, there may be indirect effects which result when a causal relation is realized. Some side effects of a fired gun are the recoil of the gun and the odor of the ignited gunpowder. It is particularly important to represent side effects which can only be realized indirectly, through a causal relation whose primary effect is something else.

The quantity level of causal rules provides a concise rendering of causal relations. Changes in independent quantities result in changes in dependent quantities through dependences. The physical level of causal rules allows an arbitrary number of preconditions and side effects to be represented for each causal relation. Preconditions and effects are either relations on physical objects or constraints on the ranges of values for quantities.

When a dependence is asserted between two quantities, this is only the first step in constructing a causal rule. The preconditions and effects which make up the physical level of the causal rule must also be identified.

The procedure for constructing the physical level of a causal rule is:

---

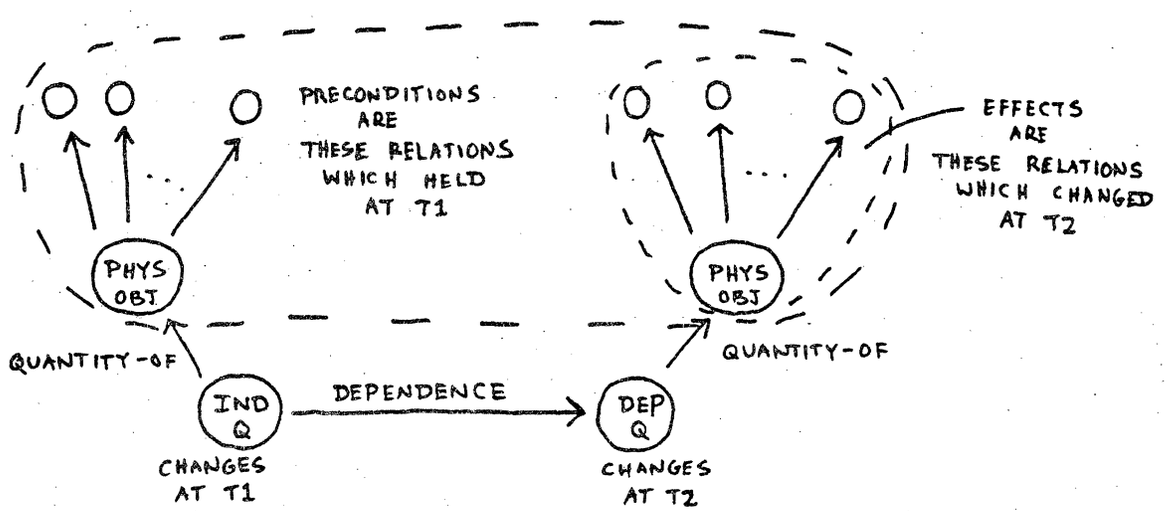
Collector:

Given the independent quantities, the dependent quantities, the associated physical objects, and the changes which are the primary cause and effect in a causal rule,

Collect the values of the quantities and relations on the physical objects which held at the time the primary cause occurred. These are the preconditions.

Collect the values of the dependent quantities and relations on the physical objects associated with the dependent quantities which changed at the time the primary effect occurred. These are the effects.

---




Preconditions and Effects

Normally, effects are assumed to be persistent. However, an effect which involves a continuous change is tracked to see if a limit value is reached. If so, this value is included in the causal rule as a long-term effect.

Earlier, it was stated that the first attempt to arrive at a causal explanation for a change in a quantity involves checking known dependences and determining if the independent quantities changed in the expected way. What JACK actually does is check known causal rules and determine if the independent quantities changed in the expected way and all preconditions were satisfied.

The procedure for identifying preconditions and effects can be either over-general or over-specific. Spurious preconditions and effects may be included. Relevant preconditions and effects may be missed. A later section which discusses how causal rules can be generalized over further experience addresses this problem.

## Imagination Orders the Explanation Hierarchy

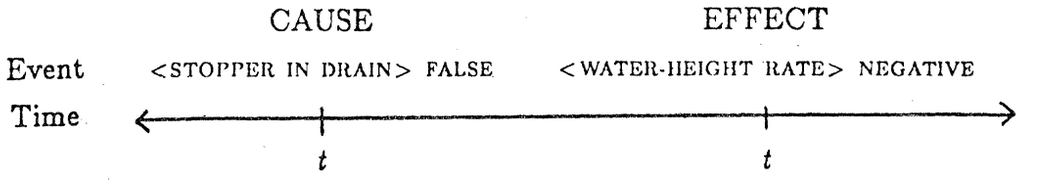
JACK first tries to explain changes in quantities by appealing to former experience encoded in existing causal rules. This kind of explanation does not involve an hypothesis.

If such an explanation is not forthcoming, JACK tries to propose a new dependence. This type of explanation assumes that the primary cause of the change in the dependent quantity is an observable change in an independent quantity. This is the nominal situation for the learning program. However, the last satisfied precondition – which may not be manifest in an observable change in a quantity – sometimes plays the role of primary cause. Because preconditions can become satisfied in different orders, different instantiations of a causal rule may display different primary causes.

Therefore, if an explanation involving a change in an independent quantity and a dependence is not forthcoming, JACK searches for a change in a relation on a physical object. This physical object and the time of the change also must satisfy the temporal and physical proximity requirement.

The explanation now is that the change establishes a precondition for a dependence whose dependent quantity is the quantity which changed. The unsatisfied precondition was preventing the causal relation from being realized. The change in the dependent quantity was latent, and the now-satisfied precondition was the important cause, not an unobservable change in an independent quantity. This type of causal explanation proposes a new dependence and a new quantity.

An example from the sink domain illustrates how this hypothesizing can work. While there is water in the basin, the stopper is removed from the drain and the water begins to fall.




Since the flow at the drain is not observable, the learning program cannot know about this influence directly. But JACK does know that the drain, which is touching the water (physical connectedness), underwent a change, namely the stopper was removed from it, as the water began to fall (simultaneity). This evidence is sufficient to propose a new quantity for the drain, and construct a new dependence and causal rule, one of whose preconditions states that the stopper must not be in the drain.

If there is no observable evidence about a cause for a change in a quantity, a final attempt to construct a causal explanation might be made by using analogy. If another situation matches well with the current one, it may support an hypothesis about an unobservable independent quantity, or even an unobservable physical object and change which establishes a precondition. The analogy would proceed by matching observable effects of the two situations, and then trying to map the causal explanation in the known situation onto the current situation. This explanation would have to satisfy the temporal and physical proximity constraint as well.

Although the use of analogy to construct causal models was not implemented, its use to extend causal models was, and is explored in Chapter 5.

JACK learns by making hypotheses to explain changes in quantities. JACK may make several attempts to construct causal explanations. Each type of explanation is more imaginative than the previous because each successive type of explanation proposes more to complete an adequate causal explanation.

In summary, these are the types of causal explanations JACK tries to construct (in order) when confronted with a change in a quantity:

The Explanation Hierarchy

- Identify a known dependence and causal rule in the existing causal model.
  What is proposed: nothing.
- Identify a change in a quantity which satisfies the temporal and physical proximity heuristics. By hypothesis, this quantity is the independent quantity.
  What is proposed: dependence, causal rule.

- Identify a change in a relation on a physical object which satisfies the temporal and physical proximity heuristics. By hypothesis, this relation is a precondition. What is proposed: independent quantity, dependence, causal rule.
- Identify a similar causal rule which explains a similar change in a similar quantity and which satisfies the temporal and physical proximity heuristics. By analogy, the causal explanation is transferable.
  What is proposed: physical object, relation, independent quantity, dependence, causal rule.

## Resolution and Boundaries

JACK's ability to construct causal explanations is limited by the level of *resolution* at which a physical system and its changes are presented and by the implicit *boundaries* [Kirsh 84] on the space of candidate causes and preconditions imposed by the causality-proposing heuristics.

JACK is provided with a structural description of a device at a single level of resolution – roughly what can be seen from the exterior of the device. JACK is not allowed to "open up" the device to know of additional components and connections. The temporal resolution is matched to the visible changes undergone by the parts of the device.

The heuristics of temporal adjacency, simultaneity, and physical connectedness allow JACK to make causal hypotheses about interactions which are visible. The same device heuristic essentially allows the learning program to hypothesize new connections between components.

The use of the heuristics is ordered so that the boundaries on the space of candidate causes and preconditions is expanded as JACK searches for a causal explanation for a change. These heuristics embody the notion of a closed system whose internal behavior is not impinged upon by events outside the system's boundaries.

## Generalizing Over Further Experience

Causal models are originally constructed on the basis of a single experience with a physical system. Any form of the causality-proposing heuristics can admit incorrect or incomplete hypotheses. More likely than not, causal models will need refinement. The next few sections discuss ways to recognize deficiencies in causal models and ways to generalize causal models to repair such deficiencies.

## Spurious Preconditions and Effects

Preconditions and effects at the physical level of a causal rule are collected by noting, respectively, what relations held, and what relations changed when the causation was manifest. Spurious effects are less likely because of the additional constraint, but this procedure does not guarantee the exclusion of either spurious preconditions or effects. However, an irrelevant precondition will never prevent a causal relation from being realized, and an irrelevant effect will not necessarily occur. Therefore, any precondition or effect which is respectively, unsatisfied or unrealized when a causal rule is otherwise intact, can be dropped. This pruning is a kind of generalizing from negative examples.

An example from the sink domain illustrates how a spurious precondition can be recognized and dropped. When JACK first attributes the recession of the water in the sink to the removal of the stopper from the drain, there is a bar of soap floating in the water. JACK includes but later drops the presence of the soap as a precondition when the water flows out again *sans* soap.

## Making Better Hypotheses

A causal rule may fail to explain apparently similar events because the causal relation it describes may subsume a chain of causality, or may itself be part of a larger causal structure. Such an incomplete causal description may be missing relevant dependences, preconditions, and effects.

When the effects listed in a causal rule do not obtain despite all known preconditions being satisfied, this is evidence that the causal model is incorrect or at least incomplete. JACK might resume the search for an hypothesis where it originally terminated and try to generate a causal explanation which covers both the new and previous events. A better idea is to try and determine why the causal model failed by comparing the situation where it failed to the situations where it did not fail. Any differences can support new hypotheses which can then be tested by the causality-proposing heuristics. Differences reveal causes in rehypothesizing just as changes reveal causes in initial hypothesizing. Rehypothesizing is a kind of generalizing from both positive and negative examples. On the other hand, JACK's initial hypotheses are based on explanations of a single positive example.

The procedure for rehypothesizing is:

---

Rehypothesizer:

Given two situations, one where a causal rule provides a causal explanation and one where it does not,

Compare the two situations.

Until an adequate causal explanation which covers both situations has been constructed,

For each difference,

Try to construct a causal explanation.

---

An example from the toaster domain illustrates how a better causal model can result when JACK is forced to rehypothesize because the current model fails.

Part of JACK's initial model of the toaster includes an influence between the position of the lever and the temperature of the coils. This model works fine until the toaster's plug is pulled from the outlet. JACK compares (see Appendix V for a description of the matcher) the state of the toaster at the time of the initial hypothesis and at the time of the failure and discovers the difference involving the plug. The plug now becomes a candidate for affecting the coils. JACK asserts a new dependence between a new quantity associated with the plug (which we might call current) and the temperature of the coils. One of the new preconditions is that the plug must be in the outlet.

To see how differences play the same role in rehypothesizing as changes do in initial hypothesizing, imagine that JACK's first experience with the toaster had involved the lever being already down and the plug being put in the outlet last. In this case, JACK would have made the better hypothesis first.

## Dependences in Devices

There are additional situations in which deficiencies in causal models can be recognized, if one assumes that dependences are always *functions* (in the mathematical sense) from the independent quantity's quantity space to the dependent quantity's quantity space. One-to-many relations between parameters of a device make little sense because they imply random behavior. This is an oversimplified, but useful teleological assumption about the nature of dependences in *designed* physical systems. If a one-to-many relation is ever observed, this is evidence that the causal model is incomplete and a better hypothesis is needed.

This assumption is buttressed by a felicity condition [VanLehn 83] which requires dependences to be *monotonic* functions. This condition guarantees one-to-one correspondences between quantity spaces across dependences, which makes some qualitative reasoning easier. For example, increasing an independent quantity must increase (or decrease) the corresponding dependent quantity. Many-to-one dependences in physical systems also can be useful, e.g., to transform a wide range of inputs into a finite set of stable states – but they were avoided in this work. Therefore, further experience with dependences should result in nothing more than the possible parallel expansions of the appropriate quantity spaces.

An example from the toaster domain illustrates how the teleological assumption about dependences can enable the learning program to recognize incomplete causal models.

The toaster produces toast of a certain darkness the first time through. JACK's initial model includes a dependence between the temperature of the coils and the darkness of the toast but this dependence cannot explain why a second piece of toast comes out lighter. On each occasion, the plug was in, the lever was down, and the coils heated up. JACK compares the two situations to find an explanation for the difference between the two pieces of toast. JACK finds that the thermostat dial was set differently in the two situations and asserts a function between the setting of the thermostat dial and the darkness of the toast. The learning program does not actually discern the thermostat mechanism or the heat exchange process which controls the darkness of the toast. However, the abstract causal relation JACK does propose is accurate to the resolution available, and useful.

This kind of analysis does not apply to rates of quantities because of another resolution limitation. Values in quantity spaces for rates are limited to negative, zero, and positive. It is not possible to determine if a quantity is changing faster this time than at a previous time.

## The Learning Session in the Sink Domain

This section contains an annotated transcript of the learning session in the sink domain. The sequence of events (the actual input to the learning program) appears in *italic* type. (The sequence of events also appears in Appendix I). The causal explanations JACK constructs to understand these events are given in **bold** type. Comments appear in normal type.

*Already, the tap, the faucet, and the basin are part of the sink.*
*The drain, the safety, and the stopper are part of the basin.*

These two sentences create a hierarchical structural description of the sink.

*The stopper is in the drain.*
*The faucet's position is closed.*
*The light-switch's setting is off.*
*The window's height is down.*

The adverb *already* signifies that the relations so far described have held since some indeterminate time in the past. The learning session proper begins here.

*Initially, the faucet's position is open.*
*The light-switch's setting is on.*
**Thinking at t=1.**

The adverb *initially* starts the internal clock at 1.

The learning program is told *a priori* which changes are due to external actions and does not try to explain them. These include turning the faucet on and off, and turning the light on and off.

*Next, a water-column appears between the tap and the basin.*
*The water-column's width is steady.*
**Thinking at t=2.**

The adverb *next* ticks the clock once.

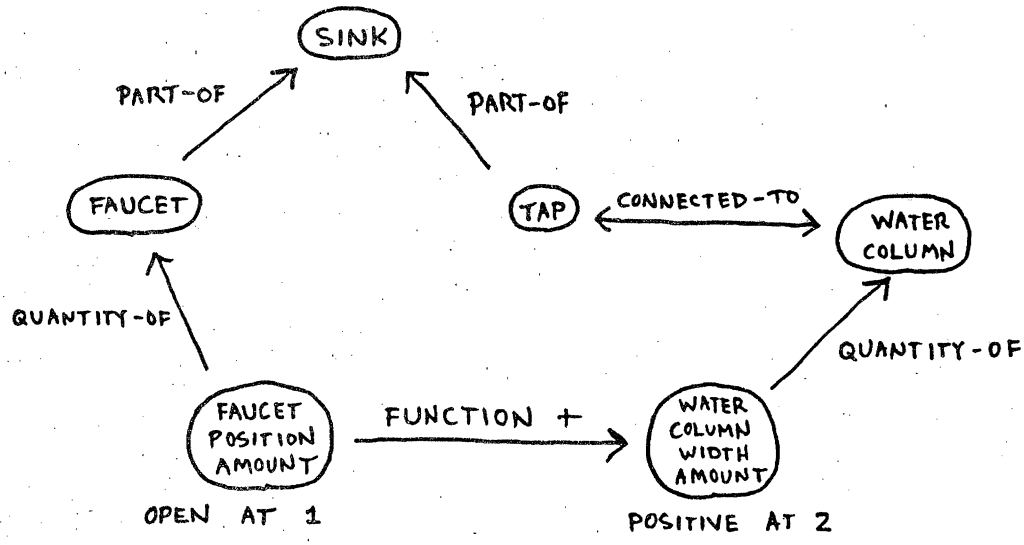
JACK observes that both the faucet's position and the light switch's setting changed at t=1. Either of these changes could explain the change in the water column's width at t=2 because both satisfy the temporal adjacency heuristic. However, the light-switch fails to satisfy either the physical connectedness heuristic or the same device heuristic with the water column. On the other hand, the faucet does satisfy the physical proximity requirement. The water column is connected to the tap and the tap and faucet are part of the sink. The same device heuristic enables JACK to hypothesize about the pipe – which is not visible – connecting the faucet and the tap.

Asserting a function between
the position of the faucet and
the width of the water-column.

This dependence satisfies
the temporal adjacency and same device heuristics.

The width of the water-column increased because
the position of the faucet increased.

When JACK asserts a dependence to construct a causal explanation, he gives the justification for the new dependence (which causality-proposing heuristics were satisfied), and the explanation supported by the new dependence.




*Next, water appears in the basin.*
*The water-column is connected to the water.*
*The water is connected to the drain.*
*The water's height is increasing.*

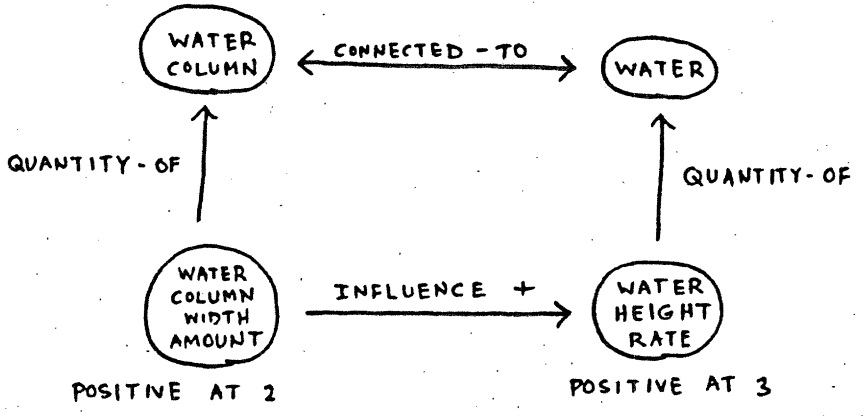
Thinking at t=3.

Asserting an influence between
the width of the water-column
and the height of the water.

This dependence satisfies
temporal adjacency and physical connectedness.

The height of the water is increasing because
the width of the water-column is positive.

This explanation is straightforward. Notice that the learning program finds a non-intuitive two-link chain of causality between the faucet, the water column, and the water in the basin. JACK treats the water column and the water as separate objects. He does not know that the water column and the water are the same "stuff". This curiosity notwithstanding, the causal explanation that the learning program finds is useful.




*Later, the water is connected to the safety.*
*The water's height is equal to the safety's height.*
*The water's height is steady.*
Thinking at t=4.
Thinking at t=5.

The adverb *later* ticks the clock twice. This creates an intermediate interval during which no additional changes occur. The water rises to the level of the safety drain during this interval.

**The height of the water has reached a stable value.**

When a changing quantity reaches a stable value, this event is included as a long-term effect in the causal rule which includes the appropriate dependence. A demon makes these addenda.

**I am proposing a new quantity for the safety.**
**Please give a name for this quantity.**
> *flow*

**Asserting an influence between**
**the flow of the safety and**
**the height of the water.**

**This dependence satisfies**
**simultaneity and physical connectedness.**

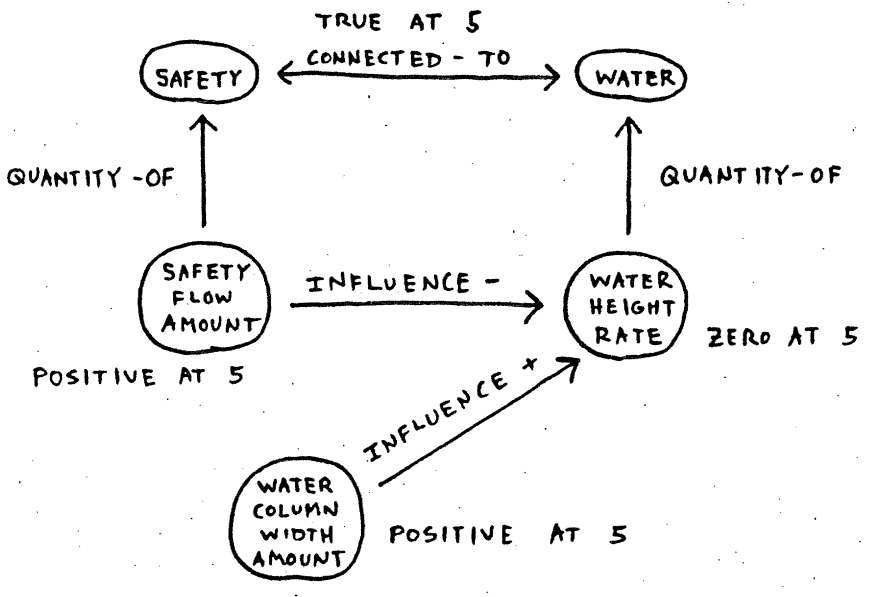
**The height of the water is not changing because**
**there is an equilibrium between**
**the flow of the safety and**
**the width of the water-column.**

The water has now reached the safety drain and has stopped rising. JACK knows that a steady quantity has two possible explanations. Either there are no influences or there are balancing influences. Since the faucet is still on, the only possible explanation is that an equilibrium state has been achieved. Unfortunately, there is no other independent quantity in sight, so the learning program must make a more imaginative conjecture. JACK tries to find an object which satisfies the physical proximity requirement with the water and which just underwent a physical change of some kind (to satisfy the temporal proximity requirement). The assumption is that the change is a newly-satisfied precondition which is now enabling a latent causal relation.

The object and change that the learning program finds are the safety drain and the fact that the water is now touching the safety drain. This is in fact an enabling

condition for water to flow out of the safety drain. There is an equilibrium state between water flowing in at the tap and flowing out at the safety drain.

Once the safety drain has been identified as the causal culprit, assigning a quantity to it and forming a new dependence to complete the causal explanation is straightforward.




*Later, the faucet's position is closed.*

Thinking at t=6.
Thinking at t=7.

*Next, the water-column disappears.*
*The water's height is decreasing.*

Thinking at t=8.

The width of the water-column decreased because the position of the faucet decreased.

The height of the water is decreasing because the flow of the safety is positive.

A previously learned dependence explains why the water column disappears when the faucet is turned off. With the faucet off, for just a moment water continues to flow out of the safety drain until enough has flowed out to actually break contact with the safety drain. For that short moment, the positive half of the equilibrium state has been broken because the water column has disappeared, but the negative half of the equilibrium state is still intact and some water flows out. JACK has no trouble understanding this situation.

*Next, the water is not connected to the safety.*
*The water's height is steady.*
Thinking at t=9.

**The height of the water has reached a stable value.**

**The height of the water is not changing because**
**there is nothing affecting it.**

The height of the water is now steady because enough water has flowed out to break contact with the safety drain. From JACK's viewpoint, a precondition on the remaining negative half of the equilibrium state has become unsatisfied, and there are now no influences on the height of the water.

*Later, soap is in the water.*

Thinking at t=10.
Thinking at t=11.

*Later, the stopper is not in the drain.*
*The window's height is up.*
*The water's height is decreasing.*

Thinking at t=12.
Thinking at t=13.

I am proposing a new quantity for the drain.
Please give a name for this quantity.
> *flow*

Asserting an influence between

the flow of the drain and

the height of the water.

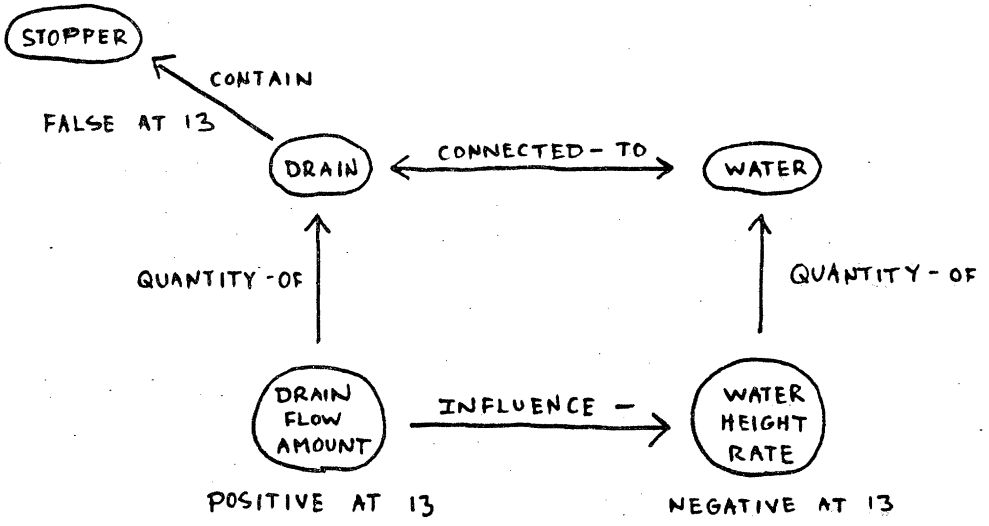
This dependence satisfies

simultaneity and physical connectedness.

The height of the water is decreasing because

the flow of the drain is positive.

The height of the water begins to fall when the stopper is pulled from the drain and the window is opened. The change in the window's height is eliminated as a candidate explanation because the window does not satisfy either the physical connectedness or the same device heuristic with the water. Since, the learning program cannot perceive flow at the drain directly, it is unable to construct a causal explanation in terms of a change in an independent quantity. Instead, JACK looks for a change in a physical object which satisfies temporal and physical proximity. The only such object and change JACK finds is the drain and the fact that the stopper has just been pulled from the drain. Assuming that this change is a just-satisfied precondition for a newly discovered causal relation, the learning program assigns a new quantity to the drain, asserts a new influence, and constructs a new causal rule. This rule includes another precondition which states that soap must be in the water because this was true when the rule was formed. This spurious precondition will be pruned by later experience.




*Later, the water disappears.*

**Thinking at t=14.**

**Thinking at t=15.**

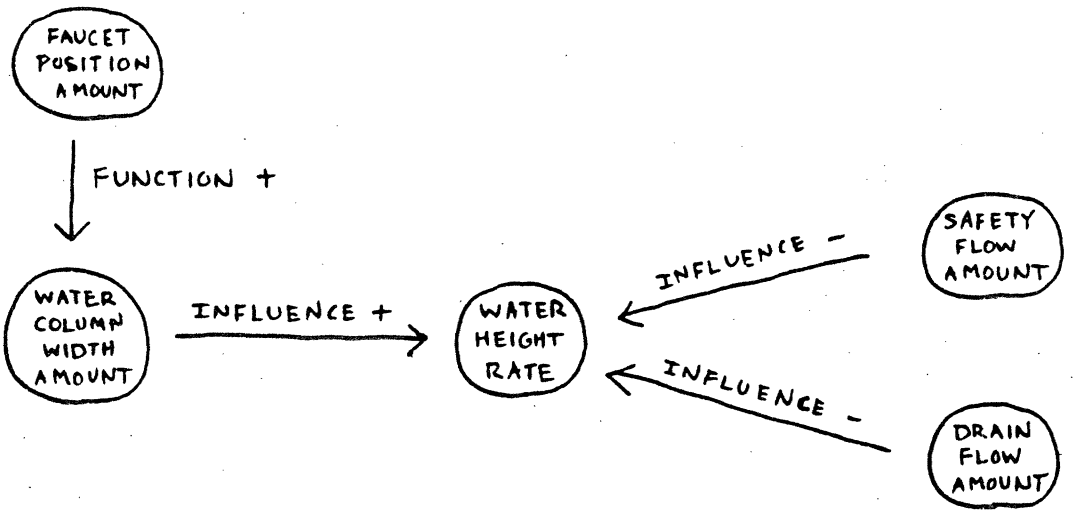
**The height of the water has reached a stable value.**

**The height of the water is not changing because there is nothing affecting it.**

When the water finally disappears, this event is included as a long-term effect of the causal rule just constructed which describes how flow at the drain causes the water's height to fall. Such long-term effects will happen as long as the preconditions of the pertinent causal rule hold persistently. In this case, the stopper must remain out of the drain.

*Finally, nothing is changing.*

This completes the initial learning session in the sink domain. The figure below shows the quantities and dependences JACK uses to causally explain the observed behavior of the sink.




These are the quantity spaces of the quantities of the sink.

Faucet Position (CLOSED -> OPEN)

Water Column Width (ZERO -> POSITIVE)

Water Height (ZERO -> BELOW-SAFETY -> SAFETY)

Safety Flow (ZERO -> POSITIVE)

Drain Flow (ZERO -> POSITIVE)

The set of causal rules which make up JACK's full causal model of the sink (including preconditions and effects) appear in Appendix II. This model has been refined over further experience. The transcript of these experiences appear in the next chapter.

## The Learning Session in the Toaster Domain

This section contains an annotated transcript of the learning session in the toaster domain. As in the previous section, the sequence of events which describes changes in the toaster over time appear in *italic* type. (This sequence of events also appears in Appendix III). The causal explanations JACK constructs for those changes appear in bold type. Comments appear in normal type.

*Already, the lever, the plug, the dial, and the slot are part of the toaster.*

*The coils are part of the slot.*

These sentences form a hierarchical structural description of the toaster.

*The coils' temperature is cold.*
*The lever's position is up.*
*The dial's setting is D.*
*The plug is in the outlet.*
*The bread is in the slot.*
*The bread's shade is white.*
*The faucet's position is closed.*
*The light-switch's setting is on.*
*The window's height is up.*

These sentences describe the state of the toaster in terms of physical relations and values of quantities.

*Initially, the lever's position is down.*
*The faucet's position is open.*
*The bread is not visible.*

**Thinking at t=1.**

*Next, the coils' temperature is increasing.*

**Thinking at t=2.**

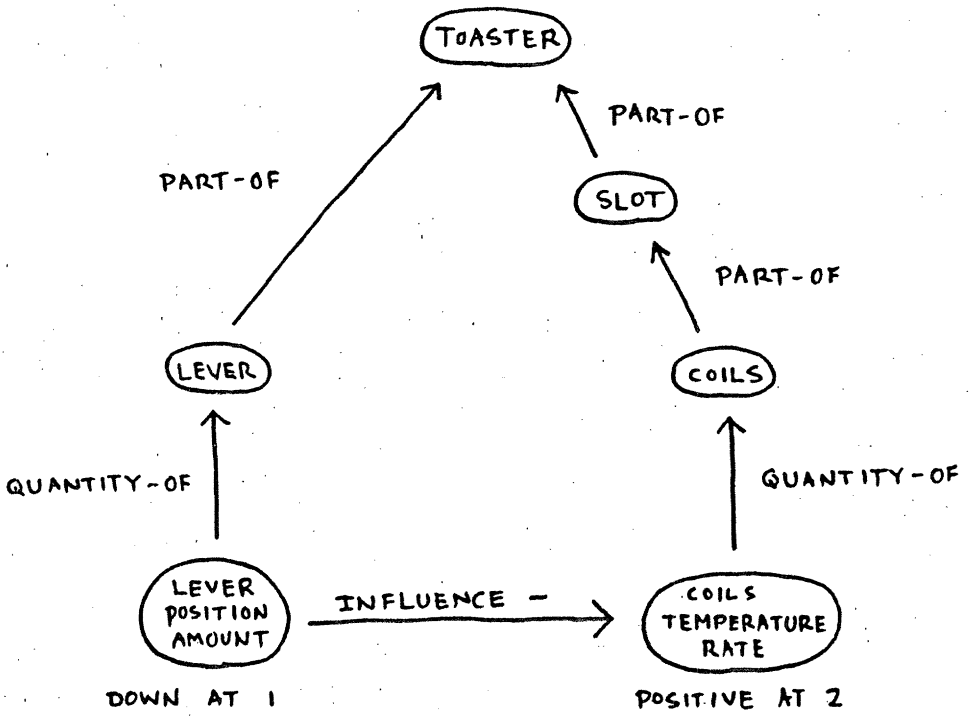
**Asserting an influence between**
**the position of the lever and**
**the temperature of the coils.**

**This dependence satisfies**
**temporal adjacency and same device.**

**The temperature of the coils is increasing because**
**the position of the lever is negative.**

Both the change in the position of the lever and the change in the position of the faucet satisfy temporal adjacency with the change in the rate of the temperature

of the coils. However, the faucet does not satisfy any physical proximity test with the coils.




The sink will continue to exhibit changes but for clarity's sake, its behavior will now be omitted.

*Later, the lever's position is up.*
*The coils' temperature is hot.*
*The coils' temperature is steady.*
*The bread is visible.*
*The bread's shade is dark.*

Thinking at t=3.
Thinking at t=4.

The temperature of the coils has reached a stable value.

This value "hot" is included as a long-term effect in the causal rule just constructed which describes how the temperature of the coils increases.

There are two changes which JACK will try to explain – the lever has popped up and the bread has turned to toast.

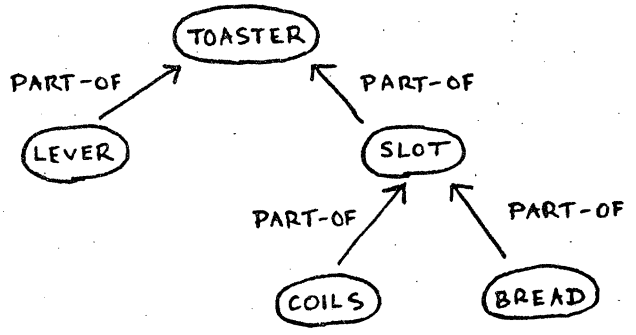
**Asserting a function between**
**the temperature of the coils and**
**the shade of the bread.**

**This dependence satisfies**
**simultaneity and same device.**

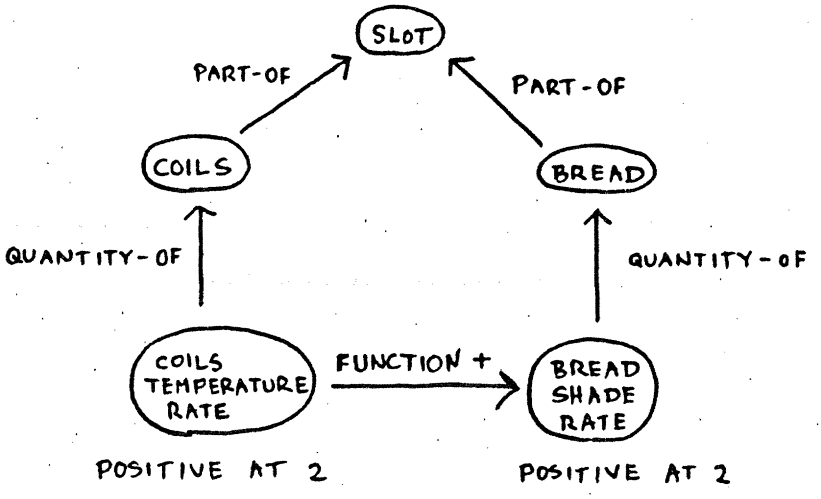
**The shade of the bread increased because**
**the temperature of the coils increased.**

JACK first makes the inference that the bread's darkness was changing continuously during the time it was not visible. JACK asserts "**The bread's shade is increasing.**" at t=2 and "**The bread's shade is steady.**" at t=4. Then JACK looks back in the sequence of events to try and construct a causal explanation.

Both the change in the position of the lever (by temporal adjacency) and the change in the temperature of the coils (by simultaneity) are candidate causes. Also, both the lever and the coils satisfy the same device heuristic with the bread. However, the bread and coils are closer in the PART-OF hierarchy than the bread and the lever. (The IN relation implies the PART-OF relation. This inference is handled by a demon).



Thus JACK chooses the coils as the causal culprit. Notice that the same device heuristic enables JACK to handle an instance of "action at a distance", in this case heat flow.




**There are more than one possible explanations for the change in the position of the lever.**

**A function between the temperature of the coils and the position of the lever.**

**This dependence satisfies simultaneity and same device.**

**The position of the lever decreased because the temperature of the coils increased.**

**or**

A function between
the shade of the bread and
the position of the lever.

This dependence satisfies
simultaneity and same device.

The position of the lever decreased because
the shade of the bread increased.

The figure above shows why JACK cannot distinguish these hypotheses. JACK now does what reasonable learning programs do in such situations – he waits for less ambiguous experience. Later, when JACK sees that the coils are always hot when the lever pops up but the pieces of toast can be of any darkness, he will be able to make a justified choice between these competing hypotheses.

*Next, the coils' temperature is decreasing.*
*The window's height is down.*

Thinking at t=5.

The temperature of the coils is decreasing because
the position of the lever is positive.

JACK uses a known dependence to construct a causal explanation. The change in the window's height is ignored because it fails the physical proximity requirement.

*Later, the coils' temperature is cold.*

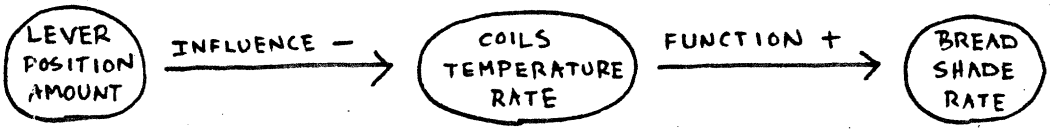
Thinking at t=6.
Thinking at t=7.

The temperature of the coils has reached a stable value.

Another long-term effect.

*Finally, nothing is changing.*

This completes the initial learning session in the toaster domain. The figure below shows the quantities and dependences JACK uses to causally explain the observed behavior of the toaster.




---

These are the quantity spaces of the quantities of the toaster.

Lever Position (DOWN -> UP)

Coils Temperature (COLD -> HOT)

Bread Shade (WHITE -> DARK)

JACK will have cause (no pun intended) to refine this initial model of the toaster when the plug is pulled from the outlet and when toast of varying darkness is produced. The transcript of these further experiences appears in the next chapter. The final causal model of the toaster appears in Appendix IV.

This chapter has shown how the learning program goes from a sequence of events describing changes in a physical system to an explicit representation of the causality which underlies the behavior of the physical system.

The goal of a learning system is not just to create new knowledge structures, but to create new knowledge structures which can support reasoning which was impossible before the learning took place. The next chapter shows that this goal has been achieved.

# CHAPTER 4
# REASONING WITH THE CAUSAL MODEL: EXPLANATION, PREDICTION, AND PLANNING

The definition of learning that guides this research is the following: Learning is the creation of useful knowledge structures to facilitate reasoning that was not possible before the learning took place.

The learning program described in this thesis constructs causal models of physical systems. The models consist of a set of causal rules, each of which describes some aspect of a physical system's behavior in terms of causal relations.

There are three kinds of causal reasoning that a causal model should support:

- explaining phenomena
- predicting phenomena
- constructing plans to generate phenomena

This section shows how the learned causal model supports these kinds of reasoning and also how these kinds of reasoning provide feedback about deficiencies in the model. When predictions are inaccurate or plans do not work, this is evidence that the causal model is incomplete and rehypothesizing is in order. Thus learning supports reasoning which drives further learning.

## Causal Rules are If-Then Rules

A causal rule consists of a set of dependences between quantities at the quantity level, and a set of preconditions and effects at the physical level. A causal rule can be restated as follows:

Quantity Level

IF [the independent quantities change in the manner prescribed by the dependences]

THEN [the dependent quantities will change in the manner prescribed by the dependences]

Physical Level

IF [the preconditions are satisfied]

THEN [the effects will occur]

The known results about rule-based inference apply to causal rules and have been exploited implicitly by the learning program all along.

Explanation, prediction, and planning done at the physical level deals with real-world objects, relations, and events. Reasoning at the quantity level can be incomplete because the full set of preconditions is omitted. However, the abstractions available at the quantity level which support hypothesizing also support qualitative reasoning which can in some cases, go beyond what is modelled explicitly in the causal rules.

## Explanation, Prediction and Planning is Done by Rule-Based Inference

Explanation of phenomena in the physical system is done by backward chaining on the set of causal rules that makes up the causal model of the physical system. This kind of explanation uses the existing causal model as is. It is different from the causal explanations which support hypotheses to create and modify the causal model.

The following is the procedure for doing explanation:

---

Explainer:

Given an event,

Find a causal rule which lists that event as an effect.

If there is no such causal rule, stop.

The preconditions of that rule and the time they hold are the explanation.

For each precondition,

Explain that precondition.

---

The following is an explanation in the sink domain for the appearance of water in the basin.

(T) <FAUCET PART-OF SINK> TRUE

(T) <FAUCET-POSITION AMOUNT> OPEN

↑

(T+1) <WATER-COLUMN CONNECTED-TO TAP> TRUE

(T+1) <WATER-COLUMN CONNECTED-TO BASIN> TRUE

(T+1) <WATER-COLUMN-WIDTH AMOUNT> POSITIVE

↑

(T+2) <WATER IN BASIN> TRUE

## An Explanation

Prediction of phenomena in the physical system is done by forward chaining on the set of causal rules.

The following is the procedure for doing prediction:

---

Predictor:

Given a state of the physical system,

Find all causal rules whose preconditions are completely satisfied.

If there are no such causal rules, stop.

The effects of these causal rules and the time(s) they hold are the prediction.

Update the state of physical system according to this set of effects.

Predict.

---

The following is a prediction from the sink domain about what will happen when water is in the basin and the stopper is removed from the drain.

(T) <WATER IN BASIN> TRUE

(T) <DRAIN CONNECTED-TO WATER> TRUE

(T) <DRAIN PART-OF BASIN> TRUE

(T) <STOPPER IN DRAIN> FALSE



(T) <WATER-HEIGHT RATE> NEGATIVE

(T+2) <WATER IN BASIN> FALSE

(T+2) <DRAIN CONNECTED-TO WATER> FALSE

(T+2) <WATER-HEIGHT AMOUNT> ZERO

## A Prediction

Planning also is done by backward chaining on causal rules. However, instead of explaining an event, the task is to achieve a goal. A plan must specify how to *make* something happen. It must describe not only the pertinent causal relations, but also the *actions* which must be taken in order to achieve a goal.

Because the planner must know about actions, it is told which states of the physical system are externally settable.

The procedure for doing planning is given below:

---

Achiever:

Given a goal to achieve,

Find a causal rule which lists that goal as an effect.

If there is no such causal rule, fail.

The preconditions of that causal rule and the time they hold is the plan.

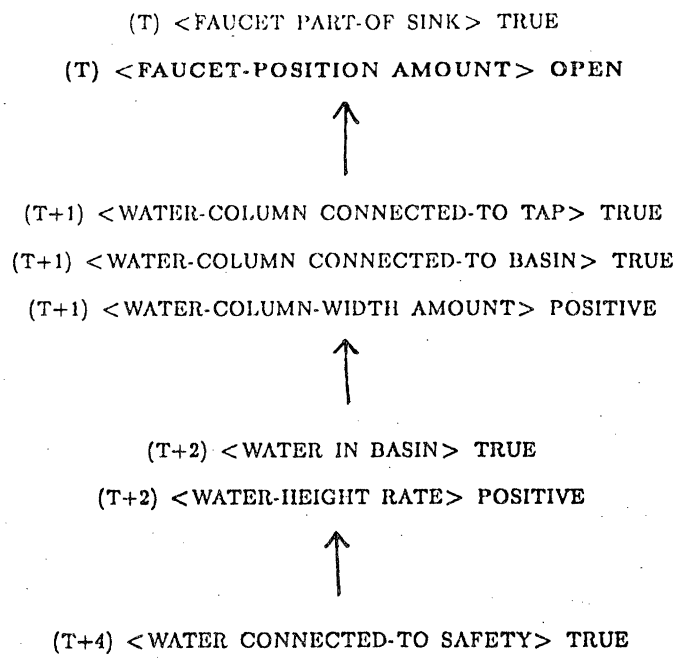
For each precondition which is neither externally settable, nor already holds,

Find a plan for achieving that precondition.

---

The following is a plan in the sink domain to make the water reach the safety

drain.




## A Plan

The planner distinguishes the actions which are at the roots of the causal chains in a plan.

Explanations, predictions, and plans are causal chains of events which are relocatable in time. Each node in one of these structures describes states that hold simultaneously. Links between nodes are justified by causal rules which describe a causal relation between a set of preconditions and a set of effects.

It may be possible to find more than one explanation for the same event, or more than one plan for the same goal, if the physical system is overdetermined. A causal model of an overdetermined system would list the same relation as an effect in more than one causal rule. However, the implemented explainer and planner do not search for multiple solutions; they stop at the first one.

Similarly, the same relation may appear as a precondition in more than one rule. Unlike the explainer and planner, the predictor finds all possible changes which can proceed from a given state of the physical system. A branching prediction violates the teleological assumption about devices not being designed to produce one-to-many behavior. This would be evidence that the causal model needs refinement.

In addition, conflicts can arise if the structure of an explanation, prediction, or plan is non-linear. A conflict would be, for example, one branch of a plan undoing what was achieved in another branch. Such conflicts might be evidence for spurious preconditions or effects or other deficiencies in the causal model. Fortunately or unfortunately, these situations did not arise in this research. Dealing with contradictions and resolving conflicts in planning has been discussed in other research, such as [Sussman 75, Sacerdoti 77, Doyle 78].

## The Planner has Two Modes: Achieve & Prevent

The planner has two modes, and in this respect it differs from the explainer and predictor, and also from many other planners. In one mode, the planner generates plans to *achieve* a desired event. This mode was discussed in the previous section. In the other mode, the planner generates plans to *prevent* an unwanted event from occurring.

Preventing a goal is harder than achieving a goal because while *any* way of making something happen is adequate, if the task is to stop something from happening, *all* the ways it can happen have to be inhibited. The preventer, given a goal to be prevented, must find all the causal rules which list that goal as an effect, and for each of these rules, it must prevent the effect from occurring. Curiously enough, breaking individual causal rules is easier than satisfying them. The achiever has to satisfy *all* of the preconditions (a conjunction) of a causal rule to ensure that its effects will be realized, while the preventer only has to deny *any* one of the preconditions (a negated conjunction is a disjunction) of a causal rule to ensure that the effects of that rule will not be realized.

Another way to prevent something from happening is to generate a normal plan to achieve a mutually exclusive state, e.g. the same relation with a different truth-value or the same quantity with a different value. The current planner does not try to do prevention planning this way. However, both modes of the planner do interact to produce complex plans which include both the achievement of some states and the denial of others.

The procedure for preventing a goal is:

---

Preventer:

Given a goal to prevent,

Find all causal rules which list that goal as an effect.

If there are no such causal rules, fail.

For each causal rule,

Find any precondition which is either externally settable, or does not hold.

The denied preconditions of these causal rules (one from each) and the time they are denied is the plan.

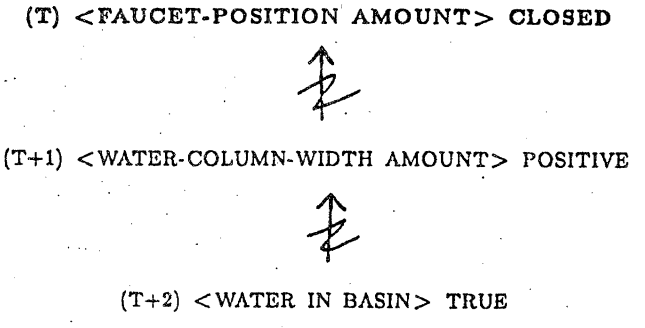
If there are no such preconditions, then

For any precondition

Find a plan for preventing that precondition.

---

The following is a plan in the sink domain to prevent water from collecting in the basin.




A Prevention Plan

Conjunctions appear at the level of preconditions of individual causal rules in plans to achieve a goal. Consequently, it is at this level that the achiever is sensitive to incompleteness of the causal model. Conjunctions appear at the level of causal rules in plans to prevent a goal. Similarly, it is at this level that the preventer is sensitive to incompleteness of the causal model.

Note that the possible incompleteness of the causal model at the level of the set of causal rules does not affect the achieve mode of the planner. Only if the achiever could not find a plan at all could a more complete model possibly make a difference.

Thus a plan generated by the achiever is guaranteed to work only if the causal rules that make up the plan are complete; a plan generated by the preventer is guaranteed to work only if the causal model itself (the set of causal rules) is complete.

Both modes of the planner are useful, not only because they do the right thing in many cases, but precisely because they can create situations in which deficiencies in the behavioral model can become explicit. If a plan to achieve something fails, this suggests a relevant precondition was missed (one involving objects and relations out of sight, for instance) during the construction of the appropriate causal rule – a precondition which is now unsatisfied. Similarly, when a plan to prevent something fails, this suggests that unknown causal relations exist. Feedback generated by failed plans indicate the need for better hypotheses to refine the existing causal model.

## Qualitative Reasoning with Quantities

Plans are always constructed at the physical level, but the quantity level can aid planning by supporting reasoning which can go beyond what appears explicitly in the causal model.

Consider the planning problem of achieving a goal which involves a state that has never been observed before. Certainly, there can be no causal rule which lists this state as an effect. However, if this state corresponds to a conjectured value for a quantity which is greater, or less, than a value in the quantity's quantity space previously thought to be a limit, then the planner can look for a causal rule which shows how the quantity can be made to change in the desired direction. Given the felicity condition that correspondences between quantity spaces across dependences are monotonic, such a plan should work as long as the preconditions are maintained while the quantity is changing. The plan may fail because the quantity achieves a stable or limit value before reaching the desired value or because the causal model is incomplete, but at least there is something to try.

This kind of reasoning allows the planner to generate a plan to produce toast of an unprecedented lighter shade by extrapolating the dependence between the setting of the thermostat dial and the shade of the resulting toast.

Recall the table which lists knowledge about second-order changes in quantities, used in hypothesizing causal relations.

| Current state | Add + | Add − | Del + | Del − |
|---|---|---|---|---|
| Constant C (0,[ ]) | I | D | x | x |
| Increase I (+,[+]) | I | E | C | x |
| Decrease D (−,[−]) | E | D | x | C |
| Equilibrium E (0,[+,−]) | E | E | D | I |

For example, a state of equilibrium can be changed to a state of increase by deleting the negative half of the equilibrium. A state of decrease can be changed to a stable state by adding a positive contribution and achieving equilibrium or by deleting the negative contribution.

The felicity condition which prohibits tradeoff situations simplifies this table considerably. Without this restriction, there would be many ambiguous entries. For example, adding a negative influence to a positive one could result in positive tradeoff, negative tradeoff, or equilibrium.

This knowledge can be used by the planner to generate plans to achieve a state for quantity which is different from its current state. The achieve mode can add contributions and the prevent mode can delete them.

An example from the sink domain illustrates how reasoning with this knowledge can facilitate planning. The planner is given the task of making the water rise above the safety drain. Because this event has never occurred, there is no causal rule which lists it as an effect. However, the planner does know that the height of the safety drain is an equilibrium value for the water's height. It reasons that the water can be made to rise above the safety drain by keeping the positive half of the equilibrium state intact and inhibiting the negative half. Unfortunately, even this reasoning is not enough because the planner can find no way to prevent flow at the safety drain. The problem is ultimately solved by using an analogy, which is the subject of the next chapter.

## Reasoning with the Sink Model

This section contains an annotated transcript of causal reasoning problems in the sink domain presented to and solved by the explainer, the predictor, and the two complementary halves of the planner – the achiever and the preventer. The ability of these reasoning programs to solve these problems serves to demonstrate that learning has taken place. When predictions prove wrong or plans do not work, this is evidence that the causal model is incomplete. In these cases, JACK is called back in to try and improve the model. The same conventions will be followed here

as in the learning session transcript. The statement of the problems to the various reasoning programs and the sequence of events appear in *italic* type. The responses of the explainer, predictor, and planner appear in bold type. Comments appear in normal type.

This reasoning session takes place immediately after the learning session. The faucet is off, the stopper is out of the drain, and there is no water in the basin.

*Later, the stopper is in the drain.*

*Predict what will happen.*

**Nothing will change.**

The predictor correctly determines that the sink is in a stable state. No causal rules have their set of preconditions completely satisfied.

*Make the water-column appear.*

**To achieve:**
<WATER-COLUMN-WIDTH AMOUNT> POSITIVE
**These must hold:**
<FAUCET PART-OF SINK> TRUE
<FAUCET-POSITION RATE> ZERO
<WATER-COLUMN-WIDTH AMOUNT> ZERO
<WATER-COLUMN-WIDTH RATE> ZERO
**These should be set:**
<FAUCET-POSITION AMOUNT> OPEN

*Stop the water-column from appearing.*

**To prevent:**
<WATER-COLUMN-WIDTH AMOUNT> POSITIVE
**These should be set:**
<FAUCET-POSITION AMOUNT> CLOSED

The same goal is presented to the achiever and the preventer. The achiever finds a causal rule which lists the desired goal as an effect. Satisfying all the preconditions of this rule constitutes a plan for making the water column appear. The planner always identifies the *actions* among the set of conditions that must be satisfied to carry out a plan.

The preventer finds the same causal rule. Since it is the only rule which lists the appearance of the water column as an effect, it is the only rule the preventer has to break. The preventer only has to inhibit a single precondition of a rule to break it.

*Later, the faucet's position is open.*

*Predict what will happen.*

Next, these will hold:
<WATER-COLUMN CONNECTED-TO BASIN> TRUE
<WATER-COLUMN CONNECTED-TO TAP> TRUE
<WATER-COLUMN-WIDTH AMOUNT> POSITIVE
Next, these will hold:
<WATER CONNECTED-TO DRAIN> TRUE
<WATER CONNECTED-TO WATER-COLUMN> TRUE
<WATER-COLUMN CONNECTED-TO WATER> TRUE
<WATER IN BASIN> TRUE
<WATER-HEIGHT RATE> PLUS
Later, these will hold:
<WATER CONNECTED-TO SAFETY> TRUE
<WATER-HEIGHT AMOUNT> SAFETY
<WATER-HEIGHT RATE> ZERO
Next, nothing will change.

After the faucet is turned on, the predictor is invoked. It correctly predicts that, as long as no further changes occur, the water column will appear and the water will rise until it reaches the level of the safety drain.

*Next, a water-column appears between the tap and the basin.*
*The water-column's width is steady.*

*Next, water appears in the basin.*
*The water-column is connected to the water.*
*The water is connected to the drain.*
*The water's height is increasing.*

So far, all of the predictor's prophecies about what would happen when the faucet was turned on have been fulfilled. The water will not quite reach the safety drain, but only because the faucet will be turned off before it gets there.

*Explain why there is water in the basin.*

<WATER IN BASIN> TRUE
**holds because:**
**First, these held:**
<FAUCET PART-OF SINK> TRUE
<FAUCET-POSITION AMOUNT> OPEN
<FAUCET-POSITION RATE> ZERO
<WATER-COLUMN-WIDTH AMOUNT> ZERO
<WATER-COLUMN-WIDTH RATE> ZERO
**Then, these held:**
<WATER-COLUMN CONNECTED-TO TAP> TRUE
<WATER-COLUMN CONNECTED-TO BASIN> TRUE
<WATER-COLUMN-WIDTH AMOUNT> POSITIVE

The explainer finds the correct causal explanation for the appearance of water in the basin. It back-chained through the same two rules that the predictor just forward-chained through. The explainer does not treat actions (such as turning on the faucet) specially.

*Next, the faucet's position is closed.*
*Next, the water-column disappears.*
*The water's height is steady.*

Thinking at t=24.

The height of the water has reached a stable value.

The height of the water is not changing because
there is nothing affecting it.

Because the earlier prediction is not entirely corroborated, JACK returns to try and explain what went wrong. If necessary, JACK will try to refine the causal model.

In this case, the preconditions for making the water rise became unsatisfied before the water reached the safety drain. No rehypothesizing is necessary.

*Make the water disappear.*

**To achieve:**

<WATER IN BASIN> FALSE

**These must hold:**

<DRAIN CONNECTED-TO WATER> TRUE
<DRAIN PART-OF BASIN> TRUE
<WATER CONNECTED-TO WATER-COLUMN> FALSE
<WATER CONNECTED-TO DRAIN> TRUE
<WATER CONNECTED-TO SAFETY> FALSE
<WATER IN BASIN> TRUE
<WATER-HEIGHT AMOUNT> BELOW-SAFETY
<WATER-HEIGHT RATE> ZERO

**These should be set:**

<DRAIN CONTAIN STOPPER> FALSE
<WATER CONTAIN SOAP> TRUE

The planner generates a plan for making the water go away.

*Later, the stopper is not in the drain.*

*Predict what will happen.*

**Nothing will change.**

This prediction is based on an unsatisfied precondition in the rule which describes how water flows out of the drain – namely, there is no soap in the water. JACK will now discover that this is a spurious precondition.

*The water's height is decreasing.*

Thinking at t=26.

This precondition is spurious:
At t=11, <SOAP IN WATER> FALSE.
At t=26, <SOAP IN WATER> TRUE.

Water flows out of the drain whether or not there is soap in the water. This precondition is flushed.

*Later, the water disappears.*

*Finally, nothing is changing.*

This completes the reasoning session in the sink domain which shows how the explainer, predictor, and planner can all use the causal model which was constructed during the learning session.

## Reasoning with the Toaster Model

This section contains an annotated transcript of solved causal reasoning problems in the toaster domain. JACK will have three opportunities to refine the causal model of the toaster when causal reasoning does not corroborate the behavior of the toaster. Again, the problems and the sequence of events appear in *italic* type. JACK's new hypotheses and the responses of the causal reasoning programs appear in bold type. Comments appear in normal type.

This reasoning session takes place immediately after the initial learning session. The toast has popped up and the coils have cooled down.

*Later, the bread is not in the slot.*
*Next, the plug is not in the outlet.*
*Next, the new bread is in the slot.*
*The bread's shade is white.*

*Next, the lever's position is down.*
*The bread is not visible.*

*Predict what will happen.*

Next, these will hold:
<COILS-TEMPERATURE RATE> POSITIVE
<BREAD-SHADE RATE> POSITIVE
Later, these will hold:
<COILS-TEMPERATURE AMOUNT> HOT
<COILS-TEMPERATURE RATE> ZERO
<BREAD IS VISIBLE> TRUE
<BREAD-SHADE AMOUNT> DARK
<BREAD-SHADE RATE> ZERO
Next, nothing will change.

Notice that the prediction does not mention the lever popping up. This is because JACK was unable to generate an hypothesis from the earlier, ambiguous experience to explain this event.

*Next, nothing is changing.*

The prediction is not corroborated. This is evidence that the causal model is incomplete. JACK compares the situation in which the causal model for the toaster was first constructed against the current situation. Any differences might explain why the model worked then but not now.

Thinking at t=14.

This precondition was missing:

At t=1, <PLUG IN OUTLET> TRUE.

At t=13, <PLUG IN OUTLET> FALSE.

**Proposing a new quantity for the plug.**

**Please give a name for this quantity.**

> *current*

**Asserting an influence between**

**the current of the plug and**

**the temperature of the coils.**

**This dependence satisfies**

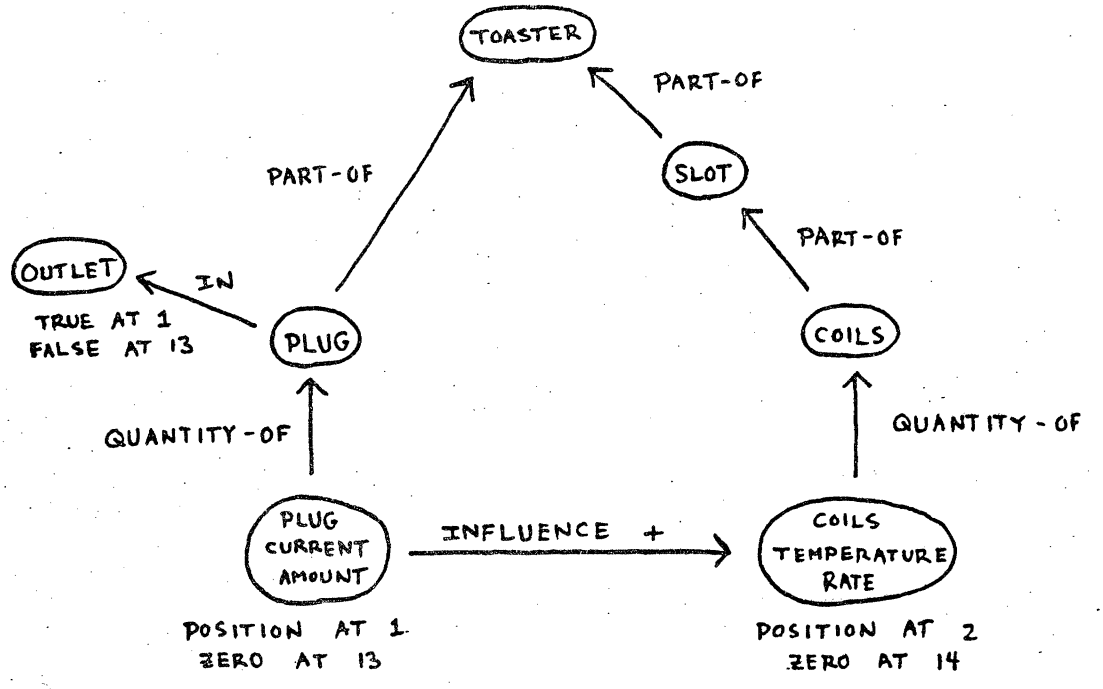
**same device.**

**The temperature of the coils is increasing because**

**the positive of the lever is negative and**

**the current of the plug is positive.**

JACK is able to generate a causal explanation based on a difference which is assumed to reveal a precondition which became unsatisfied. The plug being in/out of the outlet is the only difference JACK finds. Furthermore, the plug and the coils satisfy the same device heuristic. The temporal proximity requirement does not apply when JACK's hypotheses are generated from differences between two situations rather than from changes which are causes in a single situation. Differences are an alternate way of generating candidate causes.




*Next, the dial's setting is M.*
*The plug is in the outlet.*

*Predict what will happen.*

Next, these will hold:

<COILS-TEMPERATURE RATE> POSITIVE
<BREAD-SHADE RATE> POSITIVE

Later, these will hold:

<COILS-TEMPERATURE AMOUNT> HOT
<COILS-TEMPERATURE RATE> ZERO
<BREAD IS VISIBLE> TRUE
<BREAD-SHADE AMOUNT> DARK
<BREAD-SHADE RATE> ZERO

Next, nothing will change.

*Next, the coils' temperature is increasing.*

*Later, the lever's position is up.*

*The coils' temperature is hot.*

*The coils' temperature is steady.*

*The bread's shade is medium.*

Another prediction has gone awry. Two shades of toast (dark,medium) have resulted from apparently the same increase in the coils' temperature. (JACK cannot perceive differences in the durations of the intervals during which the coils heat up). The dependence between the coils' temperature and the darkness of the toast appears to be one-to-many. Because of the teleological assumption that dependences in devices are functions, this is evidence that the causal model is incomplete. JACK compares the two situations to try to explain the difference.

At t=2, <DIAL-SETTING AMOUNT> D

At t=16, <DIAL-SETTING AMOUNT> M

**Asserting a function between**

**the setting of the dial**

**and the shade of the bread.**

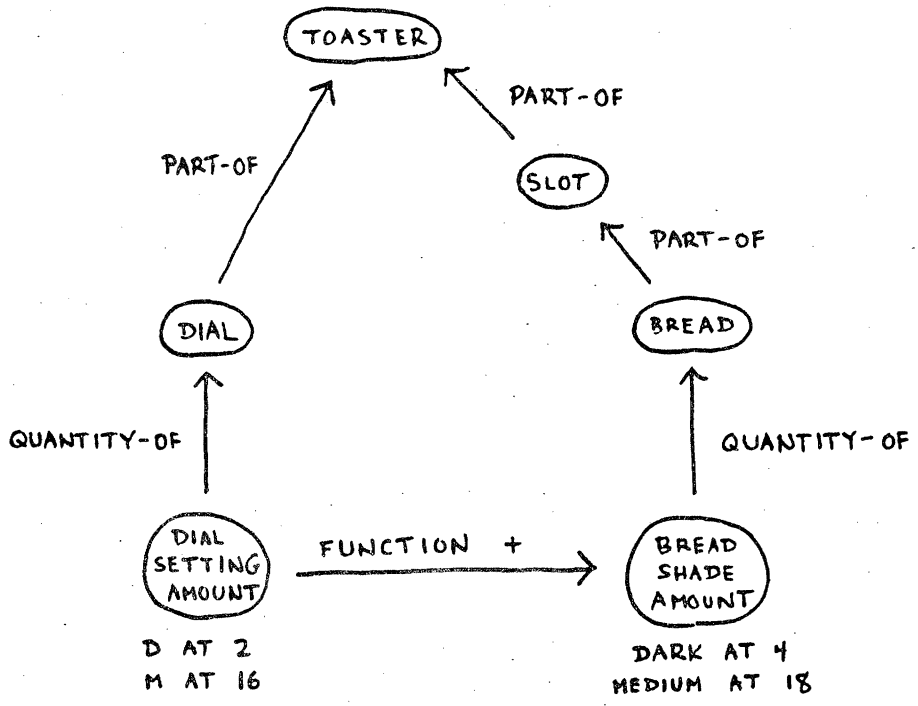
**This dependence satisfies**

**same device.**

**The shade of the bread decreased because**

**the setting of the dial decreased.**

JACK finds a new dependence which displays a satisfactory one-to-one correspondence between values of the thermostat dial and shades of the resulting toast. The thermostat dial and the bread satisfy the same device heuristic.




At t=4, <BREAD-SHADE AMOUNT> DARK
At t=18, <BREAD-SHADE AMOUNT> MEDIUM

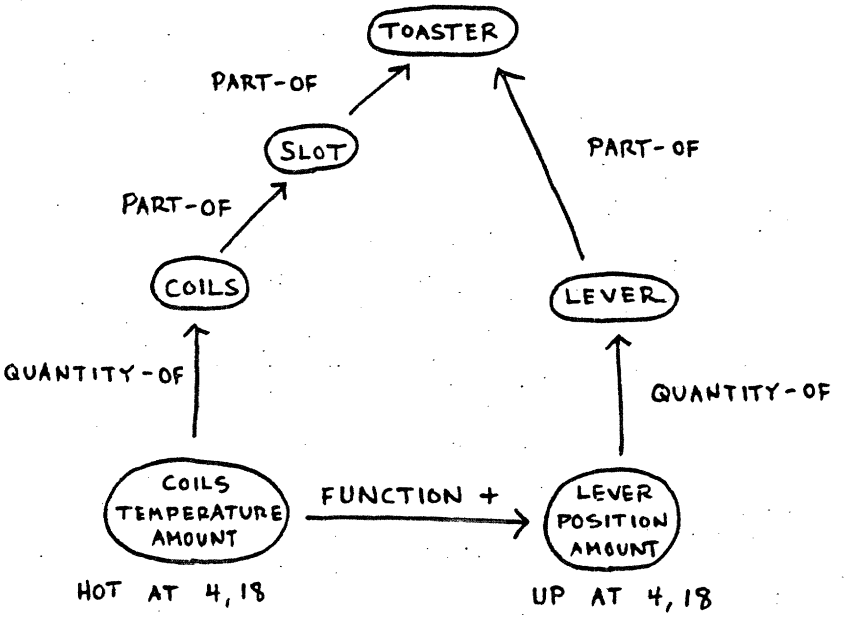
Asserting a function between
the temperature of the coils
and the position of the lever.

This dependence satisfies
same device.

The position of the lever increased because
the temperature of the coils increased.

JACK also finds an explanation for why the lever popped up. Earlier, there were two competing hypotheses – either the change in the shade of the bread or the change in the temperature of the coils caused the lever to pop up. Now

JACK has seen the lever pop up for two shades of toast. This dependence would be many-to-one, violating the felicity condition that dependences be monotonic functions, hence one-to-one. JACK chooses the remaining hypothesis – the lever pops up when the coils reach their maximum temperature.

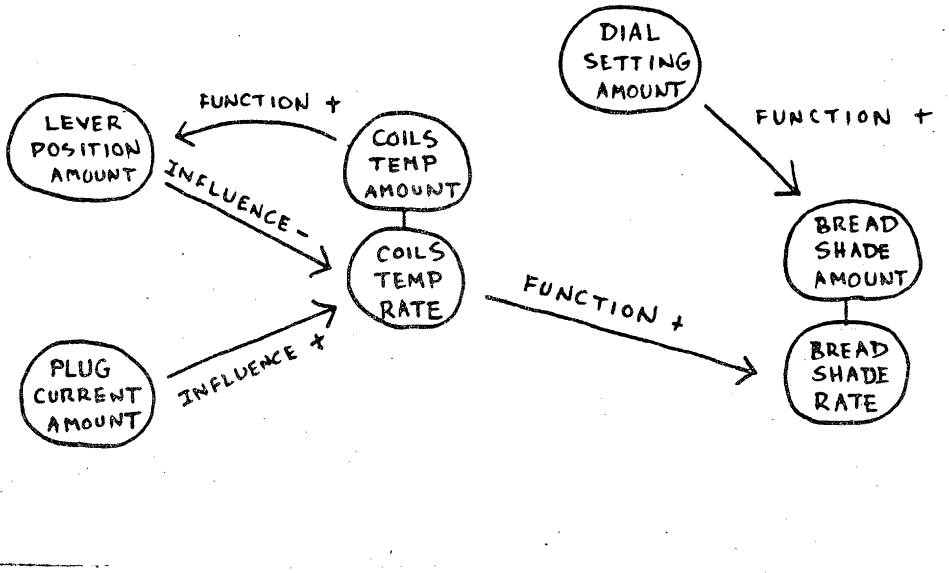



*Next, the coils temperature is decreasing.*

*Later, the coils are cold.*

*Finally, nothing is changing.*

Here is JACK's refined model of the toaster.




The quantity spaces of the quantities are:

Lever Position (DOWN -> UP)

Plug Current (ZERO -> POSITIVE)

Coils Temperature (COLD -> HOT)

Dial Setting (M -> D)

Bread Shade (WHITE -> MEDIUM -> DARK)

JACK's model of the thermostat mechanism in the toaster is abstract. JACK does not know that a coil of metal expands until a circuit is broken and that darker pieces of toast say in the toaster longer. Although the "guts" of the toaster remain unknown, JACK's model of the toaster is useful. For example, consider this planning problem:

*Make the bread's shade lighter.*

To achieve:
  <BREAD-SHADE AMOUNT> LIGHT
These must hold:
  <COILS PART-OF SLOT> TRUE
  <BREAD PART-OF SLOT> TRUE
  <DIAL PART-OF TOASTER> TRUE

<LEVER PART-OF TOASTER> TRUE
<PLUG PART-OF TOASTER> TRUE
<COILS-TEMPERATURE AMOUNT> COLD
<PLUG-CURRENT AMOUNT> POSITIVE

These must be set:

<DIAL-SETTING AMOUNT> L
<PLUG IN OUTLET> TRUE
<LEVER-POSITION AMOUNT> DOWN

JACK is able to solve this planning problem by extrapolating the correspondence due to the function between the setting of the thermostat dial and the darkness of the toast. Because the function is assumed to be monotonic, a lower setting of the dial should result in a lighter shade of toast.

This completes the reasoning session in the toaster domain.

Causal reasoning can provide feedback about deficiencies in a causal model at any stage of its evolution. In the case of the toaster model, several inadequacies were discovered when predictions proved inaccurate and plans did not work. JACK was able to generalize the model to explain these new phenomena by applying generalization rules adapted to causal models and by exploiting constraints formed from a teleological assumption about the nature of dependences in devices.

There is one more planning problem remaining in the sink domain which the planner is unable to solve. The planner will not be able to solve this problem until the causal model of the sink is extended via an analogy. The account of the planner's initial failure, the analogy, and the final successful plan appear in the next chapter.

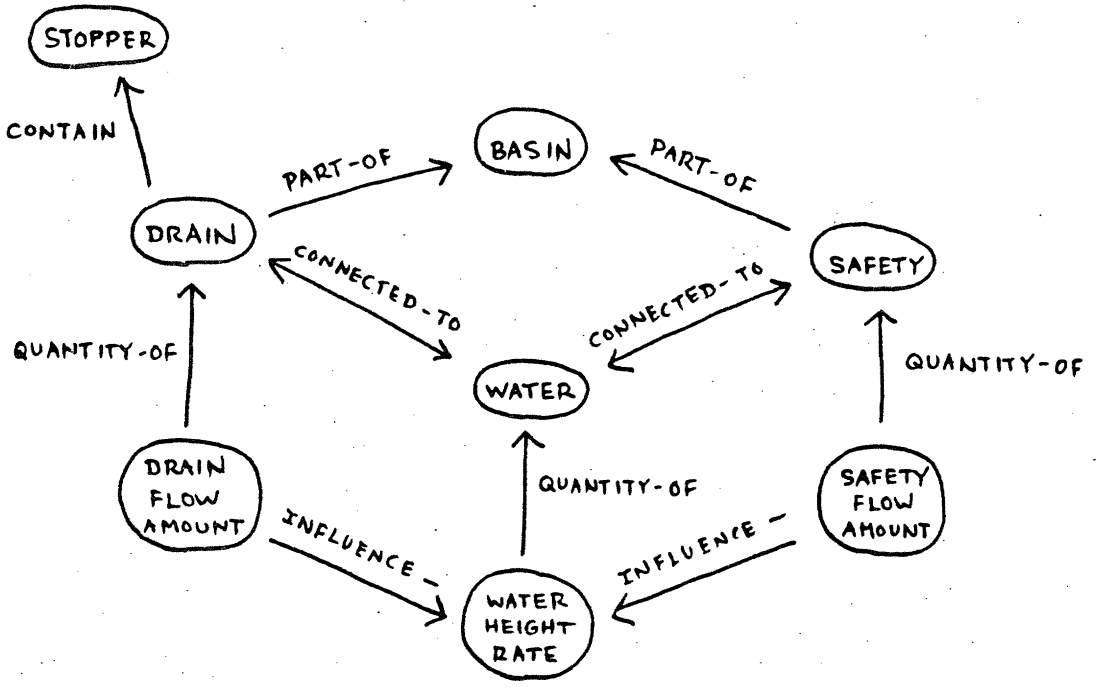
# CHAPTER 5
# EXTENDING THE CAUSAL MODEL: ANALOGY

A pervasive common sense competence is the ability to apply knowledge from former experience to new problems. Analogy involves comparing two domains, one which is well understood, and one which is the subject of current investigation. The driving assumption behind analogy is that if two domains are similar enough, then constraints which hold in one domain will also hold in the other.

Analogy can be used to extend causal models by comparing the causal relations modelled by causal rules. Successful analogies result in preconditions and/or effects being mapped over from one causal rule to another. Since causal rules support explanation, prediction, and planning, analogies enhance the capability to do these forms of causal reasoning. An example from the sink domain illustrates how this works.

The planner is given the problem of making the water rise above the safety drain. It knows that the height of the safety drain is an equilibrium value for the water's height and concludes that it must change the equilibrium state to a state of increase. This means preserving the positive half of the equilibrium and breaking the negative half. The planner quickly determines that keeping the faucet on will make the height of the water rise but it searches in vain for a way to stop the water from flowing out of the safety drain. There is no known action which can inhibit the operation of the safety drain.

An analogy with the normal drain comes to the rescue. The planner knows that water will not flow out of the normal drain when the stopper is in. The analogy leads to the discovery that the normal drain has a stopper and the safety drain does not. This knowledge is mapped over by adding a new precondition to the rule for flow at the safety drain – the safety drain must not contain a stopper either. The original planning problem can now be solved by plugging up the safety drain, a previously unknown action which is now available to the planner. The use of analogy augments the causal rule which describes flow out of the safety drain. The extended causal model enables the planner to solve a problem it would have otherwise failed on.




An Analogy

## Issues in Analogy

- What to compare.

Finding an appropriate domain to compare to the given domain is not an easy task. A solution is offered which is rather specific to this research and involves abstracting to the level of quantities and dependences where causal descriptions are summarized. This solution exposes a more general heuristic of comparing summarized descriptions before comparing detailed descriptions. This solution is only a hedge, and does not propose any memory model or indexing/retrieval scheme. These appear to be necessary elements of any general theory of selection.

- How to match.

The basic operation of analogy is comparing two domains to determine how well they match. The partial matcher presented here comes in two parts. There is a relational matcher and a causal rule matcher.

- How to evaluate a match.

What constitutes a good match? Part of this problem is solved implicitly when a good selection is made. But some parts of a concept are more important than others in a given context. This issue is not addressed here beyond requiring that matches be nearly, but not quite perfect.

- How to map.

Once an analogy has been selected, performed, and justified, the final task is to reap the results by mapping constraints over from the known domain to the evolving domain. The mapper presented here uses the results of the matcher to augment causal rules, moving knowledge in the form of preconditions and/or effects from the source rule to the target rule.

### What to Compare

The first step in doing analogy is determining what concepts to compare. Some kind of selection process should precede the matching. Otherwise, the only option is to blindly compare all pairs of known concepts in the hope of finding two that match well and form a useful analogy.

This is the selection problem. The selection problem is really two problems – relevance and retrieval. The selector must find another knowledge structure which is relevant to the reasoning task at hand and the search process should be made efficient by making candidate knowledge structures easy to access.

In principle, both of these problems can be solved simultaneously by employing an appropriate indexing scheme. Unfortunately, the indexing problem seems to be very complex and there does not appear to be a simple solution. Any knowledge structure might describe several different items and might support several different kinds of problem solving tasks. Also, different knowledge structures might describe different aspects of the same items.

Winston has noted that in less constrained analogy situations, causal relations should be matched first [Winston 80]. Because analogies in this work always and only involve causal rules, some of the problems involved in selection are implicitly solved. The selection problem for this research reduces to locating causal rules which describe similar causal relations on similar objects. Selection tries to ensure before the matcher is invoked that the knowledge structures being compared are indeed similar and that the results of the matching have a strong possibility of being useful in an analogy.

The way to avoid doing a full match on preconditions and effects of causal rules until it appears justified is to first match a description of the same knowledge which captures the essence without the details. This is exactly the difference between the quantity level and the physical level of causal rules. The way to select causal rules for analogies is to compare their quantity levels. Only if they are similar there is the matcher invoked to compare the more detailed physical levels.

The procedure for selecting causal rules for analogies is:

---

Selector:
Given a causal rule CAUSAL-RULE-1,
For each causal rule CAUSAL-RULE-2 in the set of other causal rules until success,
Compare pairs of dependences, one from CAUSAL-RULE-1 and one from CAUSAL-RULE-2 with the relational network matcher.
If all dependences match, succeed.
Call the causal rule matcher on CAUSAL-RULE-2 and CAUSAL-RULE-1.

---

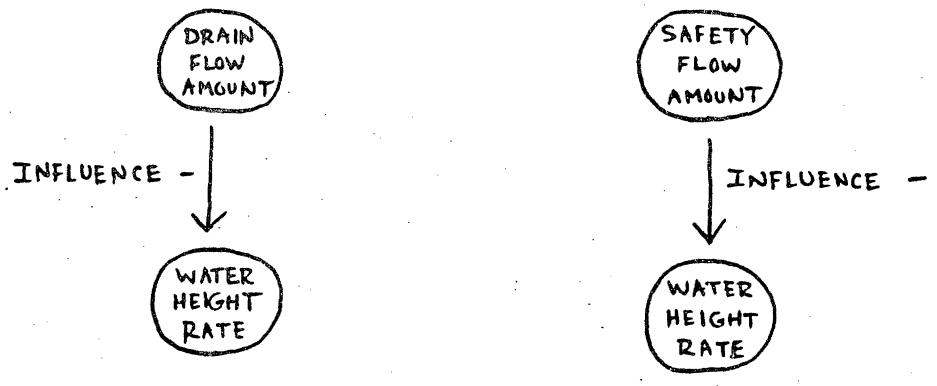
The planner stalled on the problem of making the water rise above the safety drain when it could find no action which prevents water from flowing out of the safety drain. The causal rule which describes flow at the safety drain is now identified and the selector tries to find a different but relevant causal rule which can be used in an analogy.

The quantity level of the causal rule which depicts flow at the safety drain describes a single influence between the flow at the safety drain and the height of the water.

There are five other rules to consider. Two of these rules – the ones which describe the causal links between turning the faucet on and off and the appearance and disappearance of the water column – describe discrete changes, functions rather than influences, and are quickly eliminated. Another rule describes the equilibrium at the safety drain. This rule has two dependences and cannot match. Yet another rule describes how water rises when the water column is present. This rule comes close to matching because it also describes an influence which changes the height of the water. However, the directions of the influences clash – one describes how

the water can rise, the other how it can fall.

The rule which the selector finally chooses is the rule which describes flow at the normal drain. Both rules describe negative influences on the height of the water.




A Selection

Although the procedure for doing selection for analogies presented here is highly specific to this research, it does expose a principle which is applicable to the problem in general. The idea is to find relevant knowledge structures by comparing abstract, summarized descriptions of those knowledge structures first. Only if the abstract descriptions match well is the full matcher invoked to do a detailed comparison of the knowledge structures. Thus selection can be merely another form of matching. The difference is that selection involves matching at an abstract level. The small investment made by matching at an abstract level avoids committing the matcher to doing detailed comparisons until there is some assurance that the effort will bear some fruit.

## How to Match

There are two matchers. One works on the relational network which is the foundation of our knowledge representation scheme. This matcher concerns itself with nodes and arcs in the relational network. The other matcher works on causal rules, which are built from objects and relations in the relational network. The causal rule matcher uses the results of the relational network matcher. These matchers are

not as powerful as others that have appeared in the artificial intelligence literature [Winston 80, 82, Brotsky 80], but they serve to support the use of analogy in this research. The matchers are described in Appendix V.

## How to Map

Once an analogy is selected, performed, and justified the last thing to do is to reap the results of the comparison by mapping information from one causal rule to another. The assumption behind analogy is that relations or constraints which hold in one concept will hold in another if the two are similar enough.

The mapper uses the results of the causal rule matcher. The preconditions and effects that did not match are the concern of the mapper.

The following is the procedure for mapping preconditions and effects from one causal rule to another:

Mapper:

Given a target causal rule and the lists of unmatched states (preconditions or effects), unmatched objects and matched objects from the causal rule matcher and the relational network matcher,

For each unmatched state (precondition or effect) consisted of a SUBJECT, RELATION, OBJECT and VALUE,

Construct a matching state for the target causal rule according to the following:

Map the RELATION exactly.

Map the VALUE exactly.

For the SUBJECT and OBJECT,

If they are relations, identify their values and map them as states.

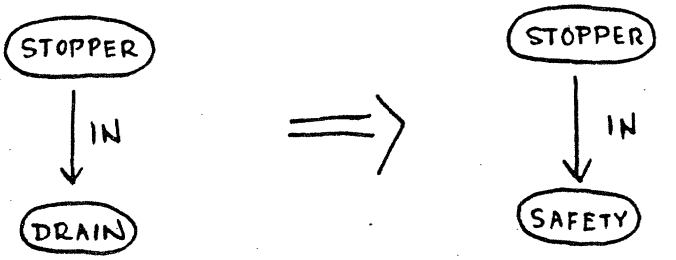
If they are objects,

If they appear in the matched objects list, map the corresponding objects.

If they appear in the unmatched objects list, generate another object in the immediate class which contains the unmatched object and map the new object.

---

There is one precondition which does not match when the rule which describes flow at the normal drain is compared to the rule which describes flow at the safety drain. This precondition says that the stopper must be out of the normal drain. The mapper maps this precondition by preserving the relation IN and the value FALSE, substituting SAFETY for DRAIN which are corresponding objects, and generating a new STOPPER.




A Mapping

## A Successful Analogy

This section contains an annotated transcript of a planning problem which originally fails and then succeeds after missing knowledge is provided by means of an analogy.

The input to the program is in *italic* type. The program's responses are in **bold** type. Comments appear in normal type.

*Make the water's height greater than the safety's height.*

The problem is to make the water's height rise above the safety drain. In its experience with the sink, the learning program has never seen the water above the safety drain, therefore there is no causal rule in the causal model which lists this state as an effect.

However, the planner does know that the height of the safety drain corresponds to an equilibrium state for the water's height. It reasons that the water can be made to rise above the safety drain by changing the equilibrium state to a state of increase, i.e., by breaking the negative half of the equilibrium state.

Hence the planner identifies the causal rule which describes flow at the safety drain – the negative half of the equilibrium state. Here it gets stuck. The planner can find no way of disabling any of the preconditions for flow at the safety drain.

**To achieve:**

<WATER-HEIGHT AMOUNT> ABOVE-SAFETY

**is not possible.**

An analogy now helps to solve this planning problem. The first step is finding another causal rule to use in the analogy. Selection is done by comparing the summarized descriptions of causality appearing at the quantity level of causal rules. The assumption is that if rules match well at the abstract quantity level, there is a good chance they will match well at the detailed physical level and be useful in an analogy.

The selector finds that the quantity level of the causal rule describing flow at the normal drain matches best with the quantity level of the causal rule describing flow at the safety drain.

<DRAIN-FLOW INFLUENCE WATER-HEIGHT> NEGATIVE
<SAFETY-FLOW INFLUENCE WATER-HEIGHT> NEGATIVE

Matched Objects: (DRAIN-FLOW SAFETY-FLOW), WATER-HEIGHT

The matcher compares DRAIN-FLOW to SAFETY-FLOW by exploring the networks surrounding these two nodes. This matching reveals the following matched and unmatched preconditions.

**Matched preconditions:**

<DRAIN CONNECTED-TO WATER> TRUE
<SAFETY CONNECTED-TO WATER> TRUE

<DRAIN PART-OF BASIN> TRUE
<SAFETY PART-OF BASIN> TRUE

<WATER CONECTED-TO WATER-COLUMN> FALSE
<WATER CONECTED-TO WATER-COLUMN> FALSE

<WATER CONNECTED-TO DRAIN> TRUE
<WATER CONNECTED-TO SAFETY> TRUE

<WATER IN BASIN> TRUE
<WATER IN BASIN> TRUE

Unmatched Preconditions:

<DRAIN CONTAIN STOPPER> FALSE

The mapper maps the relation involving the stopper over to the causal rule describing flow at the safety drain.

Proposing a stopper as in
<DRAIN CONTAIN STOPPER> FALSE

With this new information, the planner now can successfully generate a plan to make the water rise above the safety drain. The preconditions for the rule which shows how water can be made to rise must be satisfied, and some precondition for the rule which shows how water flows out the safety drain must be inhibited.

To achieve:
<WATER-HEIGHT AMOUNT> ABOVE-SAFETY
These must hold:
<WATER-COLUMN CONNECTED-TO TAP> TRUE
<WATER-COLUMN CONNECTED-TO BASIN> TRUE
<WATER-COLUMN-WIDTH AMOUNT> POSITIVE
These should be set:
<SAFETY CONTAIN STOPPER> TRUE

Analogies extend the causal model by augmenting causal rules. Problems in explanation, prediction, and planning which fail because the causal model is incomplete can become solvable after it is extended through analogies.

Although this is the only place where analogies are employed in this work, they could be applied in the learning process itself. Learning is motivated by the need to explain changes in the visual environment, and results in the construction of causal rules. Conceivably, causal explanations for changes could be based on analogies with known causal rules. This kind of analogy would be more difficult because it would involve comparing a causal rule to an unstructured situation, rather than comparing two known causal rules. This problem might be the subject of future research.

# CHAPTER 6
# LOOKING BACK, AROUND, AND AHEAD

## This Work

This section reviews the accomplishments of this thesis in terms of the issues addressed, the solutions offered, and the principles behind those solutions.

This thesis presents a learning system which hypothesizes and refines causal models of simple physical systems by constructing causal explanations for observed changes in these systems. *The problem of formulating causal hypotheses is made tractable by a set of constraints on causal relations which are embedded in the learning system.* This is the main result of this thesis.

These constraints are:

- Temporal and physical proximity.

Four heuristics capture the common sense notion that causally connected events are contiguous in space and time. Temporal proximity is tested by the *temporal adjacency* or the *simultaneity* heuristic. Physical proximity is tested by the *physical connectedness* heuristic or the weaker *same device* heuristic.

- A finite set of abstract causal explanations for changes in terms of quantities and dependences.

Shifting the representation for changes and causality to the level of quantities and dependences exposes various constraints that reduce the set of viable causal explanations. The constraints exposed by this perspicuous representation include:

- Types of changes in dependent quantities are linked to types of changes in independent quantities and types of dependences.
- The signs or directions of change of quantities and dependences have to be consistent.
- There are a finite number of explanations for second-order changes in quantities.

The set of second-order causal explanations is simplified considerably by a felicity condition which excludes tradeoff situations.

These constraints collectively define a kind of syntax of causal explanation which the learning system exploits to hypothesize causal relations.

The causal rules which make up JACK's causal models are constructed at two levels – at the quantity level in terms of independent quantities, dependences, and dependent quantities, and at the physical level in terms of preconditions and effects.

Preconditions embody the notion of enabling conditions for causal relations. They also permit causal explanations to be hypothesized in terms of the last of a set of preconditions becoming satisfied.

JACK is able to refine causal models by generalizing over further experience. The generalization rules JACK uses include:

- Given two positive examples of a causal relation, any unsatisfied preconditions or unrealized effects can be dropped.

  This is a variant of the well-known drop-condition specialization rule [Winston 75].
- Given a positive and negative example of a causal relation, any differences are likely to include a missing precondition.

  This induction rule harks back to the time-tested near-miss idea [Winston 75].

The causal models which JACK constructs support causal reasoning (explanation, prediction, and planning) which in turn provides feedback about deficiencies in the causal models. Inaccurate predictions and failed plans reveal situations where the above generalization rules can be fruitfully employed.

A teleological assumption that dependences in devices are functions, and a felicity condition that requires these functions to be monotonic together constrain dependences to be one-to-one. Thus one-to-many or many-to-one behavior also lead JACK to try to refine the existing model.

Analogies are another way to improve an existing causal model. Causal rules are compared first at the summarized quantity level, then at the physical level. Differences between otherwise well-matched causal rules are mapped over.

## Other Work

This section describes the relations between this research and other previous and current research efforts.

The representations for quantities and dependences are borrowed directly from Ken Forbus' seminal Qualitative Process Theory [Forbus 84]. Also, the causal rules which JACK constructs are reminiscent of Forbus' process descriptions.

The abstract causal explanations employed by the learning system were inspired originally by Chuck Rieger's work on representing causality [Rieger 76].

Pat Hayes' object-based *histories* [Hayes 79] implicitly include the notion of temporal and physical proximity defining boundaries on causal interactions.

The physical proximity principle is similar to Randy Davis' locality principle [Davis 83] – used to generate candidate faults in the troubleshooting of electronic circuits. Modelling and troubleshooting employ some of the same kinds of reasoning.

The inductive inference rules used to generalize causal models over experience are variants of rules introduced in Patrick Winston's landmark thesis [Winston 75]. In addition, Ryszard Michalski has treated induction comprehensively [Michalski 83] and Tom Mitchell has provided valuable insights on the induction of conjunctive concepts [Mitchell 82].

Johan de Kleer pioneered the use of causal and teleological reasoning in the domain of expert analysis of circuits [de Kleer 79]. This contrasts with the more naive modelling of physical systems in this work.

The rule-based causal reasoning programs which perform explanation, prediction, and planning, have roots which go all the way back to STRIPS [Fikes and Nilsson 71].

The use of analogy to construct and refine concepts has been investigated fruitfully in [Winston 80] and [Gentner 83].

## Future Work

This section discusses limitations of the current learning system and where appropriate, identifies solutions from other research efforts, as well as thoughts on extensions to this work.

All learning systems are limited ultimately by any fixed representation language. JACK is limited by the representation language for describing physical systems and their changes and the representation language used to describe the various constraints on causal hypotheses.

The temporal and physical proximity heuristics capture a useful common sense notion of causality but exclude at least two classes of causal relations – those that involve "action at a distance", and those that involve "delayed reactions".

Part of the problem is the limited ability to construct hierarchical descriptions. The PART-OF relation supports only crude hierarchical structural descriptions. More importantly, there is no ability to "open up" a physical system by expanding to a description at a lower level. Similarly, the time representation does not support nested intervals which could partition time at several levels of resolution. If both structure and time could be represented hierarchically, then "delayed reactions" might be explained by constructing a causal chain at a lower level of resolution. [Davis et al 82] offers ideas about representations for hierarchical structural descriptions. Allen has a hierarchical time representation [Allen 81].

JACK does successfully model an instance of action at a distance when he proposes a dependence between the temperature of the coils and the darkness of the toast. However, this is somewhat fortuitous. JACK uses the same device heuristic in this situation, effectively proposing a physical connection between the coils and the toast. Thus JACK gets the right answer for the wrong reason. JACK does not model the heat exchange as an instance of action at a distance because there is no available representation for this class of causal relations.

If there was an abstract, explicit representation of what a causal relation is, perhaps it would be possible to derive context-dependent heuristics for identifying causality – heuristics like the ones used in this thesis, but also more relaxed versions of temporal and physical proximity which would not exclude instances of action at a distance and delayed reactions. These heuristics should be ordered so that levels of resolution and boundaries denoting where the closed system ends would be systematically expanded until a viable hypothesis was constructed. Such a learning system could dynamically augment the language used to represent constraints on causal relations. This capability would address the fixed representation bottleneck problem in learning systems. These conjectures identify a difficult, but potentially fruitful area in which to expand this thesis.

Another limitation of the current representation language is the simplified set of causal explanation abstractions for understanding states of physical systems. The most complex abstraction available is the equilibrium state. An extended version of the learning system might model positive and negative tradeoff situations and make use of more complex abstractions built up from many dependences, such as feedback loops.

The role of analogy both in extending and hypothesizing causal models is another area for possible exploration. Analogies could be used to generate hypotheses which

could then be tested by some version of the temporal and physical proximity constraint. Analogies also might be useful in "opening up" a system, i.e., in hypothesizing invisible components and connections to construct causal explanations. Analogy is a huge problem which subsumes the issues of indexing, partial matching, and transferring knowledge – each a difficult problem in itself.

JACK's hypotheses are justified by satisfying the temporal and physical proximity requirements, by matching one of the abstract causal explanations, and by not violating teleological assumptions about the nature of dependences in devices. JACK can distinguish competing hypotheses only by ordering them according to the version of physical proximity they satisfy (physical connectedness or same device), and by how much must be proposed to complete one of the abstract causal explanations.

Because JACK's ability to order competing hypotheses is limited, and because models are always generalized over a finite set of experiences, JACK's theories are always sensitive to the local maximum problem. In other words, a causal model may adequately explain some finite set of experiences, yet still have latent, possibly gross deficiencies.

JACK already uses inductive inference rules for refining causal models and in the worst case, would need a dependency-directed backtracking capability for retracting hypotheses. A better way to address the local maximum problem is to give JACK the ability to gather more context-dependent justification for hypotheses to better distinguish them immediately, rather than waiting for more revealing experience. JACK needs the capability to design experiments to distinguish and test hypotheses.

The methodology of science obviously provides some abstract guidelines. The ability to design experiments relies on such skills as recognizing parameters and finding ways to isolate them. Being able to change levels of resolution and expand boundaries on the closed system can also aid in the design of experiments. In addition, analogies can suggest experiments. The issue of how to design experiments identifies a most intriguing direction in which to extend this thesis.

# REFERENCES

Allen, James F., "An Interval-Based Representation of Temporal Knowledge," *7th International Joint Conference on Artificial Intelligence*, 1981.

Brotsky, Daniel C., "Efficient Graph Matching through Exploitation of Constraint," Report AIM-600, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1980.

Davis, Randall, et al, "Diagnosis Based on Structure and Function," *National Conference on Artificial Intelligence*, 1982.

Davis, Randall, "Diagnosis via Causal Reasoning: Paths of Interaction and the Locality Principle," *National Conference on Artificial Intelligence*, 1983.

de Kleer, Johan, "Causal and Teleological Reasoning in Circuit Recognition," Report TR-529, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1979.

de Kleer, Johan and John Seely Brown, "Assumptions and Ambiguities in Mechanistic Mental Models," in *Mental Models*, Dedre Gentner and Albert L. Stevens (eds.), Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

Doyle, Richard J. and Boris Katz, "Exploring the Boundary Between Natural Language and Knowledge Representation," Report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, *forthcoming*, 1984.

Doyle, Jon, "A Truth-Maintenance System for Problem-Solving," Report TR-419, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1978.

Fikes, Richard E., and Nils J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence*, vol. 2, nos. 3 and 4, 1971.

Forbus, Kenneth D., "Qualitative Process Theory," Report TR-789, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1984, *Artificial Intelligence, to appear.*

Gentner, Dedre, "Structure-Mapping: A Theoretical Framework for Analogy," *Cognitive Science*, vol. 7, no. 2, 1983.

Hayes, Patrick J., "The Naive Physics Manifesto," in *Expert Systems in the Micro-Electronic Age*, Donald Michie (ed.), Edinburgh University Press, Edinburgh, 1979.

Katz, Boris, "A Three-Step Procedure for Language Generation," Report AIM-599, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1980.

Katz, Boris, and Patrick H. Winston, "A Two-Way Natural Language Interface," in *Integrated Interactive Computing Systems*, P. Degano and Erik Sandewall (eds.), North-Holland, Amsterdam, 1982.

Kirsh, David J., *in conversation.*

Kuipers, Benjamin, "Getting the Envisionment Right," *National Conference on Artificial Intelligence*, 1982.

McDermott, Drew V., "A Temporal Logic for Reasoning about Processes and Plans," *Cognitive Science*, vol. 6, no. 2, 1982.

Michalski, Ryszard S., "A Theory and Methodology of Inductive Learning," *Artificial Intelligence*, vol. 20, no. 3, 1983.

Michalski, Ryszard S. and Richard L. Chilausky, "Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing and Expert System for Soybean Disease Diagnosis," *International Journal of Policy Analysis and Information Systems*, vol. 4, no. 2, 1980.

Mitchell, Tom M., "Generalization as Search," *Artificial Intelligence*, vol. 18, no. 2, 1982.

Rieger, Chuck, "On Organization of Knowledge for Problem Solving and Language Comprehension," *Artificial Intelligence*, vol. 7, no. 2, 1976.

Rieger, Chuck, and Milt Grinberg, "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms," *5th International Joint Conference on Artificial Intelligence*, 1977.

Sacerdoti, Earl D., *A Structure for Plans and Behavior*, Elsevier, New York, 1977.

Simmons, Reid G., "Representing and Reasoning about Change in Geological Interpretation," Report TR-749, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1983.

Simmons, Reid G., "Temporal Representations for Planning," *Area Exam Paper*, 1984.

Sussman, Gerald J., *A Computer Model of Skill Acquisition*," Elsevier, New York, 1975.

Vere, Steven A., "Planning in Time: Windows and Durations for Activities and Goals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 3, 1983.

Weld, Daniel S., "Switching Between Discrete and Continuous Process Models to Predict Genetic Activity," Report TR-793, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1984.

Winston, Patrick H., "Learning Structural Descriptions from Examples," in *The Psychology of Computer Vision*, Patrick H. Winston (ed.), McGraw-Hill Book Company, New York, 1975. Based on a PhD thesis, Massachusetts Institute of Technology, 1970.

Winston, Patrick H., "Learning and Reasoning by Analogy," *Communications of the Association for Computing Machinery*, vol. 23, no. 12, 1980.

Winston, Patrick H., "Learning New Principles from Precedents and Exercises," *Artificial Intelligence*, vol. 19, no. 3, 1982.

# APPENDIX I
# THE SINK SCENARIO

This appendix contains the sequence of events which makes up the learning session in the sink domain.

*Already, the tap, the faucet, and the basin are part of the sink.*
*The drain, the safety, and the stopper are part of the basin.*

*The stopper is in the drain.*
*The faucet's position is closed.*
*The light-switch's setting is off.*
*The window's height is down.*

*Initially, the faucet's position is open.*
*The light-switch's setting is on.*
*Next, a water-column appears between the tap and the basin.*
*The water-column's width is steady.*
*Next, water appears in the basin.*
*The water-column is connected to the water.*
*The water is connected to the drain.*
*The water's height is increasing.*

*Later, the water is connected to the safety.*
*The water's height is equal to the safety's height.*
*The water's height is steady.*

*Later, the faucet's position is closed.*
*Next, the water-column disappears.*
*The water's height is decreasing.*
*Next, the water is not connected to the safety.*
*The water's height is steady.*

*Later, soap is in the water.*

*Later, the stopper is not in the drain.*
*The window's height is up.*
*The water's height is decreasing.*

*Later, the water disappears.*

*Finally, nothing is changing.*

*Later, the stopper is in the drain.*

*Later, the faucet's position is open.*
*Next, a water-column appears between the tap and the basin.*
*The water-column's width is steady.*
*Next, water appears in the basin.*
*The water-column is connected to the water.*
*The water is connected to the drain.*
*The water's height is increasing.*

*Next, the faucet's position is closed.*
*Next, the water-column disappears.*
*The water's height is steady.*

*Later, the stopper is not in the drain.*
*The water's height is decreasing.*

*Later, the water disappears.*

*Finally, nothing is changing.*

# APPENDIX II
# THE CAUSAL MODEL OF THE SINK

This appendix contains the six causal rules which make up the causal model of the sink. Refinements and extensions to the causal model were made at various times. The results of these changes are noted in the appropriate places.

This causal rule describes how turning the faucet on makes the water column appear.

```
CAUSAL-RULE-1

THE OBJECTS ARE
   THE FAUCET
   THE WATER-COLUMN

THE QUANTITIES ARE
   THE POSITION OF THE FAUCET
   THE WIDTH OF THE WATER-COLUMN

THE DEPENDENCES ARE
   <POSITION FUNCTION WIDTH> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <FAUCET PART-OF SINK> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <POSITION AMOUNT> OPEN
   (T) <POSITION RATE> ZERO
   (T) <WIDTH AMOUNT> ZERO
   (T) <WIDTH RATE> ZERO

THE PHYSICAL-EFFECTS ARE
   (T+1) <WATER-COLUMN CONNECTED-TO BASIN> TRUE
   (T+1) <WATER-COLUMN CONNECTED-TO TAP> TRUE

THE QUANTITY-EFFECTS ARE
   (T+1) <WIDTH AMOUNT> POSITIVE
```

This causal rule describes how the water rises eventually to the level of the safety drain as long as the water column is present (and by the previous rule, the faucet is on).

```
CAUSAL-RULE-2

THE OBJECTS ARE
   THE WATER-COLUMN
   THE WATER

THE QUANTITIES ARE
   THE WIDTH OF THE WATER-COLUMN
   THE HEIGHT OF THE WATER

THE DEPENDENCES ARE
   <WIDTH INFLUENCE HEIGHT> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <WATER-COLUMN CONNECTED-TO TAP> TRUE
   (T) <WATER-COLUMN CONNECTED-TO BASIN> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <WIDTH AMOUNT> POSITIVE
   (T) <WIDTH RATE> ZERO
   (T) <HEIGHT AMOUNT> ZERO
   (T) <HEIGHT RATE> ZERO

THE PHYSICAL-EFFECTS ARE
   (T+1) <WATER CONNECTED-TO DRAIN> TRUE
   (T+1) <WATER CONNECTED-TO WATER-COLUMN> TRUE
   (T+1) <WATER-COLUMN CONNECTED-TO WATER> TRUE
   (T+1) <WATER IN BASIN> TRUE

   (T+3) <WATER CONNECTED-TO SAFETY> TRUE

THE QUANTITY-EFFECTS ARE
   (T+1) <HEIGHT RATE> POSITIVE

   (T+3) <HEIGHT RATE> ZERO
   (T+3) <HEIGHT AMOUNT> SAFETY
```

This causal rule describes the equilibrium state that occurs when the faucet is on and the water has reached the level of the safety drain. Notice that there are two dependences of opposite sign. Also notice that there are no effects which are continuous changes. The equilibrium state is stable.

```
CAUSAL-RULE-3

THE OBJECTS ARE
   THE SAFETY
   THE WATER
   THE WATER-COLUMN

THE QUANTITIES ARE
   THE FLOW OF THE SAFETY
   THE HEIGHT OF THE WATER
   THE WIDTH OF THE WATER-COLUMN

THE DEPENDENCES ARE
   <FLOW INFLUENCE HEIGHT> NEGATIVE
   <WIDTH INFLUENCE HEIGHT> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <SAFETY CONNECTED-TO WATER> TRUE
   (T) <SAFETY PART-OF BASIN> TRUE
   (T) <WATER CONNECTED-TO WATER-COLUMN> TRUE
   (T) <WATER CONNECTED-TO DRAIN> TRUE
   (T) <WATER CONNECTED-TO SAFETY> TRUE
   (T) <WATER IN BASIN> TRUE
   (T) <WATER-COLUMN CONNECTED-TO TAP> TRUE
   (T) <WATER-COLUMN CONNECTED-TO BASIN> TRUE
   (T) <WATER-COLUMN CONNECTED-TO WATER> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <FLOW AMOUNT> POSITIVE
   (T) <FLOW RATE> ZERO
   (T) <WIDTH AMOUNT> POSITIVE
   (T) <WIDTH RATE> ZERO

THE PHYSICAL-EFFECTS ARE

THE QUANTITY-EFFECTS ARE
   (T) <HEIGHT AMOUNT> SAFETY
   (T) <HEIGHT RATE> ZERO
```

This causal rule describes how turning the faucet off makes the water column disappear. The same dependence appears here as in the rule which describes how turning the faucet on makes the water appear.

```
CAUSAL-RULE-4

THE OBJECTS ARE
  THE FAUCET
  THE WATER-COLUMN

THE QUANTITIES ARE
  THE POSITION OF THE FAUCET
  THE WIDTH OF THE WATER-COLUMN

THE DEPENDENCES ARE
  <POSITION FUNCTION WIDTH> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
  (T) <FAUCET PART-OF SINK> TRUE
  (T) <WATER-COLUMN CONNECTED-TO TAP> TRUE
  (T) <WATER-COLUMN CONNECTED-TO BASIN> TRUE
  (T) <WATER-COLUMN CONNECTED-TO WATER> TRUE

THE QUANTITY-PRECONDITIONS ARE
  (T) <POSITION AMOUNT> CLOSED
  (T) <POSITION RATE> ZERO
  (T) <WIDTH AMOUNT> POSITIVE
  (T) <WIDTH RATE> ZERO

THE PHYSICAL-EFFECTS ARE
  (T+1) <WATER-COLUMN CONNECTED-TO WATER> FALSE
  (T+1) <WATER-COLUMN CONNECTED-TO BASIN> FALSE
  (T+1) <WATER-COLUMN CONNECTED-TO TAP> FALSE

THE QUANTITY-EFFECTS ARE
  (T+1) <WIDTH AMOUNT> ZERO
```

This causal rule describes how water flows out of the safety drain until it reaches a stable height just below the safety drain. Notice that one of the preconditions in this rule is that the safety drain not contain a stopper. This precondition was not part of the original rule. It is the result of comparing the normal drain to the safety drain in an analogy.

```
CAUSAL-RULE-5

THE OBJECTS ARE
   THE SAFETY
   THE WATER

THE QUANTITIES ARE
   THE FLOW OF THE SAFETY
   THE HEIGHT OF THE WATER

THE DEPENDENCES ARE
   <FLOW INFLUENCE HEIGHT> NEGATIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <SAFETY CONNECTED-TO WATER> TRUE
   (T) <SAFETY PART-OF BASIN> TRUE
   (T) <SAFETY CONTAIN STOPPER> FALSE
   (T) <WATER CONNECTED-TO WATER-COLUMN> FALSE
   (T) <WATER CONNECTED-TO DRAIN> TRUE
   (T) <WATER CONNECTED-TO SAFETY> TRUE
   (T) <WATER IN BASIN> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <FLOW AMOUNT> POSITIVE
   (T) <FLOW RATE> ZERO
   (T) <HEIGHT AMOUNT> SAFETY

THE PHYSICAL-EFFECTS ARE
   (T+1) <SAFETY CONNECTED-TO WATER> FALSE
   (T+1) <WATER CONNECTED-TO SAFETY> FALSE

THE QUANTITY-EFFECTS ARE
   (T) <HEIGHT RATE> NEGATIVE

   (T+1) <HEIGHT AMOUNT> BELOW-SAFETY
   (T+1) <HEIGHT RATE> ZERO
```

This causal rule describes how water flows out of the normal drain. A precondition which stated that there must be soap in the water was dropped.

```
CAUSAL-RULE-6

THE OBJECTS ARE
   THE DRAIN
   THE WATER

THE QUANTITIES ARE
   THE FLOW OF THE DRAIN
   THE HEIGHT OF THE WATER

THE DEPENDENCES ARE
   <FLOW INFLUENCE HEIGHT> NEGATIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <DRAIN CONNECTED-TO WATER> TRUE
   (T) <DRAIN CONTAIN STOPPER> FALSE
   (T) <DRAIN PART-OF BASIN> TRUE
   (T) <WATER CONNECTED-TO WATER-COLUMN> FALSE
   (T) <WATER CONNECTED-TO DRAIN> TRUE
   (T) <WATER CONNECTED-TO SAFETY> FALSE
   (T) <WATER IN BASIN> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <HEIGHT AMOUNT> BELOW-SAFETY
   (T) <FLOW AMOUNT> POSITIVE
   (T) <FLOW RATE> ZERO

THE PHYSICAL-EFFECTS ARE
   (T+2) <DRAIN CONNECTED-TO WATER> FALSE
   (T+2) <WATER CONNECTED-TO DRAIN> FALSE
   (T+2) <WATER IN BASIN> FALSE

THE QUANTITY-EFFECTS ARE
   (T) <HEIGHT RATE> NEGATIVE

   (T+2) <HEIGHT AMOUNT> ZERO
   (T+2) <HEIGHT RATE> ZERO
```

# APPENDIX III
# THE TOASTER SCENARIO

This appendix contains the sequence of events which makes up the learning session in the toaster domain.

*Already, the lever, the plug, the dial, and the slot are part of the toaster.*
*The coils are part of the slot.*

*The coils' temperature is cold.*
*The lever's position is up.*
*The dial's setting is D.*
*The plug is in the outlet.*
*The bread is in the slot.*
*The bread's shade is white.*
*The faucet's position is closed.*
*The light-switch's setting is on.*
*The window's height is up.*

*Initially, the lever's position is down.*
*The faucet's position is open.*
*The bread is not visible.*
*Next, the coils' temperature is increasing.*

*Later, the lever's position is up.*
*The coils' temperature is hot.*
*The coils' temperature is steady.*
*The bread is visible.*
*The bread's shade is dark.*

*Next, the coils' temperature is decreasing.*
*The window's height is down.*

*Later, the coils' temperature is cold.*

*Finally, nothing is changing.*

*Later, the bread is not in the slot.*
*Next, the plug is not in the outlet.*
*Next, the new bread is in the slot.*
*The bread's shade is white.*

*Next, the lever's position is down.*
*The bread is not visible.*
*Next, nothing is changing.*

*Next, the dial's setting is M.*
*The plug is in the outlet.*
*Next, the coils' temperature is increasing.*

*Later, the lever's position is up.*
*The coils' temperature is hot.*
*The coils' temperature is steady.*
*The bread is visible.*
*The bread's shade is medium.*

*Next, the coils' temperature is decreasing.*

*Later, the coils are cold.*

*Finally, nothing is changing.*

# APPENDIX IV
# THE CAUSAL MODEL OF THE TOASTER

This appendix contains the three causal rules which make up the causal model of the toaster. Generalizations were made at various times and are noted in the appropriate places.

This causal rule describes how the temperature of the coils increases when the lever is pushed down and decreases when the lever pops up. JACK learned that the plug has to be in the outlet also.

```
CAUSAL-RULE-1

THE OBJECTS ARE
   THE LEVER
   THE COILS
   THE PLUG

THE QUANTITIES ARE
   THE POSITION OF THE LEVER
   THE TEMPERATURE OF THE COILS
   THE CURRENT OF THE PLUG

THE DEPENDENCES ARE
   <POSITION INFLUENCE TEMPERATURE> NEGATIVE
   <CURRENT INFLUENCE TEMPERATURE> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <LEVER PART-OF TOASTER> TRUE
   (T) <COILS PART-OF SLOT> TRUE
   (T) <PLUG PART-OF TOASTER> TRUE
   (T) <PLUG IN OUTLET> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <POSITION AMOUNT> (DOWN,UP)
   (T) <POSITION RATE> ZERO
   (T) <TEMPERATURE AMOUNT> (COLD,HOT)
   (T) <TEMPERATURE RATE> ZERO
   (T) <CURRENT AMOUNT> POSITIVE
   (T) <CURRENT RATE> ZERO

THE PHYSICAL-EFFECTS ARE

THE QUANTITY-EFFECTS ARE
   (T+1) <TEMPERATURE RATE> (POSITIVE,NEGATIVE)

   (T+3) <TEMPERATURE AMOUNT> (HOT,COLD)
   (T+3) <TEMPERATURE RATE> ZERO
```

This causal rule describes how the heating coils turn bread into toast. JACK learned that the thermostat dial controls the darkness of the toast, when the initial model could not explain why one piece of toast came out darker than another.

```
CAUSAL-RULE-2

THE OBJECTS ARE
   THE COILS
   THE BREAD
   THE DIAL

THE QUANTITIES ARE
   THE TEMPERATURE OF THE COILS
   THE SHADE OF THE BREAD
   THE SETTING OF THE DIAL

THE DEPENDENCES ARE
   <TEMPERATURE FUNCTION SHADE> POSITIVE
   <SETTING FUNCTION SHADE> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <COILS PART-OF SLOT> TRUE
   (T) <BREAD PART-OF SLOT> TRUE
   (T) <DIAL PART-OF TOASTER> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <TEMPERATURE RATE> POSITIVE
   (T) <SETTING AMOUNT> (L,M,D)
   (T) <SETTING RATE> ZERO

THE PHYSICAL-EFFECTS ARE

THE QUANTITY-EFFECTS ARE
   (T) <SHADE RATE> POSITIVE

   (T+2) <TEMPERATURE AMOUNT> HOT
   (T+2) <TEMPERATURE RATE> ZERO
   (T+2) <SHADE AMOUNT> (LIGHT,MEDIUM,DARK)
   (T+2) <SHADE RATE> ZERO
```

This causal rule describes how the lever pops up when the coils reach their maximum temperature. This is the closest JACK comes to modelling the thermostat mechanism.

```
CAUSAL-RULE-3

THE OBJECTS ARE
   THE COILS
   THE LEVER

THE QUANTITIES ARE
   THE TEMPERATURE OF THE COILS
   THE POSITION OF THE LEVER

THE DEPENDENCES ARE
   <TEMPERATURE FUNCTION POSITION> POSITIVE

THE PHYSICAL-PRECONDITIONS ARE
   (T) <COILS PART-OF SLOT> TRUE
   (T) <LEVER PART-OF TOASTER> TRUE

THE QUANTITY-PRECONDITIONS ARE
   (T) <TEMPERATURE AMOUNT> HOT
   (T) <TEMPERATURE RATE> ZERO

THE PHYSICAL-EFFECTS ARE

THE QUANTITY-EFFECTS ARE
   (T) <POSITION AMOUNT> UP
   (T) <POSITION RATE> ZERO
```

# APPENDIX V
# THE MATCHERS

This appendix describes the matchers used in generalizing and forming analogies.

## The Relational Network Matcher

The relational network matcher is given two concepts to compare. These concepts correspond to two entities in the relational network, and the matcher compares the concepts by exploring the subnetworks which the two entities are embedded in.

Relations define the template which must be common to both concepts, hence they provide the major source of constraint for the matcher. The intent of matching is to find what relational structures are shared by the two concepts. Shared relations in turn indicate which objects correspond to each other across the two concepts. Relations are matched first and objects are matched only by virtue of participating in the same relations.

The primitive structure in the relational network is the *relation*:

<SUBJECT RELATION OBJECT>

There are two dimensions of complexity in the relational network. Each SUBJECT and OBJECT can participate in an arbitrary number of relations and relations can be nested, i.e., any SUBJECT or OBJECT can itself be a full <SUBJECT RELATION OBJECT> structure. The matching is done in a bottom-up fashion, starting at two locations in the relational network and proceeding outwards. Primitive relations are encountered in pairs along the way and they must match in the following way:

- the arcs (RELATIONS) must match exactly and
- the nodes (atomic SUBJECTS and OBJECTS) must either match exactly or be shown to be in the same class (their A-KIND-OF hierarchies join.

Matching continues through the network, exploring the subnetworks surrounding the original two locations, until no further matches can be made, or the network is exhausted.

The following is the procedure for doing matching on the relational network:

---

Relational Network Matcher:

Given two locations in the relational network,

If they are relations, call the arc (relation) matcher.

If they are objects, call the node (object) matcher.

If they are not of the same type, fail.

---

---

Node Matcher:

Given two nodes in the relational network,

If the nodes are the same node, succeed.

If the nodes are in the same class (their A-KIND-OF hierarchies join), succeed.

If no match, fail.

Otherwise, call the relation pairer on the two nodes.

---

---

Arc Matcher:

Given two arcs in the relational network,

If the arcs are not the same arc, fail and stop here.

Call the network matcher on the entities at the source ends of the arcs – the SUBJECTS.

If the SUBJECTS do not match, fail and stop here.

Call the network matcher on the entities at the target ends of the arcs – the OBJECTS.

If the OBJECTS do not match, fail and stop here.

Otherwise, call the value matcher on the two arcs.

Call the relation pairer on the two arcs.

---

---

Relation Pairer:

Given two entities SUBJECT1 and SUBJECT2 in the relational network (corresponding to nodes or arcs),

For each arc RELATION1 adjoining SUBJECT1,

Identify OBJECT1 at the opposite end of RELATION1.

Collect all arcs RELATIONS2 adjoining SUBJECT2.

Collect all OBJECTS2 at the opposite ends of the RELATIONS2.

If OBJECT1 is in OBJECTS2, succeed.

Otherwise for each OBJECT2 in OBJECTS2 until success,

Call the network matcher on OBJECT1 and OBJECT2.

If no match, put RELATION1 on the unmatched relations list and put OBJECT1 on either the unmatched objects or unmatched relations list depending on whether OBJECT1 is a node or an arc.

---

The final step in matching relations is comparing values. Since relations have histories which describe how their values change, the times at which the comparison is to be made must be specified as well.

The following procedure compares values:

---

Value Matcher:

Given two matched relations RELN1 and RELN2, and two times T1 and T2,

If the value of RELN1 at T1 is the same as the value of RELN2 at T2, put the relations and values on the matched relations list, and the corresponding objects on the matched objects list, succeed.

Otherwise, put the relations and values on the unmatched relations list, and the corresponding objects on the unmatched objects list, fail.

---

The output of the relational network matcher is a set of lists showing what relations matched, what objects matched, and just as importantly, what relations and objects did not match. The unmatched relations and objects are mapped over in analogies and reveal differences which can form the basis of new hypotheses when generalizing.

## The Causal Rule Matcher

When the causal rule matcher is used to support rehypothesizing because the causal model failed, the question of what to compare is easy to answer. The causal rule which failed is compared to itself at different times.

When the causal rule matcher is used in an analogy, the results of selection – matched dependences from difference causal rules – are used to answer the question of what to compare. The results of selection tell where to "anchor" the comparison of causal rules.

The causal rule matcher uses the results of the relational network matcher. The following is the procedure for comparing causal rules:

---

Causal Rule Matcher:

Given two matched dependences from two causal rules,

Compare, using the relational network matcher, corresponding independent quantities, one from each dependence, at the times the causes occurred in the respective causal rules.

Compare, using the relational network matcher, corresponding dependent quantities, one from each dependence, at the times the effects occurred in the respective causal rules.

Isolate preconditions and effects from the list of matched and unmatched relations.

---

Since quantities and linked to physical objects by the QUANTITY-OF relation, the preconditions and effects, which are relations on these physical objects, also will be compared.