International Journal of Computer and Information Sciences, Vol. 2, No. 3, 1973

On the Problems of Time, Retrieval of Temporal Relations, Causality, and Coexistence[1]

Nicholas V. Findler[2] and David Chen[2,3]

Revised March 1973

Intelligent question-answering programs do more than retrieve "raw" data; they make deductive inferences in order to return all valid responses. They report logical inconsistencies, possibly at the data input phase. Similarly, more information is requested from the user if a question asked proves to be ambiguous. A question-answering system of the above type has been designed and implemented. Besides retrieving explicit and implicit temporal relations, the system discovers potentially causal relationships which also satisfy different time restrictions. Questions concerning a generalized concept of coexistence can also be answered. It is hoped that programs of a similar nature will become of much pragmatic use to researchers in physics, chemistry, biology, and so on, in evaluating complex, interrelated experimental data. Several additional applications for this type of program are mentioned, ranging from problems in criminology to air traffic control. The Associative Memory, Parallel Processing Language, AMPPL-II, was found rather satisfactory for the project. It is finally suggested that the system being described could serve as a component in a complex cognitive mechanism.

The work reported here was supported by the National Science Foundation Grant GJ-658.

[1] This paper, in a less complete form, was presented at the Second International Joint Conference on Artificial Intelligence, London, England, September 1971.

[2] Department of Computer Science, State University of New York at Buffalo, Amherst, New York.

[3] Both authors worked on the design of the system. The first author (N.V.F.) suggested and elaborated the subject matter and was responsible for the writing of this paper. Many refinements and improvements are due to the second author (D.C.), who has also done all the programming. An enlarged version of this work, in the form of a Master's Project, will be submitted to the State University of New York at Buffalo in partial fulfilment of the requirements of David Chen's M.S. degree.

828/2/3-1

1. INTRODUCTION

There is an overwhelming abundance of complex events in "real" life. One might look upon science as a mechanism for helping man escape from unexplained chaos. The basic assumption of the reductivist is inherent in all scientific investigations—the existence of (simple) laws that govern natural phenomena.

In the interpretation of scientific data, working hypotheses are formed that are based on *prima facie* relations between patterns of events. The belief in causality calls for the testing of these working hypotheses under a wide variety of conditions. The logic of the concept of causality requires that the scientist should, first of all, sort out predecessor–successor relations.[4] It seems obvious that, when huge masses of time-relevant data are to be analyzed, the computer should come to the scientist's aid in this nontrivial task. Yet, the passage of time, one of the most salient of human experiences, has received relatively little study in question-answering programs. The time variable is at the heart of practically all physical, biological, and psychological events. It is therefore the fundamental variable in every "process-descriptive" model.[5]

The present work is aimed at all possible temporal relations between time-dependent events—an expandable but well-defined universe. The framework of a categorical structure is built during the input phase. As more and more questions are asked, this framework is filled up with directly given and inferred data and relations between them. This fact and many other aspects of the project are in analogy with human cognition, although no attempt was made to simulate the latter. Inconsistencies and missing information are discovered and reported back to the user. (Missing information is manifested, in an indirect manner, by returning a larger set of answers.) The objectives of this work were briefly described in Ref. 1.

Finally, we note that we do not wish to enter the realm of philosophy in this work. The meaning and the role of time, in the abstract sense, are beyond the scope of these investigations.

[4] Certain paradoxical results of the theory of relativity, which would be of no concern to us here, render it possible to send signals into the past. See, for example, Terletskiy.[14]

[5] According to the usual distinction made by systems theory (see, for example, Simon[4]), systems can often be described by two types of models. The state-descriptive model tells how a system behaves now, whereas the process-descriptive model provides information as to how and through what stages it has developed from an initial configuration to the current one. The latter is usually more valuable, not only because it enables the researcher to extrapolate to the future, but also because it gives insight into the mechanisms operating within the system.

2. ON TIME-DEPENDENT EVENTS AND TEMPORAL RELATIONS

We shall not be concerned here with stochastic phenomena. (As Einstein said, "The dear God does not play with dice.") The blurring effect of probability distributions is replaced by the following paradigm:

$$\text{Event } E_i \text{ has } \begin{bmatrix} \text{relevant} \\ \text{irrelevant} \end{bmatrix} \begin{bmatrix} \text{starting time} \\ \text{duration} \\ \text{finishing time} \end{bmatrix}$$

where members of the second bracket may assume either attribute of the first bracket. Meaningful examples can be found for each of the eight ($= 2^3$) possible cases. The distinction between relevancy and irrelevancy is indeed useful since chronological data in real life are often incompletely defined because of a lack of information, measurement errors, conflicting data sources, etc.

Further, partial specification of chronological data can be given in the "irrelevant" case by saying.

$$\text{Event } E_i \text{ occurred } \begin{bmatrix} \text{fully} \\ \text{partially} \end{bmatrix} \begin{bmatrix} \text{before} \\ \text{during} \\ \text{after} \end{bmatrix} \text{ event } E_j$$

Let us now define two types of events.

(1) A *point event* takes place momentarily. However, its effect, which is not necessarily contiguous to the event, may be lasting. For example, "to wake up" is a point event (although some people may disagree with this) and its effect "to be awake" is lasting. Or, "John starts reading" is, again, a point event which is followed by a lasting action. It is more difficult to find a point event with no subsequent effect. A moot one is, for example, "He decided not to do anything."

(2) A *duration event* has distinct starting and finishing times, and a duration between the two. There are a few exceptions. Explicit or implicit truth statements often do not have starting and finishing times. Unless one accepts religious teachings or the "big bang" theory of cosmology, the "world" has no starting time.

In order to be realistic about simultaneity, we have assumed a quantized time scale. In the computer program described later 1 min is the shortest measurable time interval. (This represents no restriction because a definable, arbitrary time unit is also available; see later.) Further, we have considered the time coordinate unbounded in both directions.

We adopt in this section the convention that lower case letters refer to time points and upper case letters to time intervals. The letter e will have a special significance: e will denote a point event and E a duration event.

Suppose e_1 takes place at t_1 and e_2 at t_2. We can then say

$$e_1 \text{ occurs} \begin{bmatrix} \text{before} \\ \text{simultaneously with} \\ \text{at about the same time as} \\ \text{after} \end{bmatrix} \mathbf{e}_2 \quad \text{iff} \quad \begin{bmatrix} t_1 < t_2 \\ t_1 = t_2 \\ t_1 \cong t_2 \\ t_1 > t_2 \end{bmatrix}$$

Suppose both e_1 and e_2 occurred during T, that is, $t_1, t_2 \in T$. Then e_3, related to t_3, must have occurred during T if (weak condition)

$$t_1 \leqslant t_3 \leqslant t_2 \qquad \text{or} \qquad t_1 \geqslant t_3 \geqslant t_2$$

Also T_1 precedes T_2 iff

$$t_1 < t_2 \quad \text{for all} \quad t_1 \in T_1 \quad \text{and all} \quad t_2 \in T_2$$

When T_1 is neither before T_2 nor after T_2, there is a partial or full overlap between them. Let s denote starting times, f finishing times, and D durations. We then have with these events

$$s_i + D_i = f_i$$

for $i = 1$ and 2.

Partial overlap is found between E_1 and E_2 iff

$$s_i < s_j < f_i \qquad \text{and} \qquad f_i < f_j$$

for $(i, j) = (1, 2)$ or $(2, 1)$. (Some of the $<$ signs could be replaced selectively by $\leqslant$ or $\cong$ but we shall ignore this now for the sake of notational simplicity.)

Full overlap occurs iff

$$s_i < s_j < f_j < f_i$$

for $(i, j) = (1, 2)$ or $(2, 1)$.

The two events are *contiguous* iff

$$s_i < f_i = s_j < f_j$$

for $(i, j) = (1, 2)$ or $(2, 1)$.

Finally, E_1 and E_2 are *disjoint* iff

$$f_i < s_j$$

for $(i, j) = (1, 2)$ or $(2, 1)$.

In the above definitions, all parameters appear to be time-relevant. In the case of approximate time specifications, whether absolute or relative to other events, the system must make proper allowances for the changes in the retrieval logic.

One can establish a simple logical system on the basis of the previous definitions, using standard first-order predicate calculus plus axioms with regard to transitivity, irreflexivity, and existence of predecessors and successors. This enables the system to make simplifications on and inferences concerning temporal relations between any two of three or more events.

3. THE PROGRAM

It has been our intention to allow a most general input mode. The data to be used should be checked for consistency at a stage as early as possible. At the beginning of the program, the skeleton of a complex, hierarchical, and relational data structure is established. This would then be filled in as need arises through the questions asked. We did not frown upon redundancy in stored information whenever it shortened or simplified processing.

Many difficult decisions had to be made in regard to input mode, internal representation, data manipulation, search processes, etc. Several times we had to give up a certain avenue and start on another one. The following represents the current state of affairs, which we may improve and expand yet.

The program is divided into three major components, the input phase, the data-structuring phase, and the question-answering phase. These will now be discussed briefly.

(1) The *input phase* is concerned with three types of information:

(a) The user defines the *symbolic components* of an event in the form

(EVENT, EVTNAM, STRTIM, FINTIM, LENGTH)

where EVENT is a reserved word to denote symbolic specification; and EVTNAM, STRTIM, FINTIM, LENGTH are symbolic names of the event, its starting and finishing time, and its duration, respectively.

(b) The user also sets up *temporal relations* between symbolically defined time quantities by the predicates

(ANTE, $T1$, $T2$, $T3$,..., Tn)
(EQUI, $T1$, $T2$, $T3$,..., Tn)
(CEOUI, $T1$, $T2$, $T3$,..., Tn)

Here the first statement means that $T1$ is the *ante*cedent of $T2$, which is the antecedent of *T*3, etc.; the second one means that *T*1,*T*2, *T*3,..., are *equi*temporal; and the third one that *T*1, *T*2, *T*3,..., are *c*irca-*equi*temporal. (The *T*'s refer only to time points in case of ANTE.)

(c) Finally, the input phase *binds* the *time parameters* of the events, already symbolically specified, to numerical values. Several possibilities exist here. We can either use "natural" units (for a time point, for example, 1970–DECEMBER–31–11 o'clock–10 minutes, written in the proper format), or "defined" units the zero point and length of which are arbitrarily set by the user to suit, say, particular experimental conditions (a time period, for example, 870 units long). In one body of information the two cannot be mixed.

Further, whether natural or defined units are used, a time quantity can be numerically specified either directly, in absolute terms, or indirectly, i.e., relative to another time quantity. Verbalized examples of this are: "Event *A* started 15 units before time point *T*8,' or "Event *C* is 42 days longer than interval *D*5.'

Some comments are needed here. Since the number of days in a year (note leap years) and the number of days in a month are not constant, whenever a time interval is to be directly specified in combined units (say, years, months, days, hours, minutes) the user must define a dummy event with starting and finishing times so determined that its duration is the right one.

There are eight modes of indirect specification, as indicated by an input parameter. The relative time length that has to be added to or subtracted from another time quantity can be given in minutes, hours, days, months, years, packed year-month-day-hour-minute mode, packed month-day mode, and packed day-minute mode.

In the last three cases, the addition or subtraction starts at the rightmost unit and proceeds to the left, for the sake of unambiguity. Proper care has been taken of the transition between the Julian and Gregorian calendars, the "missing" year between 1 B.C. and 1 A.D., and similar idiosyncrasies.

There is an important, multipurpose, reserved word, *DELTA*. In the input phase it denotes an irrelevant time quantity, for which no numerical specification is to be expected. We would put for example, for time points *TA* and *TB*

(EQUI, DELTA, TA, TB)

saying that they both are irrelevant (but *not* equitemporal!) *DELTA* can also be put directly, in lieu of any symbolic name of an event being asked about. Examples to follow will make this point clear.

Nested specifications are also possible. A verbalized example of this is, "Event *A* started 20 min before the event ended that started at 8 a.m."

The first stage of checking for inconsistencies is done as soon as possible during the input phase. Contradictions may occur within or between the symbolically and numerically given data. If found, these are reported back to the user and the program aborts.

Finally, if one of the three time quantities, characterizing a duration event, starting time, finishing time and length, is missing, it is computed and recorded at this stage. In the cases in which partial chronological information is given in terms of restrictions, longest or shortest possible time lengths are computed and appropriately marked whenever possible.

(2) The *data-structuring phase* transforms the input information into an expandable format, which is particularly suitable for the last phase, during which questions are asked and answered.

Some of the SLIP list structures established initially are expanded, some others are transcribed into a sequence of content-addressable and parallel-processable AMPPL-II Relations (see Ref. 2 or 3) and then destroved. Events are characterized, individually and collectively, in reference to each other. Numerical and symbolic data are used to generate inferences and, again, inconsistencies are discovered and reported back to the user.

(3) The *question-answering phase* consists of two parts. The first one is the ANALYZER. It dissects each question and expands it into its elementary components. The ANALYZER is essentially a recursive procedure in view of the fact that complex, nested questions can also be asked. The second part is an executive routine that calls in special functions for each question component. These functions can be divided into five groups:

(a) To *retrieve the symbolic names* of the starting time and finishing time and length of a given event.

(b) To establish whether a particular *relation* between two time quantities (points or durations) or two events *is true*. The possible relations are as follows. (i) between two time points: one time point is before, at the same time as, or at about the same time as the other one; (ii) between two time durations: one time duration is longer than, of equal length as, of about equal length as, or shorter than the other one; (iii) between a time point and an event: the time point is exactly or about at the same time as the event's starting, finishing point; it occurred before, during, or after the event; (iv) between a duration and an event: the duration is longer than, of equal length as, or of about equal length as, or shorter than the event; (v) between two events: one event is partially, completely antecedent to the other; it coexists with (i.e., partially or completely overlaps), covers the other; it is contiguous to, identical to, about identical to, longer than, of equal length as, of about equal length as, or shorter than the other one.

If one of the time quantities or events is specified as *DELTA*, then all entities, either on a list specified by the user or in the universe of the system, are searched for those that satisfy the given relation. On the other hand, if a relation is specified as *DELTA*, then all valid relations between the given entities are retrieved.

(c) To find the *extremum* (extrema) of certain conditions. The latter can be: earliest, latest; longest, shortest; closest.

The first two refer to the possible range of (irrelevant) time points. The next two refer to time durations or events with one or two irrelevant limiting end points. The last condition, "closest," may refer to time points, in which case there is no ambiguity. However, the user may also look for an event (or events) whose length is the closest to a given duration. Or else, he may specify one event and look for another one that has the maximum length of overlap with the given event. If there are several, which, for example, completely cover the latter, the system can select the shortest event of these.

In general, all potentially acceptable answers are returned when the information available is not sufficient to decide on their acceptability.

(d) To establish *chains of events* where the mode of chaining must satisfy certain restrictions, such as (i) the successive events must be contiguous, i.e., the finishing time of the antecedent is the starting time of the successor; (ii) the successive events must partially or completely overlap each other; (iii) either the starting or the finishing times, specifiable by parameters, must be consecutive with successive events; (iv) the same as above but certain time durations at least must elapse between consecutive starting or finishing times (cf. problems of causality).

(e) To answer questions concerning a *generalized concept of coexistence*. There are, say, three sets of events given. The program will find a subset of the first set, each member of which must coexist (partial or complete overlap) with at least one member of the second set of events and must not coexist (disjoint events) with any member of the third set. Or, more generally, we can look for a subset of the first set, each member of which must coexist with at least one member of an arbitrary number of sets and must not coexist with *any* single member of another arbitrary number of other sets of events.

Finally, we note two additional types of side conditions that can be specified for the results. These are (1) the *logical conditions*: for *all* on a list specified or anywhere in the system (cf. the universal quantifier); find *one* (cf. the existential quantifier); Boolean AND, OR, and NOT of sets of events; and (2) the *time conditions*: the resulting time quantity or event must occur before or after a certain time point; before, during, or after an event; within a certain time length.

4. EXPERIENCE WITH THE PROGRAM

Programming was done on a CDC6400, with which SLIP and AMPPL-II have been resident on the system disk. Only those subprograms are called into the core that are needed in the program. The total code occupies less than 15k core memory. The running time, of course, depends on the size of the data base and on the complexity (levels of nesting) of the questions asked.

With a moderately sized data base and questions of the type below, an answer was obtained on the average in a few hundredths of a second if the information was directly available. However, if logical inferences concerning nested questions had to be made and the data structure had to be updated, the time required for an answer may take as long as 2.5 sec (see later).

To illustrate the notation to be employed by the user, we give a list of simple questions and a few nested ones. We note that the same question can often be formulated in several different ways.

(1) Could the start of event E_1 cause event E_2?

(RELTRUE, BEFØRE, P, (STARTIM, E1), E, E2)

Here P means that the starting time of event E_1 is a time point and E means that E_2 is an event.

(2) Does event E_1 coexist (partial or complete overlap) with event E_2 ?

(RELTRUE, CØEXIST, E, E1, E, E2)

(3) Could the completion of event E_1 immediately cause event E_2?

(RELTRUE, CØNTIGUE, E, E1, E, E2)

(4) Is event E_1 longer than event E_2?

(RELTRUE, LØNGER, E, $E1$, E, $E2$)

(5) Which events are such that (a) their start could cause event E_1?

(RELTRUE, BEFØRE, P, (STARTIM, DELTA), E, E1)

(b) they coexist with event E_1?

(RELTRUE, CØEXIST, E, DELTA, E, E1)

(c) their completion could cause event E_1?

(RELTRUE, BEFØRE, P, (FINTIM, DELTA), E, E1)

(d) their duration is longer than that of event E_1?

(RELTRUE, LØNGER, E, DELTA, E, E1)

(6) Are there sequences of events [(a) only consecutive, (b) only

contiguous, or (c) also overlapping] that lead to event E_1, and if so, which ones? (Note chain of causally connected events.):

(a) (CHAIN, CØNSECUTIVE, FINTIM, STARTIM, E1)
(b) (CHAIN, CØNTIGUE, E1)
(c) (CHAIN, ØVERLAP, E1)

(7) Given three sets of events

$$\{E_1\}: \{E_{11}, E_{12}, \ldots, E_{1m}\}$$
$$\{E_2\}: \{E_{21}, E_{22}, \ldots, E_{2n}\}$$
$$\{E_3\}: \{E_{31}, E_{32}, \ldots, E_{3p}\}$$

find a subset $\{E_4\}$ of $\{E_1\}$ such that each of its members coexists (partial or complete overlap) with at least one member of $\{E_2\}$ and does not coexist with any member of $\{E_3\}$. Notationally, $\{E_4\} := \{E_1\} \oplus \{E_2\} \neg \oplus E_3$:

(CØEXISTENC, (E11, E12,...), (E21, E22,...), (BUTNØT, E31, E32,...)

or

((E1, E11, E12,...), (E2, E21, E22,...), (E3, E31, E32,...),
(CØEXISTENCE, E1, E2, (BUTNØT, E3)))

(8) From among a set of events $\{E_1\}$: $\{E_{11}, E_{12}, \ldots\}$ which events last shortest and longest?

(EXTREMUM, E, LØNGEST, (E11, E12,...))

and

(EXTREMUM, E, SHØRTEST, (E11, E12,...))

(9) From among a set of events $\{E_1\}$: $\{E_{11}, E_{12}, \ldots\}$ which event starts or finishes earliest or latest?

(EXTREMUM, P, EARLIEST, ((STARTIM, E11), (STARTIM, E12),...))
(EXTREMUM, P, LATEST, ((STARTIM, E11), (STARTIM, E12),...))
(EXTREMUM, P, EARLIEST, ((FINTIM, E11), (FINTIM, E12),...))
(EXTREMUM, P, LATEST, ((FINTIM, E11), (FINTIM, E12),...))

(10) From among a set of events $\{E_1\}$: $\{E_{11}, E_{12}, \ldots\}$ which event lasts closest to a time interval?

(EXTREMUM, E, CLØSEST, D, R, (E11, E12,...))

where R is the name of a certain time duration and D indicates duration.

(11) We can restrict potential causal relationships between two events E_1 and E_2 by saying that E_2 must start or finish at least t_{12} time units after E_1 starts or finishes. In general, the user specifies during the input phase a matrix of restrictions the elements of which indicate the minimum time

durations that must elapse between starting or finishing times of potentially causally linked events. One or several predicates are first given of the format

(RESTRICTED, (TM1, TM2)(E1, E2,..., Ek))

where RESTRICTED is a reserved word, and TM1 and TM2 are either STARTIM or FINTIM. The predicate may be nested in an input list to arbitrary depth. The value t_{ij} indicates that TM2 of E_j can occur at least t_{ij} time units after TM1 of E_i.

Immediately following the predicate(s), the user must input the numerical values of t_{ij}, each adjoining another number, m_{ij}. The latter describe the modes of t_{ij}:

$m_{ij} = 98$ means that t_{ij} is irrelevant
$m_{ij} = 99$ is the end of the restriction matrix
$m_{ij} = -1$ denotes that t_{ij} is in "defined" units
m_{ij} having other values signifies "natural" time units

If t_{ij} is negative, the roles of TM1 and TM2 are interchanged. Further, for each relevant t_{ij} concerning E_i and E_j, the system generates a unique dummy event D_{ij} of length t_{ij} with temporal relations such that the starting time of D_{ij} coincides with TM1 of E_i and the finishing time of D_{ij} coincides with TM2 of E_j. These internally generated dummy events get names starting with $$.

The example given later will clarify this point completely.

(12) Which events take place between the end of event E_1 and the start of event E_2?

((INTSECTIØN(RELTRUE, ANTE, P, (FINTIM, E1), P, (STARTIM), DELTA)), (RELTRUE, AFTER, P, (STARTIM, E2), E, DELTA)))

or

((RELTRUE, BEFØRE, P, (FINTIM, E1), E, DELTA, (RELTRUE, ANTE, P, (FINTIM, DELTA), P, (STARTIM, E2))))

(13) Which is the longest event that lies between the finishing time of event E_1 and the starting time of event E_2?

(EXTREMUM, E, LØNGEST, (RELTRUE, BEFØRE, P, (FINTIM, E1), E, DELTA, (RELTRUE, BEFØRE, P, (FINTIM, DELTA, P, (STARTIM, E2)))).

(14) Suppose there are two sets of events, members of which coexist with events E_1 and E_2, respectively. Find the temporal relation between the finishing time of the latest event (i.e., the one that ends the latest) in the first

set and the starting time of the earliest event (i.e., the one that starts the earliest) in the second set:

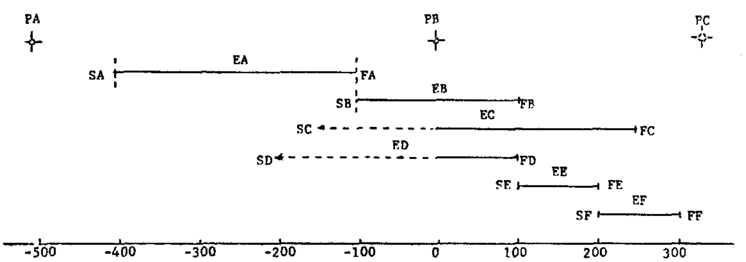
(RELTRUE, DELTA, P, (FINTIM, (EXTREMUM, E, LATEST, (RELTRUE, CØEXIST, E, DELTA, E, E1))), P, (STARTIM, (EXTREMUM, E, EARLIEST, (RELTRUE, CØEXIST, E, DELTA, E, E2))))

A few more points are to be noted here. The set-theoretic functions INTSECTIØN and UNIØN can be nested in any of the question forms. The function (VALUE, TA, TB,...) retrieves numerical values of an indefinite number of time points and/or durations. The symbolic names of these may be referenced indirectly in a nested form. For example, if we wish to find the durations of events *T*1 and *T*2, and the times of occurrence of those point events that happened before point event *P*1, we put

(VALUE, T1, T2, (RELTRUE, ANTE, P, DELTA, P, P1))

Whenever exact values are not available, a set of routines is called upon to compute the *possible time range* of the time points or durations involved.

The following diagram and excerpts from a computer output represent a run with a small data base:



Event names, names of starting and finishing times, are indicated. There are three point events, PA, PB, and PC, and six duration events, EA, EB, EC, ED, EE, and EF. The following symbolic and numerical specifications are to be given to the system:

ANTE: (PA, SA); (SD, FA, SB, PB); (FE, PC); (SD, SC, FA)
EQUI: (FB, FD, SE); (FE, SF)
CEQUI: (FA, SB)

PA=−500, PB=0, FB=100*, FC=250, FD=100*, SE=100, FE=200, LE=100*, SF=200*, FF=300*, LF=100

(Those marked with an asterisk are values computed by the system.)

```
SIX EVENTS AND THREE TIME POINTS ARE INVOLVED IN THE FOLLOWING EXAMPLE.
THE QUESTIONS ARE NOT NECESSARILY MEANINGFUL.  THEY ARE USED TO DEMONSTRATE
WHAT THE SYSTEM CAN DO.

INPUT FACTS IN LIST FORM :

  (()(EVENT,EA,SA,FA,LA)   (EVENT,EB,SB,FB,LB)   (EVENT,EC,SC,FC,LC)
     (EVENT,ED,SD,FD,LD)   (EVENT,EE,SE,FE,LE)   (EVENT,EF,SF,FF,LF)    )*
```

```
 WE WANT TO INPUT THE LEAST AMOUNT OF INFORMATION ONLY.  IN ORDER TO SEE
HOW THE SYSTEM COMBINES ABSTRACT (SYMBOLIC) AND NUMERIC INFORMATION, SOME
OF THE ENTITIES WILL BE GIVEN NUMERICAL VALUES.

 WE DEFINE 3 TIME POINTS PA, PB AND PC FOR REFERENCE.  PA ANTECEDES THE
STARTING TIME OF EVENT EA.  TIME POINT SD ANTECEDES FA THAT ANTECEDES SB
THAT ANTECEDES PB.  THE FINISHING TIME OF EVENT EF OCCURS BEFORE PC.  FB,
FD AND SE ARE EQUITEMPORAL TIME POINTS.  FA AND SB ARE CIRCA-EQUITEMPORAL.
FE IS EQUITEMPORAL TO SF.  TIME POINTS SD, SC AND FA ARE IN THIS CHRONOLO-
GICAL ORDER.  SE, FE, FC, LF, PA, AND FB WILL BE GIVEN NUMERICAL VALUES.

INPUT FACTS IN LIST FORM :

     ((PA,PB,PC)(ANTE,PA(STARTIM,EA))(ANTE,SD,FA,SB,PB)(ANTE,(FINTIM,EF)PC)
        (EQUI,FB,FD,SE) (CEQUI,FA,SB) (EQUI,FE,SF) (ANTE,SD,SC,FA)      )*
```

(The SLIP system requires an asterisk at the end of every list structure to be inputted.)

RECOGNIZED EVENTS ARE :

EVENT	STARTIM	FINTIM	LENGTH
EA	SA	FA	LA
EB	SB	FB	LB
EC	SC	FC	LC
ED	SD	FD	LD
EE	SE	FE	LE
EF	SF	FF	LF

Following this, the system prints out five list structures: ANTE (precedence ordering), LSTEQ (equitemporal), LSTCEQ (circa-equitemporal), TPL (recognized time points), and TDL (recognized time durations). For the sake of brevity these are omitted here. The relevant temporal relations are then transferred to the Simulated Associative Memory.

NUMERIC VALUE ASSIGNMENTS

INPUT DATA			COMPUTED VALUES	
			EVENT	VALUE
PA	R PB	-500		
SE		100	SE	100
FE		200	FE	200
FC	R PA	750		
SF	R SF	0		
LF		100	LF	100
PB		0	PB	0
			PA	-500
			FC	250
			FB	100
			FD	100
			SF	200
			LE	100
			FF	300

The above is partly the echo of the input, partly values inferred by the system. Next a chronological ordering of time points (list NAL) is printed but this is omitted here.

```
           FIND POSSIBLE CHAINS OF NON-OVERLAPPING AND CONSECUTIVE EVENTS THAT LEAD
          TO EVENT EF.

QUESTION IN LIST FORM :

           (  (CHAIN,CONSECUTIV. FINTIM,STARTIM,EF)  )*

A POSSIBLE CHAIN IS :
EA, EF.

A POSSIBLE CHAIN IS :
EA, EB, EF.

A POSSIBLE CHAIN IS :
ED, EF.

NO MORE CHAIN

THIS QUESTION USED   1.9700 SECONDS
```

```
          FIND POSSIBLE CHAINS OF EVERLAPPING EVENTS THAT LEAD TO EVENT EF.

QUESTION IN LIST FORM :

           (    (CHAIN,OVERLAP,EF)    )*

A POSSIBLE CHAIN IS :
ED, EC, EF.

NO MORE CHAIN

THIS QUESTION USED    .8920 SECONDS
```

```
           FIND POSSIBLE CHAINS OF CONTIGUOUS EVENTS THAT LEAD TO EF.

QUESTION IN LIST FORM :

           (     (CHAIN,CONTIGUE,EF)    )*

A POSSIBLE CHAIN IS :
EB, EE, EF.

A POSSIBLE CHAIN IS :
ED, EE, EF.

NO MORE CHAIN

THIS QUESTION USED   2.0080 SECONDS
```

```
          LIST ALL TIME POINTS THAT ANTECEDES PC AND ARE ANTECEDED BY PA.

QUESTION IN LIST FORM :

           (  (RELTRUE,ANTE,P,DELTA,P,PC,(RELTRUE,ANTE,P,PA,P,DELTA)  )    )*

THE ANSWER IS :
PB, FF, SF, FE, SE, FD, FC, FB, SB, FA, SA.

THIS QUESTION USED   2.4840 SECONDS
```

```
   FROM AMONG EVENTS EC, EB, EE, ED, EA AND FF, SELECT THOSE THAT COEXIST
WITH EB OR EC.  FROM THIS GROUP, DELETE THE EARLIEST ONE OF THOSE THAT ARE
COMPLETELY ANTECEDED BY EVENT EA.

QUESTION IN LIST FORM :

   (  (NOT (COEXISTENC (EC,EB,FE,ED,EA,EF) (FB,EC))(EXTREMUM,EARLIEST,E,
      (RELTRUE,COMPANTE,E,EA,E,DELTA)  )  ))*

THE ANSWER IS :
EF, EA, ED, EE, EC.

THIS QUESTION USED   2.1260 SECONDS



   FIND THOSE EVENTS THAT START AFTER STARTING TIME OF EA AND WHOSE
FINISHING TIME IS COVERED BY EVENT EC.

QUESTION IN LIST FORM :

   ( (INTSECTION (RELTRUE,BEFORE,P(STARTIM,EA)E,DELTA)(RELTRUE,BETWEEN,
      P(FINTIM,DELTA),E,EC)   )   )*

THE ANSWER IS :
EE, EB.

THIS QUESTION USED    .8940 SECONDS



   FIND THE LONGEST AND THE SHORTEST EVENTS OF EB,EC,EE,EF AND ED.

QUESTION IN LIST FORM :

   ( (EX,EB,EC,EE,EF,ED)(UNION  (EXTREMUM,LONGEST,E,EX)(EXTREMUM,SHORTEST,E,
      EX)  ))*

THE ANSWER IS :
EF, EE, ED, EC.

THIS QUESTION USED    .6000 SECONDS



   FIND VALUES OF TIME POINT PA, OF THOSE TIME POINTS THAT OCCUR AFTER PA,
AND OF THOSE TIME DURATIONS THAT ARE EQUAL TO OR LONGER THAN LE (THE LENGTH
OF EVENT EE).

QUESTION IN LIST FORM :

   (  (VALUE,PA,(RELTRUE,ANTE,P,PA,P,DELTA)  (RELTRUE,EQUAL,D,LE,D,DELTA)
      (RELTRUE,LONGER,D,DELTA,D,(LENTH,EE)  )  )  )*

ANSWERS ARE :
PA HAS VALUE          -500
SA HAS RANGE OF VALUE FROM        -500 TO          0
FA HAS RANGE OF VALUE FROM        -500 TO          0
SB HAS RANGE OF VALUE FROM        -500 TO          0
FB HAS VALUE           100
THE RANGE OF VALUE OF SC IS FROM -INFINITY TO      0.
FC HAS VALUE           250
THE RANGE OF VALUE OF SD IS FROM -INFINITY TO      0.
FD HAS VALUE           100
SE HAS VALUE           100
FE HAS VALUE           200
SF HAS VALUE           200
FF HAS VALUE           300
PB HAS VALUE             0
THE RANGE OF VALUE OF PC IS FROM         300 TO INFINITY.
LE HAS VALUE           100
LF HAS VALUE           100
LB HAS RANGE OF VALUE FROM        100 TO          600
THE RANGE OF VALUE OF LC IS FROM         250 TO INFINITY.
THE RANGE OF VALUE OF LD IS FROM         100 TO INFINITY.

THIS QUESTION USED    .9860 SECONDS
```

```
FROM AMONG EVENTS EA, EC, ED, EE AND EF, SELECT THOSE THAT CO-EXIST WITH
EVENTS ED OR EC AND DO NOT CO-EXIST WITH EVENT EB.

QUESTION IN LIST FORM :

( (COEXISTENC (EA,EC,ED,EE,EF) (ED,EC) (BUTNOT,EB) ) )*

THE ANSWER IS :
EF, EE, EC, EA.

THIS QUESTION USED   1.8560 SECONDS



THIS EXAMPLE DEMONSTRATE SOME FURTHER USE OF THE FUNCTIONS EXTREMUM AND
UNION.

QUESTION IN LIST FORM :

( (UNION (EXTREMUM,LONGEST,D,(RELTRUE,LONGER,D,DELTA,E,EF) ),
   (EXTREMUM,EARLIEST,P,(SA,(STARTIM,EC) (FINTIM,EF),SD,SB) ) (EA,EF,EB) ) )*

THE ANSWER IS :
EB, EF, EA, SA, LD, LC, LB.

THIS QUESTION USED   .3360 SECONDS


A PLEASURE WORKING FOR YOU, BYE.
```

To illustrate date-time specification, a data base similar to the one in the first example was used with different values. We have, for instance, SE = 1960, 01 01 (January 1, 1960), FE = 1960 12 31; LE = 365 (computed; 1960 leap year); SF = 1960 12 31, FF = 1971 5 1; LF = 3772 days (eight regular and two leap years, the five first months in 1971).

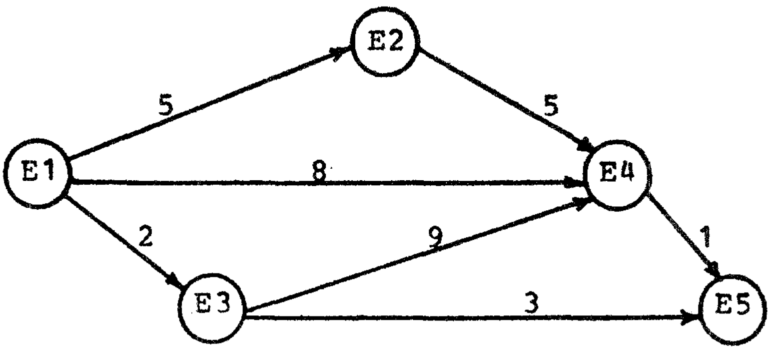
NUMERIC VALUE ASSIGNMENTS

INPUT DATA			COMPUTED VALUES	
			EVENT	VALUE
PA	194001010000	8	PA	194001010000
PB	195001010000	8	PB	195001010000
FC	197001010000	8	FC	197001010000
SE	196001010000	8	SE	196001010000
FE	196012310000	8	FE	196012310000
FF	197105010000	8	FF	197105010000
			FB	196001010000
			FD	196001010000
			SF	196012310000
			LE	3650000
			LF	37720000

```
FIND VALUES OF TIME POINT PA, OF THOSE TIME POINTS THAT OCCURS AFTER PA,
AND OF THOSE TIME DURATIONS THAT ARE EQUAL TO OR LONGER THAN LE (THE
LENGTH OF EVENT EE).
```

```
QUESTION IN LIST FORM :

            (  (VALUE,PA,(RELTRUE,ANTE,P,PA,P,DELTA) (RELTRUE,EQUAL,D,LE,D,DELTA)
                 (RELTRUE,LONGER,D,DELTA,D,(LENTH,EE)   )   )   )*

ANSWERS ARE :
PA HAS VALUE    194001010000
SA HAS RANGE OF VALUE FROM    194001010000 TO    195001010000
FA HAS RANGE OF VALUE FROM    194001010000 TO    195001010000
SB HAS RANGE OF VALUE FROM    194001010000 TO    195001010000
FB HAS VALUE    196001010000
FC HAS VALUE    197001010000
FD HAS VALUE    196001010000
SE HAS VALUE    196001010000
FE HAS VALUE    196012310000
SF HAS VALUE    196012310000
FF HAS VALUE    197105010000
PB HAS VALUE    195001010000
THE RANGE OF VALUE OF PC IS FROM    197105010000 TO INFINITY.
LE HAS VALUE         3650000
LB HAS RANGE OF VALUE FROM        36530000 TO        73050000
THE RANGE OF VALUE OF LC IS FROM        73050000 TO INFINITY.
THE RANGE OF VALUE OF LD IS FROM        36530000 TO INFINITY.
LF HAS VALUE        37720000

THIS QUESTION USED    1.3740 SECONDS

 A PLEASURE WORKING FOR YOU,  BYE.
```

As an example of using the Restriction Matrix concept, let us consider the following network of events:



For the sake of simplicity here, each event takes one time unit and the number above the edge connecting E_i and E_j represents the minimum number of time units that must elapse between the FINTIM of E_i and the STARTIM of E_j . Therefore the restriction matrix looks like

		S	T	A	R	T	I	M
F		E1	E2	E3	E4	E5		
I	E1	X	5	2	8	*		
N	E2	*	X	*	5	*		
T	E3	*	*	X	9	3		
I	E4	*	*	*	X	1		
M	E5	*	*	*	*	X		

where * indicates irrelevant entry and X is a forbidden entry. The rest is obvious from the following printout.

```
WE DEFINE 5 EVENTS E1, E2, E3, E4 AND E5.  FOR LATER DEMONSTRATION OF
THE RETRIEVAL OF ALL POTENTIALLY CAUSAL CHAINS LEADING TO EVENT E5, EACH
EVENT WILL BE GIVEN A DURATION OF 1 TIME UNIT.

INPUT FACTS IN LIST FORM :

(  ()(EVENT,E1,S1,F1,L1) (EVENT,E2,S2,F2,L2) (EVENT,E3,S3,F3,L3)
       (EVENT,E4,S4,F4,L4) (EVENT,E5,S5,F5,L5)       )*



NOW WE SPECIFY THAT RESTRICTED POTENTIALLY CAUSAL RELATIONSHIPS EXIST.
THIS IS FOLLOWED BY A MATRIX OF TIME RESTRICTIONS.

INPUT FACTS IN LIST FORM :

(      () (RESTRICTED, (FINTIM,STARTIM) (E1,E2,E3,E4,E5)))*
```

TEMPORAL RESTRICTION MATRIX ELEMENTS

T(I,1)	M(I,1)	T(I,2)	M(I,2)	T(I,3)	M(I,3)	T(I,4)	M(I,4)	T(I,5)	M(I,5)
X		5	1	2	1	8	1	*	98
*	98	X		*	98	5	1	*	98
*	98	*	98	X		9	1	3	1
*	98	*	98	*	98	X		1	1
*	99	*	*	*	*	*	*	X	

X INDICATES THAT T(I,I) SPECIFICATION IS NOT ALLOWED

* INDICATES IRRELEVANT ENTRY

RECOGNIZED EVENTS ARE :

EVENT		STARTIM		FINTIM		LENGTH	
E1		S1		F1		L1	
E2		S2		F2		L2	
E3		S3		F3		L3	
E4		S4		F4		L4	
E5		S5		F5		L5	
$$E	A	$$S	A	$$F	A	$$L	A
$$E	B	$$S	B	$$F	B	$$L	B
$$E	C	$$S	C	$$F	C	$$L	C
$$E	D	$$S	D	$$F	D	$$L	D
$$E	E	$$S	E	$$F	E	$$L	E
$$E	F	$$S	F	$$F	F	$$L	F
$$E	G	$$S	G	$$F	G	$$L	G

NUMERIC VALUE ASSIGNMENTS

INPUT DATA		COMPUTED VALUES		
		EVENT		VALUE
		$$L	A	5
		$$L	B	2
		$$L	C	8
		$$L	D	5
		$$L	E	9
		$$L	F	3
		$$L	G	1
S1	197001010000 6	S1		197001010000
L1	1 1	L1		1
L2	1 1	L2		1
L3	1 1	L3		1
L4	1 1	L4		1
L5	1 1	L5		1
		F1		197001010001

```
    WE SHALL RETRIEVE ALL POTENTIALLY CAUSAL CHAINS LEADING TO EVENT E5.
  THESE CHAINS MAY INCLUDE DUMMY EVENTS STARTING WITH $$E INDICATING THE
  EXISTENCE OF RESTRICTED RELATIONSHIP(S).

QUESTION IN LIST FORM :

           (    (CHAIN,CONSECUTIV,FINTIM,STARTIM,E5)    )*

A POSSIBLE CHAIN IS :
E1         , E5         .

A POSSIBLE CHAIN IS :
E1         , E2          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, E2          , E5          .

A POSSIBLE CHAIN IS :
E1         , E3          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E3          , E5          .

A POSSIBLE CHAIN IS :
E1         , E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , E2          , E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, E2          , E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , E3          , E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E3          , E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       C, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       D, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , E2          , $$E       D, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, E2          , $$E       D, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, $$E       D, E4          , E5          .
```

(continued)

```
A POSSIBLE CHAIN IS :
E1         , $$E       E, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , E3          , $$E       E, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E3          , $$E       E, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, $$E       E, E4          , E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       C, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       D, E5          .

A POSSIBLE CHAIN IS :
E1         , E2          , $$E       D, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, E2          , $$E       D, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       A, $$E       D, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       E, E5          .

A POSSIBLE CHAIN IS :
E1         , E3          , $$E       E, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E3          , $$E       E, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, $$E       E, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       F, E5          .

A POSSIBLE CHAIN IS :
E1         , E3          , $$E       F, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, E3          , $$E       F, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       B, $$E       F, E5          .

A POSSIBLE CHAIN IS :
E1         , $$E       G, E5          .
```

```
A POSSIBLE CHAIN IS :
E1        , E2        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, E2        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E3        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, E3        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E2        , E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, E2        , E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E3        , E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, E3        , E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      C, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      D, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E2        , $$E      D, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, E2        , $$E      D, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, $$E      D, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      E, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E3        , $$E      E, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, E3        , $$E      E, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, $$E      E, E4        , $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      C, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      D, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E2        , $$E      D, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, E2        , $$E      D, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      A, $$E      D, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      E, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , E3        , $$E      E, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, E3        , $$E      E, $$E      G, E5       .

A POSSIBLE CHAIN IS :
E1        , $$E      B, $$E      E, $$E      G, E5       .

NO MORE CHAIN

THIS QUESTION USED   9.0160 SECONDS
```

5. FURTHER RESEARCH PLANNED

Besides tidying up some parts of the code, other things can yet be done. There appear to be two interesting lines of attack within the framework of this project.

First, we could investigate statistically defined associative structures. In other words, the more often a particular pattern of events is followed by another pattern of events, the more likely it is that a causal relationship connects the two. Positive and negative reinforcement processes could lead to hypotheses of different strengths. Cyclic or near-cyclic events should also be at work.

The second area of possible further investigations would be with reference to natural-language input/output in a reasonably large subset of English. Some preliminary studies have already been conducted by Adrian Walker and one of the present authors (N.V.F.). It is, however, too early to report on these now.

As possible applications, a number of interesting ideas have come to our mind. Many crimes are committed with, and their temporary success depends on, split-second precision (at least according to TV writers). Alibis may also depend on certain critical, highly time-dependent events. One might use this type of program to prove whether a particular crime could have been committed by a certain individual or several crimes by the same person, etc.

Turning to more mundane ideas, air traffic control must be exercised under dynamically changing conditions. The program at hand could serve as a component in a computerized system. The on-line optimization of traffic light timing in a metropolitan environment may also incorporate some of our ideas. The scheduling of reconnaissance missions is another potential area of application.

We also note here that this project is closely related to the problems of Critical Path Methods and PERT. (See, for example, Ref. 5.) We may sometime look into the possibility of establishing a link between the two fields of study.

Finally, it should be pointed out that intelligent machines will necessarily have to evaluate dynamically changing environments. In order to act in an optimum manner, they must make hypotheses on causal relationships. It is suggested that the cognitive structure of such machines would contain processes similar to those described in this paper.

APPENDIX. THE SECOND GREAT TRAIN ROBBERY

Chief Inspector Joshua Haggerty of Scotland Yard looked straight into the face of his Programmer–Analyst, Miss Eleanore Langley. He could hardly contain his frustration:

—All right, we are stuck. This is the first time my intuition has failed to give any b... signal at all. If you think it may help to have a run of that Findler–Chen program with our data. I don't care—let's try it.

—Listen carefully. I am giving you all the essential information we are certain about. Well, let me tell it to you from the beginning.

—The Bank of England was to transfer some gold bullions, worth the sum of £40,000,000, to its secret Manchester vaults. The armoured lorries, under the expert leadership of the newly hired security chief, Mr. H. McNaughton, picked up the shipment at 4:25 p.m. Some of the lorries, carrying a fake load of light metal, went direct to Victoria Station. They stopped on the way for exactly 14 minutes in order to synchronize their arrival at the station with the other group of lorries that followed a carefully designed indirect route. However, as Mr. McNaughton tells us, the second group got into a bit of a traffic jam and arrived at Victoria Station only at 6:28 p.m., nine minutes later than scheduled.

—By 6:37 p.m. both the real and the fake loads were safely sitting in the armoured carriage under the close watch of McNaughton's ten toughest guards.

—The express train left right on time, at 6:49 p.m., to reach Manchester at 9:17 p.m. However, exactly six minutes after the train rolled out of Birmingham station, at 8:23 p.m., it screeched to a sudden stop. Two masked gangsters held up the loco engineer at gun point. At the same time one of them poured tear gas into the air conditioning duct of the armoured carriage. As a response to the threat of exploding the whole train with its passengers, the guards released the special doors that could be opened only from inside. Two other gangsters transferred apparently only the real bullions to a waiting lorry at a terrific speed so that the train was able to continue its journey at 8:28 p.m.

—Luckily, the engineer of the train memorised the license plate number of the holdup lorry. It was found abandoned some 150 miles away in a little forest. The condition of the roads leading to this spot is such that the lorry could not have reached the place in less than three hours and 17 minutes. They should have transferred the loot here into a bigger bus, judging by the tyre prints. But, lo and behold, we found there nothing else than another set of fake bullions!

—Oh, another piece of information—added the Chief Inspector with some apprehension in his voice.

—We have made some time-and-motion studies. It takes at least eight minutes to transfer the real stuff from one lorry to another while the lighter, fake load can be shifted in about five minutes.

—That is all I know and I doubt you can get more sense out of this with your blooming grey boxes.

Miss Langley looked unperturbed and optimistic. She sat down at the keypunch to produce a few cards, which she then attached to the end of a small box of punched cards and submitted the lot to the Scotland Yard's friendly computer. Out came the answers:

```
        THE SECOND GREAT TRAIN ROBBERY

        SYMBOLIC NAME   MEANING

        DAY-OF-ROB      DAY OF ROBBERY

        SHIFT-REAL      EVNT OF MOVING REAL GOLD FROM 1 LORRY TO ANOTHER
        L-SFT-REAL      LENGTH OF TIME REQUIRED FOR SHIFT-REAL

        SHIFT-FAKE      EVNT OF MOVING FAKE GOLD FROM 1 LORRY TO ANOTHER
        L-SFT-FAKE      LENGTH OF TIME REQUIRED FOR SHIFT-FAKE

        LORY-REAL       EVNT OF MOVING REAL GOLD FROM BANK TO VICT STATION
        S-LORYREAL      STARTING TIME FOR LORY-REAL
        F-LORYREAL      FINISH TIME OF LORY-REAL
        L-LORYREAL      LENGTH FO TIME REQUIRED FOR LORY-REAL

        LORY-FAKE       EVNT OF MOVING FAKE GOLD FROM BANK TO VICT STATION
        S-LORYFAKE      STARTING TIME OF LORY-FAKE
        F-LORYFAKE      FINISH TIME OF LORY-FAKE
        L-LORYFAKE      LENGTH OF TIME REQUIRED FOR LORY-FAKE

        DELAY-LORY      EVNT OF DELAY OF LORRIES WITH REAL GOLD
        L-DELALORY      LENGTH OF DELAY OF DELAY-LORY

        HOLD-UP         EVNT OF HOLDUP AND ROBBERY
        S-HOLDUP        STARTING TIME OF HOLD-UP
        F-HOLDUP        FINISH TIME OF HOLD-UP
        L-HOLDUP        LENGTH OF TIME REQUIRED FOR HOLD-UP

        WAIT-VICT       EVNT OF WAITING AT VICT STATION
        S-WAIT          STARTING TIME OF WAIT-VICT
        F-WAIT          FINISH TIME OF WAIT-VICT
        L-WAIT          LENGTH OF TIME OF WAIT-VICT

        TRK-TO-TRN      EVNT OF MOVING REAL AND FAKE GOLD FROM TRUCK TO TRAIN
        S-TRK-TRN       STARTING TIME OF TRK-TO-TRN
        F-TRK-TRN       FINSIH TIME OF TRK-TO-TRN
        L-TRK-TRN       LENGTH OF TIME REQUIRED FOR TRK-TO-TRN

    INPUT FACTS IN LIST FORM :

            (  (DAY-OF-ROB) (EVENT,SHIFT-REAL,DELTA,DELTA,L-SFT-REAL)
               (EVENT,SHIFT-FAKE,DELTA,DELTA,L-SFT-FAKE)
               (EVENT,LORY-REAL,S-LORYREAL,F-LORYREAL,L-LORYREAL)
               (EVENT,LORY-FAKE,S-LORYFAKE,F-LORYFAKE,L-LORYFAKE)
               (EVENT,DELAY-LORY,DELTA,DELTA,L-DELALORY)
               (EVENT,HOLD-UP,S-HOLDUP,F-HOLDUP,L-HOLDUP)
               (EVENT,WAIT-VICT,S-WAIT,F-WAIT,L-WAIT)
               (EVENT,TRK-TO-TRN,S-TRK-TRN,F-TRK-TRN,L-TRK-TRN)   )*

    RECOGNIZED EVENTS ARE :

    EVENT          STARTIM        FINTIM         LENGTH

    SHIFT-REAL     DELTA          DELTA          L-SFT-REAL
    SHIFT-FAKE     DELTA          DELTA          L-SFT-FAKE
    LORY-REAL      S-LORYREAL     F-LORYREAL     L-LORYREAL
    LORY-FAKE      S-LORYFAKE     F-LORYFAKE     L-LORYFAKE
    DELAY-LORY     DELTA          DELTA          L-DELALORY
    HOLD-UP        S-HOLDUP       F-HOLDUP       L-HOLDUP
    WAIT-VICT      S-WAIT         F-WAIT         L-WAIT
    TRK-TO-TRN     S-TRK-TRN      F-TRK-TRN      L-TRK-TRN
```

```
NUMERIC VALUE ASSIGNMENTS

INPUT DATA                                         C O M P U T E D  V A L U E S
                                                   EVENT                 VALUE

DAY-OF-ROB                    197901010000 6       DAY-OF-ROB     197901010000
L-SFT-FAKE                               5 1       L-SFT-FAKE                5
L-SFT-REAL                               8 1       L-SFT-REAL                8
S-LORYREAL R DAY-CF-ROB                425 9       S-LORYREAL     197901010425
F-LORYREAL R DAY-OF-ROB                626 9       F-LORYREAL     197901010626
S-LORYFAKE R DAY-OF-ROB                425 9       S-LORYFAKE     197901010425
F-LORYFAKE R DAY-OF-ROB                619 9       F-LORYFAKE     197901010619
L-DELALORY                               9 1       L-DELALORY                9
S-HOLDUP   R DAY-CF-ROB                823 9       S-HOLDUP       197901010823
F-HOLDUP   R DAY-CF-ROB                828 9       F-HOLDUP       197901010828
S-WAIT     R DAY-OF-ROB                637 9       S-WAIT         197901010637
F-WAIT     R DAY-CF-ROB                649 9       F-WAIT         197901010649
S-TRK-TRN  R DAY-OF-ROB                628 9       S-TRK-TRN      197901010628
F-TRK-TRN  R DAY-OF-ROB                637 9       F-TRK-TRN      197901010637
                                                   L-LORYREAL              123
                                                   L-LORYFAKE              114
                                                   L-HOLDUP                  5
                                                   L-WAIT                   12
                                                   L-TRK-TRN                 9

     WHEN COULD TRANSFER OF REAL BULLIONS TAKE PLACE ]

QUESTION IN LIST FORM :

          (  (RELTRUE,SHORTER,D,(LENTH,SHIFT-REAL),E,DELTA)  )*

THE ANSWER IS :
TRK-TO-TRN, WAIT-VICT , DELAY-LORY, LORY-FAKE , LORY-REAL .

THIS QUESTION USED    .0900 SECONDS


 A PLEASURE WORKING FOR YOU,  BYE.
```

Miss Langley went back to the Chief and casually remarked:
—I am sure you already have a warrant to arrest Mr. McNaughton.

ACKNOWLEDGMENT

The authors appreciate the comments of the referee for the Second International Joint Conference on Artificial Intelligence, and his pointing out the rich world of the calculus of tenses, and the logic of change, action, and norms. (See Refs. 6–13.[6]) We are grateful to Clement Luk, who completed the part of the program that is concerned with the matrix of restrictions.

[6] In Ref. 13, see especially Section 51.

REFERENCES

1. N. V. Findler, "A survey of seven projects using the same language," in *Artificial Intelligence and Heuristic Programming*, N. V. Findler and B. Meltzer, eds. (Edinburgh University Press, Edinburgh, 1971).
2. N. V. Findler and W. R. McKinzie, "On a new tool in artificial intelligence research," *in Proc. First Int. Conf. on Artificial Intelligence*, pp. 259–270 (Washington, D.C., 1969).
3. N. V. Findler, J. L. Pfaltz, and H. J. Bernstein, *Four High Level Extensions of FORTRAN IV: SLIP, AMPPL-II, TREETRAN, and SYMBOLANG* (Spartan Books, New York, 1972).
4. H. A. Simon, *The Sciences of the Artificial* (MIT Press, Cambridge, Mass., 1969).
5. J. J. Moder and C. R. Philips, *Project Management with CPM and PERT* (Reinhold, New York, 1964).
6. R. A. Bull, "An algebraic study of tense logics with linear time," *J. Symb. Logic* **33**:27–39 (1968).
7. H. N. Castañeda, "The logic of change, action, and norms," *J. Phil.* **62**: 333–344 (1965).
8. G. H. von Wright, "The logic of action—A sketch," in *The Logic of Decision and Action*, N. Rescher, ed. (University of Pittsburgh Press, Pittsburgh, 1966).
9. D. Davidson, "The logical form of action sentences," in *The Logic of Decision and Action*, N. Rescher, ed. (University of Pittsburgh Press, Pittsburgh, 1966).
10. C. O. Evans, "States, activities and performances," *Austral. J. Phil.* **45**: 293–308 (1967).
11. A. N. Prior, *Time and Modality* (Clarendon Press, Oxford, 1957).
12. A. N. Prior, *Past, Present and Future* (Clarendon Press, Oxford, 1967).
13. H. Reichenbach, *Elements of Symbolic Logic* (The Free Press, New York, 1947).
14. Y. P. Terletskiy, Paradoxes in the Theory of Relativity (Transl. by B. Hoffmann) (Plenum, New York, 1968).