UNIVERSITY OF PATRAS

INTERDEPARTMENTAL POSTGRADUATE COURSE IN
MEDICAL PHYSICS

THESIS

“CODIFICATION OF CLINICAL JUDGEMENT WITH THE USE OF
BELIEF NETWORKS”

by

GEORGE C. SAKELLAROPOULOS

PATRAS, 1995
SUPERVISOR:
G. NIKIFORIDIS, Assoc. Professor

EXAMINATION COMMITTEE:
G. NIKIFORIDIS, Assoc. Professor, Dept. of Medical Physics
G. DASSIOS, Professor, Dept. of Chemical Engineering
Z. PSILLAKIS, Lecturer, Dept. of Physics
To my parents, Μαρία, Χρήστο
It is unwise to prophesy either death or recovery in acute disease.

Hippocrates, Aphorisms, Sect II, No. 19

The sciences do not try to explain; they hardly even try to interpret; they mainly make models.

John von Neumann

Read not to contradict nor to believe and take for granted ... but to weigh and consider.

Francis Bacon
1. INTRODUCTION .............................................................................................................. 6
   1.1 Physicians’ Judgement .......................................................................................... 6
   1.2 Analytical Approaches ....................................................................................... 8
   1.3 Conditional Probability ..................................................................................... 10
   1.4 Method’s Overview ............................................................................................ 11
   1.5 Network Topology ............................................................................................. 12
   1.6 Cyclic Graphs ....................................................................................................... 14
      1.6.1 Chow’s method ............................................................................................ 15
      1.6.2 Star decomposition ....................................................................................... 16

2. METHOD ............................................................................................................................ 18
   2.1 Method Chosen .................................................................................................... 18
   2.2 Link Matrices ....................................................................................................... 19
   2.3 Propagation and Fusion ...................................................................................... 20
   2.4 Flow of Belief ..................................................................................................... 23
   2.5 Multiple Parents .................................................................................................. 24

3. SPECIFIC CASE ................................................................................................................ 27
   3.1 Variables ............................................................................................................. 27
   3.2 Files .................................................................................................................... 28
   3.3 Architecture Determination ................................................................................ 30
   3.4 Link Matrices ...................................................................................................... 32
   3.5 Belief Network Realization Software .................................................................. 33
   3.6 Results ................................................................................................................ 34
      3.6.1 Comparison with other architectures ......................................................... 39
      3.6.2 Implementation to Outpatient’s Department .............................................. 42

4. DISCUSSION ...................................................................................................................... 45
   4.1 Types of Data and Their Impact on Prognostic Outcome .................................... 47
   4.2 Medical Database Files ....................................................................................... 50
   4.3 Limitations .......................................................................................................... 51
   4.4 Evaluation Scales ............................................................................................... 52

5. REFERENCES .................................................................................................................... 56

6. APPENDIX ........................................................................................................................ 60
   6.1 Software Flow Diagrams ..................................................................................... 60
   6.2 Software Code Excerpts ...................................................................................... 62
1. INTRODUCTION

Judgement in medicine is undoubtedly of great importance. Among the various aspects of physician performance, the ability to reach good clinical judgement is ranked first in the priority list. It is therefore evident that any research aiming at the improvement of the physicians’ ability to make better decisions in his clinical practice should be encouraged. Moreover, the conclusions drawn from such research should be taken into account in everyday practice, thus providing the necessary feedback to more clinically relevant studies.

Clinical prediction is a central procedure in all clinical tasks. Whether this task is the making of a diagnosis, the assessment of prognosis or the determination of the therapy and the monitoring of the outcomes, clinical prediction is an indispensable activity.

Three are the main characteristics of the clinical prediction: the data available, the method used and its accuracy.

Data become available to the physician through the patient’s history, physical examination and the various laboratory tests. Many of them – symptoms, signs – are of probabilistic, uncertain character.

Human judgement and various analytical approaches comprise the methods used by physicians in evaluating this prediction. The method by which the vast majority of predictions are made, whether in clinical practice or in everyday life, is by human judgement.

Under the term human judgement we include any informal, qualitative, or not explicitly quantitative methods. One can also call these methods artistic since they much rely on the practitioner’s personal skills.

With the term actuarial, statistical, or analytical approach we refer to methods that use formal quantitative techniques to reach diagnostic decisions.

Accuracy of the prediction, on the other hand, is influenced by a number of factors: the prior probability of the disease or the outcome, the sensitivity and specificity of the test, the ability to present realistic probability estimates and the factors contributing to the deviations of the prediction from the actual outcome or disease [1].

1.1 Physicians’ Judgement

Clinical practitioners consider clinical judgement to be more an artistic and qualitative process than a formal one [2]. There are certain reasons that justify their position:

1. Data are mainly of probabilistic, uncertain character
2. There are only a few explicit rules available, hence experience alone helps to acquire the necessary ability to analyze the data
3. The established belief that while clinical practice deals with a specific case, scientific methods refer to general principles and therefore formal approaches should be ruled out.

4. The complexity of the patterns of data involved require qualitative techniques, more sensitive than formal approaches.

5. The relative lack of training in formal decision theory that often leads to wondering whether population probabilities can be applied to individual decision making.

6. The difficulty they have in assigning precise probabilities to the various diagnostic possibilities, combined with their capacity to easily scale the latter in order of subjective probability.

But do physicians have good judgement? Having “good” judgement means that their predictions regarding diagnosis, prognosis, or management are usually accurate. Being accurate could mean that they can correctly predict a continuously measured outcome, such as the length of time until a patient will begin to respond to therapy, or that they can correctly predict a discrete event, such as whether or not a patient actually has a given disease. At the extreme, having “good” judgement would suggest that no more than some minimum number of errors would be made.

Many measures have been developed and used for gauging the accuracy of human judgements. These measures tend to reflect two fundamental aspects of accuracy: the ability to discriminate patients with a given outcome from those without that outcome, and the agreement between predicted values and actual values, i.e., the degree of calibration [3, 4, 5, 6, 7].

Why might physicians’ predictions be inaccurate? One possibility is that the events we are trying to predict are in fact unpredictable. It could be that the available clinical data simply do not correlate well with the outcome we are trying to predict. Alternatively, it could be that the event is predictable, but we are using the wrong cues from the clinical data, or that we are using the right cues but giving them the wrong weights.

At this point we should distinguish between two different classes of accuracy: the accuracy of the prediction as an overall procedure and the accuracy of specific variables used in the prediction process. The latter contributes to the accuracy of the prediction and reflects the clinician’s influence on the assessment of clinical variables.

There is a number of studies showing that human information processing is limited [8] and as a consequence undertaking complex decisions using only human judgement leads to errors. When a decision maker tries to process the available data simultaneously, he usually fails to appreciate the relative importance of the data on revising his prior opinion. He updates his opinion less than he should and thus his judgement is conservative [9]. This conservatism forces the decision maker to order more tests than needed to reach a certain level of diagnostic belief. The existing positive correlations between the tests cause this redundancy.

A sequential data processing also leads the decision maker to predictions that deviate from the correct ones. Any uncertainty in the data is ignored. Every evidence is treated as perfect information [10, 11]. The result is an
overprediction: the decision maker’s belief in a certain diagnosis is larger than it would be were the data evaluated properly.

The source of this kind of error is the fact that physicians treat the several pieces of evidence at their disposal as being independent of each other while they actually are not. Taking into account the objective computational difficulty of interdependent data processing we can easily come to the conclusion that human judgement alone cannot avoid such errors. On the other hand, humans have the ability to integrate the various clues and acquire an holistic view of the situation in an intrinsic and not fully codable manner.

It will be shown that analytical methods, and especially those that embody the concept of conditional dependence among the evidence, can treat the available data in the proper manner and therefore lead to more accurate predictions.

1.2 Analytical approaches

The philosophy of the statistical approach lies in the insistence that decision rules can be made explicit and that we should make them so because, no matter whether clinical data are of probabilistic nature, it is better to combine them in a formal, explicit manner to achieve more efficient and consistent information processing, than by intuition.

The way analytical methods approach the problem of clinical prediction avoids the methodological limitations mentioned in the previous paragraphs: the process is apt to our detailed control since the steps in assessing the prediction are posed by us. This explicit process of the analytical systems leads to a high reliability, a quality not often met in intuitive approaches.

Such methods, however, when used to model broad decision tasks become very complicated, involving the use of large number of variables and the consequent calculations grow both in number and in complexity. If we are to use the analytical approach for such tasks we need to make approximations, thus reducing the overall accuracy.

Therefore, for broad decision tasks, intuitive approach leads by far to better results, while formal techniques prevail over human judgement in narrow decision tasks.

Many different methods to construct a model that approaches analytically a given medical domain have been proposed [12-17]. All of them use a limited number of clinical or laboratory variables that have been chosen by experts in the corresponding medical field or posed by the everyday practice in the hospitals. A number of outcomes is assigned to each variable and the system is fed with the information acquired when the various tests are performed.

The way this information is evaluated by the system differs from system to system. In expert systems IF-THEN-ELSE rules are implemented and thus each piece of information acquired is driving the system to a different direction according to the value or the combination of values of the variable under measurement.
In *neural networks* the situation is the opposite as no such rules are applied; the system's behaviour is determined by the combination of the pieces of information acquired and by the previous system's responses (history). Neural networks are systems that become more experienced as new cases are fed into them.

A third class of analytic systems is those called **Belief Networks**. To our opinion belief networks best simulate the process a physician follows as they use Bayesian methods in assessing beliefs in specific outcomes.

The necessity for probabilistic reasoning in medicine arises from the fact that a medical diagnosis or prognosis is rarely made with absolute certainty. The end result of the diagnostic process usually gives a “most likely” diagnosis.

During this process, clinicians take into account several types of evidence, i.e. the values of clinical and laboratory variables. They start with a knowledge of the probability of an event (e.g. specific disease) before acquiring new information – like a diagnostic test result. After new information has come, they evaluate it and alter their belief in the disease incrementally or decrementally, according to the results of laboratory tests. The interpretation of the new piece of information is actually an estimation of the probability of an event conditioned on the test result, or the calculation of a posterior probability.

The overall procedure clinicians follow should not be viewed as strictly sequential; if the laboratory results deviate from the expected ones, the doctor retrieves once again the information from the clinical examination and seeks hints that might lead him to a different diagnosis. Alternatively, the repeated retrieval of data could lead to a further support of the initial belief.

Through this process, undertaken implicitly, physicians attain the integration of all the available data and reach their decision task taking into account all the pieces of evidence. What they lack is the codification of the procedure they follow: the relative weights they assign to the various types of data at hand and the way they evaluate them remains hidden. Their assessments are always qualitative; never can a numerical value be assigned to the degree of confidence they have in their estimations after each phase of information evaluation.

Bayes' theorem explicitly simulates the above presented way of thinking. It makes use of the concept of conditional probability, which is of essential importance to the physicians. The thorough understanding of this concept is necessary to distinguish the probability of symptoms given a specific disease, \( P(S+ \mid D+) \), which medical students are generally taught, from the probability the clinician must predict, that of a specific disease given certain symptoms, \( P(D+ \mid S+) \).

Belief networks, based on Bayes’ theorem and using conditional probability files, offer the necessary codification of the clinicians’ procedures whilst preserving the integrative character of human judgement.
1.3 Conditional Probability

The use of Bayesian approach in apparently simple and well known cases shows off certain misunderstandings of several concepts. As an example we mention the PAP-test used to identify uterus’ cervix cancer. This test has both sensitivity and specificity equal to 95%. Assume we produce to a gynaecologist the following case:

“A female of unknown age undertook a PAP-test and it was found positive. No other evidence is available”

The question the doctor is asked to answer is “What is the probability for the female under examination to actually have cervix cancer?”

It might sound astonishing but most of physicians are likely to respond with a value within the interval 50% - 95% while the correct answer is 3.9%! The reason for this misjudgement is the fact that physicians ignore that no other evidence is supplied (age, possible complications) and therefore the a priori probability of this disease, which influences strongly the result, refers to the whole female population and is only 0.2 %.

Trying to represent the procedure a doctor follows to come to a decision about either a diagnostic or prognostic task using mathematical formalism, we could state that the doctor’s knowledge on the specific field is represented by a joint distribution $P(x_1, ..., x_n)$ on a set of propositional variables $x_1, ..., x_n$, while the task of drawing inferences from observations is the computation of probabilities of a small subset $H_1, ..., H_k$ of variables (hypotheses) conditioned upon a group of instantiated variables $e_1, ..., e_m$, called evidence.

Schematically, it’s the computation of $P(H_1, ..., H_k / e_1, ..., e_m)$.

If one is given the joint probability distribution $P(x_1, ..., x_n)$, the aforementioned computation is trivial, yet requiring a number of calculations that increases exponentially with the number of propositions.

It is clear that doctors do not undertake this vast number of calculations on evaluating the patient’s clinical and laboratory data before coming to a diagnosis. Rather, human mind is capable of assessing the qualitative aspect of conditional probability $P(x_i/x_j)$ [whether knowing $x_j$ alters our belief in $x_i$] and of conditional dependency [$x_j$ alters our belief in $x_i$, given $x_k$] clearly and consistently.

In order to avoid problems of underspecification of the graph dependencies or of usage of excessive number of parameters, we must find a representation satisfying the demands of completeness and consistency.

The joint-distribution representation is preferable for many reasons:

- it gives a solution to the consistency-completeness issue, coming directly from its chain-rule representation
- it is transparent to synthesis of consistent probability models and to detection of inconsistencies
- it very closely emulates the way physicians evaluate the information given to them as it uses the same mathematical elements with those used
in calculating the predictive value of a test given its sensitivity and specificity.

Because of the use of Bayes’ rules in this representation, networks of this kind are called either belief networks or Bayes networks or influence networks.

1.4 Method’s overview

Belief networks is a way to represent our perception of a specific problem field by creating a graph in which the nodes represent propositions and the links show the connections between conceptually directly related propositions. If their connection is via another variable, their relation is indirect.

Regarding the graph structure chosen to represent the medical domain of head injuries, the nodes represent clinical or laboratory variables or elements of the patient’s history. The decision upon which nodes to include was based upon generally accepted medical knowledge [18-26] and opinions of experts on the field. Every variable is assigned a finite number of outcomes. In the case of variables whose values are not discrete but continuous, their range is divided into intervals.

Our aim is to quantify these links with appropriate weighting factors so that they will correctly reflect the type and strength of the dependencies between the connected nodes. The evaluation of the weights assigned to each link connecting two neighbouring nodes passes through the calculation of conditional probabilities drawn from the patient’s file. These conditional probabilities refer to sub-groups of the patient’s population since only a cluster of the whole number of patients satisfies the necessary conditions each time.

The data file the network is based upon, therefore, is of different quality compared to the existing patients’ file. One can view this file as a collection of libraries of conditional probabilities, appropriate ones for each neighbourhood of variables.

Networks constructed in such a manner carry the knowledge of a specific field and are able to become a tool at the physician’s disposal for the interpretation of certain input data. This interpretation starts with the instantiation of a group of variables corresponding to the input data and ends with the selection of the most likely hypotheses. The network’s role is to calculate the impact of the input data on the probabilities of the posed hypotheses.

We think of a belief network as a computational structure with the links viewed as the centres of direction and flow of data in the process of updating beliefs and the nodes as separate processors having the duty to both maintain the belief parameters for the variables they represent and supervise the communication between neighbouring, hence directly related, variables.

The processors communicate with each other always but a multidirectional propagation process is triggered only when some local conditions are satisfied. Since at each computational phase inputs are coming from
neighbouring nodes only and the triggering of the nodes’ activations follow conceptually familiar courses, physicians can give meaningful interpretation to the individual phases and finally be confident in the final result.

One could argue that this calculation could be carried out by the human interpreter, in our field the physician. There are however some human characteristics that are incompatible with the task we aforementioned. Humans have difficulties in shifting quickly between alternative ways of reasoning and our focus of attention is narrow thus showing that our reasoning process progresses incrementally along pre-established pathways. Apart from that, the speed and clarity with which the interactions between the nodes are evaluated by the network by far exceed those of the physician.

1.5 Network Topology

Once a graph is created we can identify the dependencies among the variables in a relatively easy manner. The correspondence between the topology of a Bayes network and the various types of independence is presented in the following paragraphs. What is worth mentioning of is that the ordering of the variables is of our choice and we can always guarantee that the need for the graph to be complete and consistent will be fulfilled since we use a joint-distribution representation.

Let’s consider a triplet of variables $x_1$, $x_2$, $x_3$ where $x_1$ is connected to $x_3$ via $x_2$. The three variables can be connected in any of the 3 following ways :

![Diagram: Network Topology]

[1] tail-to-tail

[2] head-to-tail

[3] head-to-head

In cases 1 and 2, variables $x_1$ and $x_3$ are conditionally independent given $x_2$, $P(x_1/x_3, x_2)=P(x_1/x_2)$. In case 3, $x_1$ and $x_3$ are marginally independent, $P(x_3/x_1)=P(x_3)$, but they might become dependent given $x_2$. If we assumed that the node representing variable $x_2$ had descendants $x_4$, $x_5$ etc, then the instantiation of any of them would also make $x_1$, $x_3$ dependent.

In general, we can apply a graphical criterion for testing conditional independence between two variables $x_i$, $x_j$: if a subset of variables $S_e$ separates $x_i$ from $x_j$, then $x_i$ is conditionally independent of $x_j$, given $S_e$. Or, $P(x_i/x_j, S_e)=P(x_i/S_e)$

The subset $S_e$ separates the two variables if at least one pair of successive links along every path between these variables is blocked by $S_e$. And depending on the configuration of the two successive links we have the following definition of blockage:
• two links meeting tail-to-tail or head-to-tail at node X are blocked by $S_e$ if X is in $S_e$, and
• two links meeting head-to-head at node X are blocked by $S_e$ if neither X or any of its descendants is in $S_e$.

Any two variables in the network can therefore be viewed as conditionally independent of each other, given the appropriate set of variables $S_e$ that separates them.

Our task now is to find the necessary and sufficient set of variables $S$ that, once known, render a specific variable independent of every variable not in $S$.

Variables in $S$ can be called a variable’s screening neighbourhood and in Belief Networks it is comprised of the variable’s direct parents, direct successors and all direct parents of the latter. In tree-structured belief networks, the screening neighbourhood consists of the unique father and the immediate successors.

In figures 1.5-A and 1.5-B we can see the screening neighbourhoods of the bold-line nodes for a general belief network and a tree-structured one.

In other words, if for every variable with k direct fathers, we know the probability $P(\text{variable}/\text{father}_1, \text{father}_2, \ldots, \text{father}_k)$ we need no other information regarding the influence of other, more remote nodes on our variable. This influence is indirect and is taken into account when we assess the corresponding probability of the variable’s neighbours.

One can see now the advantages of such a way of representation:

• The graph can be constructed with ease, since we apply our perception of the problem directly, addressing a small number of variables as direct causes or consequences of another variable.
• The number of the necessary calculations is greatly reduced, since only the neighbourhood of a node takes part in the computational procedure.
• The model and its induced causality fits very well with the doctor’s own way of thinking and is therefore appropriate for a true translation of the medical knowledge on a specific field into a codified and objective analytical approach.
• The integration of the available data, characteristic of the clinician, is now attained.
1.6 Cyclic graphs

When doctors evaluate the pieces of information coming from several laboratory and clinical variables, they have the tendency to consider every variable to be dependent on all the others. To justify their position they refer either to the corresponding causal relations between the variables, based on the theoretical substrate of the specific medical domain, or to the fact that according to their experience a variation in the value of one variable is escorted by changes to the values of the rest of the variables.

In order to represent this view of the dependencies among the variables with the aid of belief networks, we would have to construct multiply connected networks, where multiple parents of common children may possess common ancestors. We shall call networks of such type cyclic graphs.

Examples of networks meeting these characteristics are shown below.

Cyclic graphs lack the simplicity of calculations and the clarity of operations existing in trees. The loops formed lead to indefinite circulation of the messages between the nodes and the process does not converge to the correct state of equilibrium.

Apart from the technical problems arising in cyclic graphs, it is useful to transform cyclic belief networks into tree-structured graphs in order to reveal the conditional dependence relations among the variables. The physicians’ belief that every variable is related—directly or not—to all the others would then be explained through the topology presented earlier: given the values of a variable’s screening neighbourhood, it becomes independent of all other variables.

This transformation can be accomplished using two different approaches:

- The first one uses only the variables that are available and recorded in our file. It is an approximation seeking method of an n-th order joint distribution by a second order one, proposed by Chow [27].
- If, apart from the usual nodes representing clinical and laboratory variables, we introduce dummy ones, we can use a procedure called “star-decomposition” that turns cyclic graphs into trees [28, 29].
1.6.1 Chow’s method

According to this method [27], any n-dimensional discrete probability distribution can be approximated by a product of n-1 second order component distributions:

\[ P(x) = P(x_{m_1} / x_{m_0}) \cdot \ldots \cdot P(x_{m_{n-1}} / x_{m_{n-2}}) \]

0 ≤ j(i) < i

Each variable may be conditioned upon at most one of the variables. \( m_1, m_2, \ldots, m_n \) is an unknown permutation of integers 1, 2, ..., n.

This kind of approximation is called a probability distribution of first order tree dependence.

As an example we present the following figure, corresponding to the expansion \( P(x) = P(x_1).P(x_2/x_1).P(x_3/x_2).P(x_4/x_2).P(x_5/x_2).P(x_6/x_5). \)

![Dependence tree diagram]

In order to decide upon the best dependence tree we need to define some parameters:

As a measure of closeness in approximating \( P \) by \( P_a \) we adopt the quantity

\[ I(P, P_a) = P(x).\ln \frac{P(x)}{P_a(x)} \]

that can be interpreted as the difference of the information contained in \( P(x) \) and that contained in \( P_a(x) \) about \( P(x) \).

\[ I(P, P_a) = 0 \quad \text{if} \quad P(x) \equiv P_a(x) \quad \text{for all} \quad x \quad \text{and} \quad I(P, P_a) > 0 \quad \text{otherwise.} \]

We seek therefore a distribution \( P_t(x) \) that minimizes the above quantity.

We define as mutual information between two variables the expression

\[ I(x_i, x_j) = \sum_{x_i, x_j} P(x_i, x_j) \cdot \log \{ \frac{P(x_i, x_j)}{P(x_i)P(x_j)} \} \]

In the graphical representation of dependence relations, to every branch of the dependence tree (i.e. to every link between two variables) we assign a branch weight \( I(x_i, x_{j(i)}) \).

It can be proved that a probability distribution of tree dependence \( P_t(x) \) is an optimum approximation to \( P(x) \) if and only if its dependence tree \( t \) has maximum weight, because minimizing the closeness measure \( I(P, P_t) \) is equivalent to maximizing the total branch weight \( \sum_{i=1}^t I(x_i, x_{j(i)}) \). Since the
branch weights are additive, the maximum weight dependence tree can thus be constructed branch by branch.

We first index the \( \frac{n(n-1)}{2} \) branches \( b_i \) emanating from \( n \) nodes according to decreasing weights. We select \( b_1 \) and \( b_2 \) and add \( b_3 \) only if the latter does not form a cycle with \( b_1 \) and \( b_2 \). In general, we continue to consider branches of successively higher indices, selecting a branch whenever it does not form a cycle with the set previously selected, and rejecting it otherwise. This procedure produces a unique solution if the branch weights are all different. If several weights are equal, multiple solutions are possible, all having the same maximum weight.

1.6.2 Star decomposition

Following this [29] procedure, the introduction of dummy nodes in the set of our variables is necessary. Dummy nodes might not just be abstract qualities invented to facilitate this transformation; an analysis of the tree obtained might enable us to decipher their true meaning, as causes of the observed values of variables. This induced causality also serves the need that humans have to construct structured models governed by intrinsic rules, open to our inspection or not.

The task of star-decomposition is feasible if we know precisely each interleaf correlation, naming as leaves the variables that are directly accessible to our judgement (clinical or laboratory variables).

The central problem of the procedure is the following:

“If we are given a tree-decomposable distribution (this is checked using certain criteria) \( P(x_1, \ldots, x_n) \), can we uncover its underlying topology and the underlying tree-distribution \( P_T(x_1, \ldots, x_n, w_1, \ldots, w_m) ? \)"

With \( w \) we denote “hidden causes”, not directly observable but facilitating the acquisition of effective causal models from empirical data. Hidden causes are viewed as dummy variables which, if held constant, induce probabilistic independence among sets of visible variables.

The tree-reconstruction is based on the observation that any 3 leaves in a tree have only one internal node that can be viewed as their centre since it lies on all the paths connecting the leaves to each other. If the centre is removed, the leaves become disconnected from each other.
The possible topologies that a quadruple of leaves in the tree may obtain are the above 4. Their difference is that in each of them different triplets share common centres. In the third topology, for example, the sharing pairs are \([(1,4,2), (1,4,3)]\) and \([(1,2,3), (4,2,3)]\) while in the fourth topology all pairs share their common centre.

Once we check whether 3 leaves form a triplet (i.e. that a common centre exists), we choose a fourth variable and seek to find to which leg of the star should this variable be joined. By testing which pairs of triplets share centres we decide on the appropriate topology and we connect the fourth variable accordingly.

Working in the same manner we attach all of our leaves and the tree distribution is uncovered.

The mathematical background of the above procedures assumes that our variables are bivalued. An extension for multivalued variables is necessary if we are to apply this method in the medical domain.
2. METHOD

2.1 Method Chosen

In order to accomplish the tasks we mentioned in the introduction (1.2, 1.4), we used a method proposed by Pearl [28]:

We created a tree-structured belief network composed of several nodes in a predefined structure. The structure was chosen on medical knowledge grounds, regarding its content (which nodes were included), and based on a certain method—presented earlier—of tree-architecture extraction regarding the specific configuration. It resembles very much the way medical doctors act on evaluating the data of the patient given to them. Two nodes that are directly connected do not necessarily imply a causality between them. Rather, their link shows an existing correlation between the two, which sometimes can be viewed as causality. The decision upon the architecture of the specific network can also be made through iterating, i.e. evaluating the results of many possible configurations and choosing the one that resembles reality best.

Each node represents a multivalued variable representing a collection of mutually exclusive hypotheses (e.g. CT findings, Glasgow Coma Scale score) or a collection of possible observations (e.g. eye response).

The Bayesian relations between the nodes are established through matrices which are appointed to each link connecting two adjacent nodes. Matrices’ elements are probabilities of the i-th outcome of one node conditioned on the j-th outcome of the node connected to the former through the specific link. The dynamic values of the updated node probabilities are denoted by \( \text{BEL}(A_i) \) which reflects the overall belief accorded to the proposition \( A=A_i \), by all data so far received (\( A \) is the variable and \( A_i \) its possible values).

We appoint to each node two kinds of vectors: \( \lambda \)-vectors are used to transmit information to the node’s father and \( \pi \)-vectors transmit information to the node’s children.

Let \( D_{\lambda} \) stand for the data contained in the tree rooted at \( A \) and \( D_{\lambda}^+ \) for the data contained in the rest of the network. \( \lambda(A_i) = P(D_{\lambda}^+ / A_i) \) represents the diagnostic or retrospective support \( A_i \) receives from \( A \)’s descendants.

\( \pi(A_i) = P(A_i / D_{\lambda}^+) \) represents the anticipatory support attributed to \( A_i \) by the ancestors of \( A \). The total strength of belief in \( A_i \) is obtained fusing these two supports via the product \( \text{BEL}(A_i) = \alpha \cdot \lambda(A_i) \cdot \pi(A_i) \), where \( \alpha \) is a normalizing factor. Pearl’s algorithm can therefore be viewed as an extension of the Bayes’ formula to a network of variables, where \( \pi \)-messages are analogous to priors and \( \lambda \)-messages are analogous to likelihoods.

The \( \lambda \)-vector of each node can be computed from the \( \lambda \)'s of its children by multiplying the latter by their respective link matrices and then multiplying the resultant vectors together, term by term. Similarly, the \( \pi \)-vector of any node
can be computed from the $\pi$ of its father and the $\lambda$’s of its siblings, after multiplication by the corresponding link matrices. A direct communication with the siblings is however unnecessary since the information required already resides at the father’s site (for the purpose of calculating its $\lambda$) and can be sent down to the requesting son.

### 2.2 Link matrices

The aforementioned link matrices can be explained now:

Let’s assume that node A represents a variable with three possible values and node B represents a variable with two possible values. Obviously $\lambda$ and $\pi$ vectors of node A will have a dimension of 3 and those of node B a dimension of 2.

The link matrix between A and B, denoted $M(B/A)$, acts as an interpreter between the two nodes, in two ways:

- **a.** reformats B’s message-vector $\lambda$ to the father (A) so that the receiver obtains it in his own configuration (from 2 to 3 dimensions).
- **b.** takes into account the collected up to now medical knowledge using conditional probabilities in order to translate the effect of the transmitted message into the language of the receiver.

In our example this matrix would be:

$$M(B/A) = \begin{pmatrix}
P(B_1 / A_1) & P(B_2 / A_1) \\
P(B_1 / A_2) & P(B_2 / A_2) \\
P(B_1 / A_3) & P(B_2 / A_3)
\end{pmatrix}$$

which is a 3x2 matrix.

Matrix $M(B/A)$ multiplied by B’s 2x1 $\lambda$-vector (column element) gives a 3x1 $\lambda$-vector of the father A.

The matrices’ elements are drawn from our overall patients’ database, which is updated after a significant number of cases, and if the population is sufficiently big, represent the probabilities of appearance of the combination of the appropriate values of variables.

Several classes of nodes can be defined:

- **Root node:** the boundary condition for the root node is established by setting $\pi(\text{root}) = $ prior probability of the root variable
- **Data node:** a variable with instantiated value. If the $j$-th state of node were observed to be true, we set $\lambda = \pi = \text{BEL} = (0, ..., 0, 1, 0, ..., 0)$ with 1 at the $j$-th position
- **Dummy node:** a node E representing virtual or judgemental evidence bearing on node F: We do not specify $\lambda(E)$ or $\pi(E)$ but instead we post a $\lambda_E(F_i) = K \cdot P(\text{observation}/F_i)$, and $K$ is any convenient constant
- **Anticipatory node:** A leaf node that has not been instantiated yet. For such variables, BEL should be equal to $\pi$ and therefore we should set $\lambda = (1, 1, ..., 1)$
It is useful to distinguish 2 different phases of the system:

- in phase 1 the system (i.e. the network of the nodes) reaches an equilibrium state, dependent upon the initial conditions posed by our patients' data file on the root node, and
- in phase 2, when new data enter the system and the procedure until a new equilibrium state is reached takes place.

New information may enter the system in two ways:

- if a variable was observed to have a certain value then the corresponding node is a data node and the procedure stated above is followed.
- if we have judgemental or subjective evidence about a variable's value then we attach a dummy node which transmits the information to the variable's node in the way we described above.

Once a node is fed with certainty on one of its outcomes through a $\lambda$-message, its belief shows certainty on that outcome and therefore it is not updated when information from other nodes is available. As a consequence, this node's children do no longer participate in the acquisition and propagation of belief, since their pathway to the rest of the network is blocked by their common father.

The natural way of designing a tree-structured belief network is therefore one that places observables at terminal positions while variables whose values are not easily and surely determined at positions in the middle of the network. This rule of thumb obviously does not exclude terminal nodes from receiving evidential or judgemental evidence.

Regarding some of the network's variables as non-observable leads to a new type of problem: how can one derive the elements of the matrix connecting such a variable with another, whether the latter is an observable or not? It is clear that our patients' file is no longer adequate but medical knowledge is also needed to fill the gap created.

### 2.3 Propagation and fusion

The assignment of the initial values for $\lambda$ and $\pi$ is followed by the propagation throughout the network of the information represented by these vectors. This is accomplished through the transmission of $\lambda$ and $\pi$ signals by a node, as soon as it becomes activated. The destination of these signals is the neighbourhood of the activated node, with the exception of the node or nodes that provoked this activation.

The information initially flows in a bottom-up fashion, meaning that the leaf nodes once activated cause the update of beliefs of their parent nodes and gradually the root node collects all the transmitted vectors. After that happens, the root node sends $\pi$ messages to its children, starting the top-
down propagation, i.e. the process that leads to the update of beliefs of one branch of the tree due to the information acquired by the other branch.

Each step of this propagation includes the fusion at each node of the information coming from its immediate descendants through a term-by-term multiplication of the corresponding \( \lambda \) messages.

The exact procedures that take place in each node are described below. Figure 2.3-A shows a hypothetical tree-structured belief network. The shaded node D is a middle node with two children.

Viewing a node as a small processor, node D looks like the one depicted in figure 2.3-B. The arrows on the top show the interactions with node’s D father (node A) while the arrows at the bottom show the corresponding interactions with its children (nodes E, F). In order for this node to communicate with its neighbours, it uses the 2 kinds of vectors we mentioned earlier: \( \lambda \) and \( \pi \) vectors. The notational conventions we follow are the following:

1. \( \pi_E(D) \) is a message from father D to son E
2. \( \pi_F(D) \) is a message from father D to son F
3. \( \pi_D(A) \) is a message from father A to son D

![Figure 2-A](image)

![Figure 2-B](image)
We can see therefore on figure 2.3-B that a node like D (in a middle position) both transmits and receives $\pi$-messages. Node D also receives and transmits $\lambda$-messages.

1. $\lambda_E(D)$ message to father D from son E
2. $\lambda_F(D)$ message to father D from son F
3. $\lambda_D(A)$ message to father A from son D

Let’s now see how this processor works:

Assume that node E is instantiated, i.e. we know the value of the variable represented by E and we feed the network with this value. As soon as node E is activated, it sends a $\lambda$-message to its father : $\lambda_E(D)$. Since node F is not activated, the $\lambda$ message it sends is of the form $\lambda_F(D)=(1,1,...,1)$.

In node D the fusion of these incoming pieces of information takes place: vectors $\lambda_E(D)$ and $\lambda_F(D)$ are multiplied term by term and their product is $\lambda(D)$, or the $\lambda$-vector of node D.

$\lambda(D)=\lambda_E(D).\lambda_F(D)$

The belief of node D will be the product of $\lambda(D)$ that was just updated and of $\pi(D)$ that had acquired its value in the previous step, appropriately normalized to unity:

$BEL(D)=a.\lambda(D).\pi(D)$

The information from E obviously has to reach both the root node of the network and the node F. Therefore, node D sends a message $\lambda_D(A)$ to his father A, and a message $\pi_F(D)$ to his other son F, containing this information.

Vector $\lambda_D(A)$ is not the same with $\lambda(D)$, neither in form or in content. Node D alters the vector $\lambda(D)$ before transmitting it for two reasons:

- Nodes D and A are of different dimension, i.e. they have different number of outcomes, and therefore the message must pass through a medium that changes its dimension
- The effect of the transmitted message needs to be translated into the language of the receiver A, to reveal its impact on A’s outcomes.

To accomplish the above tasks, $\lambda_D(A)$ is obtained by multiplying $\lambda(D)$ with the matrix of the link D–A

$\lambda_D(A) = M(D/A).\lambda(D)$

For example, if node D has 6 outcomes and node A has 5, $M(D/A)$ will be a 5x6 matrix which, when multiplied by the 6x1 $\lambda(D)$ will give the 5x1 $\lambda_D(A)$.

Vector $\pi_F(D)$ on the other hand contains the information that came from E and helps F update its belief due to changes in its father’s belief. The translation we mentioned before will take place this time inside node F : its $\pi$-message will be the product of the transpose of the link F–D matrix by $\pi_F(D)$ :

$\pi(F) = M^T(F/D).\pi_F(D)$,

where $\pi_F(D) = \lambda_E(D).\pi(D)$ or $\pi_F(D)=BEL(D) / \lambda_F(D)$
The network’s function leads to an ongoing update of nodal beliefs as time evolves, or new data enter one or more of its nodes. The belief of the root node, initially drawn from the overall patients’ database, is changing for each patient according to his personal clinical and laboratory data and leads finally to a percentage-based recommended diagnosis.

2.4 Flow of belief

The figure below shows six successive stages of the belief propagation through a simple tree.

- Stage 1: the tree is in equilibrium, ready to respond to a nodes’ activation.
- Stage 2: two terminal nodes become activated and immediately send messages to their fathers.
- Stage 3: the fathers update their beliefs and create messages to their neighbours: both to children and fathers. It’s worth noticing that the links through which messages were absorbed do not receive new messages.
- Stage 4: the root node updates its belief and posts messages to its children.
- Stage 5: the messages travel to the leaf nodes.
- Stage 6: after the leaf nodes receive the messages, the system reaches equilibrium.
2.5 Multiple parents

In this section we shall discuss the propagation scheme to graph structures permitting a node to have multiple parents, capturing “sideways” interactions via common successors. This constitutes an extension to the case of tree-structured belief networks; however, the graphs are required to be singly connected, meaning that one path at most exists between any two nodes.

The picture above shows a fragment of a singly connected network with multiple parents. Arrows show the messages sent to and from the node A. The probability distribution of each variable A in the network can be computed if three types of parameters are made available:

1. the current strength of the causal support, \( \pi_A(B) \), \( \pi_A(C) \), contributed by each incoming link to A,

2. the current strength of the diagnostic support, \( \lambda_X(A) \), \( \lambda_Y(A) \), contributed by each outgoing link from A and

3. the fixed conditional probability matrix \( P(A \mid B, C) \) which relates the variable A to its immediate ancestors.

According to the principles we followed in the case of tree-structured networks, we let each link carry two dynamic parameters, \( \lambda \) and \( \pi \), and let each node store an encoding of \( P(A \mid B, C) \).

Using these parameters we can write the fusion equation:

\[
BEL(A_i) = a \lambda_X(A_i) \cdot \lambda_Y(A_i) \cdot \sum_{jk} P(A_i \mid B_j, C_k) \cdot \pi_A(B_j) \cdot \pi_A(C_k)
\]

Our task now is to describe how the influence of new information should spread through the network. This is made through \( \lambda \) and \( \pi \) messages that node A sends to its neighbours:

- Updating \( \lambda \)

\[
\lambda_A(B_i) = \alpha \sum_j \left[ \pi_A(C_j) \sum_{k} \lambda_X(A_k) \lambda_Y(A_k) P(A_k \mid B_i, C_j) \right]
\]

We can see that only three parameters, in addition to the conditional probabilities, are needed for updating the diagnostic parameter vector \( \lambda_A(B) \): \( \pi_A(C) \), \( \lambda_X(A) \) and \( \lambda_Y(A) \).
• Updating $\pi$

$\pi_X(A_i) = \alpha \cdot \lambda_Y(A_i) \left[ \sum_{jk} P(A_i / B_j, C_k) \cdot \pi_A(B_j) \cdot \pi_A(C_k) \right]$

Again three neighbouring parameters are needed: $\lambda_Y(A)$, $\pi_A(B)$ and $\pi_A(C)$.

The above 2 equations demonstrate that a perturbation of the causal parameter $\pi$ will not affect the diagnostic parameter $\lambda$ on the same link, and vice versa. Therefore any perturbation of beliefs due to new evidence propagates through the network and is absorbed at the boundary without reflection. A new state of equilibrium will be reached after a finite number of updates.

The $\lambda$-update equation also reveals that if no data are observed below A (i.e. all $\lambda$’s pointing to A are unit vectors), then all $\lambda$’s emanating from A are unit vectors. This means that evidence gathered at a particular node does not influence its spouses (nodes of the same level, sharing common successors) until their common son gathers diagnostic support (either direct by the user or its successors). This reflects the connectivity conditions established in section 1.5.

It is interesting to approach the message-vectors $\lambda$ and $\pi$ through a configuration more general than the above that uses the analytical, index based representation. More specifically we shall try to apply the technique of the link matrices, used in tree structures, in this generalized case.

Assume that for the preceding graphic example the range of indices is the following:

• for node B: $j=1-4$ or variable B has 4 outcomes
• for node C: $k=1-3$ or variable C has 3 outcomes
• for node A: $i=1-6$ or variable A has 6 outcomes

The j-th component of the vector $\lambda_Y(B)$ can be written as:

$$\lambda_Y(B_j) = \left[ \begin{array}{c} \pi_A(C_1) \\ \pi_A(C_2) \\ \pi_A(C_3) \end{array} \right] \cdot \left[ \begin{array}{cccc} P(A_1 / B_1, C_1) & P(A_2 / B_1, C_1) & \ldots & P(A_6 / B_1, C_1) \\ P(A_1 / B_2, C_2) & P(A_2 / B_2, C_2) & \ldots & P(A_6 / B_2, C_2) \\ P(A_1 / B_3, C_3) & P(A_2 / B_3, C_3) & \ldots & P(A_6 / B_3, C_3) \end{array} \right] \cdot \left[ \begin{array}{c} \lambda_Y(A_1) \cdot \lambda_Y(A_1) \\ \lambda_Y(A_2) \cdot \lambda_Y(A_2) \\ \lambda_Y(A_3) \cdot \lambda_Y(A_3) \end{array} \right]$$

Accordingly, the i-th component of the vector $\pi_X(A)$:

$$\pi_X(A_i) = \left[ \begin{array}{c} \pi_A(B_1) \\ \ldots \\ \pi_A(B_4) \end{array} \right] \cdot \left[ \begin{array}{cccc} P(A_1 / B_1, C_1) & P(A_1 / B_2, C_1) & \ldots & P(A_1 / B_4, C_1) \\ \ldots & \ldots & \ldots & \ldots \\ P(A_1 / B_1, C_3) & P(A_1 / B_2, C_3) & \ldots & P(A_1 / B_4, C_3) \end{array} \right] \cdot \left[ \begin{array}{c} \lambda_Y(A_1) \\ \ldots \\ \lambda_Y(A_3) \end{array} \right]$$

We can see that using this representation, the conditional probability matrix is decoupled by the vectors.

In the case of tree-structured networks, the corresponding vectors $\lambda$ and $\pi$ were given by the product between a matrix and a vector. In this general case, things are more complicated: the product between matrix and vectors gives the component of the vector of interest. The latter will therefore be the
product of vectors and a third-order tensor. In the general case of \( n \)-fathers, the “matrix” stored in their common successor will be a \( n \)-th order tensor.

Belief networks with multiple parents can be used to facilitate the codification of more complex clinical tasks, like the differential diagnosis or the treatment determination, due to the sideways interaction of variables incorporated in such procedures.
3. SPECIFIC CASE

A number of methods of predicting outcome in critically ill patients has been developed [24,30,31] and more specifically about head-injury cases. The Leeds score [21] aims to identify patients whose death can be predicted with certainty, thus enabling the physician to withdraw treatment. However, a study by Feldman et al [25] has showed that the Leeds model is not infallible and cannot be relied on to "clarify the complicated emotional, moral, legal and financial issues that surround the early termination of care in seriously head-injured patients".

The use of the Glasgow Outcome Scale (GOS) [23], consisting of five exclusive outcome categories is widely accepted. Most studies, however, deal with patient outcome classification by only two broad outcome groups: good outcome, combining good recovery and moderate disability (cat. 4, 5 in GOS) and poor outcome, combining the other three categories in GOS (death, vegetative, severe disability).

Choi, Narayan et al. [20] attempted to increase the specificity of the outcome prediction by categorizing patients into one of four GOS categories, combining vegetative and death into one.

In order to develop an analytical method in assessing the outcome of patients coming to the Intensive Care Unit of a general hospital after a head injury, we formed a belief network using nine variables extracted from the records of 375 patients for the years 1992-1994.

3.1 Variables

The choice of the predicting factors – clinical variables or diagnostic tests used as indicators of outcome – is a key issue in the prognostic procedure. Titterington et al. [32] note that the choice of variables used in the prediction is more important than the method of prediction.

Currently, the vast majority of institutions worldwide rely on the Glasgow Coma Scale (GCS) score at admission [22], although it is known that certain combinations of indicators yield more accurate predictions than when factors are used singly [33, 34]. Choi et al. [20] found that accuracy rates obtained with age and the motor response alone were equally good or better than using the entire GCS score and age.

Quite often, the composite GCS score (sum of the individual scores for eye, verbal and motor response) is used to describe a patient's condition. Jagger et al [35] report that all the predictive value of the GCS lies in the motor response score, while Teasdale et al [36] stress that patients should be described by the three separate responses; the total score is merely a convenient method for summarizing data, especially for a series of patients.

At this point we should mention that the validity of a certain variable as a predicting factor is assessed up to now only in a phenomenological way. The physiological demand on the selection of the optimal indicator or combination
of indicators of patients’ outcome is yet to be exposed to us. The use of evoked potentials and of other diagnostic modalities might give a better insight to this aspect [37, 38].

The final choice of the variables was based on the previously presented literature [18-26, 30-38] and on opinions of experts, colleagues of the Department of Neurosurgery. A number of outcomes was assigned to every variable. In the case of variables that take continuous rather than discrete values, their range was divided into appropriate intervals. The variables and the respective outcomes are shown in table 1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Outcome</th>
<th>Value</th>
<th>Variable</th>
<th>Outcome</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glasgow Outcome</td>
<td>1</td>
<td>Death</td>
<td>Glasgow Outcome</td>
<td>1</td>
<td>Score 3 - 4</td>
</tr>
<tr>
<td>Scale</td>
<td>2</td>
<td>Vegetative</td>
<td>Coma</td>
<td>2</td>
<td>Score 5 - 7</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Severe Disability</td>
<td>Score 8 - 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Moder. Disability</td>
<td>Score 11-13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Good Condition</td>
<td>Score 14-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td>1</td>
<td>Midline Displac.</td>
<td>Age</td>
<td>1</td>
<td>0-10 years</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Oper.Hematoma</td>
<td></td>
<td>2</td>
<td>11-20</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Non Oper.Hemat.</td>
<td></td>
<td>3</td>
<td>21-40</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Edema</td>
<td></td>
<td>4</td>
<td>41-60</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>No findings</td>
<td></td>
<td>5</td>
<td>61-</td>
</tr>
<tr>
<td>Eye Resp.</td>
<td>1</td>
<td>Both Mydriatic</td>
<td>Hb</td>
<td>1</td>
<td>&lt;8 gr%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Different size</td>
<td></td>
<td>2</td>
<td>8-10 gr%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Normal</td>
<td></td>
<td>3</td>
<td>&gt;10 gr%</td>
</tr>
<tr>
<td>Mean Arterial Pressure</td>
<td>1</td>
<td>&lt;60 mmHg</td>
<td>Delay</td>
<td>1</td>
<td>&lt;2 hours</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>&gt;120</td>
<td>Time until hospitalization</td>
<td>2</td>
<td>2-6 hours</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>60-120</td>
<td></td>
<td>3</td>
<td>&gt;6 hours</td>
</tr>
</tbody>
</table>

| Urination           | 1       | <0.5 ml/kg/h  |                     |         |                |
|                     | 2       | 0.5-3.5       |                     |         |                |
|                     | 3       | >3.5          |                     |         |                |

### 3.2 Files

Initially the patients’ files of the Intensive Care Unit, regarding the variables of interest, were transferred manually to a personal computer. A computer file was thus created, containing the values of the nine variables, described previously, of 375 patients.

The next step involves the procedure of obtaining the conditional probability libraries, i.e., all the elements of the second order distributions $P(A_i/B_j)$ where $A_i, B_j$ are any two variables and $A_i, B_i$ the respective variable’s outcome. The data from the patients’ records were processed by special software in order to extract all the conditional probability values, necessary for the formation of the link matrices.

In the picture below we can see the screen output of the software used to hold the patients’ data and calculate the conditional probabilities:
3.3 Architecture determination

In order to reveal the underlying tree structure of a set of variables A, B, C, ... we used Chow’s method that approximates the joint distribution \( P(A, B, C, ...) \) by a product of several of its lower order component distributions. This method provides a criterion for the possible connection between two variables in the network and leads to the optimal architecture for this set of variables.

It incorporates the calculation of all the second order distributions and the assignment of branch weights to every possible link between any two variables. If we sort these weights in a descending order and keep only those links that render the set tree-like, we come to the desired optimal tree architecture.

Applying this technique in our case, we calculated the weights of 36 branches formed by 9 variables \([36=9(9-1)/2]\) and sorted them in a decreasing-value order starting from the top. We kept only those branches that did not form a cyclic graph. These 8 branches can be seen greyed in the following table. Quantities \( i \) and \( j \) show the combinations of nodes. For example the first pair of nodes, \( i=3 \) and \( j=9 \), are nodes Age and Urination. The correspondence between the nodes and the indexes is the following:

<table>
<thead>
<tr>
<th>Index</th>
<th>Node</th>
<th>Index</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GOS</td>
<td>6</td>
<td>CT</td>
</tr>
<tr>
<td>2</td>
<td>Delay</td>
<td>7</td>
<td>MAP</td>
</tr>
<tr>
<td>3</td>
<td>Age</td>
<td>8</td>
<td>Hb</td>
</tr>
<tr>
<td>4</td>
<td>GCS</td>
<td>9</td>
<td>Urination</td>
</tr>
<tr>
<td>5</td>
<td>Eye response</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>i</th>
<th>j</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>9</td>
<td>0.328653</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
<td>0.179263</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>0.155645</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0.120344</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>8</td>
<td>0.108248</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>4</td>
<td>0.081245</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>5</td>
<td>0.074232</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>6</td>
<td>0.053319</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>8</td>
<td>0.052742</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>6</td>
<td>0.036065</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>8</td>
<td>0.032294</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>3</td>
<td>0.031036</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>4</td>
<td>0.029643</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>7</td>
<td>0.026976</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>8</td>
<td>0.025660</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>5</td>
<td>0.021780</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>2</td>
<td>0.018485</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>7</td>
<td>0.017648</td>
</tr>
<tr>
<td>19</td>
<td>6</td>
<td>9</td>
<td>0.016713</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>8</td>
<td>0.016456</td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>7</td>
<td>0.014943</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>6</td>
<td>0.014197</td>
</tr>
<tr>
<td>23</td>
<td>5</td>
<td>8</td>
<td>0.014016</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>7</td>
<td>0.013510</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>9</td>
<td>0.012816</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>5</td>
<td>0.012512</td>
</tr>
<tr>
<td>27</td>
<td>3</td>
<td>7</td>
<td>0.012014</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>9</td>
<td>0.011598</td>
</tr>
<tr>
<td>29</td>
<td>8</td>
<td>9</td>
<td>0.009141</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>3</td>
<td>0.007361</td>
</tr>
<tr>
<td>31</td>
<td>2</td>
<td>4</td>
<td>0.005838</td>
</tr>
<tr>
<td>32</td>
<td>2</td>
<td>9</td>
<td>0.004451</td>
</tr>
<tr>
<td>33</td>
<td>2</td>
<td>8</td>
<td>0.002958</td>
</tr>
<tr>
<td>34</td>
<td>7</td>
<td>9</td>
<td>0.002445</td>
</tr>
<tr>
<td>35</td>
<td>2</td>
<td>7</td>
<td>0.001566</td>
</tr>
<tr>
<td>36</td>
<td>5</td>
<td>9</td>
<td>0.000887</td>
</tr>
</tbody>
</table>
Applying the optimal tree extraction algorithm, we derived the architecture depicted in figure 3.3-A. The overall procedure, from the hospital patients’ records to the network realization is shown in figure 3.3-B.
3.4 Link matrices

Having determined the tree architecture, we now calculate the appropriate link matrices:

\[
\begin{array}{ccc}
0.52 & 0.30 & 0.18 \\
0.80 & 0 & 0.2
\end{array}
\]
\[M(Delay/GOS)\]

\[
\begin{array}{ccc}
0.66 & 0.25 & 0.09 \\
0.49 & 0.42 & 0.09 \\
0.53 & 0.29 & 0.18 \\
0.49 & 0.24 & 0.27 \\
0.20 & 0.60 & 0.20
\end{array}
\]
\[M(Eye/GOS)\]

\[
\begin{array}{cccc}
0.76 & 0.18 & 0.05 & 0.01 & 0 \\
0.40 & 0.40 & 0.20 & 0 & 0
\end{array}
\]
\[M(GCS/GOS)\]

\[
\begin{array}{cccc}
0.57 & 0.36 & 0.07 & 0 & 0 \\
0.42 & 0.41 & 0.14 & 0.01 & 0.02 \\
0.43 & 0.29 & 0.13 & 0.09 & 0.06
\end{array}
\]
\[M(CT/GOS)\]

\[
\begin{array}{ccc}
0.08 & 0.18 & 0.39 \\
0.10 & 0.25 & 0.65 \\
0.09 & 0.20 & 0.71 \\
0.17 & 0.25 & 0.58 \\
0.14 & 0.58 & 0.28
\end{array}
\]
\[M(Hb/GCS)\]

\[
\begin{array}{ccc}
0.08 & 0.18 & 0.39 \\
0.06 & 0.14 & 0.54 \\
0.17 & 0.19 & 0.46 \\
0.12 & 0.26 & 0.52 \\
0.15 & 0.21 & 0.52
\end{array}
\]
\[M(Age/CT)\]

\[
\begin{array}{ccc}
0.02 & 0.40 & 0.58 \\
0.02 & 0.40 & 0.58 \\
0.06 & 0.14 & 0.54 \\
0.17 & 0.19 & 0.46 \\
0.12 & 0.26 & 0.52 \\
0.15 & 0.21 & 0.52
\end{array}
\]
\[M(Urin./Age)\]

\[
\begin{array}{ccc}
0.02 & 0.40 & 0.58 \\
0.02 & 0.40 & 0.58 \\
0.06 & 0.14 & 0.54 \\
0.17 & 0.19 & 0.46 \\
0.12 & 0.26 & 0.52 \\
0.15 & 0.21 & 0.52
\end{array}
\]
\[M(MAP/Hb)\]

\[
\begin{array}{ccc}
0.35 & 0 & 0.65 \\
0.28 & 0.02 & 0.70 \\
0.07 & 0.04 & 0.89
\end{array}
\]
\[M(Hb/GCS)\]
We remind the reader that the elements of the link matrices are conditional probabilities between the outcomes of the connected variables. If for example variable \( \text{var1} \) has outcomes \( \text{var1a}, \text{var1b}, \ldots, \text{var1n} \) and its daughter variable \( \text{var2} \) has outcomes \( \text{var2a}, \text{var2b}, \ldots, \text{var2n} \), the matrix connecting the two variables will be of the following form:

\[
M(\text{var1}/\text{var2}) =
\begin{pmatrix}
P(\text{var2a} / \text{var1a}) & P(\text{var2b} / \text{var1a}) & \cdots & P(\text{var2n} / \text{var1a}) \\
P(\text{var2a} / \text{var1b}) & P(\text{var2b} / \text{var1b}) & \cdots & P(\text{var2n} / \text{var1b}) \\
\vdots & \vdots & \ddots & \vdots \\
P(\text{var2a} / \text{var1n}) & P(\text{var2b} / \text{var1n}) & \cdots & P(\text{var2n} / \text{var1n})
\end{pmatrix}
\]

We can see that certain elements of the above matrices are equal to zero. This is due to the lack of the corresponding combinations of variables among our records of patients. None of the patients, for example, that eventually exited the Intensive Care Unit in a vegetative state had been transferred to the hospital within 2 to 6 hours after their injury occurred (line 2, column 2 of Delay/GOS matrix).

The fact that such limitations exist even in a file consisting of 375 records underlines the need for large and rich—in the sense of the variety of combinations—files to properly construct a belief network.

### 3.5 Belief Network realization software

Given the network’s architecture and the corresponding link matrices, we are now ready to actually create the Belief Network which makes use of Pearl’s algorithm. For this purpose, special software written in C++ for the Windows 3.1™ environment was developed. The software is divided in two parts: the first deals with the building of the network and the second with the network’s activation.

Both programs are easy to operate and the end user is able to enter the patient’s data with a click of a button.

In the next pages we can see screen captures from these programs. Picture 3.5-A shows the Belief Network as it is presented in the program after we construct it. In picture 3.5-B we can see the elements of the link AGE–CT matrix. Picture 3.5-C shows the Beliefs acquired by the different outcomes of the Glasgow Outcome Scale node, after the activation of the CT node with 100% confidence in its first (worst) outcome (Picture 3.5-D).

At the final stage of this research work, the different pieces of software were integrated into one compact computer program which is used to both receive the data of the hospital archives, decide on the best architecture, calculate the link matrices through the conditional probabilities and go through the propagation and fusion of the data entered by the end user.
3.6 Results

In the following paragraphs we shall present the results that the belief network of the selected architecture gave as output to the various pieces of information we fed it with. Specifically, we shall see the beliefs acquired to the different outcomes of prognosis (death, vegetative, severe disability, moderate disability, good condition) for many hypothetical cases of patients for which we know the values of either all or a part of the variables.

Let's first study the cases for which the value of only one variable is known. This will show the relative impact that each variable alone has on the final prognosis.

Table 2 summarises this study. The first column shows the variable that is used in the activation of the network. The numbers in the second column show which of the outcomes we have absolute (100%) certainty in, each time. The rest of the columns show the belief (in percentage) in the various prognostic outcomes after the network reaches equilibrium.

The table's first line shows the belief in each of GOS outcomes before any activation of the system. These values show the probability that a new patient, coming to the Intensive Care Unit, has, for each of the prognostic outcomes if no other information is available. In other words, from the fact alone that this patient has come head injured to the ICU, we can assign to him a probability of nearly 40% to die and around 48% to end up with moderate disability or in good condition.

We can make the following comments on the results presented in table 2.

- The apparently small values of belief acquired by the second outcome of GOS (vegetative state) initially, are due to the presence of a small number of such patients in the original patients' file.
- No matter which outcome of Hb or MAP or Urination we are certain in, the impact to the prognosis is negligible when any one of these is, alone, the system's input.
- Patients with age above 40 have a tendency towards worse prognosis compared to younger people, especially regarding the first outcome (death)
- Regarding the impact of the delay until hospitalisation of the patient, we can see that only prolonged delays (greater than 6 hours –outcome 3) drag the patient's prognosis to worse outcomes.
- Significant deviations from the initial values occur when variables GCS, CT and Eye are instantiated. Specifically, there is an obvious trend for the decrease of death’s probability when the system is fed with values closer to normal. The zero values should be treated as a limitation induced by our file and not as real. The respective effect is an increase in the probability for good condition.
Table 2

<table>
<thead>
<tr>
<th>Activations</th>
<th>GOS Prognostic Outcomes (Belief in %)</th>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
<th>[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>52.3</td>
<td>11.6</td>
<td>0.9</td>
<td>15.8</td>
<td>19.4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>25.3</td>
<td>14.9</td>
<td>1.8</td>
<td>31.4</td>
<td>26.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20.9</td>
<td>8.7</td>
<td>2.7</td>
<td>32.0</td>
<td>35.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13.4</td>
<td>0.0</td>
<td>0.0</td>
<td>7.4</td>
<td>79.2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>21.8</td>
<td>78.2</td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>71.4</td>
<td>7.3</td>
<td>0.9</td>
<td>10.5</td>
<td>9.9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50.4</td>
<td>13.2</td>
<td>1.3</td>
<td>21.2</td>
<td>13.9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>23.6</td>
<td>13.9</td>
<td>2.7</td>
<td>33.4</td>
<td>26.4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>14.5</td>
<td>7.8</td>
<td>0.0</td>
<td>22.3</td>
<td>55.4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>21.3</td>
<td>0.0</td>
<td>19.7</td>
<td>59.0</td>
<td></td>
</tr>
<tr>
<td>Eye</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>76.8</td>
<td>8.3</td>
<td>1.0</td>
<td>7.7</td>
<td>6.2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>47.6</td>
<td>17.6</td>
<td>4.0</td>
<td>15.2</td>
<td>15.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>19.4</td>
<td>11.1</td>
<td>0.5</td>
<td>30.3</td>
<td>38.7</td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>38.4</td>
<td>14.4</td>
<td>1.9</td>
<td>19.7</td>
<td>25.6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>37.8</td>
<td>9.3</td>
<td>0.0</td>
<td>28.9</td>
<td>24.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>47.3</td>
<td>7.0</td>
<td>1.8</td>
<td>12.9</td>
<td>31.0</td>
<td></td>
</tr>
<tr>
<td>Hb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>42.2</td>
<td>11.4</td>
<td>1.2</td>
<td>19.9</td>
<td>25.3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>39.4</td>
<td>11.5</td>
<td>1.2</td>
<td>21.2</td>
<td>26.7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>38.9</td>
<td>11.9</td>
<td>1.4</td>
<td>22.1</td>
<td>25.7</td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>32.2</td>
<td>12.5</td>
<td>1.5</td>
<td>24.2</td>
<td>29.6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>36.8</td>
<td>11.6</td>
<td>1.2</td>
<td>21.8</td>
<td>28.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>37.6</td>
<td>12.0</td>
<td>1.3</td>
<td>22.1</td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>50.0</td>
<td>10.7</td>
<td>1.2</td>
<td>18.5</td>
<td>19.6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>48.0</td>
<td>11.0</td>
<td>1.4</td>
<td>20.3</td>
<td>19.3</td>
<td></td>
</tr>
<tr>
<td>MAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>40.1</td>
<td>11.6</td>
<td>1.2</td>
<td>21.0</td>
<td>26.1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>39.2</td>
<td>11.6</td>
<td>1.3</td>
<td>21.5</td>
<td>26.4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>39.4</td>
<td>11.7</td>
<td>1.3</td>
<td>21.7</td>
<td>25.9</td>
<td></td>
</tr>
<tr>
<td>Urination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>40.6</td>
<td>11.7</td>
<td>1.3</td>
<td>21.5</td>
<td>24.9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>40.3</td>
<td>11.6</td>
<td>1.3</td>
<td>21.4</td>
<td>25.4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>38.6</td>
<td>11.8</td>
<td>1.3</td>
<td>21.8</td>
<td>26.5</td>
<td></td>
</tr>
</tbody>
</table>
The next step involves the activation of two variables for each case of patient. It is logical to expect that for combinations of an effective and a not effective variable, the prognostic beliefs will be affected mainly— if not solely— by the former.

### Table 3

<table>
<thead>
<tr>
<th>Activations</th>
<th>GOS Prognostic Outcomes (Belief in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial values</strong></td>
<td>39.5 1.3 11.7 21.6 25.9</td>
</tr>
<tr>
<td>[1] CT-1, Hb-1</td>
<td>73.7 0.8 6.9 9.3 9.3</td>
</tr>
<tr>
<td>[2] CT-1, Hb-2</td>
<td>71.4 0.9 7.2 10.3 10.2</td>
</tr>
<tr>
<td>[3] CT-1, Hb-3</td>
<td>70.8 0.9 7.5 10.9 9.9</td>
</tr>
<tr>
<td>[4] Delay-3, Age-1</td>
<td>39.3 2.1 7.6 14.8 36.2</td>
</tr>
<tr>
<td>Delay-3, Age-2</td>
<td>44.1 1.6 6.9 13.1 34.3</td>
</tr>
<tr>
<td>Delay-3, Age-3</td>
<td>45.3 1.7 7.2 13.3 32.5</td>
</tr>
<tr>
<td>Delay-3, Age-4</td>
<td>58.5 1.6 6.2 10.8 22.9</td>
</tr>
<tr>
<td>Delay-3, Age-5</td>
<td>56.8 1.9 6.5 12.0 22.8</td>
</tr>
<tr>
<td>Delay-3, GCS-1</td>
<td>60.5 1.2 6.7 9.1 22.5</td>
</tr>
<tr>
<td>Delay-3, Eye-1</td>
<td>83.3 1.2 4.5 4.3 6.7</td>
</tr>
<tr>
<td>Delay-3, CT-1</td>
<td>78.2 1.1 4.0 5.8 10.9</td>
</tr>
<tr>
<td>Delay-3, Hb-1</td>
<td>50.0 1.5 6.8 11.7 30.0</td>
</tr>
<tr>
<td>Delay-3, MAP-1</td>
<td>47.8 1.7 6.9 12.5 31.1</td>
</tr>
<tr>
<td>Delay-3, Urin.-1</td>
<td>48.5 1.8 7.0 12.9 29.8</td>
</tr>
<tr>
<td>Delay-1, Eye-1</td>
<td>74.9 1.6 10.3 7.1 6.1</td>
</tr>
<tr>
<td>Delay-1, Eye-3</td>
<td>19.0 0.7 13.8 27.9 38.6</td>
</tr>
<tr>
<td>MAP-3, Eye-3</td>
<td>19.4 0.5 11.1 30.4 38.6</td>
</tr>
<tr>
<td>GCS-5, Eye-3</td>
<td>0 0 11.1 20.7 79.3</td>
</tr>
<tr>
<td>GCS-5, Age-3</td>
<td>0 0 31.4 78.6</td>
</tr>
<tr>
<td>GCS-4, Age-3</td>
<td>12.4 0 7.3 80.3</td>
</tr>
</tbody>
</table>

In Table 3 we see examples of various combinations of variables. It is evident that in the case of CT & Hb, the results slightly differ from those by CT alone. Another point is that the variables we combine keep their behaviour regarding the trends they caused to the prognosis when used alone. The combination for example of Delay-1 & Eye-1 gives a 75% probability of death while Eye-1 alone gives 77%. The difference is due to the decreasing influence of Delay-1, which alone reduces the death probability by 1.2%.

The combinations shown greyed show the effect of combining outcomes favourable to good prognosis. A person with GCS score 11 to 15 that either has normal eye response or his age is between 21 and 40 is assigned a probability of nearly 80% for a good condition prognosis.

Let’s now see examples of multi-variable activation in Table 4.
Table 4

<table>
<thead>
<tr>
<th>Activations</th>
<th>GOS Prognostic Outcomes (Belief in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial values</td>
</tr>
<tr>
<td>Delay-1, Eye-3, GCS-4</td>
<td>5.0</td>
</tr>
<tr>
<td>Delay-1, Eye-3, GCS-4, CT-1</td>
<td>19.7</td>
</tr>
<tr>
<td>Delay-3, Eye-1, CT-1</td>
<td>93.9</td>
</tr>
<tr>
<td>Delay-3, Eye-1, Age-5</td>
<td>86.1</td>
</tr>
<tr>
<td>Gcs-1, Eye-1</td>
<td>81.1</td>
</tr>
</tbody>
</table>

The preceding tables have a common characteristic: they show that even with a part of the variables, we can reach high confidence in certain prognostic outcomes. Especially meaningful is that MAP, Urination, Hb do not contribute much to the prognostic process or their contribution is not necessary and can be surrogated by the combination of other variables.

The knowledge of the CT outcome, however, is of special interest since in our file there are no cases with a CT outcome 5 that lead to a prognosis of death or vegetative state. A certainty on the first CT outcome, on the other hand, can increase the probability of death in a case where the rest of the variables point to a good prognosis. In a case that the used variables have already pointed to a bad prognosis, the knowledge of a CT-1 or CT-2 enhances this trend and sometimes the probability of death reaches or even surpasses the 90% level.

In Table 5, another way of using the network is shown through two pairs of activations. Cases 2 and 4 differ with the previously presented cases: we are no longer 100% sure about the variable’s outcome; rather we assign a certain level of confidence in each of the outcomes. In case 2 for example, the system’s user estimates that the patient’s GCS score is 3-4 with a probability of 70%, 5-7 with 10%, etc. and feeds the network with this information. The other variables’ beliefs can be interpreted in a similar way.

The final column shows the belief acquired by each outcome of GOS after the evaluation of all the pieces of information for each case. For case 1, for example, the system gives a 93.9% probability of death, 0.5% probability of vegetative state, etc. The beliefs in outcomes 1 to 5 of GOS before any activation were 39.5%, 1.3%, 11.7%, 21.6%, 25.9% respectively, as before.

This use of the network allows the physician to introduce his subjective probability in the procedure of the belief evaluation. Since most of the variables’ outcomes are assessed in a probabilistic manner, the ability of the network to cope with input that is not 100% sure makes it more realistic and shows its close relation with the way of data evaluation followed by clinicians.

This is also proved by the fact that the prognostic results of the system agree with the prognosis given by experts when they were asked to evaluate qualitatively the same cases.
Table 5

<table>
<thead>
<tr>
<th>Case</th>
<th>Outcome</th>
<th>GCS</th>
<th>Delay</th>
<th>CT</th>
<th>Eye Resp.</th>
<th>GCS</th>
<th>Delay</th>
<th>CT</th>
<th>Eye Resp.</th>
<th>GOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>93,9</td>
</tr>
<tr>
<td>2</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>0,5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3,3</td>
<td></td>
</tr>
<tr>
<td>4</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>0,7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0,7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outcome</td>
<td>GCS</td>
<td>Delay</td>
<td>CT</td>
<td>Eye Resp.</td>
<td>GOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>70</td>
<td>20</td>
<td>65</td>
<td>90</td>
<td>83,0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>0,8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>60</td>
<td>15</td>
<td>5</td>
<td>8,9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td>5</td>
<td></td>
<td>2,8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td>5</td>
<td></td>
<td>4,5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outcome</td>
<td>Mean Art. Pres.</td>
<td>Age</td>
<td>Delay</td>
<td>GOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>57,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>11,8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td>5</td>
<td>6,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td></td>
<td>5</td>
<td>22,8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outcome</td>
<td>Mean Art. Pres.</td>
<td>Age</td>
<td>Delay</td>
<td>GOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>55,6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>1,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>13,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td></td>
<td>50</td>
<td>8,6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td></td>
<td>50</td>
<td>21,6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.6.1 Comparison with other architectures

We shall now present the differences in the results we obtain using other architectures, chosen purely on medical grounds.

In figures A, B, C we can see three such architectures. Figure D shows the architecture that we finally selected.
In the following table 6 we can see the different results we obtain by the same nodal activations for the different architectures. Since the relative position of nodes Age, GCS, Delay and MAP remains the same for architectures A and C, the respective results are the same.

<table>
<thead>
<tr>
<th>Activations</th>
<th>Architecture</th>
<th>GOS Prognostic outcomes (Belief in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Initial values</td>
</tr>
<tr>
<td>Age-3, GCS-4</td>
<td>A, C</td>
<td>36.9 2.3 9.9 24.5 26.5</td>
</tr>
<tr>
<td>-1-</td>
<td>B</td>
<td>12.3 0 0 8.6 79.2</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>12.4 0 0 7.3 80.3</td>
</tr>
<tr>
<td>Age-3, GCS-5</td>
<td>A, C</td>
<td>36.6 2.6 10.6 23.8 26.4</td>
</tr>
<tr>
<td>-2-</td>
<td>B</td>
<td>0 0 0 24.6 75.4</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>0 0 0 31.4 68.6</td>
</tr>
<tr>
<td>Delay-3, Age-1</td>
<td>A, C</td>
<td>38.7 0 8.9 12.9 39.5</td>
</tr>
<tr>
<td>-3-</td>
<td>B</td>
<td>31.7 0 13.9 20.5 33.9</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>39.3 2.1 7.6 14.8 36.2</td>
</tr>
<tr>
<td>Delay-3, Age-2</td>
<td>A, C</td>
<td>42.3 0 7.4 9.5 40.8</td>
</tr>
<tr>
<td>-4-</td>
<td>B</td>
<td>36.0 0 11.9 15.7 36.4</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>44.1 1.6 6.9 13.1 34.3</td>
</tr>
<tr>
<td>Delay-3, Age-3</td>
<td>A, C</td>
<td>43.9 3.0 6.2 15.4 31.5</td>
</tr>
<tr>
<td>-5-</td>
<td>B</td>
<td>36.3 2.2 9.7 24.5 27.3</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>45.3 1.7 7.2 13.3 32.5</td>
</tr>
<tr>
<td>Delay-3, Age-4</td>
<td>A, C</td>
<td>58.0 0 0.1 9.3 24.6</td>
</tr>
<tr>
<td>-6-</td>
<td>B</td>
<td>49.5 0 13.1 15.4 22.0</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>58.5 1.6 6.2 10.8 22.9</td>
</tr>
<tr>
<td>Delay-3, Age-5</td>
<td>A, C</td>
<td>67.4 3.5 6.4 13.2 9.5</td>
</tr>
<tr>
<td>-7-</td>
<td>B</td>
<td>57.1 2.6 10.3 21.6 8.4</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>56.8 1.9 6.5 12.0 22.8</td>
</tr>
<tr>
<td>MAP-3, Eye-3</td>
<td>A, C</td>
<td>39.5 1.3 11.7 21.6 25.9</td>
</tr>
<tr>
<td>-8-</td>
<td>B</td>
<td>38.0 1.3 11.6 22.1 27.0</td>
</tr>
<tr>
<td>Selected</td>
<td></td>
<td>19.4 0.5 11.1 30.4 38.6</td>
</tr>
</tbody>
</table>

The differences between the selected and the rest of the architectures are remarkable when a good prognosis is expected (activations 1, 2, 8). Activations 3 - 7 show slight deviations since the positions of nodes Delay and Age are almost the same in all configurations.
Table 7

<table>
<thead>
<tr>
<th>Activations</th>
<th>Architecture</th>
<th>GOS Prognostic outcomes (Belief in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Initial values</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>39.5</strong></td>
</tr>
<tr>
<td>Eye-1, CT-1</td>
<td>A, C</td>
<td>[1]</td>
</tr>
<tr>
<td>-1-</td>
<td><em>Selected</em></td>
<td>92.0</td>
</tr>
<tr>
<td>Delay-3, CT-1</td>
<td>B</td>
<td>44.0</td>
</tr>
<tr>
<td>-2-</td>
<td><em>Selected</em></td>
<td>78.2</td>
</tr>
<tr>
<td>Hb-1</td>
<td>A, B</td>
<td>39.5</td>
</tr>
<tr>
<td>-3-</td>
<td>C</td>
<td>60.0</td>
</tr>
<tr>
<td></td>
<td><em>Selected</em></td>
<td>42.2</td>
</tr>
<tr>
<td>Hb-1, Age-5</td>
<td>A, B</td>
<td>56.5</td>
</tr>
<tr>
<td>-4-</td>
<td>C</td>
<td>76.6</td>
</tr>
<tr>
<td></td>
<td><em>Selected</em></td>
<td>50.9</td>
</tr>
<tr>
<td>Hb-1, Age-5, Delay-3</td>
<td>C</td>
<td>84.3</td>
</tr>
<tr>
<td>-5-</td>
<td><em>Selected</em></td>
<td>59.4</td>
</tr>
</tbody>
</table>

Activations 1, 2 of Table 7 show the variations of the impact that the worst outcomes of Eye response, CT and Delay have on the prognosis for the different configurations. Activations 3, 4, 5 show the effect of the position of the variable in the network. The nearer to the root node a variable is, the bigger the impact.
3.6.2 Implementation to Outpatient’s Department

As we mentioned earlier, the above results come from a file of patients at the Intensive Care Unit, where only serious cases are present. It is therefore reasonable to expect a tendency of the results towards a bad outcome. What would be very interesting is to apply a similar system to the Outpatients Department and monitor patients with heavy as well as mild head injuries.

Obviously, the files on which the overall Network is based will be different and consequently the conditional probability libraries will point to a probably different architecture. The link matrices will also be different and so will be the results of the prognostic outcomes.

We are currently implementing such a procedure to the University Hospital Outpatient’s Department using the following protocol:
## PERSONAL DATA

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST NAME</td>
<td></td>
</tr>
<tr>
<td>SURNAME</td>
<td></td>
</tr>
<tr>
<td>AGE</td>
<td></td>
</tr>
<tr>
<td>TELEPHONE NUMBER</td>
<td></td>
</tr>
</tbody>
</table>

## ACCIDENT'S DATA

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAUSE OF HEAD INJURY (1: TRAFFIC, 2: FALL, 3: OTHER)</td>
<td></td>
</tr>
<tr>
<td>TIME ELAPSED SINCE ACCIDENT (IN HOURS)</td>
<td></td>
</tr>
</tbody>
</table>

## FINDINGS OF EXAMINATION

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLASGOW COMA SCALE (SUM)</td>
<td></td>
</tr>
<tr>
<td>EYE RESPONSE</td>
<td></td>
</tr>
<tr>
<td>VERBAL RESPONSE</td>
<td></td>
</tr>
<tr>
<td>MOTOR RESPONSE</td>
<td></td>
</tr>
<tr>
<td>ARTERIAL PRESSURE (mmHg)</td>
<td></td>
</tr>
<tr>
<td>COMPUTERIZED TOMOGRAPHY (CT)</td>
<td></td>
</tr>
<tr>
<td>OTHER INJURIES (sign with X)</td>
<td></td>
</tr>
<tr>
<td>THORAX</td>
<td></td>
</tr>
<tr>
<td>ABDOMEN</td>
<td></td>
</tr>
<tr>
<td>ARM-LEG-PELVIS-VERTEBRAL COLUMN</td>
<td></td>
</tr>
<tr>
<td>FACIAL CRANIUM</td>
<td></td>
</tr>
</tbody>
</table>

## COMPUTERIZED TOMOGRAPHY OUTCOMES

1. **Diffuse injury I**: No visible intracranial pathology seen on CT.
2. **Diffuse injury II**: Cisterns are present with midline shift of 0-5 mm; lesion densities present; no high- or mixed-density lesion >25 ml; may include bone fragments and foreign bodies.
3. **Diffuse injury III**: Cisterns compressed or absent with midline shift of 0-5 mm; no high- or mixed-density lesion >25 ml.
4. **Diffuse injury IV**: Midline shift >5 mm; no high- or mixed-density lesion >25 ml.
5. **Evacuated mass lesion**: Any lesion surgically evacuated.
6. **Nonevacuated mass lesion**: High- or mixed-density lesion >25 ml; not surgically evacuated.

## PATIENT FOLLOW-UP

<table>
<thead>
<tr>
<th>Outcome</th>
<th>24 HOURS</th>
<th>3 MONTHS</th>
<th>6 MONTHS</th>
<th>12 MONTHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEATH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VEGETATIVE STATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEVERE DISABILITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODERATE DISABILITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL RECOVERY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. DISCUSSION

In the preceding sections we presented the implementation of the Belief Networks as a method to analytically approach the task of clinical prediction in the medical domain of head injuries.

The objective of analytical methods obviously isn’t to replace the physician’s judgement but rather to assist the practitioner in reaching a more accurate prediction. What analytical methods can offer is a quantification of the overall process. Using analytical methods we can assign certain probability values for each prediction and not just a “most likely” choice.

Even more importantly, the codification of the evaluation process allows us to examine thoroughly the mechanism through which the various factors influence the physician’s belief in a certain prediction and reveals the major contributors in this belief. Certain clinical or laboratory variables used in everyday clinical practice retain their true importance in the prediction assessment and others prove to have little or no impact, no matter the value they attain.

But if an analytical method is to be trusted and finally adopted by clinicians, it has to embody the philosophy that intrinsically governs the course of human judgement: given the ability to quantitatively discriminate between the degrees of importance of several clinical and laboratory variables and the experience gained after many similar cases, humans manage to integrate the various cues into a most likely prediction, exploiting the majority, if not all, of the available data.

Currently used analytical methods do not consider the correlations and overlapping between the variables and thus fail to correctly manipulate the available data. Processing the various cues either sequentially or simultaneously, they end up with overpredictions or conservative judgements, respectively.

Belief networks do not bear these methodological limitations. Using Bayesian methodology, they refer to the basis of confidence assessment reasoning: how a situation changes, given a new piece of information. Through the use of joint distributions and of the conditional dependencies of the variables, belief networks evaluate the belief in the various outcomes facing every variable as a different test and updating its belief appropriately under the constraint of the previous test’s outcome, thus covering the need for integration of the data. The exact values of the beliefs in each outcome come naturally and the physician is convinced on the method’s soundness. Furthermore, the transparent way the interactions among different nodes take place and the natural way of information flow can be viewed as a training tool for the physician, either he is an expert or not, helping him to deeply understand a part –at least– of the process he uses in his mind.

Moreover, belief networks offer the possibility to become a tool for testing new types of variables whose importance is yet not exploited. With the statistical analysis of the conditional probability files and its interpretation into the respective link matrices, variables not currently considered to be
information carriers can be checked with regard to the actual impact they have on the clinical prediction procedure.

The experience gained by clinicians throughout the number of head-injury cases they deal with is transferred to the belief network in two ways:

- the choice of the variables included in the network is directly based on generally accepted protocols
- the conditional probability files, backbone of belief networks, include all the necessary information of any combination of variables and outcomes, codifying the totality of information that lies in non-exploitable form in the existing medical files.

The implementation of such systems in remote health care units, especially when the patients’ archives come from the outpatients’ department and not the intensive care unit, can give the opportunity to reach a prognosis using the knowledge acquired in a big general hospital. Another feasible task, although incorporating a larger number of variables and the existence of richer files, is to develop belief networks for diagnostic purposes and for patient treatment. When this is accomplished, a complete system for handling head injury cases will be available at the disposal of every health care unit.

The fact that the method is open to modifications regarding the architecture used, i.e. which variables are included and how they are interconnected, makes it able to accept improvements. Besides, its validity is tested every time a new-comer’s data are fed into the system by the clinical prediction it gives and its correlation with the actual outcome.
4.1 Types of data and their impact on prognostic outcome
Consider a tree configuration like the one shown in the following picture:

Nodes 4, 5, 6, 7 are called leaf nodes and normally they are the nodes that acquire data. The information acquired by leaf nodes is then transmitted to the internal nodes 2, 3 and finally reaches the root node 1.

This interpretation of a tree structure implies that internal nodes do not represent observable variables but rather quantities that are either inaccessible to our direct measurement or that mediate the impact of the leaf nodes’ activation to the root node. The method, however, that we used to reach an optimal architecture constructs trees that all their nodes denote observed variables. Any of the nodes in such trees can therefore acquire information.

Information can be of two types: specific and virtual evidence. Specific evidence corresponds to direct observations which validate with certainty the values of some variables in the network. The entry of specific evidence in a leaf node of our network is treated with the fusion and propagation mechanisms already presented in the method chapter. If the specific evidence enter an internal node, however, all the descendants of this node become inaccessible: these nodes can no longer provide updative information for their parent’s belief.

For example, specific evidence activation of node 4 in the above picture makes nodes 6, 7 unable to receive information. One can view the different levels formed in a tree as groups of variables bearing different information in the sense of the impact transferred to the root node. Nodes 2, 3, 4 form level 1 while nodes 5, 6, 7 form level 2. Nodes of level 1 are more important to the root node’s belief update than nodes of level 2. If we are certain about the value of a level-1 variable we don’t care about the values acquired by the corresponding level-2 variables.

This can be observed in the results we had in the network of the head injury case: mean arterial pressure and urination had the smallest impact on the final outcome. If we are certain about the GCS score of the patient, the knowledge of MAP and even Hb is superfluous.
At this point we should comment on the fact that the parent-child relation in our network does not always exhibit a traditional view of cause-effect connection, at least as obvious as expected.

The algorithm that lead us to this specific architecture is strongly influenced by the patients’ file. The relative position of each variable in the network is the result of procedures involving calculations between the frequencies of occurrence of the variables' outcomes or of their combinations. Apart from the intrinsic bias due to the fact that our files come from the Intensive Care Unit and not from the Outpatients Department, the small number of certain outcome combinations affects the final result.

The whole procedure of the architecture determination does not involve any medical expertise guidance. This aspect of the network could erroneously be interpreted as a handicap. From the comparison of this network with networks constructed using the medical background as a guide to “how the variables should be connected” we can see that the selected architecture leads to better results. Since no bias from pre-established beliefs is induced to the algorithm, the selected architecture is truly the one that best conforms with the available data.

Virtual evidence is the other type of incoming information. It corresponds to judgements based on undisclosed observations affecting the belief in some variables in the network. The subjective probability with which clinicians evaluate a variable’s outcome is of this type of information. As an example we can present the case of a CT-image of poor quality or the inability of the doctor to correctly read the findings hidden therein. He then feeds the system with his estimation: he is 50% sure that there is an midline shift, 30% sure that the hematoma he sees is operable, 70% sure that it is not operable and 90% sure that there is an edema. The entry of this information to the system would be made by momentarily attaching a dummy node to the CT node and posting a message of the form (0.5, 0.3, 0.7, 0.9, 0) to the latter. Note that the numbers need not sum up to unity, permitting each judgment to be formed independently of the other.

Since the clinician is unable to make a statistical analysis over a single patient, his estimations over a variable’s value are always expressed in terms of subjective probability. Obviously, in many cases the relative weight of certain values would be larger than the rest and the above example would perhaps read as (0.9, 0.1, 0, 0, 0) making it almost a certainty on the first outcome of CT.

It is therefore evident that the system's performance is greatly influenced by the user's input. All the benefits of a system based on analytical methods that were presented earlier remain latent unless it is properly fed with the corresponding information. The precision of the incoming data is reflected in the final system's output. In order to fully exploit the condensed experience of a 3-year period of head-injury cases lying in the network’s structure and link matrices, it is essential to assure that the accuracy of the subjective estimates is accordingly high. The specific network, though, avoids—with the exception of the CT—the use of variables that might be considered controversial in their estimation, thus inducing an objective way of data entry
among all the possible users, in remote health care units or in large metropolitan hospitals.

This mode of the network’s usage—the one with virtual evidence entry—does not experience the side effect of the descendant nodes inactivation. In the previous picture, a virtual evidence coming to node 3 is regarded simply as yet another message node 3 receives from its children. The dummy node that provides the virtual evidence is viewed as a momentary child node that transmits a message to its father. Node 3 will acquire a belief evaluating the contribution of all its three descendants: nodes 6, 7 and the dummy node.

The ability of the belief network to accept information in the form of virtual evidence is one of its major advantages over other analytical methods. Expert systems, for instance, are rule based and do not allow the user to be uncertain on any of the input data he feeds the system with. Given the intrinsic limitation of clinical symptoms and signs regarding the degree of confidence we have in them, we can see that belief networks approach the medical knowledge domain in a more consistent fashion.

A quite popular belief among clinicians is that the degree of confidence they have in a certain decision they make is directly proportional to the number of laboratory tests or clinical signs and examinations that support the one or the other alternative. As a consequence there is an ongoing increase in the number of lab tests per patient ordered.

The consequences of such an attitude are obvious:

- Financially: the amount of money spent is huge.
- Emotionally: the patient undergoes procedures that are sometimes painful.
- Scientifically: the abuse of tests not only undermines their prognostic or diagnostic value but also disorientates clinicians in their ability to choose the appropriate set of variables.

In most of the cases, some variables whose values the doctor wants to know through the various tests do not contribute at all to his assessment of the situation. A good doctor should be able to reach a medical decision based on the necessary data and not be misled by lab or clinical results that often contradict.

Belief networks are able to assist the doctor in this task. In constructing a belief network one has to calculate the probabilities of a variable’s outcomes conditioned upon the outcomes of the other variables. In this sense, the true connection between variables with respect to the specific medical decision is revealed.

Consider as an example our belief network of head injuries: although initially many variables were included in the network, in agreement with experts’ opinions, the results showed that some of them have negligible prognostic power. We saw that mean arterial pressure, urination and the patient’s hematocrit do not influence the final outcome, even when they acquire extreme values. Speaking always about prognosis of patients in Intensive
Care Unit, we could say that were these nodes removed from the network, there would be no impact on the network’s accuracy.

Doctors should realize that the acquisition of information through some variables might enable them to monitor the overall patient’s status but does not really contribute to their task of coming to a better prognosis.

Another aspect of this procedure is the accuracy with which the test results are obtained. If, for example, a lab test points to an outcome with a ±30% accuracy then the doctor should evaluate this information properly: the possibility of error in his decision if the latter is based on this result should be taken into account.

This holds for clinical signs and examinations too. The assessment of a patient’s GCS score by an inexperienced doctor could very well cause a false interpretation of the patient’s situation and a failure in his prognosis.

Belief networks can give solution to this problem too. They allow the entry of virtual evidence, i.e. the doctor’s subjective probability on the variable’s outcomes. The inaccuracies of the available data are handled in the proper manner by the network and the final prognostic outcome.

4.2 Medical Database Files

The medical files created from the selective retrieval of the hospital patients’ archives and the corresponding conditional probability libraries constitute the backbone of the belief network.

Their significance is evident in almost every aspect of the network’s structure and function:

1. The architecture of the network is determined by the file. More specifically, the algorithm being used to find the optimal tree structure calculates all the possible expressions of the type $P(A_i / B_j)$ where $A_i$, $B_j$ are the various outcomes of any two variables. Therefore, the importance of the file lies in the specific frequencies of occurrence of the various outcomes.

Files with different content, with respect to the conditional probability libraries, will cause the algorithm to point to different architectures. The impact to the prognostic procedure followed by our system in this case is obviously serious: changing the relative position of the nodes in the network, one changes the relative importance of the respective variables in the prognostic task. Variables whose activation caused minor changes in the root node’s (Glasgow Outcome Scale) belief might become very important in the new configuration.

Since the configuration implied by medical expertise is expected to be unique for each type of patient archives' origin (Intensive Care Unit, Outpatients Dept.), one can conclude that the respective file created by these archives will conform to this unique, yet unknown, architecture.

2. The file’s impact on the information propagation procedure.

Once the architecture is selected, the file is used for the calculation of the elements of the link matrices or the higher-order tensors, in the case of
tree-structured or multiple parents belief networks respectively. These matrices comprise the weights assigned to the internodal links so that information coming from a variable is guaranteed to cause the appropriate impact to the belief of the prognostic variable.

If the link matrices’ elements are altered due to changes to the file, the pieces of information are evaluated differently and therefore the same nodal activations lead to different belief updates.

In the specific network for the case of head injured patients of the ICU, the link matrices use 23 subpopulations of the initial 375 records, specifically those for the outcomes of GOS, GCS, Hb, CT and Age.

For example, one subpopulation is that composed by the patients having the third outcome of GCS, another one by the patients having the second outcome of CT and so on. These subpopulations are used for the calculation of conditional probabilities of the type P(A / GCS=3) or P(A / CT=2) respectively. The number of subpopulations is influenced by both the number of clinical and lab variables, the number of the variables’ possible outcomes and the specific architecture. Due to the fact that this number becomes easily big, the initial file should be large enough so that every subpopulation contains adequate quantity of records so that the relative frequencies are correctly interpreted into probabilities.

We can see that the database file is used as a form of a posteriori knowledge of a number of cases, that enables us to reach credible conclusions regarding the patients’ prognosis. But in order for the data coming from a new case of a head-injured patient of the ICU to be comparable with the existing database files, we should maintain the same protocol under which the latter were acquired, regarding both the time of data acquisition and the classification of the observed values into the preselected outcomes. For example, only data from the moment of arrival of the patient to the ICU should be taken into account; the values of certain variables are likely to change as time passes.

A characteristic of the database file our belief network is based upon is that it was created by experts in the field. Therefore, the prognostic outcome suggested by the network for a new case is based on the best available resources. The application of such a system in remote health care units or in other locations that lack the presence of physicians trained in this medical specialization serves the transfer of quantified experience acquired in a regional general hospital.

4.3 Limitations

Next we shall present the limitations and the problems arising from both the philosophy of the method and its application, especially from the data files used.

One limitation, common to all models of analytical approach, is the one posed to the number of variables used. Contrary to the process implicitly a physician uses, belief networks exploit the predictive ability of only a limited number of main variables. Although this limitation sounds like a disadvantage of the
analytical methods in general, it is important to state that the results obtained
by these methods are comparable or even better than the ones using the
physician’s judgement. Moreover, the modularity of the model, in the sense of
adding or removing some variables, is easily implemented with appropriate
software and can explicitly show how much a certain variable contributes to
the prediction of a specific outcome. The use of excessive number of
variables would disorientate us from the confinement of the main contributors
to the final prediction.

The fact that we assigned a finite —and as small as reasonably achievable—
number to each node’s (i.e. variable’s) outcomes serves a mainly technical
objective: the size of the matrices would grow rapidly if the number of
outcomes became bigger with consequences not only to the time needed for
the calculations but also to the compactness of the model.

Apart from that, the introduction of discrete values or well defined intervals for
the variables’ outcomes can show off the relative importance of our choice of
them to the final prediction. Speaking with examples, it might prove more
useful to divide a person’s blood pressure into 3 intervals than into 5, using
the variation in the prediction as a criterion of usefulness.

A different kind of problem, of probabilistic nature, arises due to the demand
for large populations of data as it is posed by the matrices’ elements. We
remind the reader that a matrix element is of the form \( P(A/B_j) \) or of the form
\( P(A/B_j, C_k, ...) \) for multiple parents architectures. It is obvious that the
population these elements refer to is not the initial population but a part of it,
specifically the portion of the group of patients that satisfy the conditions
\( B=B_j \) or the conditions \( B=B_j \ \text{and} \ C=C_k \ \text{and} ... \). Since our basic assumption
was that, given the adequate size of our database, we can switch from
relative frequencies of occurrence to probabilities, we can see that this hardly
holds true for —at least— some of the combinations of outcomes. The solution
to this difficulty is profound: we need larger and richer patient data files.

If, instead of the ICU, we used patient files from the Outpatient’s Department,
the variety of the existing combinations would be significantly larger. Cases
like “final outcome = good condition AND Glasgow score=8-10 or 11-15” that
now are rarely met would be more common and therefore the corresponding
conditional probabilities would be larger. The impact of such a possibility
would be reflected not only to the link matrices, which are directly influenced
by these probabilities, but also to the choice of the architecture, since the
algorithm used to extract the best tree-approximation internally uses these
probabilities to calculate the branch weights.

4.4 Evaluation Scales
Evaluation scales are often used in clinical practice. Either the purpose is to
describe the severity of a situation or the classification of patients’ progress,
scales offer a compact and neat solution. In order to place a patient or a
situation on a certain level of the scale, we assign to him a score which is the
result of an algorithm that combines data of different aspects of the patient’s
situation.
As an example, let's consider the Glasgow Coma Scale. Three are the components of this scale:
- the eye response (values 1 to 4)
- the verbal response (values 1 to 5)
- the motor response (values 1 to 6)
Depending on the degree of each response and the conditions under which this response is accomplished (voluntary, under pain etc) the clinician assigns to the patient a value for each response. The range of values of the responses can be seen above. For example, a patient might have the following values for his responses:
1. Eye = 2 Verbal = 3 Motor = 5
2. Eye = 3 Verbal = 4 Motor = 3
3. Eye = 1 Verbal = 1 Motor = 2
The algorithm for this scale is the simplest existing: In order to calculate a head-injury patient's score, one sums up the three sub-scores.
The overall score for the above cases would thus be:
1. GCS score = 10
2. GCS score = 10
3. GCS score = 4
Obviously, there more than one combinations of the three components giving the same sum. We can already see that cases 1 and 2 have the same GCS score, although their component values are different.
The question posed naturally is whether a score coming from the summation of three sub-scores can efficiently and correctly describe a patient's condition. Are patients of cases 1 and 2 above sharing the same condition, or the resolving power of the scale is limited and therefore we cannot trust it in all the cases?
Moreover, GCS score is one of the key variables in evaluating a head-injured patient's prognosis. Do patients of the same overall GCS score but with different component configuration share a unique prognosis?
To rephrase the problem, let e, v, m be the values of eye, verbal and motor response respectively. The GCS score will be given by their sum, GCS=e+v+m. Let's consider the case of a certain value of coma scale score GCS=a and let also b be a prognostic outcome. The probability of a patient having this outcome of prognosis given that his GCS score is a will be:
$$P(GOS=b \mid GCS=a) = P (GOS=b \mid e+v+m=a)$$
Does this probability remain constant no matter which are the exact values of e, v, m that form the GCS score e+v+m=a?
Assume we are interested in the probability associated with a specific coma scale score GCS=10 and a specific prognostic outcome, GOS=1 or death. The 17 possible allowed permutations of e, v, m that sum up to 10 are shown in the following table:
Under the term $C_i$ we denote the $i$-th configuration of the values $e$, $v$, $m$. The probability discussed previously will be:

$$P(GOS=1 / GCS=10)$$  \[1\]

and must be compared to the 17 individual probabilities

$$P(GOS=1 / C_i)$$  \[2\]

For example, the corresponding probability for the 6th configuration will be

$$P[GOS=1 / (e=2, v=3, m=5)]$$  \[3\]

This comparison can be made easily if we are given the patient’s file which contains the patient’s prognosis and GCS score, both the sum and the separate components.

What we expect to find is a more–or–less agreement between the probabilities given by [1] and most of the probabilities given by [2]. This would support the wide approval of the Glasgow scale, but would also point to possible deviations for certain response combinations, for which the probability for the specific prognosis might be bigger or smaller than the one assigned to the sum.

The need of a belief network arises from the fact that certain quantities involved can be only subjectively estimated, contrary to the rest that their values can be known without doubt. We can never obtain useful conclusions from a simple analysis of the patient’s file if, for example our data is of the form: “Verbal response is definitely 3, Motor response also definitely 2 but for the eye response we are 40% sure that it is 2, 40% sure that it is 3 and 20% that it is 1”.

This kind of data may however, be entered directly into a belief network and provide us with better accuracy about the prognosis we seek. A tree-structured belief network sufficient for this problem is seen in the figure below:

![Belief Network Diagram]

**Table 2**

<table>
<thead>
<tr>
<th>$C_i$</th>
<th>$e$</th>
<th>$v$</th>
<th>$m$</th>
<th>$C_i$</th>
<th>$e$</th>
<th>$v$</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>13</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>14</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

$GCS = e + v + m = 10$
In order to construct this network, one needs to calculate the elements of the link matrices, which are the “inverse” probabilities of those we mentioned earlier, meaning that the conditioning is on prognosis:

\[ P(\text{Eye}=x \mid \text{GOS}=y) \text{ or } P(\text{Verbal}=x \mid \text{GOS}=y) \text{ or } P(\text{Motor}=x \mid \text{GOS}=y) \]

The matrix M(eye/GOS) will for example be:

\[
M(\text{Eye}/\text{GOS}) =
\begin{bmatrix}
 p(\text{eye} = 1/ \text{GOS} = 1) & p(\text{eye} = 2/ \text{GOS} = 1) & p(\text{eye} = 3/ \text{GOS} = 1) & p(\text{eye} = 4/ \text{GOS} = 1) \\
 p(\text{eye} = 1/ \text{GOS} = 2) & p(\text{eye} = 2/ \text{GOS} = 2) & p(\text{eye} = 3/ \text{GOS} = 2) & p(\text{eye} = 4/ \text{GOS} = 2) \\
 p(\text{eye} = 1/ \text{GOS} = 3) & p(\text{eye} = 2/ \text{GOS} = 3) & p(\text{eye} = 3/ \text{GOS} = 3) & p(\text{eye} = 4/ \text{GOS} = 3) \\
 p(\text{eye} = 1/ \text{GOS} = 4) & p(\text{eye} = 2/ \text{GOS} = 4) & p(\text{eye} = 3/ \text{GOS} = 4) & p(\text{eye} = 4/ \text{GOS} = 4) \\
 p(\text{eye} = 1/ \text{GOS} = 5) & p(\text{eye} = 2/ \text{GOS} = 5) & p(\text{eye} = 3/ \text{GOS} = 5) & p(\text{eye} = 4/ \text{GOS} = 5)
\end{bmatrix}
\]
5. REFERENCES


32. Titterington DM, Murray GD, Murray LS, et al. Comparison of
discrimination techniques applied to a complex data set of head-injured

33. Choi SC, Ward JD, Becker DP. Chart for outcome prediction in severe

34. Narayan, RK, Becker, DP, Enas, GG, et, al Predicting outcome in severe

35. Jagger J, Jane JA, Rimel R. The Glasgow Coma Scale: to sum or not to

36. Teasdale G, Jennett B, Murray L, Murray G. Glasgow Coma Scale : to

early multimodality evoked potentials in severely head-injured patients. A

38. Narayan RK, Greenberg RP, Miller JD, et al. Improved confidence of
outcome prediction in severe head injury. A comparative analysis of the
clinical examination, multimodality evoked potentials, CT scanning and
6. APPENDIX

6.1 Software flow diagrams

Figure 6-A: Flow diagram of network construction software
Figure 6-B: Flow diagram of network activation software
6.2 Software code excerpts

1. Network construction software

/* CREATE VECTORS, MATRIX, BELIEF VARIABLES */
struct point
{
    char title[61];
    int psa, psa1;
    long next[10];
    long prev;
    long outc[10];
    int mesa, katv, ycos;
    float val[10][10];
    float bel[10];
    float lin[10], pin[10], lout[10];
    float pout[10][10];
    float lprev[10];
    char closed, flag1, flag2, flag3;
    COLORREF fg, bg;
    int eidos;
} Xpoint;

/* OPEN A FILE IN SUBDIRECTORY IBIS? */
int my_open(char nm[], int a, int b)
{
    char fg[100];
    if (CASE < 0) return(-1);
    wsprintf(fg, "IBIS%d%c%s", CASE, 92, nm);
    return(open(fg, a, b));
}

/* SET A NEW NODE. DEFAULT VALUES FOR MATRIX AND VECTORS */
/* SET NO CHILD AND NO OUTCOMES */
/* SET PARENT TO pr */
void new_point(COLORREF c1, COLORREF c2, char *ti, long pr, struct point *p)
{
    int f, x;
    p->fg = c1;
    p->bg = c2;
    for (f = 0; f < 10; f++)
    {
        p->next[f] = -1;
        p->outc[f] = -1;
    }
    for (f = 0; f < 10; f++)
    {
        p->bel[f] = 0;
        p->lin[f] = 0;
        p->lprev[f] = 0;
        p->lout[f] = 0;
        for (x = 0; x < 10; x++)
        {
            p->val[f][x] = 0;
            p->pout[f][x] = 0;
        }
    }
    p->psa = 0;
    p->psa1 = 0;
    p->mesa = 0;
    p->katv = 0;
    p->ycos = 1;
    p->prev = pr;


```c
strcpy(p->title, ti);
p->eidos = 0;
}
/* CALCULATE AND RETURN THE NUMBER OF NODES */
long posa_point()
{
    int fd;
    long te;
    fd = my_open("point.plh", O_RDONLY, 0x180);
    te = filelength(fd) / sizeof(point);
    close(fd);
    return (te);
}

/* FIND A DELETED NODE*/
long null_point()
{
    int fd;
    long l, te, l1, l2 = -1;
    fd = my_open("point.plh", O_RDONLY, 0x180);
    te = filelength(fd) / sizeof(point);
    for (l = 0; l < te; l++) {
        lseek(fd, l * sizeof(point) + 105, 0);
        read(fd, &l1, 4);
        if (l1 == -5) {
            l2 = l;
            l = te + 2;
        }
    }
    close(fd);
    return (l);
}

/* DELETE A NODE AND ITS CHILDREN */
void set_null_point0(int fd, long l)
{
    int x;
    long l1 = -5;
    lseek(fd, l * sizeof(point) + 105, 0);
    l1 = -5;
    x = -1;
    lseek(fd, (l + 1) * sizeof(point) - 2, 0);
    write(fd, &x, 2);
    for (x = 0; x < 10; x++) {
        lseek(fd, l * sizeof(point) + 65 + 4 * x, 0);
        read(fd, &l1, 4);
        if (l1 >= 0) set_null_point0(fd, l1);
    }
}

void set_null_point(long l)
{
    int fd;
    fd = my_open("point.plh", O_RDONLY, 0x180);
    set_null_point0(fd, l);
    close(fd);
}

void read_point(long a, struct point *p)
{
    int fd;
    fd = my_open("point.plh", O_RDONLY, 0x180);
    lseek(fd, a * sizeof(point), 0);
```
read(fd,p,sizeof(point));
close(fd);
}
long write_point(long a,struct point *p)
{
  int fd;
  long l;

  if (a<0) a=null_point();

  fd=my_open("point.plh",0x8104,0x180);
  if (a<0) l=filelength(fd)/sizeof(point);
  else l=a;
  lseek(fd,l*sizeof(point),0);
  write(fd,p,sizeof(point));
  close(fd);
  return(l);
}
/* ROUTINES OF THE DIALOG BOX FOR THE ENTRY OF MATRIX VALUES */
class Tvalues:public TDialog
{
  public:
    struct point p,p1;
    long poio;
    int f,m,x,fd;
    char fg[100];
  Tvalues(PTWindowsObject Ap,LPSTR At,long pp);
  BOOL CanClose();
  virtual void SetupWindow();
};
Tvalues::Tvalues(PTWindowsObject Ap,LPSTR At,long pp) :TDialog(Ap,At)
{
  poio=pp;
  read_point(pp,&p);
}
/* SHOW THE PREVIOUSLY STORED VALUES */
void Tvalues::SetupWindow()
{
  TDialog::SetupWindow();
  fd=my_open("outcom.plh",0x8104,0x180);
  for (f=0;f<10;f++) if (p.outc[f]>=0) {
    lseek(fd,p.outc[f]*100,0);
    ::read(fd,fg,96);
  SetDlgItemText(HWindow,211+f,fg);
  }
  if (p.prev>=0) {
    read_point(p.prev,&p1);
    for (f=0;f<10;f++) if (p1.outc[f]>=0) {
      lseek(fd,p1.outc[f]*100,0);
      ::read(fd,fg,96);
    SetDlgItemText(HWindow,201+f,fg);
    }
    close(fd);
    for (f=0;f<10;f++) for (x=0;x<10;x++) if (p.val[f][x]!=0){
      sprintf(fg,"%-10.2f",p.val[f][x]);
    SetDlgItemText(HWindow,100+f*10+x,fg);
    }
  }
/* ACQUIRE NEW VALUES */
BOOL Tvalues::CanClose()
{
    BOOL ret=TRUE;
    double s;
    int n=0,t;
    for (f=0;f<10;f++) {
        s=0;
        for (x=0;x<10;x++) {
            GetDlgItemText(HWindow,100+f*10+x,fg,10);
            p.val[f][x]=atof(fg);
            s+=p.val[f][x];
        }
        write_point(poio,&p);
        if (ret==FALSE) SetFocus(GetDlgItem(HWindow,100+n*10));
        return(ret);
    }
}

/* VALUES GIVEN BY USER IN “PREPARE” MODE ENTER NODE 0 (“START POINT” NODE) */
class Ttimes:public TDialog
{
public:
    struct point p;
    int f,m,x,fid;
    char fg[100];
    Ttimes(PTWindowsObject Ap,LPSTR At);
    BOOL CanClose();
    virtual void SetupWindow();
};
Ttimes::Ttimes(PTWindowsObject Ap,LPSTR At) :TDialog(Ap,At)
{
    read_point((long)0,&p);
}

/* SHOW THE PREVIOUSLY STORED VALUES */
void Ttimes::SetupWindow()
{
    TDialog::SetupWindow();
    SetDlgItemText(HWindow,200,p.title);
    for (f=0;f<10;f++){
        sprintf(fg,"%10.2f",p.pin[f]);
        SetDlgItemText(HWindow,101+f,fg);
    }
}

/* ACQUIRE NEW VALUES */
BOOL Ttimes::CanClose()
{
    float pset[10];
    int fd;
    for (f=0;f<10;f++){
        GetDlgItemText(HWindow,101+f,fg,10);
        pset[f]=p.pin[f]=atof(fg);
        write_point((long)0,&p);
        fd=my_open("setval.plh",0x8104,0x180);
        ::write(fd,pset,sizeof(pset));
        ::close(fd);
        return(TRUE);
    }
/* “NEW CASE” OR “EDIT CASE” USER’S CHOICE */
class TnewCase:public TDialog
{
public:
    struct {
        char name[61];
        char author[61];
        char date1[10],date2[10];
    } cas;
    int f,m,fd,te,poio;
    char fg[100];
    TnewCase(PTWindowsObject Ap,LPSTR At,int pp);
    BOOL CanClose();
    virtual void SetupWindow();
};
TnewCase::TnewCase(PTWindowsObject Ap,LPSTR At,int pp) :TDialog(Ap,At)
{
    poio=pp;
}
void TnewCase::SetupWindow()
{
    TDIGIT::SetupWindow();
    if (poio>=0) {
        fd=::open("case.plh",0x8004,0x180);
        lseek(fd,(long)poio*sizeof(cas),0);
        ::read(fd,&cas,sizeof(cas));
        ::close(fd);
        SetDlgItemText(HWindow,101,cas.name);
        SetDlgItemText(HWindow,102,cas.author);
        SetDlgItemText(HWindow,103,cas.date1);
        SetDlgItemText(HWindow,104,cas.date2);
    }
}
BOOL TnewCase::CanClose()
{
    int f;
    GetDlgItemText(HWindow,101,fg,61);
    if (fg[0]==0) return(FALSE);
    strcpy(cas.name,fg);
    GetDlgItemText(HWindow,102,fg,61);
    strcpy(cas.author,fg);
    GetDlgItemText(HWindow,103,fg,10);
    strcpy(cas.date1,fg);
    GetDlgItemText(HWindow,104,fg,10);
    strcpy(cas.date2,fg);
    fd=::open("case.plh",0x8104,0x180);
    if (poio<0) te=filelength(fd)/sizeof(cas);
    else te=poio;
    lseek(fd,(long)te*sizeof(cas),0);
    ::write(fd,&cas,sizeof(cas));
    ::close(fd);
    f=CASE;
    CASE=te;
    wsprintf(fg,"IBIS%d",CASE);
    mkdir(fg);
    if (poio<0) {
        struct point p;
new_point(MColors[4],MColors[0],"Start point",(long)-1,&p);
    write_point((long)0,&p);
}
CASE=f;
return(TRUE);

/* "SELECT CASE" USER'S CHOICE */
class TselectCase:public TDialog
{
    public:
    struct {
        char name[61];
        char author[61];
        char date1[10], date2[10];
    } cas;
    PTListBox tl;
    int f,m,fd,te;
    char fg[100];
    TselectCase(PTWindowsObject Ap, LPSTR At);
    ~TselectCase();
    void SelectCaseFonts(int f);
    BOOL CanClose();
    virtual void SetupWindow();
    virtual void new_case()=[ID_FIRST+102];
    virtual void edit_case()=[ID_FIRST+103];
};
void TselectCase::SelectCaseFonts(int f)
{
    char fg[100];
    int fd,rr=0;
    static CHOOSEFONT cfTemp;
    static LOGFONT lfTemp;
    sprintf(fg,"ibis%cf.plh",f+48);
    fd=::open(fg,0x8004,0x180);
    if (fd>=0) {
        ::read(fd,m_font,sizeof(m_font));
        close(fd);
        rr=1;
    }
    for (fd=0;fd<2;fd++) {
        lfTemp=m_font[fd];
        ifTemp=m_font[fd];
        ifTemp.IStructSize = sizeof( CHOOSEFONT );
        ifTemp.hwndOwner = HWindow;
        ifTemp.hDC = 0;
        ifTemp.lpLogFont = &lfTemp; // Store the result here
        ifTemp.Flags = CF_EFFECTS | CF_FORCEFONTEXIST |
                    CF_SCREENFONTS | CF_INITTOLOGFONTSTRUCT | CF_APPLY;
        if (rr==0 & fd==0) ifTemp.Flags &= ~CF_INITTOLOGFONTSTRUCT;
        ifTemp.rgbColors = RGB(0,0,0); // Color and font dialogs use the same color
        ifTemp.iCustData = 0;
        ifTemp.lpfnHook = NULL;
        ifTemp.lpTemplateName = NULL;
        ifTemp.hInstance = 0;
        ifTemp.lpszStyle = NULL;
        ifTemp.nFontType = SCREEN_FONTTYPE;
        ifTemp.nSizeMin = 0;
        ifTemp.nSizeMax = 0;
        if( ChooseFont( &ifTemp ) == TRUE ) {
            m_font[fd]=lfTemp;
            if (rr==0 & fd==0) m_font[fd+1]=lfTemp;
TselectCase::TselectCase(PTWindowsObject Ap, LPSTR At) : TDialog(Ap, At)
{
    tl = new TListBox(this, 101);
}
TselectCase::~TselectCase()
{
    delete(tl);
}
void TselectCase::SetupWindow()
{
    TDialog::SetupWindow();
    fd = ::open("case.plh", 0x8004, 0x180);
    if (fd >= 0) {
        te = filelength(fd) / sizeof(cas);
        for (f = 0; f < te; f++) {
            lseek(fd, (long)f * sizeof(cas), 0);
            ::read(fd, &cas, sizeof(cas));
            tl->AddString(cas.name);
        }
        ::close(fd);
    }
}

/* CALL TnewCase CLASS (USER PRESSED THE "NEW CASE" BUTTON)*/
void TselectCase::new_case()
{
    if (GetApplication()->ExecDialog(new TnewCase(this, "NEW_CASE", -1)) == IDOK) {
        fd = ::open("case.plh", 0x8004, 0x180);
        te = filelength(fd) / sizeof(cas);
        te--;
        lseek(fd, (long)te * sizeof(cas), 0);
        ::read(fd, &cas, sizeof(cas));
        tl->AddString(cas.name);
        ::close(fd);
    }
}

/* CALL TnewCase CLASS (USER PRESSED THE "EDIT CASE" BUTTON)*/
void TselectCase::edit_case()
{
    f = tl->GetSelIndex();
    if (f >= 0) {
        if (GetApplication()->ExecDialog(new TnewCase(this, "NEW_CASE", f)) == IDOK) {
            fd = ::open("case.plh", 0x8004, 0x180);
            lseek(fd, (long)f * sizeof(cas), 0);
            ::read(fd, &cas, sizeof(cas));
            tl->DeleteString(f);
            tl->InsertString(cas.name, f);
            ::close(fd);
        }
    }
}
/* USER PRESSED "O.K." BUTTON */
BOOL TselectCase::CanClose()
{
    int f;
    f=tl->GetSelIndex();
    if (f<0) return(FALSE);
    CASE=f;
    SelectCaseFonts(f);
    return(TRUE);
}

/* NODES' OUTCOMES */
class Toutcomes:public TDialog
{
public:
    PTListBox tl;
    struct point p;
    long poio,poy,te;
    int f,m,fd;
    char fg[100];
    Toutcomes(PTWindowsObject Ap,LPSTR At,long pp);
    ~Toutcomes();
    BOOL CanClose();
    virtual void SetupWindow();
};
Toutcomes::Toutcomes(PTWindowsObject Ap,LPSTR At,long pp) :TDialog(Ap,At)
{
    poio=pp;
    read_point(pp,&p);
    tl=new TListBox(this,112);
}
Toutcomes::~Toutcomes()
{
    delete (tl);
}

void Toutcomes::SetupWindow()
{
    TDialog::SetupWindow();
    SetDlgItemText(HWindow,101,p.title);
    fd=my_open("outcom.plh",0x8104,0x180);
    te=filelength(fd)/100;
    for (f=0;f<te;f++) {
        lseek(fd,(long)f*100,0);
        ::read(fd,fg,96);
        ::read(fd,&poy,4);
        if (poy!=poio && poy>=0) tl->AddString(fg);
    }
    for (f=0;f<10;f++) if (p.outc[f]>=0) {
        lseek(fd,p.outc[f]*100,0);
        ::read(fd,fg,96);
        ::read(fd,&poy,4);
        SetDlgItemText(HWindow,102+f,fg);
    }
    ::close(fd);
}

BOOL Toutcomes::CanClose()
{
    fd=my_open("outcom.plh",0x8104,0x180);
    for (f=0;f<10;f++) {
        GetDlgItemText(HWindow,102+f,fg,61);
        if (fg[0]!=0) {
            if (p.outc[f]>=0) {
                lseek(fd,p.outc[f]*100,0);
            }
        }
    }
}
/* USER PRESSED “OUTCOMES” */
void Tprocess::setoutcom()
{
    write_point(poio,&p);
    GetApplication()->ExecDialog(new Toutcomes(this,"OUTCOMES",poio));
    read_point(poio,&p);
}

/* USER PRESSED “VALUES” */
void Tprocess::setvalues()
{
    write_point(poio,&p);
    GetApplication()->ExecDialog(new Tvalues(this,"VALUES",poio));
    read_point(poio,&p);
}

/* USER PRESSED “O.K.” */
BOOL Tprocess::CanClose()
{
    int m,dy,ycos=1;
    long prv;
    HCURSOR cc;
    cc=SetCursor(LoadCursor(NULL,IDC_WAIT));
    for (f=0;f<10;f++) {
        m=SendDlgItemMessage(HWindow,124+f,BM_GETCHECK,0,0);
        if (m==BF_CHECKED) SendDlgItemMessage(HWindow,114+f,BM_SETCHECK,FALSE,0);
    }
    GetDlgItemText(HWindow,101,fg,61);
    strcpy(p.title,fg);
    for (f=0;f<10;f++) {
        GetDlgItemText(HWindow,104+f,fg[0],60);
        if (fg[0]!=0) {
            if (p.next[f]>=0) {
                read_point(p.next[f],&np);
                ycos+=np.ycos;
                strcpy(np.title,fg);
                m=SendDlgItemMessage(HWindow,124+f,BM_GETCHECK,0,0);
                if (m==BF_CHECKED) np.eidos=2;
                else np.eidos=0;
                write_point(p.next[f],&np);
            }
            else {
                new_point(p.fg,p.bg,fg,poio,&np);
                m=SendDlgItemMessage(HWindow,124+f,BM_GETCHECK,0,0);
                if (m==BF_CHECKED) np.eidos=2;
                else np.eidos=0;
                np.mesa=p.mesa+1;
                ycos+=np.ycos;
                p.next[f]=write_point((long)-1,&np);
            }
        }
        else {
            if (p.next[f]>=0) {
                read_point(p.next[f],&np);
                np.prev=-5;
                ycos+=np.ycos;
                p.next[f]=-1;
                p.eidos=-1;
                set_null_point(p.next[f]);
            }
        }
    }
    for (f=0;f<10;f++) if (p.next[f]>=0) {
        m=SendDlgItemMessage(HWindow,124+f,BM_GETCHECK,0,0);
if (ycos>1) ycos--;
dy=ycos-p.ycos;
p.ycos=ycos;
write_point(polo,&p);
if (dy!=0) prv=p.prev;
else prv=-1;
while (prv>=0) {
    read_point(prv,&p);
    p.ycos+=dy;
    write_point(prv,&p);
    prv=p.prev;
}
KATV=0;
ypolo_katv(0);
SetCursor(cc);
return(TRUE);

//----------------------------------------------------------------------------------
/* MAIN LOOP: WAITING FOR ANY RESPONSE FROM THE USER */
class StartW:public TBWindow
{
    public:
    long l;
    char fg[100];
    int dn,moving;
    StartW(PTWindowsObject Ap,LPSTR At);
    virtual void SetupWindow();
    virtual void Paint(HDC dc,PAINTSTRUCT _FAR & ps);
    virtual void deije(HDC dc);
    virtual void deije_case();
    long find_poio(int x,int y,int *ei);
    virtual void WMLButtonDown(RTMessage m)=[WM_FIRST+WM_LBUTTONDOWN];
    void S0()=[CM_FIRST+200];
    void S1()=[CM_FIRST+101];
    void S2()=[CM_FIRST+102];
    void S3()=[CM_FIRST+103];
    void S5()=[CM_FIRST+105];
};

void StartW::deije_case()
{
    InvalidateRect(HWindow,NULL,TRUE);
    UpdateWindow(HWindow);
}
void StartW::SetupWindow()
{
    TBWindow::SetupWindow();
}

void StartW::S0()
{
    GetApplication() ->ExecDialog(new TDialog(this,"ABOUT"));
}
StartW::StartW(PTWindowsObject Ap,LPSTR At):TBWindow(Ap,At)
{
    Attr.Style|=WS_HSCROLL|WS_VSCROLL| WS_MAXIMIZE;
    moving=0;
void StartW::find_poio(int x,int y,int *ei)
{
    long s,te;
    long f=-1;
    int xa,ya,yc;
    struct point p;
    if (MSTYLE==3) return(-1);
    te=posa_point();
    for (s=0;s<te;s++) {
        read_point(s,&p);
        if (p.prev>=-1) {
            xa=p.mesa*DX1+DX2;
            ya=p.katv*DX3+DX4;
            yc=(p.ycos-1)*DX3+(DX3-DX4);
            if (MSTYLE==1) {
                ya=ya+(yc/2)-(DX3/2);
            }
            if (MSTYLE==2);
            if (MSTYLE>=20) {
                if (MSTYLE<yc) {
                    ya+=((yc-MSTYLE)/2);
                    yc=MSTYLE;
                }
            }
            if (x>=xa && x<=xa+DX1-DX2)
            if (y>=ya && y<=ya+yc) {
                strcpy(fg,p.title);
                *ei=p.eidos;
                f=s;
                s=te+2;
            }
        }
    }
    if (f<0) fg[0]=0;
    return(f);
}

void StartW::WMLButtonUp(RTMessage m)
{
    int x,y,k;
    long l;
    x=m.LP.Lo+(int)(Scroller->XPos*Scroller->XUnit);
    y=m.LP.Hi+(int)(Scroller->YPos*Scroller->YUnit);
    l=find_poio(x,y,&k);
    if (l>=0) {
        if (GetApplication()->ExecDialog(new Tprocess(this,"PROCESS",l))==IDOK) {
            deije_case();
        }
    }
}

" SKETCH THE NETWORK */
void StartW::deije(HDC dc)
{
struct point p;
long f,te;
int fd;
int x,y,yc,ya,xa,xb,yb;
RECT R,R1;
HBRUSH bra;
HFONT of,nf;
HPEN hpen,hopen;
f=CreateFontIndirect((LPLOGFONT)&m_font[0]);
of=(HFONT)SelectObject(dc,nf);

for (f=0;f<te;f++) {
    read_point(f,&p);
    if (p.prev>=-1) {
        x=p.mesa*DX1+DX2;
        y=p.katv*DX3+DX4;
        yc=(p.ycos-1)*DX3+(DX3-DX4);
        xa=x;
        ya=y+yc/2;
        if (MSTYLE==1) {
            y=ya-DX3/2;
            yc=DX3;
        } else if (MSTYLE>=20) {
            if (MSTYLE<yc) {
                y+=((yc-MSTYLE)/2);
                yc=MSTYLE;
            }
        }
        else if (MSTYLE==2) {
            int ss;
            ss=5+HIWORD(GetTextExtent(dc,p.title,strlen(p.title)));
            x=p.mesa*DX2;
            y=p.katv*ss;
            yc=ss-2;
        }
    }
    R1.left=x;
    R1.top=y;
    R1.right=x+DX1-DX2;
    R1.bottom=y+yc;
    bra=CreateSolidBrush(p.fg);
    FillRect(dc,(LPRECT)&R1,bra);
    DeleteObject(bra);
    bra=CreateSolidBrush(p.bg);
    FrameRect(dc,(LPRECT)&R1,bra);
    DeleteObject(bra);
}
if (DX1>30) {
    SetBkMode(dc,TRANSPARENT);
    SetTextColor(dc,p.bg);
    R.left=x+4;
    R.right=x+DX1-DX2;
    R.top=y+2;
    R.bottom=y+yc;
    DrawText(dc,p.title,strlen(p.title),&R,DT_LEFT|DT_WORDBREAK);
}
if (p.prev>=0) {
read_point(p.prev,&p);
  x=p.mesa*DX1+DX2;
  y=p.katv*DX3+DX4;
  yc=(p.ycos-1)*DX3+(DX3-DX4);
  xb=x+DX1-DX2;
  yb=y+yc/2;
  SetBkMode(dc,TRANSPARENT);
if (MSTYLE!=3) {
  hpen=CreatePen(PS_SOLID,1,RGB(0,0,0));
  hopen=(HPEN)SelectObject(dc,hpen);
  MoveTo(dc,xa,ya);
  LineTo(dc,xb,yb);
  SelectObject(dc,hopen);
  DeleteObject(hpen);
}
DeleteObject(SelectObject(dc,of));
}

void StartW::Paint(HDC dc,PAINTSTRUCT _FAR & ps)
{
  if (CASE<0) {
  }
  else {
    deije(dc);
  }
}

void StartW::S1()
{
  if (GetApplication()->ExecDialog(new TselectCase(this,"SELECT_CASE"))==IDOK) {
    AssignMenu("MENU_1");
    deije_case();
  }
}

void StartW::S2()
{
  CloseWindow();
}

void StartW::S3()
{
  long l,te;
  struct point p;
  int f,x;
  te=posa_point();
  for (l=0;l<te;l++) {
    read_point(l,&p);
    for (f=0;f<10;f++) {
      p.lin[f]=1;
      p.iprev[f]=1;
      p.bel[f]=0;
      p.lout[f]=1;
      p.pin[f]=1;
      for (x=0;x<10;x++) p.pout[f][x]=1;
    }
    p.closed=0;
    p.flag1=0;
    p.flag2=0;
    p.flag3=0;
    write_point(l,&p);
  }
}
GetApplication()->ExecDialog(new Ttimes(this,"TIMES"));
CASE=-1;
AssignMenu("MENU_0");
deje_case();

/* INITIAL ROUTINE FOR WINDOWS */
class TTestApp : public TApplication
{
public:
    TTestApp(LPSTR AName, HINSTANCE hInstance, HINSTANCE hPrevInstance,
        LPSTR lpCmdLine, int nCmdShow)
        : TApplication(AName, hInstance, hPrevInstance, lpCmdLine, nCmdShow) {};
    virtual void InitMainWindow();
    virtual void InitInstance();
};

void TTestApp::InitInstance()
{
    TApplication::InitInstance();
}

void TTestApp::InitMainWindow()
{
    MainWindow = new StartW(NULL, " Author ");
}

int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{
    TTestApp TestApp("Author", hInstance, hPrevInstance,
        lpCmdLine, nCmdShow);
    TestApp.Run();
    return TestApp.Status;
}
2. Network activation software

int CASE=-1;

/* ROUTINE FOR MULTIPLICATION OF MATRICES OF THE FORM c[i]=a[i][j]*b[j] */
void polapl(float c[], float a[][10], float b[], int o)
{
    int f, x;
    float pp[10][10], n;
    for (f=0; f<10; f++) for (x=0; x<10; x++) {
        if (o==0) pp[f][x]=a[f][x];
        else pp[f][x]=a[x][f];
    }
    for (f=0; f<10; f++) {
        n=0;
        for (x=0; x<10; x++) {
            n+=pp[f][x]*b[x];
        }
        c[f]=n;
    }
}

void polapl1(float c[], float a[], float b[])
{
    int f;
    for (f=0; f<10; f++) c[f]=a[f]*b[f];
}

/* CALCULATE ALL VALUES FROM NODE “poio” AND ITS SUCCESSORS */
void bres_ola(long poio, int o)
{
    struct point p, p1;
    int f, x, s;
    float tout[10], tlin[10], tmp[10];
    read_point(poio, &p);
    s=1;
    if (o!=0) {
        s=0;
        for (f=0; f<10; f++) {
            p.lprev[f]=p.lin[f];
            p.lin[f]=1;
        }
    }
    if (p.flag1==0) {
        for (f=0; f<10; f++) if (p.next[f]>=0) {
            read_point(p.next[f], &p1);
            s++;
            for (x=0; x<10; x++) {
                p.lin[x]=p1.lout[x];
            }
        }
    }
    if (UELV_EKT==1) {
        fprintf(stdprn,"lin ");
        for (f=0; f<10; f++) fprintf(stdprn,"%.4f ",p.lin[f]);
        fprintf(stdprn,"\n\n");
    }
}
if (p.prev>=0) {
    read_point(p.prev,&p1);
    s=-1;
    for (f=0;f<10;f++) if (p1.next[f]==poio) s=f,f=12;
    if (s>=0) for (f=0;f<10;f++) {
        p.pin[f]=p1.pout[s][f];
    }
}
else {
}

polapl(p.lout,p.val,p.lin,0);
polapl(tout,p.val,p.pin,1);

    polapl1(p.bel,tout,p.lin);
    for (x=0;x<10;x++) if (p.next[x]>=0) {
        for (f=0;f<10;f++) tlin[f]=1;
        for (f=0;f<10;f++) if (p.next[f]>=0 && f!=x) {
            read_point(p.next[f],&p1);
            for (s=0;s<10;s++) {
                tlin[s]*=p1.lout[s];
            }
            polapl1(tmp,tout,tlin);
            for (f=0;f<10;f++) {
                p.pout[x][f]=tmp[f];
                if (UELV_EKT==1) fprintf(stdout,"%6.4f ",tmp[f]);
            }
        }
    }
write_point(poio,&p);

/* CHANGE VALUES OF NODE “poio” DUE TO CHANGES TO NODE “apo” */

void enhmervse(long poio,long apo)
{
    int f;
    if (apo<0) bres_ola(poio,0);
    else bres_ola(poio,1);
    read_point(poio,&Xpoint);
    // if (p.closed==1) return;
    for (f=0;f<10;f++) if (Xpoint.next[f]>=0 && Xpoint.next[f]!=apo) {
        enhmervse(Xpoint.next[f],poio);
        read_point(poio,&Xpoint);
    }
    if (Xpoint.prev>=0 && Xpoint.prev!=apo) {
        enhmervse(Xpoint.prev,poio);
        read_point(poio,&Xpoint);
    }
}

/* INHIBIT INFORMATION BACK-FLOW TO A NODE DUE TO ITS OWN ACTIVATION */

void kleise(long poio)
{
    int f;
    read_point(poio,&Xpoint);
    Xpoint.closed=1;
    write_point(poio,&Xpoint);
    for (f=0;f<10;f++) if (Xpoint.next[f]>=0) {
        kleise(Xpoint.next[f]);
        read_point(poio,&Xpoint);
    }
class Tvalues1:public TDialog
{
public:
    struct point p,p1;
    long poio;
    int f,m,x,fd,kanei;
    double dd;
    char fg[100];
    Tvalues1(PTWindowsObject Ap,LPSTR At,long pp,int kn);
    BOOL CanClose();
    virtual void SetupWindow();
};

Tvalues1::Tvalues1(PTWindowsObject Ap,LPSTR At,long pp,int kn) :TDialog(Ap,At)
{
    poio=pp;
    kanei=kn;
    read_point(pp,&p);
}

void Tvalues1::SetupWindow()
{
    TDialog::SetupWindow();
    SetCaption(p.title);
    if (kanei==2) SetDlgItemText(HWindow,120,"CLOSED");
    dd=0;
    for (f=0;f<10;f++) dd+=p.bel[f];
    for (f=0;f<10;f++) if (p.lin[f]!=0)
        sprintf(fg,"%10.2f",p.lin[f]);
    SetDlgItemText(HWindow,101+f,fg);
    sprintf(fg,"%10.4f",p.pin[f]);
    SetDlgItemText(HWindow,121+f,fg);
    if (dd!=0) sprintf(fg,"%10.4f",p.bel[dd]/dd);
    else fg[0]=0;
    SetDlgItemText(HWindow,131+f,fg);
}

BOOL Tvalues1::CanClose()
{
    int m;
    if (kanei==2) return(TRUE);
    for (f=0;f<10;f++)
    { 
        GetDlgItemText(HWindow,101+f,fg,10);
        p.lin[f]=atof(fg);
    }
    m=(int)SendDlgItemMessage(HWindow,111,BM_GETCHECK,0,0);
    if (m==BF_CHECKED) p.flag1=1;
    write_point(poio,&p);
    if (m==BF_CHECKED) kleise(poio);
    return(TRUE);
}

class TselectCase:public TDialog
{// ----------------------------------------------

class StartW:public TBWindow
{ public:
    long l;
    char fg[100];
    HCURSOR h0,hw;
}
StartW(PTWindowsObject Ap,LPSTR At);
virtual void SetupWindow();
virtual void Paint(HDC dc,PAINTSTRUCT _FAR & ps);
virtual void deije(HDC dc);
virtual void deije_case();
long find_poi(int x,int y,int *ei);
virtual void WMPaint(RTMessage Msg)=[WM_FIRST+WM_PAINT];
virtual void WMSize(RTMessage Msg)=[WM_FIRST+WM_SIZE];
virtual void WMLButtonUp(RTMessage m)=[WM_FIRST+WM_LBUTTONDOWN];
virtual void WMSetCursor(RTMessage m)=[WM_FIRST+WM_SETSCURSOR];

void S0()=[CM_FIRST+200];
void S1()=[CM_FIRST+101];
void S3()=[CM_FIRST+103];
};
void StartW::SetupWindow()
{
    TBWindow::SetupWindow();
}
void StartW::S0()
{
    GetApplication()->ExecDialog(new TDialog(this,"ABOUT"));
}

/* ROUTINE FOR “RESET” FUNCTION THAT RETURNS THE NETWORK TO THE “PREPARE” STATE */
void StartW::S3()
{
    long l,te;
    struct point p;
    int f,x;
    te=posa_point();
    for (l=0;l<te;l++)
    {
        read_point(l,&p);
        for (f=0;f<10;f++)
        {
            p.lin[f]=1;
            p.lprev[f]=1;
            p.bel[f]=0;
            p.lout[f]=1;
            p.pin[f]=1;
            p.lout[f]=1;
            for (x=0;x<10;x++) p.pout[f][x]=1;
        }
        p.closed=0;
        p.flag1=0;
        p.flag2=0;
        p.flag3=0;
        write_point(l,&p);
    }
    float pset[10];
    int fd;
    read_point((long)0,&p);
    fd=my_open("setval.plh",0x8104,0x180);
    ::read(fd,pset,sizeof(pset));
    ::close(fd);
    for (fd=0;fd<10;fd++) p.pin[fd]=pset[fd];
    write_point((long)0,&p);
    enhmvse((long)0,-1);
}
StartW::StartW(PTWindowsObject Ap,LPSTR At):TBWindow(Ap,At)
/* DATA ENTRY ROUTINE: IF A NODE IS SELECTED WITH THE MOUSE, ASKS FOR USER INPUT
AND INFORMS THE NETWORK FOR THE CHANGES */
void StartW::WMLButtonUp(RTMessage m)
{
    int x, y, k;
    long l;
    x = m.LP.Lo + (int)(Scroller->XPos*Scroller->XUnit);
    y = m.LP.Hi + (int)(Scroller->YPos*Scroller->YUnit);
    l = find_poi(x, y, &k);
    if (l >= 0) {
        if (GetApplication()->ExecDialog(new Tvalues1(this, "VALUES1", l, k)) == IDOK) {
            if (k != 2) enhmervse(l, (long)-1);
            deije_case();
        }
    }
    deije_case();
}

void StartW::S1()
{
    if (GetApplication()->ExecDialog(new TselectCase(this, "SELECT_CASE")) == IDOK) {
        AssignMenu("MENU_1");
        deije_case();
    }
}

/* WINDOWS RELATED ROUTINES */
class TTestApp : public TApplication
{
public:
    TTestApp(LPSTR AName, HINSTANCE hInstance, HINSTANCE hPrevInstance,
        LPSTR lpCmdLine, int nCmdShow)
    : TApplication(AName, hInstance, hPrevInstance, lpCmdLine, nCmdShow) {}
    virtual void InitMainWindow();
    virtual void InitInstance();
};

void TTestApp::InitInstance()
{
    TApplication::InitInstance();
}

void TTestApp::InitMainWindow()
{
    MainWindow = new StartW(NULL, "Animator");
}

int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{
    TTestApp TestApp("Animator", hInstance, hPrevInstance,
        lpCmdLine, nCmdShow);
    TestApp.Run();
    return TestApp.Status;