INFORMATION TO USERS

This was produced from a copy of a document sent to us for microfilming. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help you understand markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure you of complete continuity.

2. When an image on the film is obliterated with a round black mark it is an indication that the film inspector noticed either blurred copy because of movement during exposure, or duplicate copy. Unless we meant to delete copyrighted materials that should not have been filmed, you will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed the photographer has followed a definite method in "sectioning" the material. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For any illustrations that cannot be reproduced satisfactorily by xerography, photographic prints can be purchased at additional cost and tipped into your xerographic copy. Requests can be made to our Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases we have filmed the best available copy.

University Microfilms International
300 N. ZEEB ROAD, ANN ARBOR, MI 48106
18 BEDFORD ROW, LONDON WC1R 4EJ, ENGLAND
MARTIN, ANNE WILLS

A GENERAL ALGORITHM FOR DETERMINING LIKELIHOOD RATIOS IN CASCADED INERENCE

Rice University

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106

Ph.D. 1981
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages
2. Colored illustrations, paper or print
3. Photographs with dark background
4. Illustrations are poor copy
5. Pages with black marks, not original copy
6. Print shows through as there is text on both sides of page
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements
9. Tightly bound copy with print lost in spine
10. Computer printout pages with indistinct print ✓
11. Page(s) ________ lacking when material received, and not available from school or author.
12. Page(s) ________ seem to be missing in numbering only as text follows.
13. Two pages numbered _________. Text follows.
14. Curling and wrinkled pages
15. Other________________________________________
RICE UNIVERSITY

A GENERAL ALGORITHM FOR DETERMINING
LIKELIHOOD RATIOS IN CASCADED INFERENCE

by

ANNE WILLS MARTIN

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE:

David A. Schum, Professor of Psychology
Chairman

William C. Howell, Chairman
and Professor of Psychology

John W. Brelsford, Professor of Psychology

Robert M. Thrall, Professor of Mathematical Sciences

HOUSTON, TEXAS
August, 1980
A GENERAL ALGORITHM FOR DETERMINING
LIKELIHOOD RATIOS IN CASCADED INFERENCE

by

Anne Wills Martin

Abstract

In cascaded inference tasks there is not a direct logical connection between an observable event (datum) and the hypothesis of interest. Instead there is interposed at least one logical reasoning stage, consisting of intervening variables or intermediate event states. This paper is concerned with the modification or extension of Bayes' rule to render it more specific as a normative model for cascaded inference. In particular, the work reported here is directed towards simplifying the task of the researcher who wishes to use Bayes' rule as a standard for inferential behavior and of the analyst who wishes to use task decomposition in aiding inference. This is achieved by the development of some general principles of inference, the use of concepts from graph theory for the representation of inference tasks, and the application of computer technology.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Developments in Cascaded Inference</td>
<td>3</td>
</tr>
<tr>
<td>Principles for the Derivation of Adjusted Likelihood Ratios</td>
<td>31</td>
</tr>
<tr>
<td>A Computer Program for Inference Aiding</td>
<td>42</td>
</tr>
<tr>
<td>References</td>
<td>63</td>
</tr>
<tr>
<td>Figures</td>
<td>66</td>
</tr>
<tr>
<td>Appendix I</td>
<td>82</td>
</tr>
<tr>
<td>Appendix II</td>
<td>87</td>
</tr>
</tbody>
</table>
Introduction

Bayes' rule serves as a prescription for the revision of opinion on the basis of inconclusive evidence, or data that are consistent with more than one hypothesis. As a normative model, it can be used both as a standard to which human performance in inference tasks can be compared, and as an aid to inference. In its simplest form, Bayes' rule is applied to an inference about some hypothesis (H) on the basis of a relevant datum (D) as follows:

\[ P(H|D) = \frac{P(D|H)P(H)}{P(D)} \]  \hspace{1cm} (1)

where the vertical bar (|) is read "given". \( P(H|D) \), the probability of hypothesis \( H \) given datum \( D \) is the posterior probability, or the probability of the hypothesis after \( D \) is known to have occurred; the prior probability, \( P(H) \) is the probability of the hypothesis before the occurrence of \( D \) is known. \( P(D|H) \) is called the likelihood, and is the probability of the occurrence of \( D \) when \( H \) is true; \( P(D) \) is the normalizing constant, and is equivalent to \( P(D|H)P(H) + P(D|H^C)P(H^C) \), where \( H^C \) is "\( H \) complement" or "not \( H \"; the normalizing constant insures that the probabilities of mutually exclusive and exhaustive
events sum to one, as is required by the axioms of probability theory. Application of Bayes' rule in the above form requires that the likelihood, \( P(D|H) \), be known or estimated. In many inference tasks, however, the logical connection between \( D \), the observable event or datum and \( H \), the hypothesis, is indirect; that is, \( D \) is probabilistically linked to \( H \) via other variables which are circumstantially related to \( H \). For such cases, in which we refer to the task being performed as hierarchical, multi-stage, or cascaded inference, \( P(D|H) \) is usually not known, and the inference task is sufficiently complex that direct estimation of the likelihood is intellectually difficult and prone to error. This paper is concerned with the modification or extension of Bayes' rule to render it more specific as a normative model for cascaded inference. In particular, the work reported here is directed toward simplifying the task of the researcher who wishes to use Bayes' rule as a standard, and of the analyst who wishes to use Bayesian methods in aiding inference; this is achieved by the use of general principles of inference and application of computer technology.
Developments in Cascaded Inference

Among psychologists, interest in cascaded inference grew out of work in single-stage inference of the type modeled by Bayes' rule. It became apparent that in some situations in which application of Bayes' rule is desirable, the datum in the light of which opinion should be revised is not known with certainty to have occurred—Dodson (1961) developed a modified Bayes' Theorem (MBT) to handle such cases in the context of threat evaluation. Dodson's MBT can be expressed as follows. Given a set \( \{H_1, H_2, \ldots, H_i, \ldots, H_n\} \) of mutually exclusive and exhaustive hypotheses, and a datum \( E \) such that \( E \) may be in any of the states \( E_1, E_2, \ldots, E_k, \ldots, E_q \) where the elements of \( \{E_1, E_2, \ldots, E_k, \ldots, E_q\} \) are mutually exclusive and exhaustive, the expected probability of some hypothesis, \( H_i \), after consideration of \( E \) is

\[
\sum_{k=1}^{q} P(E_k)P(H_i|E_k),
\]

which is equal to

\[
\sum_{k=1}^{q} P(E_k) \left[ \frac{P(H_i)P(E_k|H_i)}{\sum_{\alpha=1}^{n} P(H_\alpha)P(E_k|H_\alpha)} \right]. \tag{2}
\]

Dodson extended his MBT to model situations in which
probability revision is performed on the basis of a number of independent data \( \{E_1, E_2, \ldots, E_j, \ldots, E_m\} \), each of which may take on states \( E_{j1}, E_{j2}, \ldots, E_{jk}, \ldots, E_{jq} \).

where \( v_j \), the set \( \{E_{j1}, E_{j2}, \ldots, E_{jk}, \ldots, E_{jq}\} \) is mutually exclusive and exhaustive. If the probability of \( H_i \) is revised upon consideration of \( E_1, E_2, \ldots, E_{j-1} \) in turn, then Dodson's formula for the probability of hypothesis \( H_i \) upon consideration of \( E_j \) is:

\[
P_j(H_i) = \sum_{k=1}^{q_j} P(E_{jk}) \left[ \frac{P_{j-1}(H_i)P(E_{jk}|H_i)}{\sum_{\alpha=1}^{n} P_{j-1}(H_\alpha)P(E_{jk}|H_\alpha)} \right]
\]

(3)

Gettys and Wilke (1969) pointed out that, in order to parallel Bayes' rule, the posterior probability in MBT ought to be conditioned upon some observable event. They therefore reformulated the problem to include \( \omega \), an observable event which is probabilistically linked to datum \( E \).

For a mutually exclusive and exhaustive set of hypotheses \( \{H_1, H_2, H_i, \ldots, H_n\} \) and a datum \( E \) which may take on values \( \{E_1, E_2, \ldots, E_k, \ldots, E_q\} \), assume that, for each known event \( \omega_k \), \( \omega_k \) is conditionally independent of each element of the set \( \{H_1, H_2, \ldots, H_i, \ldots, H_n\} \), given \( E_k \), i.e. \( P(\omega|E_kH_i) = P(\omega|E_k) \) i,k. Then
\[ P(H_i | \omega) = P(H_i) \sum_{k=1}^{q} \frac{P(E_k | \omega) P(E_k | H_i)}{P(E_k)} \]  

where \( P(E_k) = \sum_{\alpha=1}^{n} P(H_\alpha) P(E_k | H_\alpha) \).

In Gettys and Wilkes extension of their formulation of MBT, to a multiple-input model, they corrected a deficiency which had been found in Dodson's (1961) multiple-data model (Equation 3); Dodson's model, in using \( P_{j-1}(H_i) \) in the normalizing constant (where \( P_{j-1}(H_i) \) is the probability of \( H_i \) after consideration of datum j-1) for the computation of \( P_j(H_i) \), is not path-independent, as data are given different weights according to the order in which they are considered as well as to their probabilities under the various hypotheses. For the Gettys-Wilke formulation, assume again that \( \{H_1, H_2, \ldots, H_i, \ldots, H_n\} \) is mutually exclusive and exhaustive. Let \( \{E_1, E_2, \ldots, E_j, \ldots, E_m\} \) be a set of data, each of which may take on states \( E_{j1}, E_{j2}, \ldots, E_{jk}, \ldots, E_{jq_j} \), where \( E_j \), the set \( \{E_{j1}, E_{j2}, \ldots, E_{jk}, \ldots, E_{jq_j}\} \), is mutually exclusive and exhaustive. Further, let \( \{\omega_1, \omega_2, \ldots, \omega_j, \ldots, \omega_m\} \) be observable events, each of which is diagnostic of a corresponding datum, so that \( \omega_j \) is related to \( E_j \) for all \( j \). If we assume that the \( \omega_j \) are conditionally independent given the hypotheses, i.e. \( P(\omega_1, \omega_2, \ldots, \omega_j, \ldots, \omega_m) \),
\[ \omega_m|H_i \} = \frac{1}{m} \prod_{j=1}^{m} P(\omega_j|H_i), \forall i, \] and that each \( \omega_j \) is conditionally independent of the hypotheses given any state of \( E_{jk} \), i.e. \( \forall j, P(\omega_j|H_i, E_{jk}) = P(\omega_j|E_{jk}) \) for \( 1 \leq i \leq n \) and \( 1 \leq k \leq q_j \), then

\[
P(H_i|\omega_1, \omega_2, \ldots, \omega_j) = C_j P(H_i|\omega_1, \omega_2, \ldots, \omega_{j-1}) \times \left\{ \sum_{k=1}^{q_j} \frac{P(E_{jk}|\omega_j) P(E_{jk}|H_i)}{P(E_{jk})} \right\}^{j-j}
\]

(5)

where \( C_j \) is a normalizing constant for the \( \omega \)'s equalling

\[
\prod_{\ell=1}^{j} P(\omega_{\ell})
\]

\[
P(\omega_1, \omega_2, \ldots, \omega_{\ell}, \ldots, \omega_j)
\]

and \( P(E_{jk}) \) normalizes \( P(E_{jk}|H_i)P(H_i) \) over the hypotheses, equalling \( \sum_{i=1}^{n} P(H_i)P(E_{jk}|H_i) \). Since the Gettys-Wilke formulation uses the prior probability \( P(H_i) \) instead of the posterior \( P_{j-1}(H_i) \), Dodson's lack of path independence is avoided.

Both the Dodson and the Gettys and Wilke formulations are modifications of Bayes' rule as shown in Equation 1. Bayes rule can also be expressed in an odds version:
\[
\frac{P(H_1|D)}{P(H_2|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_2)P(H_2)},
\]

(6)

where \(H_1\) and \(H_2\) are mutually exclusive hypotheses. The ratio \(P(H_1|D)/P(H_2|D)\) is called the posterior odds, \(P(H_1)/P(H_2)\) is called the prior odds, and \(P(D|H_1)/P(D|H_2)\) is the likelihood ratio. Since the normalizing constant is the same for both \(P(H_1|D)\) and \(P(H_2|D)\), it is cancelled out. Since the ratios \(P(D|H_1)/P(D|H_2)\) and \(P(H_1)/P(H_2)\) are separable, it is apparent that all of the information necessary to determine the amount and strength of odds revision [the change from \(P(H_1)/P(H_2)\) to \(P(H_1|D)/P(H_2|D)\)] is encoded in the likelihood ratio for \(D\), which is denoted \(L_D\). Schum and DuCharme (1971) made use of this in order to restrict their study of cascaded inference to likelihood ratios. In particular, their interest was in the likelihood ratio for \(D^*\), a report of the occurrence of event \(D\) which is related to hypotheses \(H_1\) and \(H_2\). This likelihood ratio adapted to cascaded inference, \(P(D^*|H_1)/P(D^*|H_2)\) is denoted \(L_{D^*}\) to distinguish it from the simple likelihood ratio \(L_D\).

The Schum-DuCharme formulation for the adjusted likelihood of \(D^*\) in the case in which \(D^*\) is assumed to be conditionally independent of \(H_1\) and \(H_2\) given \(D\), the occurrence of event \(D\), or \(D^C\), its nonoccurrence, is:
\[
\Lambda_{D^*} = \frac{P(D|H_1) + \frac{P(D^*|D^c)}{P(D^*|D) - P(D^*|D^c)}}{P(D|H_2) + \frac{P(D^*|D^c)}{P(D^*|D) - P(D^*|D^c)}}.
\]

(7)

for \( P(D^*|D) \neq P(D^*|D^c) \)

Schum and Pfeiffer (1973) extended the likelihood formulation to model a more complex case in which the observable event is a composite report, \( R \), made up of reports \( S_1, S_2, \ldots, S_i, \ldots, S_n \) from \( n \) sensors. The intermediate datum, \( D \), may take on values \( D_1, D_2, \ldots, D_j, \ldots, D_N \), where the class \( \{D_1, D_2, \ldots, D_j, \ldots, D_N\} \) is mutually exclusive and exhaustive.

Under the assumption that the class \( \{S_1, S_2, \ldots, S_i, \ldots, S_n\} \) is conditionally independent given \( D_j \) for any \( j, 1 \leq j \leq N \),

\[
P(R|D_j) = \prod_{i=1}^{n} P(S_i|D_j).
\]

The hypotheses, \( H_1, H_2, \ldots, H_k, \ldots, H_m \) are assumed to be mutually exclusive. We also assume that \( \{S_1, S_2, \ldots, S_i, \ldots, S_n\} \) is conditionally independent, given \( D_j \), of \( H_k \) for \( 1 \leq j \leq N \) and \( 1 \leq k \leq m \). For two hypotheses, \( H_u \) and \( H_v \), the adjusted likelihood ratio, \( P(R|H_u)/P(R|H_v) \) can be computed as follows:
\[ \Lambda_R = \frac{\sum_{j=1}^{N} P(R|D_j)P(D_j|H_u)}{\sum_{j=1}^{N} P(R|D_j)P(D_j|H_v)}. \] (8)

In order to investigate the effect on the adjusted likelihood ratio of the credibility of several sources, Schum and Kelly (1973) examined a special case of the Schum-Pfeiffer (1973) formulation. This case incorporates reports from \( n \) sources which/who report whether or not event \( D \) has occurred. Formally, let \( R = \{R_1, R_2, \ldots, R_i, \ldots, R_n\} \), where every \( R_i \) has one of the two values \( D_i^* \) and \( D_i^{c*} \). As required by the rules of probability theory, \( \{D, D^c\} \) is mutually exclusive and exhaustive. The hypotheses under consideration, \( H_1 \) and \( H_2 \) are mutually exclusive. The report class \( \{R_1, R_2, \ldots, R_i, \ldots, R_n\} \) is assumed to be conditionally independent given \( D \) or \( D^c \), and the elements of \( \{R_1, R_2, \ldots, R_i, \ldots, R_n\} \) are assumed to be conditionally independent, given \( D \) or \( D^c \), of \( H_1 \) and \( H_2 \). Using the notation of signal detection theory, let:

- \( h_i \), the hit rate of source \( i \), = \( P(D_i^*|D) \),
- \( m_i \), the miss rate of source \( i \), = \( P(D_i^{c*}|D) \),
- \( c_i \), the correct rejection rate for source \( i \), = \( P(D_i^{c*}|D^c) \), and
\( f_i \), the false positive rate for source

\( i, = P(D_i^x|D^c) \).

For the case in which all \( n \) sources report that \( D \) has occurred, i.e. \( R_i = D_i^x \lor V_i \), \( i \leq 1 \leq n \), a likelihood ratio can be computed as follows:

\[
\Lambda_R = \frac{P(D|H_1) + \left[ \prod_{i=1}^{n} \frac{h_i}{f_i} - 1 \right]^{-1}}{P(D|H_2) + \left[ \prod_{i=1}^{n} \frac{h_i}{f_i} - 1 \right]^{-1}}, \tag{9}
\]

assuming that \( \prod_{i=1}^{n} \frac{h_i}{f_i} \neq 1 \), and that, for \( 1 \leq i \leq n \), \( f_i \neq 0 \).

If all \( n \) sources report the nonoccurrence of \( D \), i.e. \( R_i = D_i^c \) for all \( i \), where \( 1 \leq i \leq n \), then

\[
\Lambda_R = \frac{P(D|H_1) + \left[ \prod_{i=1}^{n} \frac{m_i}{c_i} - 1 \right]^{-1}}{P(D|H_2) + \left[ \prod_{i=1}^{n} \frac{m_i}{c_i} - 1 \right]^{-1}}, \tag{10}
\]

assuming that \( \prod_{i=1}^{n} \frac{m_i}{c_i} \neq 1 \), and that for no \( i, i \leq 1 \leq n \) it is the case that \( c_i = 0 \).

When the sources give contradictory testimony, they are divided into two sets, \( I \) and \( J \). Set \( I \) contains all of the sources reporting the occurrence of \( D \),
i.e., \( I \equiv \{ i : R_i = D_i^* \} \), and \( J \) contains all of the sources reporting the nonoccurrence of \( D \), so that \( J \equiv \{ i : R_i = D_i^c \} \). The adjusted likelihood ratio for the aggregate report \( R \) in this case can be written:

\[
\Lambda_R = \frac{P(D|H_1) + \left[ \prod_{i \in I} \frac{h_i}{f_i} \right] \left[ \prod_{i \in J} \frac{m_i}{c_i} \right] - 1}{P(D|H_2) + \left[ \prod_{i \in I} \frac{h_i}{f_i} \right] \left[ \prod_{i \in J} \frac{m_i}{c_i} \right] - 1}^{-1},
\]

(11)

for \( \left[ \prod_{i \in I} \frac{h_i}{f_i} \right] \left[ \prod_{i \in J} \frac{m_i}{c_i} \right] \neq 1 \), and neither \( f_i \) nor \( c_i \) equal to zero, \( 1 \leq i \leq n \).

The Schum-Kelly equations are all in the form

\[
\Lambda_R = \frac{P(D|H_1) + V}{P(D|H_2) + V},
\]

where \( V \) is the term that incorporates source credibility. This form is used to make clear the relationship between the impact of the event being reported, and the credibilities of sources reporting its occurrence. For example, in equation 9, if \( \frac{1}{n} \prod_{i=1}^{n} \frac{h_i}{f_i} \) is very large, meaning that the probability of all \( n \) sources reporting the occurrence of \( D \) when \( D \) has occurred is much greater than the
probability of their reporting D's occurrence when D has not occurred, V is very small, and \( \Lambda_R \) is close to \( L_D; \) \( L_D \) serves as a bound on \( \Lambda_D^* \). For \( \prod_{i=1}^{n} h_i > \prod_{i=1}^{n} f_i \), as the value of \( \prod_{i=1}^{n} \frac{h_i}{f_i} \) approaches one, so that \( \Lambda_R \) approaches one, and the aggregate report is seen to cause almost no opinion revision. When \( \prod_{i=1}^{n} h_i \) is less than \( \prod_{i=1}^{n} f_i \), V is negative, and as the value of \( \prod_{i=1}^{n} \frac{h_i}{f_i} \) approaches zero, \( \Lambda_R \) approaches \( \frac{1-P(D|H_1)}{1-P(D|H_2)} \). Thus the Schum-Kelly equations educate intuition about the behavior of the adjusted likelihood ratio, of which the value prescribes amount and direction of opinion revision, as a function of source credibility.

By far the most general formulation of the cascaded inference paradigm comes from Kelly and Barclay (1973). All formulations discussed so far in this paper have been for two-stage inference; in the Gettys and Wilke (1969) case, from \( \omega \) to \( E \) and from \( E \) to \( H \), and in the Schum-DuCharme (1971) and Schum-Kelly (1973) cases, from \( D^* \) to \( D \) and \( D \) to \( H \). Kelly and Barclay cover the \( n \)-stage case, where \( n \) is any positive integer. In addition, the Kelly-Barclay formulation does not require conditional independence assumptions of the type that must be
made for applicability of models discussed previously.

However, the generality of the model is bought only at great cost both to the analyst or researcher attempting to apply it, and to the user who is being aided. In their report, Kelly and Barclay develop the model by beginning with restrictive conditional independence assumptions, yielding a relatively simple algorithm for inference, and then relaxing the assumptions, necessitating a great deal more complexity for computation. Discussion in this paper will be limited to the more general algorithm and one particular class of applications. The inference tasks under discussion are described by the tree in Figure 1. Observable events or predictor variables $D^1, D^2, \ldots, D^m$ allow us to make inferences about the state of intermediate variable $e^n$, which in turn allows us to make an inference about intermediate variable $e^{n-1}$, and so on; intermediate variable $e^1$ allows us to make an inference about $h$, the hypothesis. The states of the data, $D^1, D^2, \ldots, D^m$ are known; the hypothesis may be in any one of the mutually exclusive states $h_1, h_2, \ldots, h_j, \ldots, h_{N_0}$; $e^1$ may be in any one of $N_1$ mutually exclusive states, $e_2$ be in any of $N_2$ mutually exclusive states, and so on to $e^n$ which may be in any one of $N_n$ mutually exclusive states. The inputs to the computational routine are in the form of conditional probability estimates arranged in various
matrices. One necessary collection of matrices is the set of $E^i$ matrices which describe the probabilistic relationships of the intermediate variables $e^i$ to each other and to the hypothesis $h_k$. For $2 \leq i \leq n$ there are $N_1 \times N_2 \times \ldots \times N_{i-2} E^1$ matrices to be estimated, or one for each combination of the possible states of $h, e^1, e^2, \ldots, e^{i-2}$. There is only one $E^1$ matrix; the $E^1$ matrix contains conditional probabilities for combination of the possible states of $h$ and $e^1$:

$$
E^1 = \begin{bmatrix}
    P(e_1^1 | h_1) & \cdots & P(e_k^1 | h_1) & \cdots & P(e_{N_1}^1 | h_1) \\
    \vdots & \ddots & \vdots & \cdots & \vdots \\
    \vdots & & \ddots & \cdots & \vdots \\
    P(e_1^1 | h_j) & \cdots & P(e_k^1 | h_j) & \cdots & P(e_{N_1}^1 | h_j) \\
    \vdots & \ddots & \vdots & \cdots & \vdots \\
    \vdots & & \ddots & \cdots & \vdots \\
    P(e_1^1 | h_{N_0}) & \cdots & P(e_k^1 | h_{N_0}) & \cdots & P(e_{N_1}^1 | h_{N_0})
\end{bmatrix}
$$

For $2 \leq i \leq n$, each $E^i$ matrix is of the form
\[
E(p, \ldots, q, j) = e_{1}^{i-1} \left[ \begin{array}{c}
P(e_{1}^{i} | e_{1}^{i-1}) \ldots P(e_{k}^{i} | e_{1}^{i-1}) \ldots P(e_{N_{i}}^{i} | e_{1}^{i-1}) \\
\vdots \\
P(e_{1}^{i} | e_{u}^{i}) \ldots P(e_{k}^{i} | e_{u}^{i}) \ldots P(e_{N_{i}}^{i} | e_{u}^{i}) \\
\vdots \\
P(e_{1}^{i} | e_{N_{i-1}}^{i-1}) \ldots P(e_{k}^{i} | e_{N_{i-1}}^{i-1}) \ldots P(e_{N_{i}}^{i} | e_{N_{i-1}}^{i-1})
\end{array} \right]
\]

where \((p, \ldots, q, j)\) denotes that each of the probability estimates is conditional not only on some \(e_{u}^{i-1}\), but also on specific values \(e_{p}^{i-2}, \ldots, e_{q}^{i}, h_{j}\); that is, \(P(e_{k}^{i} | e_{u}^{i})\) is in fact \(P(e_{k}^{i} | e_{u}^{i-1} e_{p}^{i-2} \ldots e_{q}^{i} h_{j})\). Since there is one matrix to be estimated for \(e_{1}^{i}\), and \(N_{0} \times N_{1} \times \ldots \times N_{i-2}\) matrices to be estimated for all other intermediate variables \(e_{i}^{i}\), the total number of \(E_{i}\) matrices is

\[
1 + \sum_{i=2}^{n} \left( \sum_{\alpha=0}^{i-2} N_{\alpha} \right)
\]

Since each of the \(E_{i}\) contains \(N_{i} \times N_{i-1}\) estimates, the number of conditional probability estimates to be made has the potential to grow very large.

Along with the \(E_{i}\) matrices, the model requires that a number of data matrices be estimated. Each data matrix \(D\) is of the form
\[ D^1 \ldots D^2 \ldots D^m \]
\[
\begin{bmatrix}
e_1^n P(D^1|e_1^n) \ldots P(D^2|e_1^n,D^1) \ldots P(D^m|e_1^n,D^1,D^2 \ldots D^{m-1}) \\
\vdots \\
\vdots \\
e_k^n \\
\vdots \\
e_n^n P(D^m|e_n^n,D^1,D^2 \ldots D^{m-1})
\end{bmatrix}
\]

\[ \mathcal{D}(\ell,p,\ldots,q,j) = \begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
e_k^n \\
\vdots \\
e_n^n
\end{bmatrix}
\]

and there are \( \prod_{\alpha=0}^{n-1} N_\alpha \) of them.

Each data matrix \( \mathcal{D}(\ell,p,\ldots,q,j) \) is transformed into a column vector \( \overline{D}(\ell,p,\ldots,q,j) \) by taking the product of the elements of each row of \( \mathcal{D}(\ell,p,\ldots,q,j) \), i.e., element \( k \) of \( \overline{D}(\ell,p,\ldots,q,j) \) is the product of the elements in the \( k \)th row of \( \mathcal{D}(\ell,p,\ldots,q,j) \). Thus if we let \( D = D^1 \cap D^2 \cap \ldots \cap D^m \), the \( k \)th element of \( D(\ell,p,\ldots,q,j) \) is \( P(D|e_k^n,e_p^{n-1},\ldots,e_q^1,h_j) \).

For any given value of \( i \), all of the \( E^i \) matrices are placed into an \( E^{i*} \) matrix of dimensions \( (N_0 \times N_1 \times \ldots \times N_{i-1}) \) by \( (N_0 \times N_1 \times \ldots \times N_i) \). The arrangement is such that the first row of \( E^{i*}(1,1,\ldots,1) \) is contained in the first row, columns 1 through \( N_i \) of \( E^{i*} \);
the second row of $E_i^1(1,1,\ldots,1)$ is contained in row 2 of $E_i^*$, columns $N_i+1$ to $2N_i$, and so on. The first row of $E_i^1(2,1,\ldots,1)$ occupies row $N_i-1+1$, columns $N_i-1 \times N_i+1$ through $N_i-1 \times N_i + N_i$; the last row of $E_i^*$, row $N_0 \times N_1 \times \cdots \times N_i-1$ contains the elements of $E_i^1(N_{n-2},N_{n-3},\ldots,N_1,N_0)$ in columns $N_0 \times N_1 \times \cdots \times N_i-1 \times N_i+1$ to $N_0 \times N_1 \times \cdots \times N_i$. In general, the $k$th row of $E_i^1(p,\ldots,q,j)$ is placed in row $(p \times \cdots \times q \times j_{i-1}) \times N_i-1+k$, columns $(p \times \cdots \times q \times j \times N_i-1-1) \times N_i+1$ to $(p \times \cdots \times q \times j \times N_i-1 \times N_i)$. In each row of $E_i^*$, all elements which are not transformed from an $E_i^1$ matrix by the above rule are assumed to be zero.

The $D_i$ vectors are placed into an $(N_0 \times N_1 \times \cdots \times N_n)$ by $(N_0 \times N_1 \times \cdots \times N_{n-1})$ matrix in a different fashion. Each $D_i$ vector is transferred to a column of $D_i^*$. Column one, rows 1 to $N_i$ of $D_i^*$ contains $D_i(1,1,\ldots,1,1)$; $D_i(2,1,\ldots,1,1)$ is placed in column 2, rows $N_i+1$ to $2n$, and so on;

$D_i(N_{n-1},N_{n-2},\ldots,N_1,N_0)$ is placed in column $N_{n-1} \times N_{n-2} \times \cdots \times N_1 \times N_0$, rows $(N_0 \times N_1 \times \cdots \times N_{n-2} \times N_{n-1}-1) \times N_{n+1}$ to $N_0 \times N_1 \times \cdots \times N_{n-2} \times N_{n-1} \times N_n$ of $D_i^*$. Generally, $D_i(\ell,p,\ldots,q,j)$ occupies column $\ell \times p \times \cdots \times q \times j$, rows $(\ell \times p \times \cdots \times q \times j-1)N_{n+1}$ to $\ell \times p \times \cdots \times q \times j \times N_n$ of $D_i^*$ and any element of $D_i^*$ not defined by this rule is assumed to be zero.
$$D^{n*} \text{ and } E^{n*} \text{ are used to complete the diagonal matrix } T^{n-1}$$

$$T^{n-1} = E^{n*}D^{n*}.$$ 

The elements of the \((N_0 \times N_1 \times \ldots \times N_{n-1})\) square matrix are placed into a new matrix \(D^{n-1*}\) by placing the first \(N_{n-1}\) diagonal elements of \(T^{n-1}\) into column one, rows 1 to \(N_{n-1}\), placing elements \(N_{n-1}+1\) to \(2N_{n-1}\) in column two, rows \(N_{n-1}+1\) to \(2N_{n-1}\) and so on; the last \(N_{n-1}\) elements of \(T^{n-1}\) are placed in column \(N_0 \times N_1 \times \ldots \times N_{n-2}\), row \((N_0 \times N_1 \times \ldots \times N_{n-2}-1)\times N_{n-1}+1\) to \(N_0 \times N_1 \times \ldots \times N_{n-2}\times N_{n-1}\). As usual, any elements not defined by the transfer procedure are assumed to equal zero. After the operations to fill \(D^{n-1*}\) are completed, \(T^{n-2} = E^{n-1*}D^{n-1*}\) is computed. The elements of \(T^{n-2}\) are used to fill the \(N_0 \times N_1 \times \ldots \times N_{n-2}\) by \(N_0 \times N_1 \times \ldots \times N_{n-3}\) matrix \(D^{n-2*}\). Then \(T^{n-3}\) is computed and the procedures all repeated until \(D^{0*}\) is obtained. \(D^{0*}\) is the column vector containing the likelihoods; that is, the \(j\)th element of \(D^{0*}\) is 

\[P(D|h_j)\] 

where \(D\) is the joint event \(D^1 \cap D^2 \cap \ldots \cap D^m\). These likelihoods may be used in likelihood ratios for the odds-likelihood form of Bayes' rule, or incorporated into Bayes' rule in simple form.

The Kelly-Barclay model (1973) is normative, and certainly very general. It does not, however, say a great deal about the task faced by the inference-maker, except
that the task may be extremely complex and require a larger amount of time. Schum (1977) used likelihood ratio equations to model specific tasks which might be performed by an inference maker, especially a factfinder in a court trial. Juridical inference tasks are especially amenable to study within a cascaded inference framework, as almost all evidence in court trials comes as testimony \((D^*)\), or reports of events \((D)\) which are relevant to the guilt \((H_G)\) or innocence \((H_I)\) of the defendant. The case of multiple reports about the same event was covered in Schum and Kelly (1973). For the one-witness case, if we let \(h = P(D^*|D)\) and \(f = P(D^*|D^C)\), and assume that \(D^*\) is conditionally independent of \(H_I\) and \(H_G\) given \(D\) or \(D^C\), we can express the adjusted likelihood ratio for \(D^*\) as:

\[
\lambda_{D^*} = \frac{P(D|H_G) + \frac{h}{f} - 1}{P(D|H_I) + \frac{h}{f} - 1},
\]

(12)

for \(h \neq f\) and \(f \neq 0\).

Equation 12 is formally equivalent to Equation 7, from Schum and DuCharme (1971) and to Equation 9 for \(n = 1\), from Schum and Kelly (1973). \(\lambda_{D^*}\) has a familiar form, \((P(D|H_A)+V)/(P(D|H_B)+V)\). The credibility term, \(V\), and thus the adjusted likelihood ratio, \(\lambda_{D^*}\), varies with \(h/f\) in the same way that \(V\) in Equation 9 varies with
\[
\prod_{i=1}^{n} \frac{h_i}{f_i}. \text{ According to Schum (1977), there are some cases, such as bias, in which a witness' hit and false positive rates might be dependent upon the guilt or innocence of the defendant. For these cases, let } h_G = P(D^*|DH_G), \text{ the witness' hit rate under the hypothesis of guilt; let } f_G = P(D^*|D^C H_G); h_I = P(D^*|DH_I) \text{ and } f_I = P(D^*|D^C H_I). \text{ Then}
\]

\[
\Lambda_{D^*} = \frac{P(D|H_G)[h_G-f_G] + f_G}{P(D|H_I)[h_I-f_I] + f_I}.
\]

(13)

Note that when \( h \) and \( f \) are conditional upon the hypothesis, there is no separable \( V \) term. Rather than being dependent on \( P(D|H_I), P(D|H_G) \), and \( h/f \), \( \Lambda_{D^*} \) is affected by the two likelihoods of \( D \) and the sizes of the terms \( h_G, f_G, h_I, \) and \( f_I \). Schum (1977) also attempted to model the task of making an inference about defendant's guilt or innocence on the basis of hearsay evidence. Schum's hearsay paradigm is illustrated in Figure 2. Witness B makes report \( D^*_{BA} \), a report that out-of-court declarant, A, has made report \( D^*_{A} \); that is, B says that A has reported the occurrence of \( D \), an event which is probative of the guilt or innocence of the defendant. The pair of reports \( D^*_{A} \) and \( D^C_{A} \) is assumed to be exhaustive; i.e. A is assumed to have made a report about the
occurrence or nonoccurrence of event D. B's report is assumed to be conditionally independent of the hypothesis and the occurrence or nonoccurrence of event D, given $D_A^*$ or $D_A^{C*}$; A's report is assumed conditionally independent of the hypothesis given D or $D^C$. If we let $h_B = P(D_{BA}^*|D_A^*)$, $f_B = P(D_{BA}^*|D_A^{C*})$, $h_A = P(D_A^*|D)$, and $f_B = P(D_A^*|D^C)$, the likelihood ratio for $D_{AB}^*$, $P(D_{AB}^*|H_G)/P(D_{AB}^*|H_I)$ can be expressed as

\[ \Lambda_{D_A^*} = \frac{P(D|H_G) + V}{P(D|H_I) + V}, \]  

(14)

where

\[ V = \left[ \frac{h_A}{f_A} - 1 \right]^{-1} + \left[ \frac{(h_B/f_B - 1)(h_A/f_A)}{(h_B/f_B - 1)(h_A/f_A)} \right]^{-1}, \]

for $h_A \neq f_A$, $f_B \neq f_B$, and $f_A \neq 0$.

Martin (1979) formulated the hearsay problem in a slightly different fashion, as illustrated in Figure 3. In-court witness Y makes report $(D_X^*)_Y$, or reports that X has made a report $(D_X^*)$ that D has occurred. Note that the complement of $D_X^*$ is not assumed to be $D_X^{C*}$, but is $D_X^{C*}$; X is not assumed to have made a report to Y. Let $h_Y = P[(D_X^*)_Y|D_X^*]$, $f_Y = P[(D_X^*)_Y|D_X^{C*}]$, $h_X = P(D_X^*|D)$, and $f_X = P(D_X^{C*}|D)$. Making the usual conditional
independence assumptions, that \((D^*_X)_Y^*\) is conditionally independent of the hypothesis and the occurrence or non-occurrence of \(D\) given \(D^*_X\) or \(D^{*c}_X\), and that \(D^*_X\) and \(D^{*c}_X\) are conditionally independent of the hypothesis given \(D\) or \(D^c\), we can express the adjusted likelihood ratio for \((D^*_X)_Y^*\) as

\[
\Lambda(D^*_X)_Y^* = \frac{P(D|H_G) + \left\{ \begin{array}{l}
h_X + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1} \\
f_X + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1}
\end{array} \right\}}{P(D|H_I) + \left\{ \begin{array}{l}
h_X + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1} \\
f_X + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1}
\end{array} \right\}}
\]

for \(f_Y \neq 0\) and \(h_Y \neq f_Y\).

As in the Schum and Kelly (1973) and Schum (1977) models, Equation 15 can be decomposed into event probabilities and a creditibility term. Let
\[ V = \left( \frac{h_X + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1}}{f_X + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1}} - 1 \right)^{-1}, \]  

(16)

\[ f_Y \neq 0 \text{ and } h_Y \neq f_Y. \]

The hit and false positive rates for \( X \) can be replaced by their definitions, \((h_X = P(D_X^*|D)\) and \( f_X = P(D_X^*|D^C)\)), so that Equation 16 becomes

\[ V = \left( \frac{P(D_X^*|D) + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1} - 1}{P(D_X^*|D^C) + \left[ \frac{h_Y}{f_Y} - 1 \right]^{-1}} \right)^{-1} \]

(16a)

\[ = \left[ \frac{P[(D_X^{*})^*|D]}{P[(D_X^{*})^*|D^C]} - 1 \right]^{-1}. \]

Thus \( V \) is seen to contain an adjusted likelihood ratio which itself has separable event probability and credibility terms. The occurrence of the form

\[ \frac{A + [F/G - 1]^{-1}}{B + [F/G - 1]^{-1}} \]
models for both two-stage and three-stage inference is suggestive of the existence of a more generalized model. Martin (1979) derived an equation for use in a model for n-stage inference. The n-stage paradigm is illustrated in Figure 4; event $D_n$ is used in making an inference about $D_{n-1}$ which is used in turn to make an inference about $D_{n-2}$, and so on until $D_1$ is used in making an inference about $D_0$. \{D_0, D_0^c\} might be considered to be hypotheses and $D_n$ might be considered to be the observable datum; however this is unnecessary. If we assume that every event below the class \{D_1, D_1^c\} in Figure 4 is conditionally independent, given either member of the class one level above it, of all events more than one level above it, then

$$\frac{P(D_j|D_i)}{P(D_j^c|D_i^c)} = \frac{P(D_{i+1}|D_i)[P(D_j|D_{i+1}) - P(D_j|D_{i+1}^c)] + P(D_j|D_{i+1})}{P(D_j|D_i^c) P(D_{i+1}|D_i^c)[P(D_j|D_{i+1}) - P(D_j|D_{i+1}^c)] + P(D_j|D_{i+1}^c)}. \quad (17)$$

A proof is included in Martin (1979).

Equation 17 is equivalent to

$$\frac{P(D_j|D_i)}{P(D_j^c|D_i^c)} = \frac{P(D_{i+1}|D_i) + \left[ \frac{P(D_j^c|D_{i+1})}{P(D_j^c|D_{i+1}^c)} - 1 \right]^{-1}}{P(D_{i+1}|D_i^c) + \left[ \frac{P(D_j|D_{i+1})}{P(D_j|D_{i+1}^c)} - 1 \right]^{-1}}, \quad (17a)$$
for \( P(D_j | D_i^c) \neq 0 \) and \( P(D_j | D_{i+1}) \neq P(D_j | D_{i+1}^c) \).

In order to compute the likelihood ratio for \( D_n \), \( P(D_n | D_0) / P(D_n | D_0^c) \), one simply uses Equation 17a iteratively:

For \( i = 0 \) and \( j = n \),

\[
\frac{P(D_n | D_0)}{P(D_n | D_0^c)} = \frac{P(D_1 | D_0) + \left[ \frac{P(D_n | D_1)}{P(D_n | D_1^c)} - 1 \right]^{-1}}{P(D_1 | D_0^c) + \left[ \frac{P(D_n | D_1)}{P(D_n | D_1^c)} - 1 \right]^{-1}};
\]

now we need to compute \( P(D_n | D_1) / P(D_n | D_1^c) \), so we let \( j = n \) and \( i = 1 \) and apply equation 17a.

\[
\frac{P(D_n | D_1)}{P(D_n | D_1^c)} = \frac{P(D_2 | D_1) + \left[ \frac{P(D_n | D_2)}{P(D_n | D_2^c)} - 1 \right]^{-1}}{P(D_2 | D_1^c) + \left[ \frac{P(D_n | D_2)}{P(D_n | D_2^c)} - 1 \right]^{-1}};
\]

To find \( P(D_n | D_2) / P(D_n | D_2^c) \) we let \( j = n \) and \( i = 2 \), and apply Equation 17a. This process is repeated until \( P(D_n | D_{n-1}) / P(D_n | D_{n-1}^c) \) has been computed. Figure 5 shows the equation, generated by repeated application of Equation 17, for the adjusted likelihood ratio in a four-stage inference case. Fortunately, such equations need not be written if one has the aid of a computer; Equation
17 is easily programmed as a recursive function.

A number of empirical studies have been performed in which subjects' unaided cascaded inferences, or direct estimates of odds, likelihoods, or likelihood ratios for observable events not having direct logical connections to the hypothesis in question, have been compared to odds, likelihoods or likelihood ratios computed from probabilities appropriate to the task at hand (e.g. Snapper and Fryback (1971); Schum, DuCharme, and DePitts (1973); Youssef and Peterson (1973), Gettys, Kelly, and Peterson (1973)). All of these studies were performed using abstract two-stage inference tasks which can be modeled using the equations derived by Gettys and Wilke (1969) or Schum and DuChárm (1971). Schum and Martin (1980), wishing to study human performance in less abstract and more complex inference tasks, presented subjects with scenarios containing evidence of the sort used in court trials. Since many, though not all, of the tasks performed in juridical inference are of more than two stages and/or do not meet very restrictive conditional independence requirements, the Gettys and Wilke (1969) and Schum and DuCharme (1971) equations are inadequate to model fully the complexities of such tasks. The Kelly-Barclay (1973) algorithm, on the other hand, with its complete relaxation of conditional independence assumptions, is unwieldy for
modeling this "mixed" evidence. Therefore, specific models were developed to formalize the inference tasks necessary for each set of evidence. The logical structure of the task for one scenario is shown in Figure 6. $A_1^*$ is the report given by the first witness in the case; $A$ is the event to the occurrence of which the first witness testifies; $B_2^*$ is the report of the second witness that event $B$ has occurred; and $B$ is the occurrence of event $B$, and so on; $H_1$ and $H_2$ are the hypotheses of defendant's guilt and his/her innocence, respectively. Arrows connecting the various elements (known events, intermediate event classes, and the hypothesis class) of the structure represent nonindependencies. These nonindependencies may be either conditional or unconditional (stochastic). For example, the arrows linking each of the classes $\{A,A^C\}$, $\{B,B^C\}$, and $\{D,D^C\}$ to the hypothesis class represent unconditional nonindependencies; i.e. $P(A|H_1)$ is not assumed to equal $P(A)$, $P(B|H_1)$ is not assumed to equal $P(B)$, and so on. Similar nonindependencies are represented by the arrows linking $A_1^*$ to $\{A,A^C\}$, $B_2^*$ to $\{B,B^C\}$, $E_3^*$ to $\{E,E^C\}$, $\{E,E^C\}$ to $\{D,D^C\}$, and $D_4^*$ to $\{D,D^C\}$. The arrow linking $B^*$ to $A^*$ represents a conditional nonindependency. $A^*$ and $B^*$ are not conditionally independent given $H_1$ or $H_2$; that is, we may not assume that $P(B^*|H_1A^*)$ equals $P(B^*|H_1)$. The linkage
between \( \{B, B^C\} \) and \( \{A, A^C\} \) is similar; the members of \( \{B, B^C\} \) are not conditionally independent of the members of \( \{A, A^C\} \) given \( H_1 \) or \( H_2 \); also similar is the linkage between \( D_4^* \) and \( E_3^* \). The linkage between \( \{E, E^C\} \) and \( \{H_1, H_2\} \) represents a lack of conditional independence between the members of \( \{E, E^C\} \) and the members of \( \{H_1, H_2\} \) given \( D \) or \( D^C \), i.e. we may not assume that \( P(E|DH_1) = P(E|D) \). Arrows are not present for all nonindependencies among the elements of the structure. For example, one might expect that none of the reports \( A_1^*, B_2^*, E_3^* \), and \( D_4^* \) is stochastically independent of the hypothesis class, although no arrows link these reports to \( \{H_1, H_2\} \). This is because each report is conditionally independent of the hypothesis class in the chain of reasoning from the report to the hypothesis class.

The inference structure determines the probabilistic ingredients which must be assessed, as well as a formally appropriate algorithm for combining these conditional probabilities to compute a likelihood ratio for each report in the structure. An arrow leading away from an event (report) or event class tells us that the probability of the event or member of the event class must be assessed conditional upon the events or members of event classes (the hypothesis class is an event class) at the other end of the arrow. The number of arrows leading away from an
event or event class is the number of conditioning events in each conditional probability assessed for the event or event class. To determine the number of conditional probability assessments which are formally necessary for a given event, we find $n$, the number of arrows leading away from the event, and $k$, the number of the $n$ arrows which lead to known events (in this case, reports). The number of assessments necessary is $2^{n-k}$. For example, the probability of $B^*$ must be assessed conditional upon each possible combination of its conditioning events. This means that we must assess $P(B^*|A^*B)$ and $P(B^*|A^*B^C)$, a total of two or $2^{2-1}$ assessments. For $B$, we assess $P(B|AH_1)$, $P(B|A^C H_1)$, $P(B|AH_2)$, and $P(B|A^C H_2)$ for a total of four, or $2^{2-0}$ assessments. Note that there is no need to assess the corresponding conditional probabilities of $B^C$. All of the intermediate event classes are binary, i.e. composed of two mutually exclusive and exhaustive events; knowledge of the conditional probabilities for one member of the class determines the corresponding conditional probabilities for the other member. The binary nature of the event classes serves to decrease both the number of conditional probability assessments which must be made and the number of terms in the formulae for computation of adjusted likelihood ratios for the various reports.
The likelihood ratio for \( R \equiv \{A^*_1, B^*_2, E^*_3, D^*_4\} \), the set of testimony for the case represented by Figure 6, is

\[
\Lambda_R = \Lambda_{A^*_1} \Lambda_{B^*_2} \Lambda_{E^*_3} \Lambda_{A^*_1B^*_2} \Lambda_{D^*_4} \Lambda_{A^*_1B^*_2E^*_3}
\]

However, since both \( E^*_3 \) and \( D^*_4 \) are conditionally independent of \( A^*_1 \) and \( B^*_2 \) given \( H_1 \) or \( H_2 \),

\[
\Lambda_{E^*_3|A^*_1B^*_2} = \Lambda_{E^*_3}, \quad \text{and} \quad \Lambda_{D^*_4|A^*_1B^*_2E^*_3} = \Lambda_{D^*_4|E^*_3}, \quad \text{so that}
\]

\[
\Lambda_R = \Lambda_{A^*_1} \Lambda_{B^*_2} \Lambda_{E^*_3} \Lambda_{A^*_1B^*_2} \Lambda_{D^*_4} \Lambda_{E^*_3}
\]

Figure 7 shows formally appropriate equations for each of the adjusted likelihood ratios which are multiplied to obtain \( \Lambda_R \). The study described by Schum and Martin (1980) involved having subjects respond to twelve cases, each of which could be modeled by one of six different inference structures of the type exemplified by the structure in Figure 6. Each of the six structures represents a different selection and mixture of evidence items varying in number of stages of inference and patterns of conditioning. The average number of evidence items per structure is five; seventeen different equations, of which four are shown in Figure 7, are necessary to model the formally appropriate inference processes for the collection of six cases.
Principles for the Derivation of Adjusted Likelihood Ratios

Derivation of specific formulae for the computation of adjusted likelihood ratios is tedious, as is programming them in order to analyze data, aid inference, and/or perform sensitivity analyses. The usual procedures for such derivation begins with an application of the definition for conditional probability. In deriving a formula for the likelihood ratio of \( A_i^* \) in Figure 6, for example, we find that, for \( i = 1,2 \)

\[
P(A^*|H_i) = \frac{P(A^*H_i)}{P(H_i)}
\]

Next, the compound event of which the probability is in the numerator is expressed as the disjoint union of the members of a partition. This is achieved by taking the intersection of the compound event with the disjoint union of the members of the partition composed of the possible combinations of events (other than members of the compound event in question) which condition the event for which the likelihood ratio is being found, or which condition events which condition the event for which the likelihood ratio is being found. For \( A_i^* \), the only further events which must be considered are the members of \( \{A_iA^C_i\} \). Thus the desired expression of \( A_i^*H_i \) is \( A_i^*H_i \cap (A_iA^C_i) \), which is equivalent to \( A_i^*AH_i \cup A_i^*A^CH_i \), and since the probability of the
disjoint union of a number of events is the sum of the probabilities of the events, \( P(\mathcal{A}_i^* \mathcal{A}_i \cup \mathcal{A}_i^* \mathcal{A}_i^C \mathcal{H}_i) = P(\mathcal{A}_i^* \mathcal{A}_i \mathcal{H}_i) + P(\mathcal{A}_i^* \mathcal{A}_i^C \mathcal{H}_i) \).

\[
P(\mathcal{A}_i^* | \mathcal{H}_i) = \frac{P(\mathcal{A}_i^* \mathcal{A}_i \mathcal{H}_i) + P(\mathcal{A}_i^* \mathcal{A}_i^C \mathcal{H}_i)}{P(\mathcal{H}_i)}
\]

The next step is to apply the product rule to both numerator and denominator as many times as is necessary in order to obtain appropriate conditional probabilities. In our example, repeated application of the product rule yields

\[
P(\mathcal{A}_i^* | \mathcal{H}_i) = \frac{P(\mathcal{A}_i^* | \mathcal{A}_i \mathcal{H}_i)P(\mathcal{A}_i | \mathcal{H}_i)P(\mathcal{H}_i) + P(\mathcal{A}_i^* | \mathcal{A}_i^C \mathcal{H}_i)P(\mathcal{A}_i^C | \mathcal{H}_i)P(\mathcal{H}_i)}{P(\mathcal{H}_i)}
\]

Next, we simplify the formula; \( P(\mathcal{H}_i) \) cancels out, and \( P(\mathcal{A}_i^* | \mathcal{A} \mathcal{H}_i) = P(\mathcal{A}_i^* | \mathcal{A}) \), so

\[
P(\mathcal{A}_i^* | \mathcal{H}_i) = P(\mathcal{A}_i^* | \mathcal{A})P(\mathcal{A} | \mathcal{H}_i) + P(\mathcal{A}_i^* | \mathcal{A}_i^C)P(\mathcal{A}_i^C | \mathcal{H}_i).
\]

Since \( P(\mathcal{A}_i^C | \mathcal{H}_i) = 1 - P(\mathcal{A} | \mathcal{H}_i) \), we can express \( P(\mathcal{A}_i^* | \mathcal{H}_i) \) as

\[
P(\mathcal{A}_i^* | \mathcal{A})P(\mathcal{A} | \mathcal{H}_i) + P(\mathcal{A}_i^* | \mathcal{A}_i^C)[1 - P(\mathcal{A} | \mathcal{H}_i)],
\]

which equals
\[ P(A|H_1)\left[P(A^*|A) - P(A^*|A^c)\right] + P(A^*|A^c), \]

so that

\[
\]

\[
P(A|H_1) + \left[\frac{P(A^*|A)}{P(A^*|A^c)} - 1\right]^{-1}
\]

\[
= \frac{P(A|H_1) + \left[\frac{P(A^*|A)}{P(A^*|A^c)} - 1\right]^{-1}}{P(A|H_2) + \left[\frac{P(A^*|A)}{P(A^*|A^c)} - 1\right]^{-1}}.
\]

(18)

The example used above is a two-stage inference task; the procedure described can be used for a task with any number of stages. However, application to a very long inference chain is rather time-consuming. It is thus desirable to simplify the procedure in whatever ways are possible. One method is suggested by the following result.

**Theorem 1:** For any events \( D_i \) and \( D_j \), and any binary event class \( \mathcal{D}_k \equiv \{D_k,D_k^c\} \), \( P(D_j|D_i) = P(D_k|D_i)[P(D_j|D_kD_i) \]

\[ - P(D_j|D_k^cD_i)] + P(D_j|D_k^cD_i). \]

The proof is simple, merely following the procedure outlined above.
Proof: By the definition of a conditional probability,

\[ P(D_j | D_i) = \frac{P(D_j D_i)}{P(D_i)} . \]

Since \( D_k \equiv \{ D_k, D_k^C \} \) is a partition, \( P(D_j D_i) = P[D_j D_i \cap (D_k \cup D_k^C)] \), which by the distributive law for intersection, equals \( P(D_j D_k D_i \cup D_j D_k^C D_i) \). Since the probability of the disjoint union of two events is the sum of the probabilities of the events, \( P(D_j D_i) = P(D_j D_k D_i) + P(D_j D_k^C D_i) \).

Applying the product rule, we find that \( P(D_j D_i) = P(D_j | D_k D_i)P(D_k D_i) + P(D_j | D_k^C D_i)P(D_k | D_i)P(D_i) \), so that \( P(D_j D_i) = P(D_j | D_k D_i)P(D_k D_i) + P(D_j | D_k^C D_i)P(D_k | D_i) \). Since \( P(D_k | D_i) = 1 - P(D_k^C | D_i) \), we find that \( P(D_j | D_i) = P(D_j | D_k D_i)P(D_k D_i) + P(D_j | D_k^C D_i)[1 - P(D_k^C | D_i)] \), which equals \( P(D_k | D_i)[P(D_j | D_k D_i) - P(D_j | D_k^C D_i)] + P(D_j | D_k D_i) \).

The above result is a generalization of Equation 17, requiring no conditional independence assumptions. To demonstrate its utility, assume that the classes \( D_k \), \( 1 \leq k \leq n-1 \) in Figure 8 are binary. Begin by letting \( D_k \equiv D_1, D_j \equiv D_n, \) and \( D_i \equiv D_0 \). Applying Theorem 1, we find that

\[ P(D_n | D_0) = P(D_1 | D_0)[P(D_n | D_1 D_0) - P(D_n | D_1^C D_0)] + P(D_n | D_1^C D_0). \]
Now we wish to find \( P(D_n | D_1D_0) \) and \( P(D_n | D_{1C}D_0) \). To find \( P(D_n | D_1D_0) \), we hold \( j \) fixed at \( n \), increment \( k \) to 2, and let \( D_1 \equiv D_1D_0 \), a compound event. Then applying Theorem 1, we find that

\[
P(D_n | D_1D_0) = P(D_2 | D_1D_0)[P(D_n | D_2D_1D_0) - P(D_n | D_{2C}D_1D_0)] + P(D_n | D_{2C}D_1D_0).
\]

\( P(D_n | D_{1C}D_0) \) is found similarly, letting \( D_1 \) be the compound event \( D_{1C}D_0 \):

\[
P(D_n | D_{1C}D_0) = P(D_2 | D_{1C}D_0)[P(D_n | D_2D_{1C}D_0) - P(D_n | D_{2C}D_{1C}D_0)] + P(D_n | D_{2C}D_{1C}D_0).
\]

The next terms that we wish to be able to find are \( P(D_n | D_2D_1D_0) \), \( P(D_n | D_{2C}D_1D_0) \), \( P(D_n | D_2D_{1C}D_0) \), and \( P(D_n | D_{2C}D_{1C}D_0) \). This can be done by holding \( j \) at \( h \), letting \( k = 3 \), and applying Theorem 1 four times, once letting \( D_1 \equiv D_2D_1D_0 \), once letting \( D_1 \equiv D_{2C}D_1D_0 \), and so on; the results of the four applications of Theorem are eight terms which must be defined, requiring eight applications of the theorem.

The process is continued until \( k = n-1 \), and \( 2^n \) terms have been defined.

Theorem 1 provides useful equations for likelihoods under many different conditioning patterns. Consider the
inference structure in Figure 9. We need not assume that
$$D_{n_1}$$
is conditionally independent, given $$D_{n_1-1}$$ or
$$D_{n_1-1}^C$$, of $$D_0$$ or of the members of any of the classes
$$\mathcal{D}_{q_1} \equiv \{ \mathcal{D}_{q_1}^C \}$$ for $$1 \leq q < n-1$$, or that $$D_{n_1+n_2}$$ is con-
ditionally independent, given $$D_{n_1+n_2-1}$$, of $$D_0$$ or of the
members of $$\mathcal{D}_{q_2}$$ for $$n+1 \leq q_2 < n+n-1$$, and so on in
order to obtain using Theorem 1, formally appropriate
and useful forms for the likelihoods of the various events
using Theorem 1. In fact, so long as, for any
$$D_{n_1+n_2+\ldots+n_r}, \ 2 \leq r \leq m$$, either

1) $$D_{n_1+n_2+\ldots+n_r}$$ and the members of $$\mathcal{D}_{q_r}$$, for
$$n_1+n_2+\ldots+n_{r-1} \leq q_r < n+n+\ldots+n_r$$ are
conditionally independent given $$D_0$$, of the
members of $$\mathcal{D}_{q_1}, \mathcal{D}_{q_2}, \ldots$$ and $$\mathcal{D}_{q_{r-1}}$$, for
$$1 \leq q_1 < n_1, \ n_1+1 \leq q_2 < n_1+n_2$$, and so on,
or

2) $$D_{n_1+n_2+\ldots+n_r}$$ and the members of $$\mathcal{D}_{q_r}$$,
$$n_1+n_2+\ldots+n_{r-1}$$ are conditionally independent,
given $$D_0$$, of $$D_{n_1}, D_{n_1+n_2}, \ldots, D_{n_1+n_2+\ldots+n_{r-1}}$$,

the form of the equation for the likelihood obtained using
Theorem 1 is valid and useful. However, if neither 1) nor
2) above holds, application of Theorem 1 yields a likelihood
ratio which, while formally appropriate, has a feature which is undesirable in empirical applications.

This can be demonstrated using the report $B_2^*$ in Figure 6. Since $B_2^*$ is not assumed to be conditionally independent given $H_1$ or $H_2$, of $A_1^*$, a likelihood of $B_2^*$ is in fact $P(B_2^*|A_1^*H_1)$. Applying Theorem 1, letting $D_1 = A_1^*H_1$, $D_j = B_2^*$, and $D_k = B$, we obtain

$$P(B_2^*|A_1^*H_1) = P(B|A_1^*H_1)[P(B_2^*|BA_1^*H_1) - P(B_2^*(B^C A_1^*H_1))]$$

$$+ P(B_2^*|B^C A_1^*H_1).$$

We now wish to find $P(B|A_1^*H_1)$. We let $D_j = B$, $D_i = A_1^*H_1$, and, since $B$ and $B^C$ are not assumed to be conditionally independent of $A$ or $A^C$ given $H_1$ or $H_2$, we let $D_k = A$. Then

$$P(B|A_1^*H_1) = P(A|A_1^*H_1)[P(B|A A_1^*H_1) - P(B|A^C A_1^*H_1)]$$

$$+ P(B|A^C A_1^*H_1).$$

We need not complete our derivation of a decomposition formula for $P(B^*|A_1^*H_1)$, as the undesirable feature of the formula has already been exposed: the term $P(A|A_1^*H_1)$ is a posterior probability, and we wish to assess only likelihoods. Therefore, we need a different generalization for situations in which neither 1) nor 2) above holds.
**Theorem 2:** For any events $D_i, D_j, D_k$, and any binary event class $D_m = \{D_m, D_m^c\}$,

\[
P(D_j | D_k D_i) = \frac{P(D_j | D_k D_i^c)}{P(D_k | D_i)} \cdot \frac{P(D_k D_m D_i)}{P(D_k D_i)} - \frac{P(D_j | D_m D_i^c)}{P(D_k | D_i)} \cdot \frac{P(D_k D_m D_i)}{P(D_k D_i)} + \frac{P(D_j | D_m D_i^c)}{P(D_k | D_i)} \cdot \frac{P(D_k D_m D_i)}{P(D_k D_i)} \bigg/ P(D_k | D_i).
\]

**Proof:** First apply the definition of a conditional probability, obtaining

\[
P(D_j | D_k D_i) = \frac{P(D_j D_k D_i)}{P(D_k D_i)}.
\]

Now use $D_m$ and $D_m^c$ to partition $D_j D_k D_i$.

\[D_j D_k D_i \equiv D_j D_k D_m D_i \cup D_j D_k D_m D_i^c, \quad \text{and}\]

\[
P(D_j | D_k D_i) = \frac{P(D_j D_k D_m D_i) + P(D_j D_k D_m D_i^c)}{P(D_k D_i)}.
\]

Applying the product rule, we obtain

\[
P(D_j | D_k D_i) = \left\{ \frac{P(D_j | D_k D_m D_i) P(D_k | D_m D_i) P(D_m | D_i) P(D_i)}{P(D_k | D_i) P(D_m | D_i) P(D_m | D_i)} + \frac{P(D_j | D_k D_m D_i^c) P(D_k | D_m D_i^c) P(D_m | D_i^c) P(D_m | D_i)}{P(D_k | D_i) P(D_m | D_i) P(D_m | D_i)} \right\} / P(D_k | D_i) P(D_m | D_i) P(D_m | D_i)
\].
Since $P(D_i)$ occurs in both numerator terms and in the denominator, it cancels out $P(D_m^c|D_i^c) = 1 - P(D_m|D_i)$, so that

$$P(D_j|D_k D_i) = \left\{ P(D_m|D_i)P(D_j|D_k D_m D_i)P(D_k|D_m D_i) ight\}$$

$$+ \left\{ P(D_m|D_i)[P(D_j|D_m D_i)P(D_k|D_m D_i)]/P(D_m|D_i) \right\}$$

$$= \left\{ P(D_m|D_i)[P(D_k|D_m D_i)P(D_j|D_k D_m D_i)] - P(D_k|D_m D_i)P(D_j|D_k D_m D_i) \right\}$$

$$+ P(D_k|D_m D_i)P(D_j|D_k D_m D_i)}/P(D_k|D_i).$$

For an inference structure such as the one shown in Figure 6, information is normally processed sequentially. Therefore, when $Λ_{B^*_2|A^*_1}$ is computed, the value of $Λ_{A^*_1}$ has already been determined. Finding $Λ_{B^*_2|A^*_1}$ requires application of both Theorem 1 and Theorem 2. In the following example, a formula will be found only for the likelihood $P(B^*_2|A^*_1 H_1)$; $Λ_{B^*_2|A^*_1} = P(B^*_2|A^*_1 H_1)/P(B^*_2|A_1 H_2)$, and $P(B^*_2|A_1^* H_2)$ is found in exactly the same fashion as $P(B^*_2|A^*_1 H_1)$. To derive a formula for $P(B^*_2|A_1^* H_1)$, first apply Theorem 2, letting $D_i ≡ H_1$, $D_k ≡ A^*_1$, $D_j ≡ B^*_2$, and $Y_m ≡ \{A, A^c\}$. It is found that
\[ P(B_2^*|A_H^1) = \left\{ P(A_H^1)\left[ P(A_1^*|A_H^1)P(B_2^*|A^*A_H^1) - P(A_1^*|A_C^1)P(B_2^*|A_1^*A_C^1) \right] \right\} \]

\[ + \frac{P(A_1^*|A_C^1)P(B_2^*|A_1^*A_C^1)}{P(A_1^*|H_1)}. \]

Completion of the derivation yields a rather lengthy equation. Suffice it to point out that to obtain the finished product, two applications of Theorem 1 are necessary. In both applications, let \( D_j \equiv B_2^* \), and \( D_k \equiv B \); for the first application, let \( D_i \equiv A^*A_H^1 \), and for the second let \( D_i \equiv A^*A_C^1 \).

The terms \( P(B_2^*|A_1^*A_H^1) \) and \( P(B_2^*|A_1^*A_C^1) \) are replaced by the results of the applications of Theorem 1.

A combination of the principles embodied in Theorems 1 and 2 makes it possible to specify a likelihood ratio equation or set of likelihood equations for any cascaded inference task in which there are two mutually exclusive hypotheses and all possible intermediate events belong to binary classes. The drawback to strict use of these principles, however, is similar to one of the disadvantages of the Kelly-Barclay (1973) model in that the provider of likelihood estimates may be called upon to supply many more estimates than are in fact necessary for formal appropriateness. In the example above, for instance, if care were not taken, the estimator might be asked for
\( P(A^*|A H_1) \), \( P(A^*|A^C H_1) \), \( P(A^*|A H_2) \), and \( P(A^*|A^C H_2) \),
even though \( A^* \) is conditionally independent of \( H_1 \) and \( H_2 \) given \( A \) or \( A^C \) and thus the two estimates \( P(A^*|A) \) and \( P(A^*|A^C) \) would suffice. This problem can be avoided if decisions about the likelihoods to be assessed are made using the inference structure rather than the likelihood ratio equations. One simply assesses the probability of a given event conditional only upon those events belonging to event classes to which there are arrows from the event in question.
A Computer Program for Inference Aiding

Once an appropriate structure for the inference task has been determined, most of the aiding function can be allocated to a computer. In this section, a description will be given of the necessary preparatory work for such allocation, along with the ways in which the inference task can be computer aided.

As is usual in any type of problem-solving, the problem must first be formulated. Computer use requires that the problem be formulated in very specific terms; for the aiding functions discussed here, the formulation must be embodied in an appropriate inference structure.

The initial components of the inference structure are known events and hypotheses. One begins to build the structure by defining any intermediate events or intervening variables which mediate the relationships of events and hypotheses. After the intermediate events have been defined, they are placed in chains between the known events and the hypotheses in such an order that, for the event classes in a given chain from a known event to the hypothesis, if event class $X$ mediates the relationship between event $Y$ and the hypothesis, the class $X$ occupies a position which is closer to the hypothesis than is the position of event or class $Y$. For every chain, one arrow
is drawn to the hypothesis from the event class closest to
the hypothesis. The arrows added in this process repre-
sent stochastic nonindependencies. There are certain to
be other stochastic nonindependencies among events in the
structure; it is desirable to express these other stochas-
tic nonindependencies using patterns of conditional inde-
pendency and nonindependency. Therefore, the next step
in the building of the structure is conditional
nonindependency checking. Beginning at the bottom of a
given chain, we check to see whether event $X$ is condi-
tionally independent of the members of class $Y$, which
is two or more stages closer to the hypothesis, given the
possible combinations of the members of the classes occur-
ring between $X$ and $Y$ in the chain. For this purpose
we consider the hypothesis class to be the event class
closest to the hypothesis. Any conditional nonindependen-
cies found are represented using arrows. A similar process
is used in checking for conditional nonindependencies bet-
ween classes of events occurring in different chains; if
event $X$ is not conditionally independent of some event
$Y$ in another chain given the possible combinations of the
events above $X$ in the chain to which $X$ belongs, the
conditional nonindependency is represented using an arrow.

For the aiding program described here, it is
assumed that likelihood information is to be processed
sequentially, so that in Figure 6, for example, $A^*_A$ would be determined first; this would be followed by computation of $A^*_B|A^*_i$, and so on. The inference structure must be built with this in mind. One requirement is that any nonindependencies between two events be expressed using an arrow from the event which appears later in the structure to the event occurring previously, where heuristically, $X$ occurs later in the structure than $Y$ if $Y$ belongs to the same chain as $X$ and is closer to the hypothesis or $Y$ belongs to a chain which occurs before (to the left of) the chain to which $X$ belongs.

The computer represents the inference structure as a matrix, which is possible because the structure is, or can be represented as, a graph, and a graph in turn can be represented by a matrix.

Kelly and Barclay (1973) required that the inference structure be representable as a tree, which is a special case of the type of graph that is required for appropriate use of Theorems 1 and 2 in computer applications, the directed acyclic graph, or acyclic digraph.

According to Harary, Norman, and Cartwright (1965), a digraph satisfies the following axiom system.

The primitives are:

$P_1$: A set $V$ of elements called "points".

$P_2$: A set $X$ of elements called "lines".
P₃: A function f whose domain is X and whose range is contained in V.

P₄: A function s whose domain is X and whose range is contained in V.

The axioms are:

A₁: The set V is finite and not empty.

A₂: The set X is finite.

A₃: No two distinct lines are parallel.

A₄: There are no loops.

The functions f and s identify the first and second points of a given line. Suppose, for example, that there is a line $x_1$ from some point $v_i$ to another point $v_3$. Then $fx_1$ is $v_i$ and $sx_1$ is $v_3$. Two lines $x_i$ and $x_j$ are parallel if $fx_i = fx_j$ and $sx_i = sx_j$. A loop is a line $x_i$ such that $fx_i = sx_i$.

Any line $x_i ∈ X$ is fully described as the ordered pair of $fx_i$ and $sx_i$. If, for example, $fx_i = v_j$ and $sx_i = v_k$, then $x_i$ is the ordered pair $(v_j, v_k)$, often written simply $v_j v_k$. Specification of lines in such a fashion allows us to define the term path; a path from $v_i$ to $v_q$ is a set of lines $v_i v_j v_k \ldots v_{n', p'} v_{p'} v_{q'}$ where the points $v_i, v_j, v_k, \ldots, v_{n', p'}, v_{p'}, v_{q'}$ are distinct from one another. A cycle is a path from $v_i$ to $v_q$ along with a line from $v_q$ to $v_i$; i.e. a cycle is a set of
lines \( v_i v_j v_k \), \( \ldots \), \( v_n v_p v_q v_i \), where the points \( v_i, v_j, v_k, \ldots, v_p, v_q \) are distinct from one another. An **acyclic digraph** is a digraph which contains no cycles.

To represent a digraph as a matrix, the points \( v_1, v_2, v_3, \ldots, v_k, \ldots, v_n \) are arranged in numerical order of their subscripts. The adjacency matrix, \( A \) is an \( n \times n \) matrix in which an entry \( a_{ij} \) is equal to one if the digraph contains a line \( x \in X \) such that \( fx = v_i \) and \( sx = v_j \); otherwise \( a_{ij} = 0 \). We note that the diagonal elements at the adjacency matrix of a digraph must all equal zero, since there are no loops in a digraph.

A digraph is a special case of a **relation** for which the axiom system does not disallow loops. In a relation, the existence of line \( x \in X \) implies that \( fx \text{ Rsx} \).

Figure 10 shows a digraph representation of the case of which the structure is shown in Figure 6; a line in this digraph signifies the relation "is conditioned by". \( H_1 \) is represented by \( v_0 \), \( A \) is represented by \( v_1 \), and \( A^*_1 \) is represented by \( v_2 \); \( B \) is represented by \( v_3 \), \( B^*_2 \) is represented by \( v_4 \), and so on; \( D^*_4 \) is represented by \( v_8 \). Since all of the intermediate event classes are assumed to be binary, no generality is lost when a non-independency between two classes is expressed as a nonindependency between one member of one class and one member of
the other class; the other three nonindependencies are implied. The hypothesis $H_1 (v_0)$ and the known events $A_i^* (v_2)$, $B_i^* (v_4)$, $E_i^* (v_7)$, and $D_i^* (v_9)$ are distinguished from the intermediate events; this is merely a convention to be used for inference structures, since it is desirable to know what inferences are to be made. Another convention which is necessary has to do with the numbering of the vertices (points) in the graph. The hypothesis is always designated $v_0$; for any two events lying on a given path to the hypothesis, if event $X$, represented by $v_j$ lies on the path from $Y$ (represented $v_i$) to the hypothesis, then $j < i$, and otherwise $i < j$. For any two events $X (v_j)$ and $Y (v_i)$ which do not share a path to the hypothesis, if the path from $X$ to the hypothesis lies to the left of the path from $Y$ to the hypothesis, then $j < i$; otherwise $j > i$. The idea of "left" means nothing in graph theory; however, in an inference structure it is very important, as the spatial relationship of paths determines the order of information processing. The adjacency matrix for the structure in Figures 6 and 10 is shown in Figure 11. The adjacency matrix of an inference structure will be called a structure matrix. The structure matrix shown in Figure 11 is easily seen to be lower triangular. This condition insures that the digraph is acyclic, and is required for the type of computer aiding to be discussed
here; it is achieved by previously discussed specifications for the expression of nonindependency and ordering of vertices. The requirement that, in an inference structure, arrows may run only upward or to the left, treats "is conditioned by", a relation which is in fact symmetric, as an antisymmetric relation; if the relation were treated as symmetrical, the digraph would not be acyclic.

If the task of inference aiding is to be automated, the inference structure must be communicable to the computer. One way to do this, of course, is simply to input the structure matrix. Alternatively, as is the case in CASPRO, an interactive PL/I program for inference aiding, one may first input the names of the hypotheses and known events, and then go through the inference structure in vertex-subscript order entering each event along with the names of its conditioning events. The program uses character-string matching to build the structure matrix. Once the structure has been communicated in this fashion, the program can elicit the decomposed likelihoods necessary for computation of adjusted likelihood ratios. For any intermediate or known event in the structure, \( 2^{od-k} \) decomposed likelihoods must be elicited, where \( od \) is the out-degree of the event, the number of arrows originating at the event, and \( k \) is the number of arrows which lead from the event to known events. In order to elicit the appropriate likelihoods,
The program first generates all possible combinations of the members of intermediate event classes and the hypothesis class which receive an arrow from the event of which likelihoods are being elicited; then, for each possible combination, the user is asked to give the probability of the event conditional upon any known events and the combination. For any event, likelihoods are elicited in last-moves-fastest order, where "last" refers to the event having the highest vertex subscript. The likelihoods are stored in the order in which they are elicited in a two-dimensional array $L$ (not to be confused with $L_D$, the likelihood ratio of D), in which row $i$ contains the likelihood estimates for the event having vertex subscript $i$. For example, when likelihoods have been elicited for $A, A^*, B_1$ and $B_2^*$ in Figure 6, $L$ should have the following appearance containing a value for each probability term:

$$
\begin{bmatrix}
P(A|H_1) & P(A|H_2) \\
P(A^*|A) & P(A^*|A^C) \\
P(B|H_1A) & P(B|H_1A^C) & P(B|H_2A) & P(B|H_2A^C) \\
P(B_2^*|A^*B) & P(B_2^*|A^*B^C)
\end{bmatrix}
$$

After all of the decomposed likelihoods are assessed, the program should calculate a likelihood ratio for each of the known events using computational versions of Theorems 1 and 2.
However, the number of likelihoods elicited for any event may be fewer than the number required for computation. Suppose our entire structure were composed of $H_1$, $A$, and $A^*$ from Figure 6; then the likelihood ratio equation derived using Theorem 1 would be:

$$\Lambda_{A^*_1} = \frac{P(A|H_1)[P(A^*_1|AH_1)-P(A^*_1|ACH_1)]+P(A^*_1|ACH_1)}{P(A|H_2)[P(A^*_1|AH_2)-P(A^*_1|ACH_2)]+P(A^*_1|ACH_2)}.$$

If conditional independence of $A^*_1$ and $\{H_1,H_2\}$ given $A$ or $A^C$ could not have been assumed, then $\ell$ would be

$$\begin{bmatrix}
P(A|H_1) & P(A|H_2) \\
P(A^*_1|H_1A) & P(A^*_1|H_1A^C) & P(A^*_1|H_2A) & P(A^*_1|H_2A^C)
\end{bmatrix}.$$ 

The computer algorithm for likelihood ratio computation based on Theorem 1 includes the instruction

$$\ell_{ij} = \ell_{ij} \times (\ell_{i+1,2j-1} - \ell_{i+1,2j}) + \ell_{i+1,2j}.$$

Thus for the $\ell$ shown above,

$$\ell_{11} = \ell_{11} (\ell_{21}-\ell_{22}) + \ell_{22}, \text{ and }$$

$$\ell_{12} = \ell_{12} (\ell_{23}-\ell_{24}) + \ell_{24},$$

and $\Lambda_{A^*_1} = \ell_{11}/\ell_{12}$.
When conditional independence of $A_i^*$ and $\{H_1, H_2\}$ gives $A$ or $A^C$ can be assumed, $L$ is

$$
\begin{bmatrix}
P(A|H_1) & P(A|H_2) \\
P(A^*|A) & P(A^*|A^C)
\end{bmatrix},
$$

and the computational algorithm cannot be applied, since $\ell_{23}$ and $\ell_{24}$ are undefined. Once all of the likelihoods have been elicited, however, the purpose of the conditional independence assumptions has been served, as the number of assessments has been minimized. The program proceeds to "unassume" many of the independencies using processes which can be given the name **pseudo-conditioning**.

The pseudo-conditioning procedures are most easily explained in terms of the acyclic digraph which is isomorphic to the inference structure. The first stage of pseudo-conditioning begins with $v_n$, the vertex corresponding to the last event in the structure. If the set of all $x_i \in X$ such that $fx_i = v_n$ contains more than one line, then the set of $v_j \in V_J \subseteq V$ such that $sx_i = v_j$ contains more than one point. A target event $v_t$ is selected by taking the maximum over the subscripts $j$ for the set $V_J$; i.e. $v_t = V_J(\max j)$. Next the set of $v_j$ is divided into two subsets, the first being the set $V_K \subseteq V$, such that, for the set of $x_i$ for which $fx_i = v_n$, $sx_i = v_k \in V_K$ if
and only if \( s_{x_i} \neq v_t \). The second subset consists simply of the vertex \( v_t \). Now the target event is checked for conditioning against \( v_c = v_{(\max_k k)} \); if there is no line \( x_i \in X \) such that \( f_{x_i} = v_t \) and \( s_{x_i} = v_c \), then pseudo-conditioning must occur. If \( v_c \) represents a known event, then element \( a_{tc} \) in the structure matrix is changed from zero to one; this is equivalent to inserting line \( v_tv_c \) in the digraph. If \( v_c \) does not represent a known event, then the number of entries in row \( t \) of the array \( L \) is doubled by duplicating the entries already present and arranging them in an appropriate order; and \( a_{tc} \) in the structure matrix is changed from zero to one. Now the procedure is performed substituting \( v_{n-1} \) for \( v_n \) and using the revised \( A \) (structure matrix) and \( L \). When the procedure has been performed for all \( v_i \) such that \( i = n, \ n-1, \ n-2, \ldots, 3,2, \) the first stage of pseudo-conditioning is complete. The second stage begins by setting \( v_b \) to \( v_0 \), which represents a member of the hypothesis class. Initially, the vertex \( v_c \), representing the first event which "is conditioned by" \( v_b \), is found; \( v_c = v_{(\min_j j)} \) where \( v_j \in V_j \subset V \) if and only if there exists a line \( x_i \in X \) such that \( s_{x_i} = v_b \) and \( f_{x_i} = v_j \). Once \( v_c \) has been identified, it must be insured that for every vertex \( v_k \) for which the line \( v_kv_c \) exists, there also exists a line \( v_kv_b \). If there is no line \( v_kv_b \) for some \( v_k \), i.e.
there is no \( x_i \in X \) such that \( fx_i = v_k \) and \( sx_i = v_b \), then \( v_k \) must be pseudo-conditioned by \( v_b \). If \( v_b \) should happen to represent a known event, the pseudo-conditioning would consist merely of setting \( a_{kb} \) to 1. When \( v_b \) is not a known event, the number of entries in row \( k \) of \( L \) must be doubled as in stage one and \( a_{kb} \) is set to one. When pseudo-conditioning has been achieved for every \( v_k \) which "is conditioned by" \( v_c \), a new set \( V_j \) is found, excluding \( v_c \) and including all \( v_j \) such that \( j > c \) for which \( a_{jb} = 1 \) in the revised \( A \). Then a new \( v_c \) is found by the same method as the initial \( v_c \), and the checking for the existence of line \( v_b v_k \) for every \( k \) such that there is a line \( v_c v_k \) is repeated, pseudo-conditioning wherever necessary. The process of finding a new \( v_c \) and implementing any necessary pseudo-conditioning by \( v_b \) recurs until no further \( v_c \) can be found. At this point, the value of \( b \) is incremented by one and the entire process is repeated. The second stage processes are performed for every \( v_b \) such that \( 0 \leq b \leq n-2 \).

The effect of the two stages of pseudo-conditioning is to insure that, for every \( v_i > 0 \), if there is a path from \( v_i \) to \( v_j \), \( j < i \), then there is also a line \( v_i v_j \). The estimates contained in \( L \) reflect in number and order, but not in value, the new independence "unassumptions" made in the pseudo-conditioning process, and each row of \( L \)
contains the correct number of entries for application of a formally appropriate computational algorithm derived from Theorems 1 and 2. Figure 12 shows the structure matrix for the case represented in Figures 6 and 10 after all necessary pseudo-conditioning has been achieved. The array in Figure 13 is the revised array $L$ for a structure which contains only the subset $H_1, A, A^*_f, B$, and $B^*_f$ of the points in the original structure in Figure 6. For any given likelihood ratio calculation, the computational algorithm treats only the rows of $L$ which contain likelihoods of the events by which event $X$ is conditioned and the row which contains the likelihoods of $X$. These rows are placed into a buffer array, $B$.

The computational routine is, in words:

Start at the next-to-bottom row of the proper array $B$. If the event of which the likelihoods appear in this row as a known event, then replace each member $b_{ij}$ of the row $(i)$ with $b_{ij} \times b_{i+1,j}$;

If the event is not known, then each $b_{ij}$ in row $i$ is replaced by

$$b_{ij} \times (b_{i+1,2j-1} - b_{i+1,2j}) + b_{i+1,2j}$$

Repeat this procedure for rows $i-1$, $i-2$ and so on until $b_{11}$ and $b_{12}$ have been replaced.
Then if there are no likelihoods of known events above the last row, \( \Lambda = b_{11}/b_{2} \). Otherwise \( \Lambda \) is \( b_{11}/b_{12} \) divided by the product of the \( \Lambda \)'s for all known events for which there are likelihoods above the last row.

If \( \Lambda_{A_{1}^{*}} \) is to be computed, then

\[
B = \begin{bmatrix}
P(A|H_{1}) & P(A|H_{2}) \\
P(A^{*}|A) & P(A^{*}|A^{C}) & P(A^{*}|A) & P(A^{*}|A^{C})
\end{bmatrix}
\]

The next-to-bottom row is row 1; applying the routine, \( b_{11} \) takes on the value of \( P(A|H_{1})[P(A_{i}^{*}|A) - P(A_{i}^{*}|A^{C})] + P(A_{i}^{*}|A^{C}) \), and \( b_{12} \) takes on the value of \( P(A|H_{2})[P(A^{*}|A) - P(A_{i}^{*}|A^{C})] + P(A^{*}|A^{C}) \). Since \( b_{11} \) and \( b_{12} \) have been computed, no repetitions are necessary, and \( \Lambda_{A_{1}^{*}} = b_{11}/b_{12} \). If \( \Lambda_{B_{2}^{*}} \) is to be computed then the array \( B \) is the array \( L \) in Figure 13. The computational routine is applied for rows 3, 2 and 1 in turn; then \( \Lambda_{B_{2}^{*}} \) is \( b_{11}/b_{12} \)/\( \Lambda_{A_{1}^{*}} \), which is in fact \( \Lambda_{B_{2}^{*}}|A_{1}^{*} \).

For any structure, once a likelihood ratio has been calculated for every known event, the likelihood ratios are multiplied together to obtain the likelihood ratio for the entire case. Appendix 1 contains the output from a sample application of the program CASPRO to a numerical example for the case illustrated in Figure 6; Appendix 2 is a listing of the program.
Discussion

Theorems 1 and 2 can be used by human beings to derive equations for adjusted likelihood ratios, although it is not clear, in most cases, that strict application of the theorems results in substantial savings of time and/or effort for the person deriving the equations. It can be argued, of course, that for very complex inference tasks, use of the principles embodied in the theorems to decompose the task of equation derivation could result in a decrease in error rate, and thus a time savings. The major advantage of the two theorems, however, is the ease with which they can be translated into a generalized computational algorithm. Prototype computer program CASPRO, described in the previous section and listed in Appendix 2 implements the computational algorithm and renders the derivation of likelihood ratio equations unnecessary for many instances of inference aiding and empirical research in inference; in fact, the task of the human inference aider can often be reduced to building the inference structure and communicating it to the computer.

The strength of CASPRO is derived from two sources, the utilization of the structure matrix and the succinct generalized computational algorithm. When the structure matrix is used to elicit decomposed likelihoods, in many
cases the number of estimates necessary can be decreased substantially from the number of estimates necessary when no conditional independence assumptions are made for the events in a given path, as in the Kelly-Barclay model. In fact, if the hypothesis class is binary and the structure contains only those lines which are absolutely essential to the definition of relationships among the elements of the structure, the number of decomposed likelihoods elicited by CASPRO is the minimum necessary for formally appropriate likelihood ratio computation. Besides being used to decide which likelihoods to elicit, information contained in the structure matrix is used by the program to revise the contents of the matrix and to aid in insuring that the number of likelihoods for any event conforms to the number required by the computational algorithm. Finally, the structure matrix is used to determine exactly which likelihoods are fed to the computational algorithm. Since the computational algorithm can treat any set of likelihoods fed to it by the properly pseudo-conditioned inference structure, the task of choosing computational methods to determine likelihood ratios need not be performed by a person.

It should be pointed out that CASPRO, although it could be classified as a performance-oriented artificial intelligence program, is not designed as an artificial intelligence program. While the overt behavior of the
program may simulate that of a human inference analyst, and the program performs many of the tasks of the analyst, the internal functioning of the program is not meant to simulate human thought. In fact, the program is inefficient relative to the human analyst in some ways. For example, if a computed term occurs more than once in an equation, a human would normally compute the term once and use the value for all other occurrences of the term. CASPRO, on the other hand, often recomputes such terms. The program is not "aware" of what it is doing; however, the methods used by CASPRO are formally appropriate, and even though CASPRO uses many more information processing steps than would a person, it is still orders of magnitude faster than the human analyst.

As noted above, CASPRO is a prototype program; the features of CASPRO could be used in several different programs which might be useful to various consumers. One two-part program which could be of use to the empirical researcher would involve first eliciting the structure of some inference task to be performed by subjects and printing out a list of the likelihoods to be elicited from the subjects, then saving the structure for future use; at a later point, the likelihoods could be input to a three dimensional array \( L \) in which the third dimension represents "subject", pseudo conditioning operations would be
performed and likelihood ratios would be computed.

In another version, the program might save the original structure and L arrays, performing the pseudo-conditioning operations on duplicates which would be used for the computational routine. The original arrays could be used to accept changes in the values of the likelihood estimates for sensitivity analysis. This version would be especially useful to the inference aider.

Still another version might be helpful to the formal inference researcher. After the structure is input, the program would generate the character strings which in CASPRO are the names of the likelihoods to be input; the character strings would not be used to elicit likelihoods, but would be placed in an array similar to L in CASPRO; then instead of computing likelihood ratios, the program would use a routine similar to the computational routine to concatenate the character strings into longer strings containing parenthesis and symbols for multiplication, division, addition, and subtraction as connections; the result of such a program would be a set of likelihood ratio equations formally appropriate to the task.

In the examples used in the previous section, the hypothesis class could be assumed to be binary; however, no assumption to that effect has been made. So long as the hypothesis class is composed of mutually exclusive
events, CASPRO can be used to compute likelihood ratios based upon any pair of the hypotheses. A set of \( n-1 \) (where \( n \) is the number of members in the hypothesis class) likelihood ratios can yield sufficient information for the determination of likelihood ratios based upon every pair of hypotheses.

When the hypothesis class is not binary, it is possible for an independency assumption to hold for only a subclass of the hypothesis class. If CASPRO or a similar program is to be applicable in such a case, care must be taken that the inference structure contains lines which reflect any conditional nonindependency relationships involving either of the hypotheses upon which the likelihood ratio is based.

CASPRO can treat appropriately classes of tasks having structures less restricted than those required by Kelly and Barclay (1973), whose algorithm is designed for inference trees. In this sense, CASPRO is more general than the Kelly-Barclay algorithm. However, the Kelly-Barclay algorithm can handle appropriately tasks having intervening variables for which there are more than two possible values while use of CASPRO requires that intermediate event classes be binary. The major disadvantages of the Kelly-Barclay algorithm are its incomprehensibility to the aider or researcher and the huge number of
decomposed likelihood estimates that are required. Both of these difficulties might be mitigated by application of some of the techniques used in CASPRO. An inference tree is an acyclic graph and can be expressed as a digraph. Thus it is possible to represent the tree as a lower triangular adjacency matrix. A program could be written to use such an adjacency matrix to elicit the adjusted likelihood ratios, pseudo-condition the events, duplicate estimates where necessary and apply the incomprehensible but programmable Kelly-Barclay algorithm in an appropriate fashion. As has been discussed previously, the existence of a conditional nonindependency relationship involving one member of a class composed of three or more events does not imply the involvement of every member of the class in such a relationship; however, for purposes of elicitation it would have to be assumed that all members of each class containing events involved in a conditional nonindependency relationship are involved in the same type of relationship. Thus the use of the structure matrix for likelihood elicitation is not guaranteed to minimize the number of estimates; however, the number of estimates necessary could be decreased substantially in many cases.

The techniques of task decomposition and formally appropriate reaggregation can be used for aiding in many different contexts in which inferences must be made on the
basis of inconclusive evidence. Prime examples of areas in which inference tasks are cascaded are intelligence analysis and medical diagnosis; however, there have been few applications of the formal theory of cascaded inference to the diagnostic tasks performed in these areas, partially because the complexities of task decomposition and of the decomposed task itself often seem to outweigh the advantages of using appropriate procedures. It is hoped that programs such as CASPRO, along with other computer applications using the theory developed in this paper, by rendering the use of formally appropriate procedures for aiding inference less difficult and time-consuming, will also render such procedures more acceptable both to decision analysis and to other people whose work involves complex inference tasks.
REFERENCES

Dodson, J. D., Simulation system design for a TEAS simulation research facility. AFCRL 1112, PCR-R-194, Planning Research Corporation, Los Angeles, California, 1961.


Martin, A. W., Cascaded inference and hearsay. Rice University Department of Psychology Research Report #79-03, 1 December 1979.


Schum, D. A., Martin, A. W., Empirical studies of cascaded inference in jurisprudence: Methodological consid-
erations. Rice University Department of Psychology Research Report #80-01, 30 May 1980.


FIGURE CAPTIONS

Figure 1. Kelly and Barclay's "single-path tree".

Figure 2. Schum's hearsay paradigm

Figure 3. Martin's structuring of the hearsay problem.

Figure 4. An n-stage inference chain.

Figure 5. A likelihood ratio equation for a four-stage inference task.

Figure 6. The structure of a complex inference task.

Figure 7. Equations for the computation of adjusted likelihood ratios of events in Figure 6.

Figure 8. The general n-stage inference chain.

Figure 9. A partially-defined generalized inference structure.

Figure 10. An acyclic digraph isomorphic to the inference structure in Figure 6.

Figure 11. A matrix representation of the structure in Figure 6.
Figure 12. The structure matrix for Figure 6 after pseudo-conditioning.

Figure 13. The array $L$ for part of Figure 6 after pseudo-conditioning.
Figure 1
Figure 2
Figure 3
Figure 4
Figure 6
\[ \Lambda_{A_1} = \frac{P(A^*|H_1)}{P(A^*|H_2)} = \frac{P(A|H_1) + \frac{P(A^*_1|A)}{P(A^*_1|A^c)} - 1}{P(A|H_2) + \frac{P(A^*_1|A)}{P(A^*_1|A^c)} - 1} \]

\[ \Lambda_{B_2^*|A_1^*} = \frac{P(B_2^*|A_1^*H_1)}{P(B_2^*|A_1^*H_2)} = \frac{K_1 [P(B|A_{H_1}) - P(B|A_{C_{H_1}^1})] + P(B|A_{C_{H_1}^1}) + \frac{P(B_2^*|A_1^*B)}{P(B_2^*|A_1^*B^c)} - 1}{K_2 [P(B|A_{H_2}) - P(B|A_{C_{H_2}^1})] + P(B|A_{C_{H_2}^1}) + \frac{P(B_2^*|A_1^*B)}{P(B_2^*|A_1^*B^c)} - 1} \]

Where: \[ K_k = P(A|A_{H_{H_k}}) = \frac{P(A|H_k) P(A^*_1|A)}{P(A|H_k) [P(A^*_1|A) - P(A^*_1|A^c)] + P(A^*_1|A^c)} \]

for \( k = 1,2 \).

\[ \Lambda_{E_3^*} = \frac{P(E_3^*|H)}{P(E_3^*|H)} = \frac{P(D|H_1) [P(E|DH_1) - P(E|D_{C_{H_1}^1})] + P(E|D_{C_{H_1}^1}) + \frac{P(E_3^*|E)}{P(E_3^*|E^c)} - 1}{P(D|H_2) [P(E|DH_2) - P(E|D_{C_{H_2}^1})] + P(E|D_{C_{H_2}^1}) + \frac{P(E_3^*|E)}{P(E_3^*|E^c)} - 1} \]

\[ \Lambda_{D_{4,3}^*|E_3^*} = \frac{P(D_{4,3}^*|E_{3,3}^*H_1)}{P(D_{4,3}^*|E_{3,3}^*H_2)} = \frac{T_1 + \frac{P(D_{4,3}^*|E_{3,3}^*D)}{P(D_{4,3}^*|E_{3,3}^*D^c)} - 1}{T_2 + \frac{P(D_{4,3}^*|E_{3,3}^*D)}{P(D_{4,3}^*|E_{3,3}^*D^c)} - 1} \]

Figure 7
Figure 7 (continued)

Where: \( T_k = P(D|E^*H_k) \)

\[
P(D|H_k) \left[ P(E|DH_k) + \frac{P(E^*|E)}{P(E^*|E^c)} - 1 \right]^{-1} \\
= \frac{P(D|H_k)[P(E|DH_k)-P(E|D^cH_k)]+P(E|D^cH_k)}{P(D|H_k)[P(E|DH_k)-P(E|D^cH_k)]+P(E|D^cH_k)} + \left[ \frac{P(E^*|E)}{P(E^*|E^c)} \right]^{-1}
\]

for \( k = 1,2 \).
Figure 8
Figure 9
<table>
<thead>
<tr>
<th></th>
<th>$H_1$</th>
<th>$A$</th>
<th>$A_1^*$</th>
<th>$B$</th>
<th>$B_2^*$</th>
<th>$D$</th>
<th>$E$</th>
<th>$E_3^*$</th>
<th>$C$</th>
<th>$D_4^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$A_1^*$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$B$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$B_2^*$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$D$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$E$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$E_3^*$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$D_4^*$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 11
<table>
<thead>
<tr>
<th></th>
<th>$H_2$</th>
<th>$A$</th>
<th>$A_1^*$</th>
<th>$B$</th>
<th>$B_2^*$</th>
<th>$D$</th>
<th>$E$</th>
<th>$E_3^*$</th>
<th>$D_4^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$A$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$A_1^*$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$B$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$B_2^*$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$D$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$E$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$E_3^*$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$D_4^*$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 12
\[
\begin{bmatrix}
P(A|H_1) & P(A|H_2) \\
P(A^*_1|A) & P(A^*_1|A^c) & P(A^*_1|A) & P(A^*_1|A^c) \\
P(B|H_1A) & P(B|H_1A^c) & P(B|H_1A) & P(B|H_1A^c) \\
P(B^*_2|B) & P(B^*_2|B^c) & P(B^*_2|B) & P(B^*_2|B^c) & P(B^*_2|B) & P(B^*_2|B^c)
\end{bmatrix}
\]

Figure 13
APPENDIX I. CASPRO OUTPUT EXAMPLE

IN THIS PHASE, THE INFERENCE STRUCTURE IS INPUT. NAMES OF EVENTS, INTERMEDIATE EVENTS, OR HYPOTHESES SHOULD BE ENTERED IN SINGLE QUOTES.

WHAT ARE THE TWO HYPOTHESES?

'h1' 'h2'

HOW MANY RELEVANT EVENTS ARE KNOWN TO HAVE OCCURRED?

4

WHAT ARE THEY?
(ENTER FROM LEFT TO RIGHT IN STRUCTURE.

'a1#' 'b2#' 'c3#' 'd4#'

NOW ENTER, STREAM BY STREAM, FROM LEFT TO RIGHT, AND EVENT BY EVENT, FROM TOP TO BOTTOM, THE INTERMEDIATE AND KNOWN EVENTS IN THE STRUCTURE. ALSO ENTER INFORMATION ABOUT CONDITIONING PATTERNS. ENTER EVENT.

3

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.

1 'h1'

ENTER EVENT.

'a1#'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.

1 'a'
ENTER EVENT.
:
'b'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.
:
2 'h1' 'a'

ENTER EVENT.
:
'b2*'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.
:
2 'a1*' 'b'

ENTER EVENT.
:
'c'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.
:
1 'h1'

ENTER EVENT.
:
'c'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.
:
2 'h1' 'd'

ENTER EVENT.
:
'e3*'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.
:
1 'e'

ENTER EVENT.
:
'dc4*'

ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES.
:
2 'e3*' 'd'
IN THIS PHASE, LIKELIHOODS ARE ELICITED.

ENTER : \( P(a_1|h_1) \)

\( .5 \)

ENTER : \( P(a_1|h_2) \)

\( .05 \)

ENTER : \( P(a_1^*|l_a) \)

\( .9 \)

ENTER : \( P(a_1^*|l_a^*) \)

\( .01 \)

ENTER : \( P(b_1|h_1^*) \)

\( .7 \)

ENTER : \( P(b_1|h_1^*|a) \)

\( .4 \)

ENTER : \( P(b_1|h_2^*) \)

\( .6 \)

ENTER : \( P(b_1|h_2^*|a) \)

\( .3 \)

ENTER : \( P(b_2^*|l_a^*|b) \)

\( .9 \)
ENTER : P(b2*la1**b) 85
.01
ENTER : P(d1h1)
1.0
ENTER : P(d1h2)
.01
ENTER : P(elh1d)
.8
ENTER : P(elh1'd)
0.0
ENTER : P(elh2d)
.75
ENTER : P(elh2'd)
.4
ENTER : P(e3*1e)
.5
ENTER : P(e3*1'e)
.4
ENTER : .6

ENTER :

.7

\text{P}(d\text{c}^*1\text{e}3* d) \quad \text{P}(d\text{c}4*1\text{e}3* d)

\text{LAMBDA } a1* = 8.34862451
\text{LAMBDA } b2* = 1.26656935
\text{LAMBDA } e3* = 1.09004203
\text{LAMBDA } d\text{c}4* = 0.85846570

\text{OVERALL LAMBDA=}
\text{READY}

9.8948700428832852
APPENDIX II. CASPRO PROGRAM LISTING

```
CASP:  PROCEDURE OPTIONS(MAIN);  
     DECLARE (E12:0.024), KNOWN(20), CES(20) VARYING,  
         CBUFF(15) CHARACTER(34) VARYING,  
         (PROD, LUFF(24, 128), LAMBDA(20));  
     FLOAT(13), STRUCT(0.12, 0.024) FIXED(1) INIT  
         (0.25); DCL LIKE(24, 120) FLOAT,  
         KNOWN(24) FIXED(1) INIT (25); DCL  
         (CINDEX(1), CINDEX(24, 0), NKNOWN(20), NEST(24);  
         I) FIXED, (X, Y, X, Y, X, Y, X, Y) FIXED,  
         (1, 1, 1, 1, 1, 1, 1) FIXED,  
         (1, 1, 1, 1, 1, 1, 1, 1) FIXED,  
         (1, 1, 1, 1, 1, 1, 1, 1) FIXED,  
         (1, 1, 1, 1, 1, 1, 1, 1) FIXED;  
     END;  
     GENER: PROCEDURE(E, EV, CINDEX, NKNOWN, X, CBUFF);  
     DCL E12(24), CHARACTER(5) VARYING, CBUFF(15) CHARACTER(24)  
     VARYING, PIX BIT(20), SPIX BIT(20) VARYING, CAR  
     CHARACTER(34) VARYING, I, NOX, NKNOWN, A, J;  
     DCL CINDEX(1), CINDEX(24, 0) FIXED, NEST FIXED;  
     NKNOWN=KNOWN;  
     NEST=2**NE;  
     CAR=CAR+CAR;  
     IF X=KNOWN THEN DO;  
         DO I=1 TO NKNOWN;  
             CAR=CAR+CAR1;  
         END;  
     END;  
     END;  
     DO =1 TO NEST;  
     CBUFF(I) = (X);  
     END;  
     DO J=1 TO NEST;  
         PI = UNSPEC(J);  
         SPIX = SUBSTR(PIX, 1, 1);  
         DO I=1 TO LENGTH(SPIX);  
             IF SUBSTR(SPIX, 1, 1) = CBUFF(J) =  
             CBUFF(J) = CBUFF(J + 1);  
             ELSE CBUFF(J) = CBUFF(J + 1);  
             END;  
         END;  
     END;  
     END;  
     DO =1 TO NEST;  
     CBUFF(I) = CBUFF(I + 1);  
     END;  
     RETURN;  
     END GENER;  
     UPDATE: PROCEDURE(MATRIX, NO, INDEX2);  
     DECLARE (MATRIX(0, 24), 24, NO, J, K, INDEX2(24, 0)) FIXED;  
     K=1;  
     DO =1 TO NO;  
         IF MATRIX(NO, J) = 1 THEN DO;  
             INDEX2(NO, J) = K;  
             K = K+1;  
             END;  
         END;  
     END;  
     RETURN;  
     END UPDATE;  
     SIGVEC: PROCEDURE(MATRIX, NO, NANT);  
     DCL MATRIX(24, 128) FLOAT,  
     I FIXED,  
     (NO, J, NPART, LPART, NPART(10), A2, A1) FIXED,  
     CBUFF(128) FLOAT;  
     DO =1 TO L;  
     CBUFF(I) = MATRIX(NO, J);  
     END;  
```

87
PUT SKIP (2);
PUT SKIP LIST (* WHAT ARE THE TWO HYPOTHESES?*);
GET LIST (E(1,K)=E(2,K));
PUT SKIP LIST (* HOW MANY RELEVANT EVENTS ARE KNOWN TO HAVE OCCURRED?*);
GET LIST (NSTREAM);
PUT SKIP LIST (* WHAT ARE THEY?*);
GET LIST ([KNOWN][I] DO [I] TO NSTREAM);
PUT SKIP LIST (* ENTER NUMBER OF CONDITIONING EVENTS AND THEIR NAMES*);
GET LIST (N(J),CES(J)) DO J=1 TO N);
DO J=1 TO N;
DO K=0 BY 1 WHILE (S>W);
IF E(i.K)=CES(J) THEN DO;
GET LIST (STRUCT(I,K)=1;
S=S+1:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END:
END1
PROD=11
KNOBUFF[0]=0;
DO K=1 TO NSTREAM;
  KNI=11;
  DO I=1 TO NCOND(KNOWIND(K));
    DO J=1 TO 2*(1=SUM(KNOBUFF[I]=1));
      LBUFF[I,J]=LINDEX2(KNOWIND(K),I,J);
    END;
  END;
  IF KMCINDEX2(KNOWIND(K),I,J)=1 THEN DO;
    KNOBUFF[I]=11;
    SW=0;
    DO INDI=1 BY 1 WHILE (SW=0);
      IF CINDEX2(KNOWIND(K),I,J)=NOWIND(INDI) THEN DO;
        KB(KNI)=INDI;
        KNI=KNI+11;
        SW=1;
      END;
    END;
  END;
END;
ELSE KNOBUFF[I]=0;
END;
KNI=KNI+2;
LAMBOA(K)=$LAMBOA(K)$;
DO IND=1 TO KNI;
  LAMBOA(K)=LAMBOA(K)/LAMBOA(KB(IND));
END;
END;
PUT SKIP EDIT(* LAMBOA **.LAMBOA(K)** = *LAMBOA(K)**
(A(B)+A(B).F(18.8)));
PROD*PRODLAMBOA(K);
END;
PUT SKIP;
PUT SKIP EDIT(* OVERALL LAMBOA**.PROD(A(10)+F(32.16)));
END CASPRO: