



The University of Sydney

Neural-Logic Belief Networks

Technical Report Number 486

August 1994

Boon Toh Low and Norman Y Foo

ISBN 0 86758 929 9

**Basser Department of Computer Science
University of Sydney NSW 2006**

Neural-Logic Belief Networks (NLBN)

Boon Toh LOW Norman Y. FOO

Knowledge Systems Group
Basser Department of Computer Science
University of Sydney
NSW 2006, Australia

email : boontoh@cs.su.oz.au norman@cs.su.oz.au

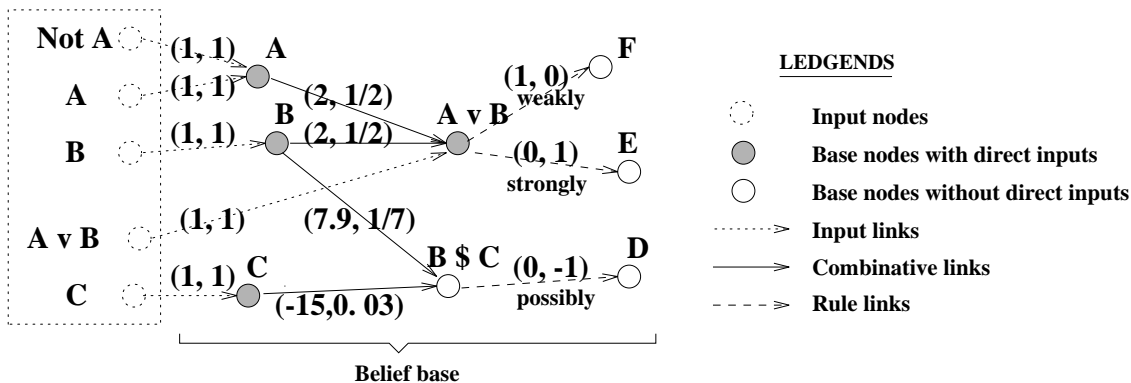
Abstract

This technical report presents a hybrid symbolic-neural network for representing and reasoning about commonsense beliefs called a Neural-Logic Belief Network (NLBN). Concepts are represented by nodes and relations amongst the concepts are represented by numerical directed links from one node to another. The network computation functions are similar to those of a neural network and classical logical relations such as AND, OR, etc., as well as human biased relations can be modelled by the network. The belief network is four-valued, that is a proposition is either being believed, its negation is believed, unknown or contradictory. Unlike the material implication in logic systems, defeasible IF-THEN rules in this formalism are unidirectional and well behaved; they only express the intended declarative semantics. A set of network operators similar to that of the belief revision operators have been defined and they allow knowledge to be dynamically updated in the knowledge system. This formalism has been implemented using Prolog on a unix machine. Its hybrid network architecture makes inferencing deterministic and efficient.

1 Introduction

The AGM [1; 10] approach to belief change has received wide attention in Artificial Intelligence. This logic is symbolic and is built on a set of postulates that capture rational belief change. It is therefore firmly in the logistic paradigm. The opposing paradigm of neural computation has its own wide following. In this approach the key notion is distributed and parallel activity in a network. Claims have been made for either paradigm that are presumably unachievable by the other. Recently [35] there have been attempts to exploit the relative strengths of each paradigm and combine them to improve performance. This technical report is a contribution toward this reconciliation.

Here, we study a hybrid symbolic-neural network architecture for belief representation called a *Neural-Logic Belief Network* (NLBN). It defines a set of network update operators that are analogs to the belief change operators in the AGM system. It is a finite acyclical directed network with nodes representing *propositions* and links (edges) representing *relations* amongst propositions. The figure below shows an example of a NLBN.



This belief network formalism utilizes basic neural network computation for feed-forward information propagation with symbolic degrees computation for *combinative-links* and a preferential selective computation for other links. Instead of a non-linear thresholding function from traditional classification neural networks (e.g. a sigmoid function for a back propagation neural net [7; 25; 18]), this architecture employs a linear thresholding function for transforming input values into outputs similar to that of a perceptron [27]. This linear transformation allows direct interpretation of the weights associated with links in the networks symbolically, which is the central theme of discussions in section 4. The following section gives a definition for a NLBN and its computation functions are presented in section 3. Section 5 discusses a set of network update operators for dynamic belief updates.

2 Definitions of a Propositional Belief Network

A *Neural-Logic Belief Network* (NLBN)¹ is a four-valued state based knowledge representation and reasoning system [21; 22].

Definition 1 A NLBN is a six tuple $\langle O, P, \Xi, \Lambda, \Sigma, \Pi \rangle$, where

- O is a finite set of n ($n \geq 0$) nodes $\{o_1, o_2, \dots, o_n\}$.

Each node o_i ($0 < i \leq n$) is associated with a node value which is a 2-tuple: (*proposition-value*, *degree-of-belief*). The *Proposition-value* is an ordered pair (t_{o_i}, f_{o_i}) where

$$(t_{o_i}, f_{o_i}) \in \{(1, 0), (0, 0), (0, 1)\}.$$

The *degree-of-belief* value, denoted as $\text{deg}[o_i]$, is an ordinal measurement which represents the belief certainty or degree of support for the proposition associated with the node. It is usually expressed in linguistic terms (it can also be numerical values) forming a total asymmetric order (TAO).² In a TAO, let 0 (zero, i.e. no degree at all) denote the weakest value and “true” denote the maximum degree. A typical example (this linguistic order will be used throughout this thesis) of TAO is:

0 < *possibly* < *weakly* < *probably* < *usually* < *normally* < *mostly* < *strongly* < *surely* < *definitely* < *true*.

There are two types of nodes: *input nodes* and *base nodes*. Input nodes are nodes without any incoming links and receive direct external inputs while all other nodes in a NLBN are base nodes.

- P is a finite set of m ($m \geq 0$) linguistically expressed *propositions* $\{p_1, p_2, \dots, p_m\}$, which can be simple concepts (basic concepts) such as “*bird*,” “*airplane*,” “John loves Mary,” or, compound concepts that group simple concepts into propositions with some relational connectives such as “*bird OR airplane*.” Propositions are expressed in their natural conceptual form which are mostly assertive or positive: for example, “*it’s a fine day*.” Their natural form can also be negative, such as “*unwanted child*.”
- Ξ is an association function. It is a mapping from nodes to propositions, $\Xi : O \rightarrow P$. The set of all propositions of base nodes, B , is called the *belief base*. The association function Ξ satisfies the following uniqueness of names requirement for base nodes:

$$\forall i, j \text{ where } 0 < i, j \leq n, \text{ if } i \neq j \text{ and } \Xi(o_i), \Xi(o_j) \in B \text{ then } \Xi(o_i) \neq \Xi(o_j).$$

The propositions of input nodes do not have to satisfy this criteria. Since each base node o_i can be uniquely named by its proposition, its node value is also denote by a $((t_{\Xi(o_i)}, f_{\Xi(o_i)}), \text{deg}[\Xi(o_i)])$.

¹It has been implemented using Prolog on a unix machine.

²TAO is an ordering relation ($<$) that is total with respect to a set of beliefs, transitive, asymmetric and non-reflexive.

- Λ is the acyclical finite network configuration which is a set of two groups of directed links in a NLBN: $\{ \textit{combinative-links}, \textit{alternative-links} \}$. A link connecting from node o_i to node o_j ($0 < i \leq n$, $0 < j \leq n$ and $i \neq j$) is denoted as $\lambda_{o_i o_j}^q$ where $\lambda_{o_i o_j}^q \in \Lambda$ and $q > 0$ is a link-identification-index for cases where there are more than one link from o_i to o_j . Each link $\lambda_{o_i o_j}^q$ is associated with a pair of weights: $(u_{o_i o_j}^q, v_{o_i o_j}^q)$ where $u_{o_i o_j}^q, v_{o_i o_j}^q \in \{\text{Real Numbers}\}$. For each alternative-link, there is an additional link-degree-of-belief value “deg[link]” which take the same degree values in TAO as the node degree-of-belief values.

There are two types of alternative-links: *input-links*—which are used to propagate external input values from input nodes to their respective base nodes with the same proposition, and *rule-links*—which serve as information channels between conditions of rules to the conclusions.

- Σ is a finite set of computation functions for network (node value) propagation. These functions take the inputs from all links to any node o_i ($0 < i \leq n$) and map them into the output node values: $(\textit{proposition-values}, \textit{degree-of-belief values})$. These values are then propagated out through the outward links of node o_i to other nodes.

More precisely, let $\Delta(o_i)$ be the set of nodes that have links feeding into o_i and for any node $o_j \in \Delta(o_i)$ with k links to o_i ($k \geq 1$ and k is finite), $\Upsilon(o_j)$ is the set of link values (link-weights and link degree(s)) of links $\lambda_{o_j o_i}^q$ for all $0 < q \leq k$. Then

$$\Sigma_{\Delta(o_i)} : ((t_{o_j}, f_{o_j}), \text{deg}[o_j]) \times \Upsilon(o_j) \rightarrow ((t', f'), \text{deg}')$$

and

$$\Sigma \text{ is } \bigcup_{o_i \in O} \Sigma_{\Delta(o_i)}.$$

- Π is the set of operations on the network for belief change. After an operation $\pi \in \Pi$ with respect to a belief b on a Neural-Logic Belief Network (O, P, A) , the network is changed to to a new state (O', P', A') .

$$\pi(b) : (O, P, A) \rightarrow (O', P', A') \quad \square$$

The following definition defines a *belief state* which is the set of beliefs currently held by a NLBN.

Definition 2 (Belief state) Let the first value of the proposition-value (t, f) represent positive information, the second value of the tuple represent negative information, an assignment of “1” to t or f represent there is positive or negative information respectively, and “0” represent no conclusive information. Given a NLBN with a belief base B , a *belief state* S is a collection of finite propositions in positive and/or negative form (the key) with their associated degree-of-belief values:

$\forall p$ if $p \notin B$ then $p \notin S$, and
 $\forall o_i, 0 < i \leq n$ where $\Xi(o_i) \in B$,
 if $(t_{\Xi(o_i)}, f_{\Xi(o_i)})=(1, 0)$ and $\deg[\Xi(o_i)]>0$ then $(\Xi(o_i), \deg[\Xi(o_i)]) \in S$;
 if $(t_{\Xi(o_i)}, f_{\Xi(o_i)})=(0, 1)$ and $\deg[\Xi(o_i)]>0$ then $(\neg\Xi(o_i), \deg[\Xi(o_i)]) \in S$;
 if $(t_{\Xi(o_i)}, f_{\Xi(o_i)})=(0, 0)$ and $\deg[\Xi(o_i)]=0$ then $(\Xi(o_i), deg) \notin S$ and
 $(\neg\Xi(o_i), deg) \notin S$;
 if $(t_{\Xi(o_i)}, f_{\Xi(o_i)})=(0, 0)$ and $\deg[\Xi(o_i)]>0$ then $(\Xi(o_i), \deg[\Xi(o_i)]) \in S$ and
 $(\neg\Xi(o_i), \deg[\Xi(o_i)]) \in S$;

where deg denotes any valid degree value and $\neg\Xi(o_i)$ denotes the negated form of the proposition $\Xi(o_i)$. The default node value of any proposition not represented by a NLBN is defined as:

$$\forall p, \text{ if } p \notin B \text{ then } (t_p, f_p)=(0, 0) \text{ and } \deg[p]=0. \quad \square$$

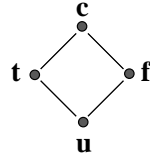
Since any proposition p in a belief set S is unique and p is the key for the entry $(p, \deg[p])$, $\deg[p]$ can be omitted from all references to membership relations for S . For example, $(p, \deg[p]) \in S$ is simplified into $p \in S$, and $(\neg q, \deg[\neg q]) \notin S$ is simplified into $\neg q \notin S$.

Definition 3 (Four truth values) Given a NLBN with a belief state S , the belief of any proposition p is mapped into four classes of truth values: $\{ \mathbf{t}$ (true), \mathbf{f} (false), \mathbf{u} (unknown), \mathbf{c} (contradictory) $\}$ as shown in the following table:

Conditions	Truth value	Description
$p \in S$	\mathbf{t}	p is believed in the belief state at $\deg[p]$.
$\neg p \in S$	\mathbf{f}	the negation of the proposition $\neg p$ is believed at $\deg[p]$.
$p \notin S, \neg p \notin S$	\mathbf{u}	p is unknown with respect to the current belief state.
$p \in S, \neg p \in S$	\mathbf{c}	p has contradictory beliefs at $\deg[p]$ in the belief state.

Node values of $((1, 0), 0)$ and $((0, 1), 0)$ do not have proper meanings and they are invalid inputs and node values. \square

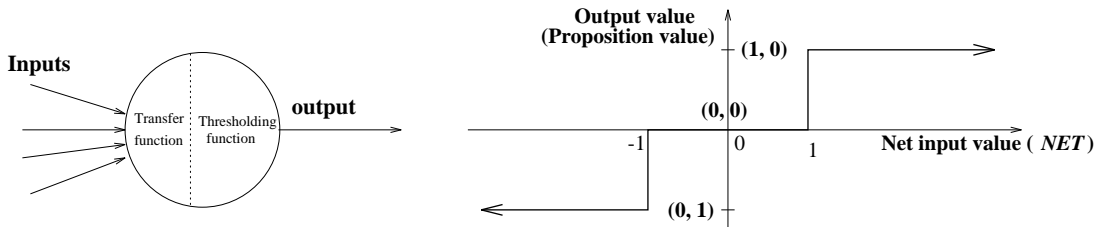
This four-valued mapping for a NLBN corresponds to the following lattice [12] and [13](A4 lattice):



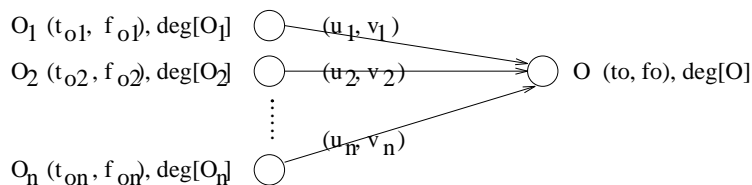
3 Computation Functions

This section presents the set of deterministic computation functions Σ for Neural-Logic Belief Networks. The combinative input computation functions for a NLBN are basic

neural-net feed-forward computations [32; 7] with symbolic degree-of-belief computations. Instead of the usual non-linear functions (e.g. sigmoid function $\frac{1}{1+e^{-x}}$), a NLBN employs a linear thresholding function for each node as shown below:



Definition 4 (Computation functions for combinative inputs) In the typical segment of a NLBN in the figure below, a typical node o has n inputs from n combinative-links.



There are two mutually exclusive sets \mathbf{E} , \mathbf{I} for node o . The set \mathbf{E} is a collection of all positive inputs (positive votes, excitatory inputs):

$$\mathbf{E} = \left\{ E_{o_i} \mid \begin{array}{l} E_{o_i} = |f_{o_i} \times v_i| \text{ where } f_{o_i} \times v_i < 0, \text{ otherwise} \\ E_{o_i} = t_{o_i} \times u_i \text{ where } t_{o_i} \times u_i \geq 0, \text{ for all } 0 < i \leq n \end{array} \right\}$$

The set \mathbf{I} is a set of all negative inputs (negative votes, inhibitory inputs):

$$\mathbf{I} = \left\{ I_{o_i} \mid \begin{array}{l} I_{o_i} = |t_{o_i} \times u_i| \text{ where } t_{o_i} \times u_i < 0, \text{ otherwise} \\ I_{o_i} = f_{o_i} \times v_i \text{ where } f_{o_i} \times v_i \geq 0, \text{ for all } 0 < i \leq n \end{array} \right\}$$

The *Transfer Function* that computes the net input value of node o is given by the difference of total positive input E and total negative input I :

$$NET = E - I$$

where the total positive input for node o is $E = \sum_{i=1}^n E_{o_i}$ and the total negative input for node o is $I = \sum_{i=1}^n I_{o_i}$

A *Thresholding Function* is given to determine the proposition-value for node o (see above figure):

$$(t_o, f_o) = \begin{cases} (1, 0) & \text{if } NET \geq 1 \\ (0, 1) & \text{if } NET \leq -1 \\ (0, 0) & \text{otherwise} \end{cases}$$

The degree-of-belief value for node o — $\text{deg}[o]$ —depends on the proposition-value (t_o, f_o) .

- If $(t_o, f_o) = (0, 0)$ then

$$\deg[o] = \max(\deg[o_i]) \quad \text{for all } 0 < i \leq n \text{ where } t_{o_i} = 0 \text{ and } f_{o_i} = 0$$

$\max(\deg[o_i])$ returns the highest degree-of-belief value according to the total asymmetric order (TAO) for the beliefs.

- If $(t_o, f_o) = (1, 0)$ then

$$\deg[o] = \begin{cases} \deg[o_k] & \text{if } E_{o_k} > 1, \deg[o_k] \geq \deg[o_i] \text{ for all } 0 < i \leq n, \\ & E_{o_k}, E_{o_i} \in \mathbf{E} \text{ and } i \neq k, & \text{otherwise} \\ \deg[o_j] & \text{if } \text{MIN}[(\sum_{j=1}^k E_{o_j}) \geq 1], \deg[o_j] \geq \deg[o_i], E_{o_j} \neq 0, E_{o_i} \neq 0, \\ & E_{o_j}, E_{o_i} \in \mathbf{E}, \text{ for all } 0 < i \leq n \text{ where } i \neq j. \end{cases}$$

where $\text{MIN}[(\sum_{j=1}^k E_{o_j}) \geq 1]$ is the minimum value of the sum $\sum_{j=1}^k E_{o_j}$ (with a distinct value of j) for all non-zero E_{o_j} that is greater or equal to 1 (the threshold) summed by descending order of the corresponding $\deg[o_j]$, in which $\deg[o_j] \geq \deg[o_{j+1}]$.

- If $(t_o, f_o) = (0, 1)$ then

$$\deg[o] = \begin{cases} \deg[o_k] & \text{if } I_{o_k} > 1, \deg[o_k] \geq \deg[o_i] \text{ for all } 0 < i \leq n, \\ & I_{o_k}, I_{o_i} \in \mathbf{I} \text{ and } i \neq k, & \text{otherwise} \\ \deg[o_j] & \text{if } \text{MIN}[(\sum_{j=1}^k I_{o_j}) \geq 1], \deg[o_j] \geq \deg[o_i], I_{o_j} \neq 0, I_{o_i} \neq 0, \\ & I_{o_j}, I_{o_i} \in \mathbf{I}, \text{ for all } 0 < i \leq n \text{ where } i \neq j. \end{cases}$$

where $\text{MIN}[(\sum_{j=1}^k I_{o_j}) \geq 1]$ is defined in the similar way above. \square

The above degree-of-belief computation can be visualized diagrammatically. For example, suppose node o has eight inputs from o_1, \dots, o_8 . Three of them, o_1, o_4 and o_7 , give positive inputs in descending order of degree-of-belief values (i.e. $\deg[o_1] > \deg[o_4] > \deg[o_7]$), another three nodes, o_6, o_8 and o_2 , give negative inputs in descending order of degree-of-belief values, o_3 and o_5 have a proposition-value of $(0, 0)$ with $\deg[o_3] > \deg[o_5]$ and they give zero inputs; this is illustrated graphically in figure 1.

If the sum of all positive inputs E is greater than the sum of all negative inputs I and their difference ($NET = E - I$) is greater than the threshold of 1, then measuring from the topmost input with the threshold of 1, the degree-of-belief of node o is the degree of the input it intersects ($\deg[o] = \deg[o_4]$ in this example). If the sum of negative inputs exceeds that of the positive inputs, a similar visualization can be drawn. If NET is less than the threshold of 1, then the maximum degree of the inputs with a proposition-value of $(0, 0)$ is chosen as the degree value for node o . In this example it is $\deg[o_3]$ as it is the highest degree with a proposition-value of $(0, 0)$ on the horizontal degree axis.

Node input values transmitted by alternative-links are computed individually with the link degrees acting as the capping constraints to input degrees. Each of the computed value is a possible candidate for the final node value where the one with the strongest degree-of-belief value is selected as the final value.

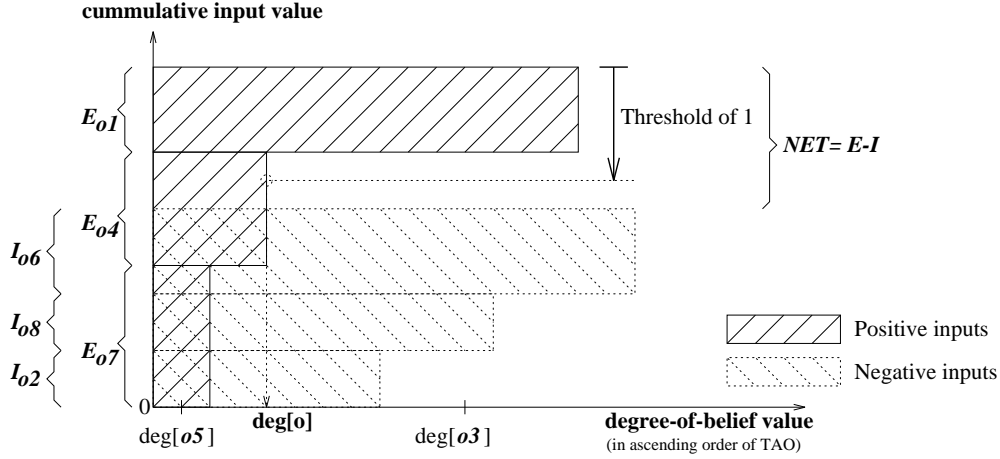


Figure 1: Computation of degree-of-belief value.

Definition 5 (Computation function for Alternative-inputs) The input node value received by a base node o (whose node value is $((t_o, f_o), \text{deg}[o])$) from another base node p with a proposition-value (t_p, f_p) via an alternative-link with link-weights of (u_p, v_p) and the link degree-of-belief value of $\text{deg}[\text{link}]$ is given by:

$$(t_o, f_o) = \begin{cases} (1, 0) & \text{if } (t_p \times u_p - f_p \times v_p) \geq 1 \\ (0, 1) & \text{if } (t_p \times u_p - f_p \times v_p) \leq -1 \\ (0, 0) & \text{otherwise} \end{cases}$$

$$\text{deg}[o] = \begin{cases} \text{deg}[p] & \text{if } \text{deg}[p] \leq \text{deg}[\text{link}] \text{ and } (t_o, f_o) \neq (0, 0) \\ \text{deg}[\text{link}] & \text{if } \text{deg}[p] > \text{deg}[\text{link}] \text{ and } (t_o, f_o) \neq (0, 0) \\ 0 & \text{otherwise} \end{cases}$$

Note that in the above computation functions, if the resulting proposition-value is $(0,0)$ and the degree value is “0” then it is a **u** otherwise it is a **c**. When there are n inputs from n alternative-links to a base node o , the network values from each input $((t_i, f_i), \text{deg}[i]), 0 < i \leq n$ is first computed according to the above computation functions, then the subsumptive *selection function* below is used to pick up the strongest input among these alternatives as the computed node value.

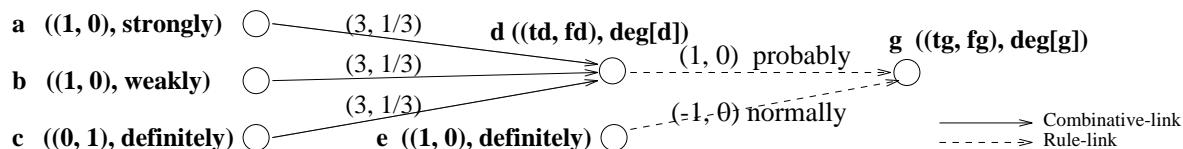
$$(t_o, f_o) = \begin{cases} (t_k, f_k) & \text{if } \text{deg}[k] > \text{deg}[i], 0 < k \leq n, k \neq i, \text{ for all } 0 < i \leq n \\ (t_j, f_j) & \text{if } (t_j, f_j) = (t_k, f_k), \text{deg}[j] = \text{deg}[k], \text{ for some } j, k, 0 < j, k \leq n, \\ & j \neq k, \text{deg}[j] > \text{deg}[i], j \neq i, k \neq i, \text{ for all } 0 \leq i \leq n \\ (0, 0) & \text{otherwise} \end{cases}$$

$$\text{deg}[o] = \begin{cases} \text{deg}[k] & \text{if } \text{deg}[k] \geq \text{deg}[i], 0 < k \leq n, k \neq i, \text{ for all } 0 < i \leq n \\ 0 & \text{otherwise} \end{cases}$$

□

If the inputs to a base node o are given by combinative-links together with alternative links, the input values from all combinative-links shall first be processed to yield at a combined node value (for node o) using computation functions in definition 4. This computed node value is treated as one of the alternative input node values by the selection function in definition 5 to arrive at a final node value for node o .

Example 3.1 The figure below shows an example segment of a NLBN.



Using the computation functions in definition 4, node value of node d is computed as follow:

$$E = E_a + E_b + E_c = 1 \times 3 + 1 \times 3 + 0 \times 3 = 6$$

$$I = I_a + I_b + I_c = 0 \times \frac{1}{3} + 0 \times \frac{1}{3} + 1 \times \frac{1}{3} = \frac{1}{3}$$

$$NET = E - I = 6 - \frac{1}{3} = 5\frac{2}{3} > 1$$

Therefore, the proposition-value of node d , $(t_d, f_d) = (1, 0)$.

Since $deg[a] > deg[b]$ (strongly $>$ weakly), and $E_a = 3 > 1$, $deg[d] = deg[a] = \text{strongly}$.

The node value of node g is computed using computation functions in definition 5. There are two rule-links: one from node d and the other one from node e . Input node value from node d is $((1, 0), \text{probably})$ since $deg[d] > deg[\text{link}]$ (strongly $>$ probably) and

$$(t_d \times u_d - f_d \times v_d) = (1 \times 1) - (0 \times 0) = 1$$

and input node value from node e is $((0, 1), \text{normally})$ since $deg[e] > deg[\text{link}]$ (definitely $>$ normally) and

$$(t_e \times u_e - f_e \times v_e) = (-1 \times 1) - (0 \times 0) = -1.$$

Using the selection function in definition 5, the stronger input node value from node e (normally $>$ probably) becomes the node value for node g , i.e.

$$(t_g, f_g) = (0, 1) \text{ and } deg[g] = \text{normally.} \quad \square$$

With the set of computation functions Π , a NLBN can be propagated in two modes:

- To compute the node value of a proposition p , a *backward chaining* mode similar to that of a Prolog query [17; 34] is used. Starting from base node p , its node value is computed in a nested way from all its incoming links until the relevant input nodes are reached. If p does not exist in the NLBN then a default node value of $((0, 0), 0)$ is assigned.

- A *forward propagation* mode can be used to propagate network values according to the direction of the directed links starting from all input nodes to all base nodes until all base nodes without out-going links are reached. This is similar to a feed-forward computation in a typical neural network [7; 32].

Theorem 6 (Convergence of NLBN) Given an acyclic NLBN, the network computation functions in definitions 4 and 5 provides a stable belief state. \square

Proof: As the computation functions in definitions 4 and 5 are deterministic, there are finite number of nodes in a NLBN (definition 2) and there is no closed loop in the network:

- In a forward propagation mode, all node values are propagated from the input nodes forward to their base node and the process continues from the base nodes to other based nodes connected by directed links. As there are only finite number of nodes, the last inference will reach a node at the maximum depth in this one-pass computation, and the network settles at a set of node values. If the same propagation is repeated, with exactly the same inputs at the input node, the same network structure and the same computation functions, it can be easily seen that it will produce the same belief state. Therefore the NLBN converges and settles at a stable belief state.
- In a backward chaining mode, starting from a proposition, the network inference back-chains to the relevant input nodes where no more backward chaining is possible, then the inferences for this selected tree of network is propagated to the proposition queried in the same way as the forward propagation computations. Without any change in the NLBN, any repetition of the query with the same network segment and inputs will result in the same belief state. The NLBN therefore converges with a stable belief state. \diamond

4 Representing Relations

This section shows how logical relations [33] and other relations are represented by different combinations of directed links in a NLBN.

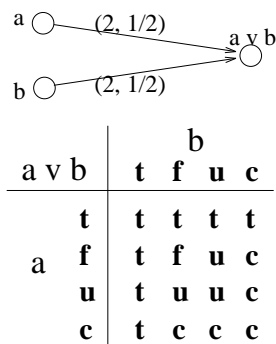
4.1 Relations Represented by Combinative-links

The four-valued logical relations modeled by the combinative-links of a NLBN are direct extensions of Kleene's strong three-valued logic [16]. If truth value classes **c** (*contradictory*) and **u** (*unknown*) are considered as a single sub-class with a proposition-value of (0, 0), then the logical relations represented by the belief networks are identical to Kleene's strong three-valued logic.

Logical relation OR (\vee):

A two-input logical OR, for example $a \vee b$, can be represented by a NLBN with two combinative-links from the base nodes of the disjuncts a and b to their disjunction base node $a \vee b$. Each of these links is assigned with a tuple of link weights $(2, \frac{1}{2})$ as shown in the figure below. Permutations of all combinations of proposition-values for node a and b give the left truth table below where functions $\max(\dots)$ and $\min(\dots)$ return the maximum degree value and minimum degree value of the arguments respectively according to the total order TAO defined for beliefs in the belief state. With definition 2 that maps node values to four truth values, the left truth table can be more generally represented by the right table.

a	b	$a \vee b$	$deg[a \vee b]$
(1, 0)	(1, 0)	(1, 0)	$\max(deg[a], deg[b])$
(1, 0)	(0, 0)	(1, 0)	$deg[a]$
(1, 0)	(0, 1)	(1, 0)	$deg[a]$
(0, 0)	(1, 0)	(1, 0)	$deg[b]$
(0, 0)	(0, 0)	(0, 0)	$\max(deg[a], deg[b])$
(0, 0)	(0, 1)	(0, 0)	$deg[a]$
(0, 1)	(1, 0)	(1, 0)	$deg[b]$
(0, 1)	(0, 0)	(0, 0)	$deg[b]$
(0, 1)	(0, 1)	(0, 1)	$\min(deg[a], deg[b])$

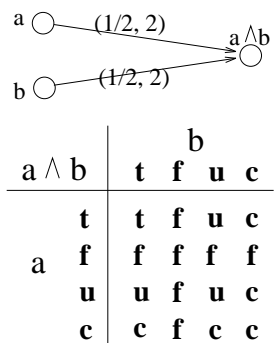


For an n -input OR, each combinative-link from the disjunct (sub-expression) to the disjunction (logical expression) is assigned with a link weight of $(2, \frac{1}{n})$.

Logical relation AND (\wedge):

In a NLBN, the network constructions for logical AND relations are symmetrical to that of OR relations. For example, a two-input AND $a \wedge b$ is represented in the similar way as a two-input OR relation except that the tuple of link-weights associated with the combinative-links are in reversed order as shown below.

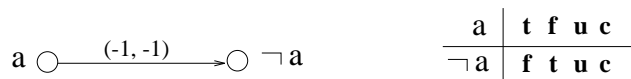
a	b	$a \wedge b$	$deg[a \wedge b]$
(1, 0)	(1, 0)	(1, 0)	$\min(deg[a], deg[b])$
(1, 0)	(0, 0)	(0, 0)	$deg[b]$
(1, 0)	(0, 1)	(0, 1)	$deg[b]$
(0, 0)	(1, 0)	(0, 0)	$deg[a]$
(0, 0)	(0, 0)	(0, 0)	$\max(deg[a], deg[b])$
(0, 0)	(0, 1)	(0, 1)	$deg[b]$
(0, 1)	(1, 0)	(0, 1)	$deg[a]$
(0, 1)	(0, 0)	(0, 1)	$deg[a]$
(0, 1)	(0, 1)	(0, 1)	$\max(deg[a], deg[b])$



Similar to logical OR relations, an n -input AND relation will have combinative link-weights of $(\frac{1}{n}, 2)$.

Logical negation NOT (\neg):

To negate a proposition in a NLBN, we are effectively changing the proposition-value of a node either from (1, 0) to (0, 1) or from (0, 1) to (1, 0), and when the value is (0, 0) its negation is also (0, 0). Its degree-of-belief value will remain unchanged. For example, a base node a can be negated to $\neg a$ with the following network construction with a tuple of combinative link-weights of (-1, -1).



Note that using this representation, a $\neg\neg a$ has the exact truth values as a and they are considered as equivalent in a NLBN. The definition for negation is the same as Belnap's four-valued scheme v [13; 3].

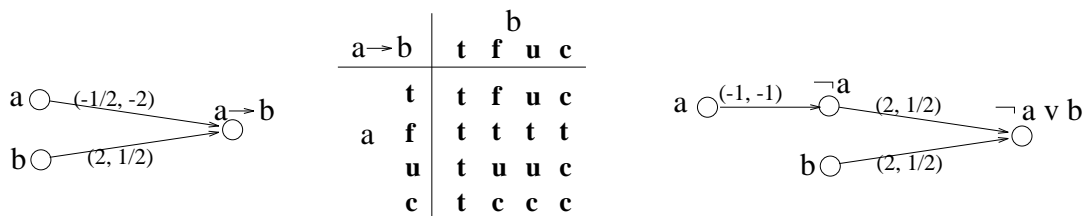
Definition 7 (Logical Relations) The *basic logical connectives* for a NLBN are AND, OR and NOT: $\{\wedge, \vee, \neg\}$ and all logical relations are expressed in Conjunctive Normal Form (CNF) [2; 33]:

$$\bigwedge_{i=1}^k \left(\bigvee_{j=1}^l \alpha_{ij} \right)$$

where k, l are the numbers of conjuncts and disjuncts in the expression respectively, and α_{ij} is a proposition p or the negated form of the proposition $\neg p$. \square

Material Implication (\rightarrow):

A material implication logical relation, for example, $a \rightarrow b$, can be represented by the left NLBN segment below:



It can be easily verified that both the left network segment and the right network segment, which represents relation $\neg a \vee b$, give the same truth table above. This corresponds to the equivalent property of the two logical expressions $(a \rightarrow b) \equiv (\neg a \vee b)$ in classical logic.

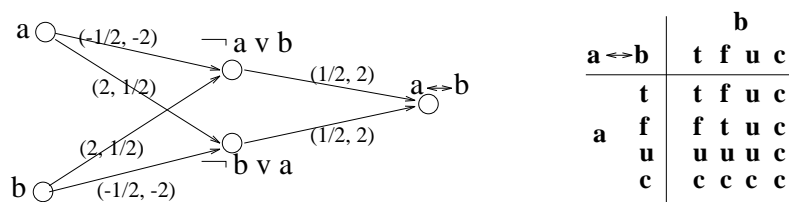
To preserve uniqueness of concepts in a belief state S , a proposition will only be expressed exclusively in its natural concept form (either positive form or negative form) and nodes with a negated proposition do not actually exist in a NLBN. The construction

for a negation (negated node) only serves as an intermediary step for constructing other more complex logical relations such as the material implication relation above.

The negated intermediary nodes can be eliminated by the *Pre-not* or *Post-not* rules [31] with identical network outputs. In the above example for implication relation, a *Pre-not* rule is applied: the two sets of link-weights immediately before and after the negated node are multiplied, $(-1, -1) \times (t, f) = (-t, -f)$, the resulting link-weights are swapped around, that is $(-f, -t)$, and applied to the new combinative-link that replaces the negated node and its incoming and outgoing links. A *Post-not* rule is the same as the *Pre-not* procedures except no swapping of resulting link-weights is carried out. These rules are applicable to nested links and multiple inputs.

Logical relation IFF (\leftrightarrow):

The logical relation IFF (if-and-only-if) can be expressed as a conjunction between two reversal material implications and this can be further reduced to the three basic logical connectives for a NLBN. For example, $a \leftrightarrow b$ can be represented as $(\neg a \vee b) \wedge (\neg b \vee a)$ as follows:



From the diagonal values of the four-valued truth table above, since truth values of the same class do not always give class **t**, and also that $a \vee \neg a$ does not always give a **t** (which can be easily verified using the truth table for an \vee relation), it is in line with the strong Kleene logic that with a standard interpretation, there is no tautology for the four-valued scheme represented by a NLBN [30].

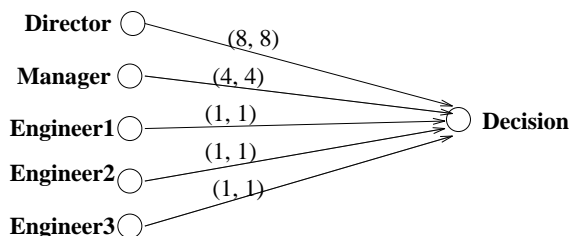
Representation of other common logical relations such as **Exclusive OR (XOR)** and **Exclusive And (XAND)** are given in [22].

Note that the direct extensions of Kleene's strong three-valued logical relations by a NLBN is different from Belnap's four-valued scheme v [13; 3] which is also made along the strong Kleene lines. The basic difference is that the v scheme assigns class **f** to the AND relation for a **u** and a **c** while a NLBN assigns it with **c**. Based on the network nature of logical relation constructions, a NLBN treats a logical relation as competitive and co-operative inputs from its sub-expressions. This is in tune with the intuitionist's constructive philosophy [14; 9] that if the final node value of the logical relation is a positive outcome then there must be positive input(s) to support it; similarly, a negative outcome require negative input(s) to support it. As contradictory class **c** contributes to neither a positive nor a negative outcome, it is intuitive that a **c** instead of a **f** should result from an AND relation with a **u** and a **c** in a NLBN.

Owing to the numerical combinative nature of the directed links, a NLBN provides a natural way of expressing relations other than standard logical relations.

Human Biased Relations:

Human biased relations can be represented readily by a NLBN. For example, let us consider a decision making relation of an authoritative organization consisting of a director, a manager and three engineers represented by a simple NLBN segment with five combinative-links below:



This network segment says that the director has absolute authority over everybody, the manager’s decision is above the engineers if the director does not make any decision, and the engineers can make decision based on majority votes if the director and the manager do not interfere. This straight forward network representation cannot be directly translated into a classical logical expression. For those relations that can be translated, for example, the majority decision making of the three engineers in the above example, the logical expressions are usually lengthy, inefficient with respect to both representation and reasoning, and most important of all, they fail to capture the intuitive semantics. Some commonly used human biased relations such as “**at-most m ,**” “**at-least m ,**” “**Majority-win,**” . . . , etc., can be more naturally and efficiently represented using a NLBN [22; 4; 5] than classical logical expressions.

4.2 Relations Represented by Alternative-links

There are two types of alternative-links: input-links and rule-links. External inputs are propagated onto their corresponding base nodes using input-links.

External Input:

Each external input for a proposition a with a truth value of \mathbf{t} , \mathbf{f} or \mathbf{u} , it is represented by an input node a and a node value $((t_a, f_a), \text{deg}[a])$. An input with a truth value of \mathbf{u} is equivalent to two opposite inputs: a \mathbf{t} and a \mathbf{f} , with the same degree-of-belief value. The input information from each input node is propagated via an input-link to the base node of the same proposition. This is a one-to-one connection and every input-link is associated with a link-weight of $(1, 1)$ and a link degree of “*true*” (the maximum degree in TAO) as follows:



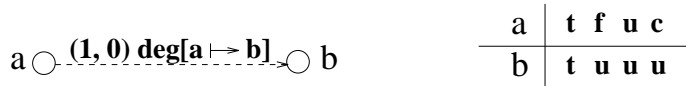
The truth table above shows that without other input, the base node will have identical node value as its intended input from the input node(s). It is possible for any base node a to have more than one external input. In such cases the selection function in definition 5 is used to determine the node value of a .

IF-THEN Rule (\mapsto):

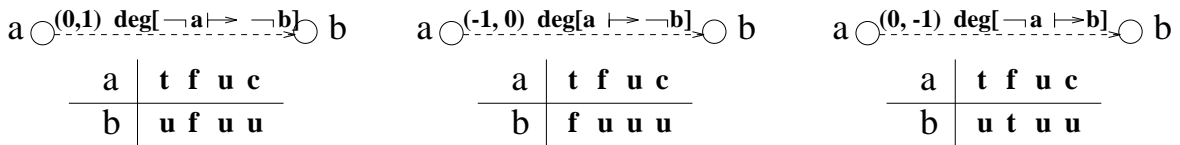
In human reasoning, rules of the following form are common:

- IF *you are a good boy today* THEN *I will definitely take you to the toy world*;
- IF *the sky is covered with thick dark clouds* THEN *normally it will rain*;
- IF *I am wealthy* THEN *I'll probably be happy*.

In a NLBN, this type of IF-THEN rules are considered to be semantically weaker than conditionals expressed by material implications (\rightarrow). A rule “IF a THEN b (certainty of the rule),” written as $a \mapsto b$ ($\text{deg}[a \mapsto b]$), means if we believe a then we can derive b with respect to the rule degree-of-belief value. In this rule a is the *condition* and b is the *conclusion*. Unlike a material implication $a \rightarrow b$, the part $\neg b \mapsto \neg a$ is not automatically assumed in a rule $a \mapsto b$ in a NLBN. This corresponds to the more natural and intuitive use of IF-THEN rules in commonsense reasoning where attempts were made to correct the “misbehaved” material implication for expressing commonsense rules in the nonmonotonic reasoning community [11; 23]. The relation $a \mapsto b$ ($\text{deg}[a \mapsto b]$) is represented in a NLBN by a rule-link with link-weights of $(1, 0)$ as follows:



and the default degree value for any IF-THEN rule is the maximum degree in the TAO — “*true*.” The condition and conclusion nodes need not be simple propositions. They can be logical expressions or other more complex relations. An IF-THEN rule in the form of $\neg a \mapsto \neg b$ will have a link weight of $(0, 1)$, and for $a \mapsto \neg b$ (or $\neg a \mapsto b$), the link weight is $(-1, 0)$ (correspondingly $(0, -1)$). These rule-links are illustrated below:



It is assumed that no proposition should appear as whole or in part in the condition as well as the conclusion of an IF-THEN rule such as $a \mapsto a$, $a \mapsto \neg a$, $a \mapsto (a \wedge b \wedge c)$, $(a \vee b) \mapsto a$, $(a \wedge b) \mapsto (a \vee c)$, ..., etc.

4.3 The Subsumptive Characteristics

The computation functions in definitions 4 and 5 give a NLBN the following characteristic:

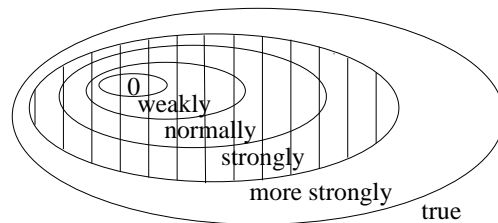
Theorem 8 (Belief Subsumption) Given a proposition p in a NLBN. If there are n ($n > 0$) input beliefs p_1, \dots, p_n to p via a set of combinative-links or any other types of alternative-links where beliefs p_1, \dots, p_n are about the same proposition p but may be with different truth values and degree-of-belief values, and their degree-of-belief values form an order: $\text{deg}[p_1] \leq \text{deg}[p_2] \leq \dots < \text{deg}[p_n]$, then the unique strongest belief p_n *subsumes* all weaker beliefs and the final belief for p takes the input of p_n with $\text{deg}[p_n]$ in the belief state.

If there are more than one input belief with the same strongest degree-of-belief value and the same truth value then p will take any value from these identical inputs.

If there are more than one input beliefs having the same strongest degree-of-belief value (> 0) but are of different truth values, then p will be assigned to a *contradictory* state with the strongest degree-of-belief value.

Otherwise the belief has no positive or negative input and it is assigned with a truth value of *unknown*. \square

This subsumptive behavior of beliefs can be represented by a *system of spheres* stratified by the degree-of-belief values in TAO where weaker spheres are enclosed by stronger spheres [20]. The inner most sphere contains *unknown* beliefs with “0” as their degree-of-beliefs and they will always be subsumed by other stronger beliefs. The diagram below shows that beliefs in the “strongly” sphere subsumes beliefs in weaker spheres:



Under the assumption that there is only one TAO in a belief system, which implies that all beliefs can be freely combined by some connectives to form complex expressions, the subsumptive principle can be applied to different propositions. Given any two propositions a and b where $\text{deg}[b] > \text{deg}[a]$, the interpretation of the logical AND relation $a \wedge b$ is that beliefs in both a and b must exist at the same instance. This means that the degree-of-belief value for $a \wedge b$ depends on the co-existence of the area subsumed by a and that subsumed by b which gives us the intersection of the two spheres where

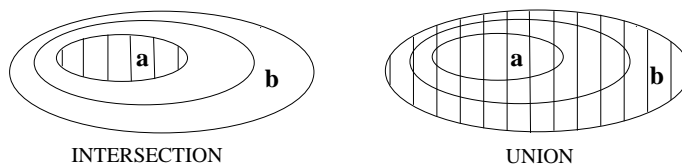
$$\begin{aligned} \text{deg}[a \wedge b] &= \text{deg}[\text{Sphere_of}(a) \cap \text{Sphere_of}(b)] \\ &= \min(\text{deg}[a], \text{deg}[b]) = \text{deg}[a] \end{aligned}$$

as shown in the left figure below.

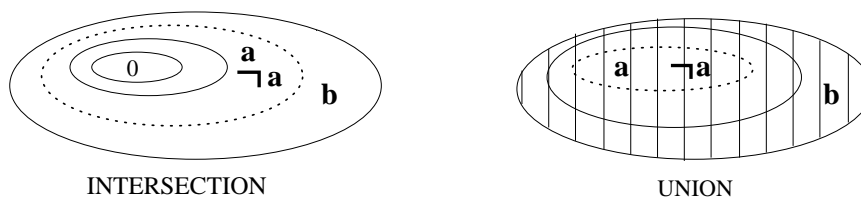
The interpretation of a logical OR relation $a \vee b$ under the subsumptive principle takes the union of the two respective spheres. This represents the area where either a is believed (the area subsumed by a) or b is believed (the area subsumed by b) (or both). This gives us

$$\begin{aligned} \text{deg}[a \vee b] &= \text{deg}[\text{Sphere_of}(a) \cup \text{Sphere_of}(b)] \\ &= \max(\text{deg}[a], \text{deg}[b]) = \text{deg}[b] \end{aligned}$$

as shown in the right figure below:



As characterized by theorem 8, if a proposition a is in a contradictory state, it is interpreted as having a belief a together with another belief $\neg a$ at the same current degree-of-belief value. Diagrammatically, we have a and $\neg a$ at the same sphere where none of them can subsume each other and this gives us a contradictory sphere denoted by the spheres in dotted lines in the figure below. A contradictory sphere cannot be properly intersected with other spheres. In such cases, they give a null result, which can be interpreted as having a zero degree-of-belief value. However, a contradictory sphere can be subsumed by another stronger sphere in a union operation. If a union or intersection operation results in a weakest sphere of zero degree which indicates there is insufficient information to arrive at a definite belief, the strongest contradictory sphere(s) then subsumes this null result. In the following figure, the result of intersecting spheres of a and b where a is at a contradictory state is a zero degree while a union makes sphere of b subsume the weaker contradictory sphere of a .



The result of intersecting a contradictory sphere with a non-zero sphere results in a null set, i.e. a degree of zero.

A contradictory sphere can be subsumed by stronger sphere(s) in union operations.

This gives us the following characteristic for logical AND and OR relations in a NLBN:

Theorem 9 (Subsumptive AND and OR)³ For an n -ary AND expression $a_1 \wedge a_2 \wedge \dots \wedge a_n$ in a NLBN, where for all $0 < i \leq n$, a_i is not a contradictory belief,

³In this four-valued mode, there is no tautology [30]. For example, $a \vee \neg a$ may have “*unknown*” or “*contradictory*” as its truth value.

the degree-of-belief value for the conjunction is the minimum degree value of all the conjuncts:

$$\text{deg}[a_1 \wedge a_2 \wedge \dots \wedge a_n] = \min(\text{deg}[a_1], \dots, \text{deg}[a_n]).$$

For an n -nary OR expression $a_1 \vee a_2 \vee \dots \vee a_n$, the degree-of-belief value for the disjunction is the maximum degree value of all the disjuncts when for any $0 < i \leq n$, if a_i has the maximum degree value then it must not be a contradictory belief unless all other disjuncts a_j ($0 < j \leq n, j \neq i$) are either negative beliefs, unknown or contradictory:

$$\text{deg}[a_1 \vee a_2 \vee \dots \vee a_n] = \max(\text{deg}[a_1], \dots, \text{deg}[a_n]). \quad \square$$

Proof: These can be directly derived from definition 4 and the network construction for AND and OR relations in a NLBN in section 4.1. \diamond

5 Operations for Dynamic Belief Change

One of the most important parts of a belief system is its capability to transform belief states dynamically. A NLBN is a state-based belief representation system and it has a set of six types of network update operators with deterministic update operations:

$$\Pi = \{ \textit{Add}, \textit{Remove}, \textit{Forget}, \textit{Revise}, \textit{Update}, \textit{Not-Conclude} \}.$$

A new belief is incorporated into the network via an *Add* operation and a replacement of existing belief without changing the network configuration is carried out by an *Update* operation. A *Remove* operation deletes a particular view of an existing belief without removing its conceptual representation while a *Forget* operation removes all views to the belief and expel it from the belief network whenever possible. A *Revise* operation removes all current views to a belief and assert the new belief to it. The *Not-Conclude* operator allows us temporarily not to conclude a positive or negative view about any belief which is necessary for reasoning about alternatives without destroying existing beliefs. We believe that these are the fundamental operators necessary for proper belief change operations. Additional operators may be defined when the needs arise.

Performing an update operation for π with respect to a belief b , where $\pi \in \Pi$, on a NLBN with a belief state S is denoted as $\pi(b, S)$ and the resulting belief state is denoted as S_b^π . There are four types of beliefs accepted by a NLBN:

- (1) input propositions,
- (2) logical expressions,
- (3) arbitrary human biased relations, and
- (4) IF-THEN rules,

and not all beliefs are applicable (meaningfully) to a particular type of update operation. Detailed procedures of each operations are defined below.

5.1 Add a Belief

Adding a belief b to a belief state S , written as $Add(b, S)$, is the most fundamental operation for a NLBN to take in a belief. This operation represents simple inclusion of new beliefs into the epistemic state. Whether b will be believed in S_b^{Add} depends on the current beliefs in S . The procedures for an Add operation are sub-divided into the following four categories depending on the type of belief to be added:

1. Add an input (Add_i)

When belief b to be added is an external input with a node value of $((t_b, f_b), \deg[b])$, the operation $Add_i(b((t_b, f_b), \deg[b]), S)$ is sequentially as follows:

- i. Create an input node and assign proposition b with the associated node value $((t_b, f_b), \deg[b])$ to this node. In practice a source field is created for each input node to differentiate inputs.
- ii. If base node b exists, create an input-link with link-weights of $(1, 1)$ and a link degree of “*true*” (the maximum degree in TAO) from input node b to base node b .
- iii. Otherwise create a base node b and perform input-link creation in the same way above.

Valid node values to be added for an Add_i operation are: $((1, 0), \deg[b] > 0)$, $((0, 1), \deg[b] > 0)$ and $((0, 0), \deg[b])$. If the node value of an input is $((0, 0), \deg[b] > 0)$ (contradictory input), then two order-independent Add_i operations are performed: one with a node value $((1, 0), \deg[b])$ and the other $((0, 1), \deg[b])$.

2. Add a logical expression (Add_{le})

A NLBN only accepts any logical expression b in CNF (definition 7). This restricts an $Add_{le}(b, S)$ operation to the following three types.

A. Logical expression b is a simple proposition “ a ”:

Perform an $Add_i(b((t_b, f_b), \deg[b]), S)$ operation.

B. Logical expression b is a pure disjunction “ $\alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_j$ ”:

If the proposition-value of b is $(1, 0)$ or $(0, 0)$:

- i. Perform an $Add_i(b((t_b, f_b), \deg[b]), S)$.
- ii. Perform an $Add_i(\alpha_i((0, 0), 0), S)$ for every α_i where $0 < i \leq j$.
- iii. For each base node α_i , link a combinative-link with link-weights of $(2, \frac{1}{j})$ to base node b .

As we do not know which disjunct is positively believed, step (ii) assigns a \mathbf{u} to all disjuncts. Since an input node with the corresponding node value is created for the disjunction in step (i), the resulting disjunction will have its proper node value.

If the proposition-value of b is $(0, 1)$ then it is equivalent to “ $\neg\alpha_1 \wedge \neg\alpha_2 \wedge \dots \wedge \neg\alpha_j$ ” (note that $\neg\neg\alpha$ is equivalent to α). In such cases, it is assumed that all conjuncts $\neg\alpha_i$ ($0 < i \leq j$) are believed at $\text{deg}[b]$ and $Add_i(\alpha_i((0, 1), \text{deg}[b]), S)$ instead of $Add_i(\alpha_i((0, 0), 0), S)$ are performed in the above procedures.

C. Logical expression b is a conjunction “ $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_k$ ”:

If the proposition-value of b is $(0, 1)$ or $(0, 0)$:

- i. Perform an $Add_i(b((t_b, f_b), \text{deg}[b]), S)$.
- ii. Perform an $Add_le(\alpha_i((0, 0), 0), S)$ for every α_i where $0 < i \leq k$ and $((0, 0), 0)$ is the associated node value.
- iii. For each base node α_i , link a combinative-link with link-weights of $(\frac{1}{j}, 2)$ to base node b .

If the proposition-value of b is $(1, 0)$ it is assumed that all conjuncts are believed at $\text{deg}[b]$ and $Add_le(\alpha_i((1, 0), \text{deg}[b]), S)$ instead of $Add_le(\alpha_i((0, 0), 0), S)$ operations are performed in the above procedures.

3. Add an arbitrary human biased relation (Add_a)

An typical arbitrary human biased relation b consists of l inputs: $\{a_1, a_2, \dots, a_l\}$, and each a_i has a link weight of (u_i, v_i) .

- i. Perform an $Add_i(b((0, 0), 0), S)$.
- ii. Perform an $Add_i(a_i((0, 0), 0), S)$ for every a_i where $0 < i \leq l$.
- iii. For each base node a_i , link a combinative-link with link-weights of (u_i, v_i) to base node b .

4. Add an IF-THEN rule (Add_r)

An IF-THEN rule “ $a \mapsto b$ ” consists of two parts: a condition a and a conclusion b , with a degree of $\text{deg}[a \mapsto b]$.

- i. Perform an $Add(a, S)$ and an $Add(b, S)$ with a node value of $((0, 0), 0)$ each.
- ii. Link a rule-link from base node a to base node b with link-weights of $(1, 0)$ and a link degree of $\text{deg}[a \mapsto b]$.

For rules of the forms IF-THEN-Not, IF-Not-THEN and IF-Not-THEN-Not, rule link-weights of $(-1, 0)$, $(0, -1)$ and $(0, 1)$ are used respectively.

5.2 Remove a Belief

When a positive- or negative- view of a proposition b in a NLBN with a belief state S is not wanted, it can be expelled by a $Remove(b(t_b, f_b), S)$ operation where $(t_b, f_b)=(1, 0)$ represents the positive view of b is removed and $(t_b, f_b)=(0, 1)$ represents the negative view of b is removed. If b does not exist in a NLBN then there is no change to the network. There are two categories:

1. Remove an input ($Remove_i$)

A $Remove_i(b((t_b, f_b), deg[b]), S)$ operation removes either a positive input (input nodes b with a node value of $((1, 0), deg[b])$) or a negative input (input nodes b with a node value of $((0, 1), deg[b])$) to b . A $Remove_i(b((0, 0), deg[b]), S)$ has no effect on the belief network.

- i. Delete the input node b with a node value of $((t_b, f_b), deg[b])$ from the NLBN.
- ii. Delete its input-link to base node b from the NLBN.
- iii. If the input node does not exist, then there is no change to the NLBN.

2. Remove a view from a proposition ($Remove_p$)

To remove a view from a proposition in a NLBN is to discard all current beliefs contributing to this view. A contributing belief can be defined as it is either a non-zero positive network input for a positive final node value of $(1, 0)$ or a negative network input for a negative final node value of $(0, 1)$ from the computation functions for a NLBN (definitions 4 and 5. Removal of contributing beliefs always starts from the weakest view upward. This is to prevent unnecessary removal of beliefs in a NLBN.

A $Remove_p(b(t_b, f_b), S)$ operation removes either all positive views or negative views to the proposition b .

- i. Perform $Remove_i(b((t_b, f_b), deg[b]), S)$ operations for all values of $deg[b]$.
- ii. If the input proposition-value from all its sub-expressions according to computation functions in definition 4 is the same as (t_b, f_b) , then for each sub-expression of b contributing to (t_b, f_b) , rank them in ascending order according to TAO and perform a $Remove_i$ to these sub-expressions starting from the weakest degree upward. After each round of $Remove_i$ for the sub-expression(s) in a particular level of TAO, use the computation function in definition 4 to check whether the combined input still contribute to the view to be removed. If it does then continue with the $Remove_i$ operations for the next higher up sub-expression(s) in the order. If it no longer contributes to the view, stop the process.
- iii. If there are m alternative inputs $((t_i, f_i), deg_i)$ ($0 < i \leq m$) from a base node a via m rule-link to b contributes to the view to be removed and $(t_i, f_i) = (t_b, f_b)$, then

- if $\text{deg}[a] \leq \text{Max}(\text{deg}[\text{link}_i])$ where $\text{Max}(\text{deg}[\text{link}_i])$ denotes the maximum degree that can be propagated by this set of rule-links, then perform a *Remove*($a(t_a, f_a), S$) operation,
- otherwise perform a *Forget_r*($a \mapsto b(\text{deg}[a \mapsto b]), S$) operation for each of these contributing rule-links.

This procedure assumes that the degree of a rule (or a set of rules) is $\text{Max}(\text{deg}[\text{link}_i])$ and preferences are given to rules (the more generalized beliefs) than direct inputs (the more specific beliefs to derive the view point) when their degrees are equal.

5.3 Forget a Belief

To delete a belief b from a NLBN with a belief state S , a *Forget*(b, S) operation is used. There are two types of *Forget* operations:

1. Forget a proposition (*Forget_p*)

This operation assures that all beliefs supporting both the positive- and negative views to b are deleted and if applicable, node b is also deleted.

- Perform a *Remove_p*($b(1, 0), S$) and a *Remove_p*($b(0, 1), S$).
- If there is no out-going link from base node b , then delete node b .

2. Forget an IF-THEN rule (*Forget_r*)

To forget an IF-THEN rule “ $a \mapsto b(\text{deg}[a \mapsto b])$,”

- Delete the rule-link from node a to node b with link-weights of $(1, 0)$ and a link degree of $\text{deg}[a \mapsto b]$.
- If base node a or b does not have any incoming or out-going link (except one pseudo $((0, 0), 0)$ input-link), then delete the base node.

Other types of IF-THEN rules (with different link-weights) are forgotten in similar ways.

5.4 Revise According to a Proposition

Revising a NLBN with a belief state S according to a proposition b using the operation *Revise*($b((t_b, f_b), \text{deg}[b]), S$), will result in all existing supporting positive- and negative views to b being deleted and the new belief $b((t_b, f_b), \text{deg}[b])$ will be in the new belief state S_b^{Revise} .

- i. If b does not exist in the NLBN, perform an $Add_i(b((t_b, f_b), \text{deg}[b]), S)$.
- ii. Otherwise, perform a $Remove_p(b(1, 0), S)$, a $Remove_p(b(0, 1), S)$ and an $Add_i(b((t_b, f_b), \text{deg}[b]), S)$.

5.5 Update an Existing Belief

There are two types of update operations:

1. Update an input ($Update_i$)

An $Update_i(b((t_b, f_b), \text{deg}[b]', \text{deg}[b]), S)$ operation only changes the degree-of-belief value of an input from the existing $\text{deg}[b]'$ to the new $\text{deg}[b]$ where its proposition-value remains unchanged.

- i. If the input node b with a node value of $((t_b, f_b), \text{deg}[b]')$ does not exist in the NLBN, perform an $Add_i(b((t_b, f_b), \text{deg}[b]), S)$.
- ii. Otherwise, change the value of $\text{deg}[b]'$ at the input node to $\text{deg}[b]$.

2. Update an IF-THEN rule ($Update_r$)

Similarly, an update to an IF-THEN rule “ $a \mapsto b$ ($\text{deg}[a \mapsto b]$)” only changes its $\text{deg}[\text{link}]$ value.

- i. If the IF-THEN rule does not exist, perform an $Add_r(a \mapsto b (\text{deg}[a \mapsto b]), S)$.
- ii. Otherwise, modify the existing $\text{deg}[\text{link}]$ on the rule-link to the new $\text{deg}[a \mapsto b]$.

5.6 Not to Conclude a Proposition

A proposition b in a NLBN with a belief state S can be temporarily suppressed using the *Not-Conclude* operator.

- i. If base node b does not exist in the NLBN, there is no change to the network.
- ii. Otherwise, perform an $Add_i(b((0, 0), \text{true}), S)$ operation where “true” is the maximum degree in TAO.

This operation will cancel out all inputs from the strongest existing belief(s).

In the above procedures for the six network update operations, to maintain network convergence, we assume that network values of all affected nodes are propagated forwardly after each operation.

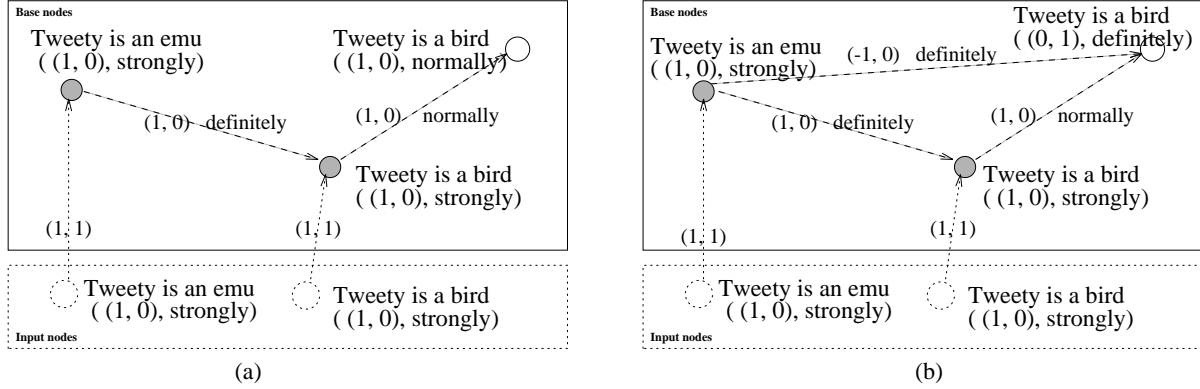


Figure 2: The NLBN representation of Tweety the bird example.

6 Examples

Let us use three examples to illustrate how commonsense knowledge can be represented and reasoned in a NLBN. The first example is a modified version of the popular “birds-fly” example for nonmonotonic reasoning.

Example 6.1 Tweety the bird: If we have the following two beliefs:

- (1) IF *Tweety is a bird* THEN *Tweety flies*. (normally)
- (2) IF *Tweety is an emu* THEN *Tweety is a bird*. (definitely)

and in addition we also believe that

- (3) *Tweety is a bird*. (strongly)
- (4) *Tweety is an emu*. (strongly)

then after four *Add* operations to a NLBN with a belief state S :

$$\begin{aligned}
 &Add_r(Tweety\ is\ a\ bird \mapsto Tweety\ flies\ (normally), S) \\
 &Add_r(Tweety\ is\ an\ emu \mapsto Tweety\ is\ a\ bird\ (definitely), S) \\
 &Add_i(Tweety\ is\ a\ bird((1, 0), strongly), S) \\
 &Add_i(Tweety\ is\ an\ emu((1, 0), strongly), S)
 \end{aligned}$$

we shall have the NLBN segment as shown in figure 2(a) where “*Tweety flies*” (normally) can be deduced. The belief state is:

$$S = \{Tweety\ is\ a\ bird\ (strongly), Tweety\ is\ an\ emu\ (strongly), Tweety\ flies\ (normally)\}.$$

If we later know that emu is a type of non-flying birds:

- (5) IF *Tweety is an emu* THEN Not *Tweety flies*. (definitely)

and after another *Add* operation

Tom's belief states at different time:

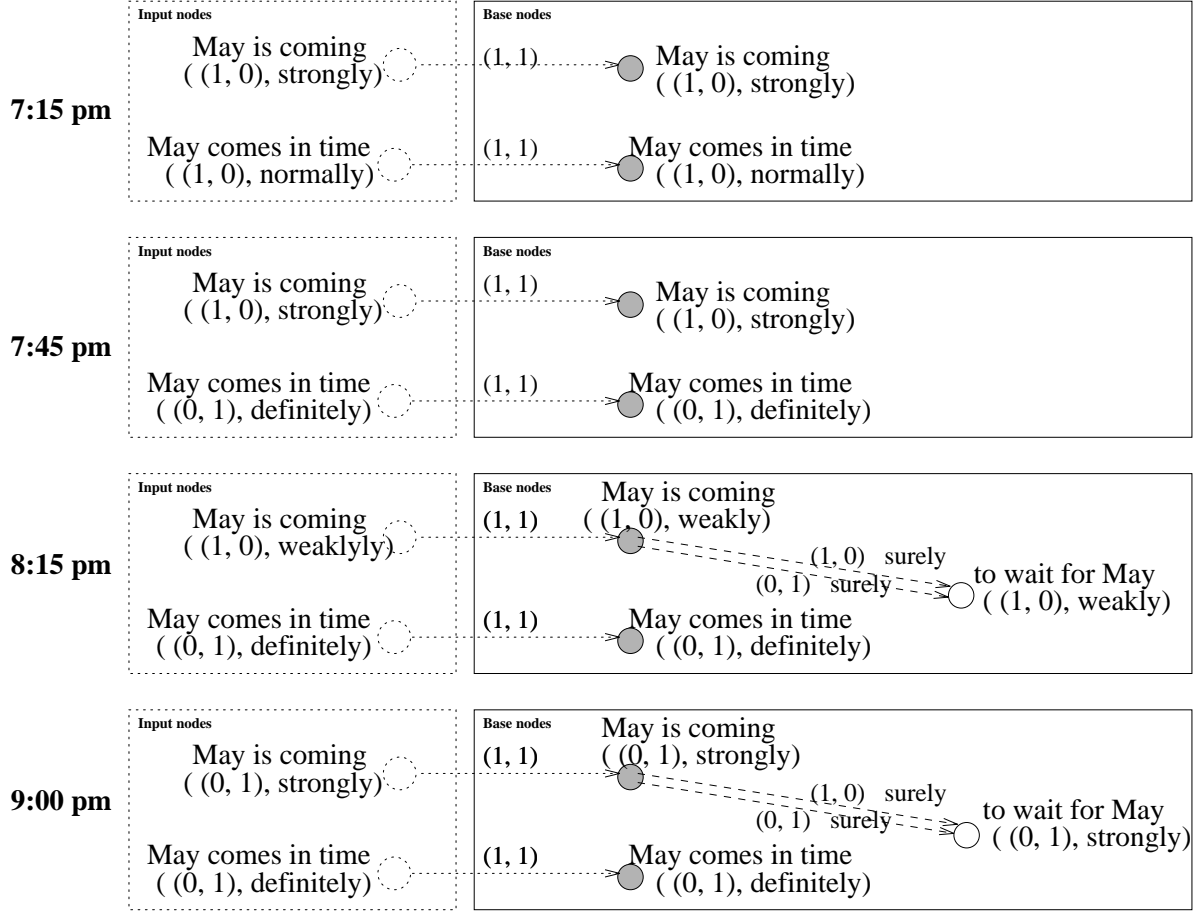


Figure 3: The revision of epistemic states for the waiting problem.

$$Add_r(Tweety \text{ is an emu} \mapsto \neg Tweety \text{ flies (definitely)}, S)$$

the final belief network is shown in figure 2(b) where “Not-*Tweety flies*”(definitely) is believed and it suppresses the previous belief of the flying ability of an emu. The belief state becomes:

$$S = \{ Tweety \text{ is a bird (strongly)}, Tweety \text{ is an emu (strongly)}, \neg Tweety \text{ flies (definitely)} \}. \square$$

A NLBN can be used to model the epistemic state transition of an agent (self belief modification), such as Tom's beliefs in the following example:

Example 6.2 Tom's appointment: Suppose Tom made an appointment with May at 7:30pm for a show and he believed strongly that she was coming. May was normally punctual so Tom, arrived at 7:15pm at the meeting place, had the following beliefs as shown in figure 3:

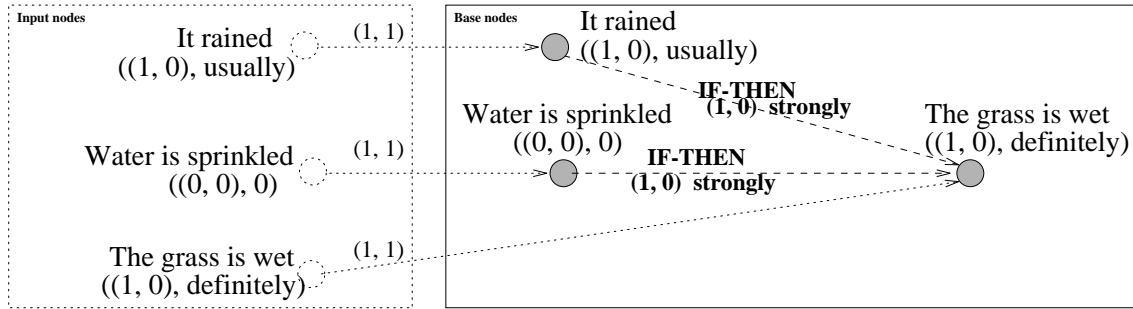


Figure 4: The NLBN representation of the wet lawn example (I).

- (1) *May is coming.* (strongly)
- (2) *May comes in time.* (normally)

At about 7:45pm, May had not turned out but Tom would still believe strongly that she was coming. He then revised his belief to:

- (2') *Not-May comes in time.* (definitely)

After another half-an-hour, Tom still believed that “*May is coming*” but the degree-of-belief has been reduced to “weakly.” He had to make up his mind whether to continue waiting for May so he asserted the following two rules:

- (3) IF *May is coming* THEN *to wait for May.* (surely)
- (4) IF *Not-May is coming* THEN *Not-to wait for May.* (surely)

At about 9:30pm, Tom got impatient and believed that

- (5) *May is not coming.* (strongly)

So he revised his belief state accordingly and concluded strongly that “*Not-to wait for May*” and went home. The epistemic revision is reflected in the belief network segments showing Tom’s belief states at different time periods after appropriate network update operations (*Add* (1) and (2) at 7:15pm, *Revise* (2’) at 7:45pm, *Add* (3) and (4) at 8:15pm, and then *Revise* (5) at 9:00pm) in figure 3. \square

A NLBN can be used to reason and explain situations:

Example 6.3 Wet lawn:

If Bryan looked out the window in the early morning and found that the lawn was wet. If he assumed that two reasons — *sprinkling* or *rain* — could strongly cause the lawn to be wet in the morning, then in his mind, he would have the following belief and two rules:

- (1) *The lawn is wet.* (definitely)
- (2) IF *it rained* THEN *the lawn is wet.* (strongly)
- (3) IF *water is sprinkled* THEN *the lawn is wet.* (strongly)

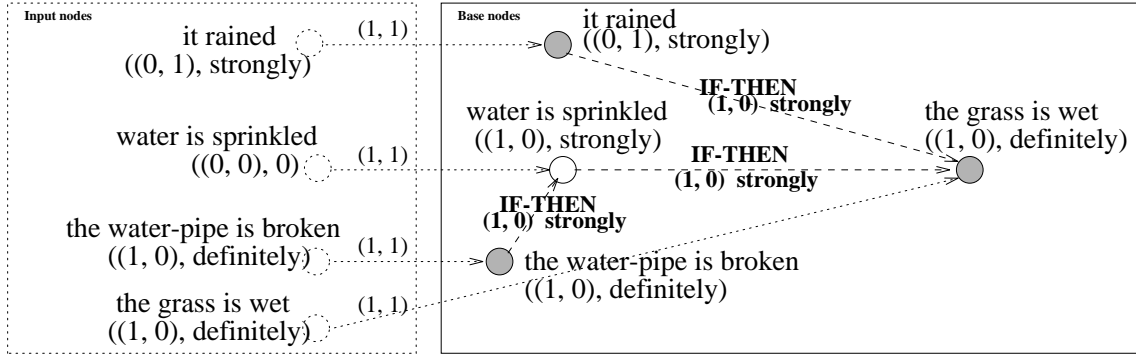


Figure 5: The belief network segment of the wet lawn example (II).

Suppose the weather forecast yesterday said that there would be showers in the morning and he believed that the weather forecast is usually correct, he would have the belief:

(4) *it rained.* (usually)

From the above, if four *Add* operations to the rules and propositions are performed, we would have the belief network segment representing Bryan's beliefs as shown in figure 4 and believe that the rain wet the lawn with a degree of "*usually*" via the rule-link representing belief (2).

If later in the morning, the weather announcement then said that it did not rain the night before, then Bryan will revise his belief to:

(5) Not-*it rained.* (strongly)

This caused his belief state to have no known reason to support the observation that the lawn was wet. As he walked out to work later, if he saw someone repairing a broken water-pipe next to the lawn, then he would immediately add to his beliefs that:

(6) *the water-pipe is broken.* (definitely)

and if he thought that a broken water-pipe would sprinkle lots of water out of the damaged joint, then he would have the following rule:

(7) IF *the water-pipe is broken* THEN *water is sprinkled onto the lawn.*
(strongly)

Now the reason for the wet lawn becomes "*the water-pipe next to the lawn is broken*" and after another three *Add* operations for beliefs (5) to (7), his epistemic state can be represented as a NLBN segment as shown in figure 5. \square

7 Discussion

Similar to the frame structure for knowledge representation [26] and nodes in Truth Maintenance Systems [8; 6], a NLBN distinguishes between representation of concepts (propositions) and their beliefs (node values). A concept can thus exist more intuitively in the belief base with one or more versions of beliefs (possibly contradictory), or, without any belief input (unknown). This is untypical of the classical logic based knowledge representation systems such as [24; 28; 1; 15] where representation of a proposition p in a belief base means it is believed. In a NLBN, input beliefs to a proposition are either direct evidence from the external world (inputs received via input nodes) or information derived from the beliefs of other propositions (inputs propagated from other base nodes). Another distinct difference is that a NLBN models commonsense rules as IF-THEN rules which are directed rule-links amongst base nodes. They are network configuration rather than a proposition and are therefore not included in the set of propositions of belief state S . They are belief structures rather than the resulting beliefs.

In a NLBN, the proposition-value $(0,0)$ denotes two truth values (see definitions 2 and 3), that is given a base node b :

$$\begin{aligned} (t_b, f_b) = (0, 0) \text{ with } \text{deg}[b] = 0 & \text{ means a } \mathbf{u} \text{ (unknown),} \\ (t_b, f_b) = (0, 0) \text{ with } \text{deg}[b] > 0 & \text{ means a } \mathbf{c} \text{ (contradictory).} \end{aligned}$$

Here, \mathbf{c} is considered to be information-wise inferior to having definite positive or negative beliefs (class \mathbf{t} or \mathbf{f}), and to prevent it from contributing to undesirable network inferences for deducing derived beliefs, a proposition-value of $(0, 0)$ instead of $(1, 1)$ is chosen to represent truth value class \mathbf{c} .

This principle seems contradictory to the selection function in definition 5 in which the input with the highest degree-of-belief value is selected as the final node value disregarding whether it is a contradictory, positive or negative input value. This wrong perception can be explained below. When there are opposite inputs (with both positive and negative beliefs) with the same highest degree value, it is defined as *contradiction* for the final node value in definition 5. A contradictory input can always be considered as having two opposite inputs (a positive input and a negative input) at the same degree. Therefore a contradictory input with the maximum degree interpreted as having two opposite inputs will give the exact result as selecting any input with a highest degree in definition 5. This has been characterized by theorem 8.

This theorem expresses the subsumptive behavior of a NLBN where the strongest belief(s) is preferred over weaker ones when there are more than one input to a proposition. This characteristic makes the belief network a nonmonotonic belief representation system as previous conclusions can be invalidated (subsumed) by new beliefs. This bias is motivated by how human beings generally make selections amongst contradictory information to a proposition in commonsense reasoning. For example, if you believe that normally there is no strong wind during this season and yesterday was a normal day, then you will believe “Not-*it has strong wind yesterday*” with a degree of “*normally*.”

If the weatherman now announces that “*it did not have strong wind yesterday*” and you believe him strongly, then you now have two supporting beliefs for “*Not-it has strong wind yesterday.*” Although combination of beliefs is a psychologically complex and unclear process, human beings tend not to combine weak beliefs into strong ones by simple linear summation. In this example we may accept that the belief stays at a degree of “*strongly*” although there is another supporting degree “*normally.*” Later, if your neighbor, Lyn, tells you that her umbrella was blown off by strong wind when she was at the park yesterday and you believe in her, then you have contradictory beliefs. If you believe Lyn more than the weatherman, say you “*definitely*” believe her, then naturally you will suppress the weaker information from the weatherman as it may be due to negligence or imprecise data collection of some weather stations for localized strong wind, and your weak commonsense rule where you know that exceptions to the rule are possible, and conclude that “*it has strong wind yesterday*” (definitely).

Theorem 9 re-expresses the subsumptive behaviors when logical AND and OR relations are considered. This theorem is in principle similar to the *MIN* and *MAX* operators for plausibility measures in fuzzy mathematics [37; 38] for logical AND and OR when the degrees of all inputs are known.

It is also important to note that a filtering process is built into the basic computation function for alternative-links in definition 5. A contradictory beliefs input to an alternative-link will always result in an unknown output. This filtering mechanism is biased towards the way defeasible commonsense IF-THEN rules are used in the real world that if the condition of a rule is in a contradictory state, we often do not want to derive any belief from it. To ensure a truth value of \mathbf{c} is transmitted via an alternative-link for external inputs, a contradictory input have to be converted into two opposite inputs of the same degree as shown in section 4.2.

With all logical expressions in CNF and all propositions are unique, any logical expression can be represented with three layers of base nodes: a lower layer of literals, the next layer of disjunctive relations and the third layer of conjunctive relations.

Suppose a proposition p is positively believed in a NLBN with a belief state S , that is $p \in S$ and $\neg p \notin S$, there is no way to deduce the belief of its negation $\neg p$. If $\neg p$ can be derived from other beliefs or direct inputs in the NLBN then its degree must be weaker than the current belief for p , i.e. $\text{deg}[\neg p] < \text{deg}[p]$. This is very different from many well-known knowledge representation systems. For example, if p is in a belief set K of AGM Logic [1; 10] with a certain level of epistemic entrenchment, then $\neg p$ is not in K and the epistemic entrenchment for $\neg p$ is 0 (zero). In probabilistic reasoning [29], if the probability of belief p is $P(p)$ then the belief for $\neg p$ is $1 - P(p)$; and in fuzzy reasoning [37; 36] the membership of the of $\neg p$ is $1 - U(p)$ where $U(p)$ is the membership degree of p . Similar observations are also true if $\neg p$ is believed.

In summary, this technical report defines the basic syntax of a Neural-Logic Belief Network (NLBN), its neural network-like computation functions for network value propagation, the representation of some typical logical and other relations, and a set of network update operations for dynamic belief change. It is shown that NLBN can be

used to model simple commonsense default reasoning, belief state transition of a reasoning agent and to provide simple explanations to different epistemological situations. Extensions of this formalism for representing predicated beliefs are discussed in [19].

References

- [1] C. E. Alchourron, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50:510–530, 1985.
- [2] J. L. Bell and M. Machover. *A Course in Mathematical Logic*. North-Holland, Elsevier Science Publishers, Amsterdam, 1977.
- [3] Nuel D. Belnap. A useful four-valued logic. In J. Michael Dunn and George Epstein, editors, *Modern Uses of Multi-Valued Logic*, pages 8–37. Reidel, 1977.
- [4] S. C. Chan, L. S. Hsu, S. Brody, and H. H. Teh. Neural three-valued logic networks. In *Proceedings of the International Joint Conference on Neural Networks*, Washington D. C., 1989.
- [5] S. C. Chan, L. S. Hsu, K. F. Loe, and H. H. Teh. Neural logic networks. In Omid M. Omidvar, editor, *Progress in Neural Networks (Vol. II)*. Albex Publication, 1993.
- [6] J. de Kleer. An assumption based TMS. *Artificial Intelligence J.*, 28:127–162, 1986.
- [7] Joachim Diederich, editor. *Artificial Neural Networks: Concept Learning*. IEEE Computer Society Press, Los Alamitos, California, 1990.
- [8] Jon Doyle. A truth maintenance system. *Artificial Intelligence J.*, 12:231–272, 1979.
- [9] M. Dummett. *Elements of Intuitionism*. Oxford University Press, Oxford, 1977.
- [10] Peter Gärdenfors. *Knowledge in Flux : Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, Massachusetts, 1988.
- [11] Matthew L. Ginsberg, editor. *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann, 1985.
- [12] Matthew L. Ginsberg. Multi-valued logics. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 243–247, 1986.
- [13] Anil Gupta and Nuel D. Belnap. *The Revision Theory of Truth*. MIT Press, Cambridge, Massachusetts, 1993.
- [14] A. Heyting. *Intuitionism, an Introduction*. North Holland, Amsterdam, 1956.

- [15] Hirofumi Katsuno and Alberto O. Medelzon. A unified view of propositional knowledge base update. In *Proceedings of the 11th IJCAI'89*, pages 1413–1419. Morgan Kaufmann, 1989.
- [16] Stephen Cole Kleene. *Introduction to Metamathematics*. North Holland, Amsterdam, 1964.
- [17] Feliks Kluzniak and Stanislaw Szpakowicz. *Prolog for Programmers*. Academic Press, London, 1985.
- [18] T. Kohonen. An introduction to neural computing. *Neural Networks*, 1:3–16, 1988.
- [19] Boon Toh Low. Neural-logic belief networks—a tool for knowledge representation and reasoning. In *Proceedings of the 5th IEEE International Conference on Tools with Artificial Intelligence*, pages 34–37, Boston, 1993.
- [20] Boon Toh Low. A subsumptive inference network for belief representation. to appear in the Proceedings of the Third Pacific Rim International Conference on Artificial Intelligence, Beijing, China,, 1994.
- [21] Boon Toh Low and Norman Y. Foo. Towards a network for representing beliefs. In *Proceedings of the Second International Computer Science Conference*, pages 85–91, Hong Kong, 1992. IEEE.
- [22] Boon Toh Low and Norman Y. Foo. A network formalism for commonsense reasoning. In *Proceedings of the Sixteenth Australian Computer Science Conference*, pages 425–434, Brisbane, 1993.
- [23] Witold Lukaszewicz. *Non-Monotonic Reasoning: Formalization of Commonsense Reasoning*. Ellis Horwood, New York, 1990.
- [24] Drew McDermott and Jon Doyle. Non-monotonic logic I. *Artificial Intelligence J.*, 13:41–72, 1980.
- [25] Pankaj Mehra and Benjamin W. Wah, editors. *Artificial Neural Networks: Concepts and Theory*. IEEE Computer Society Press, Los Alamitos, California, 1992.
- [26] Marvin Minsky. A framework for representing knowledge. In P. H. Winston, editor, *The Psychology of Computer Vision*, New York, 1975. McGraw-Hill.
- [27] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Massachusetts, 1969.
- [28] Robert C. Moore. Possible-world semantics for autoepistemic logic. In *Proceedings of AAAI, Workshop on Non-Monotonic Reasoning*, pages 344–454, New Paltz, NY, 1984.

- [29] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, 1988.
- [30] Nicholas Rescher. *Many-Valued Logic*. McGraw-Hill, New York, 1969.
- [31] T. J. Reynolds, H. H. Teh, and B. T. Low. Programming in neural logic. In *Proceedings of the First Pacific International Conference on Artificial Intelligence*, Nagoya, Japan, November 1990.
- [32] David E. Rumelhart. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, Massachusetts, 1986.
- [33] Joseph R. Shoenfield. *Mathematical Logic*. Addison-Wesley, Readings, Massachusetts, 1967.
- [34] Leon Sterling and Ehud Shapiro. *The Art of Prolog*. The MIT Press, Cambridge, Massachusetts, 1986.
- [35] Geoffrey G. Towell and Jude W. Shavlik. Refining symbolic knowledge using neural networks. In *Proceedings of the International Workshop on Multistrategy Learning*, 1991.
- [36] Pei Zhuang Wang. *Fuzzy Sets and Falling Shadows of Random Sets*. Beijing Normal University Press, 1985.
- [37] Lotfi A. Zadeh. Knowledge representation in fuzzy logic. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):89–100, 1991.
- [38] Tamás Zétényi, editor. *Fuzzy Sets in Psychology*. North Holland, Amsterdam, 1988.