January 29, 2024

Self-organizing Logic

進藤　啓介

SHINDO Keisuke

kshindo999@gmail.com

Abstract

Self-organizing logic is a theoretical system of a logical circuit operation simulator aimed at correcting and generating the order circuit by probability. This approach forms a logical circuit using a bidirectional Beigian network. The normal mapping is described using the reverse transport of this two-way logical operation, which allows a hierarchical sequencial circuit to be described. Furthermore, the propagation of the network probability is expressed by the propagation probability of the bib. The opposite direction of propagation is defined based on the theorem of Bayes. This probability calculation enables feedback based on strict probability theory to the network, enabling accurate logical operation insertion. Furthermore, the activity value called the assumption vector is proposed to this network. By selecting a network node by comparing the element of this assumed vector, a minimal new associative link is generated regardless of the scale to be observed. It is expected that these elements can be generated quickly and efficiently in real world data processing and predictive model construction.

Keywords

machine learning,bidirectional baysian network,sequencial circuit

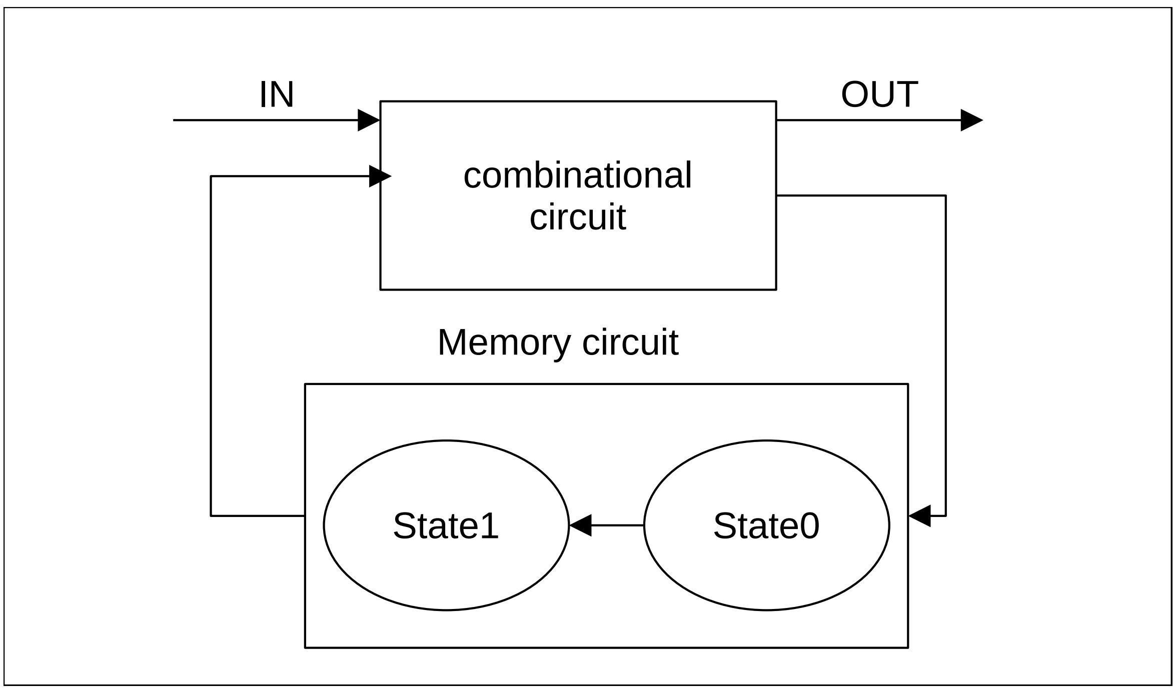
Contents

1 Components of self-organizing logic

1.1 Bidirectional Logic Operations and Linked Nodes

1.1.1 classical sequential circuit

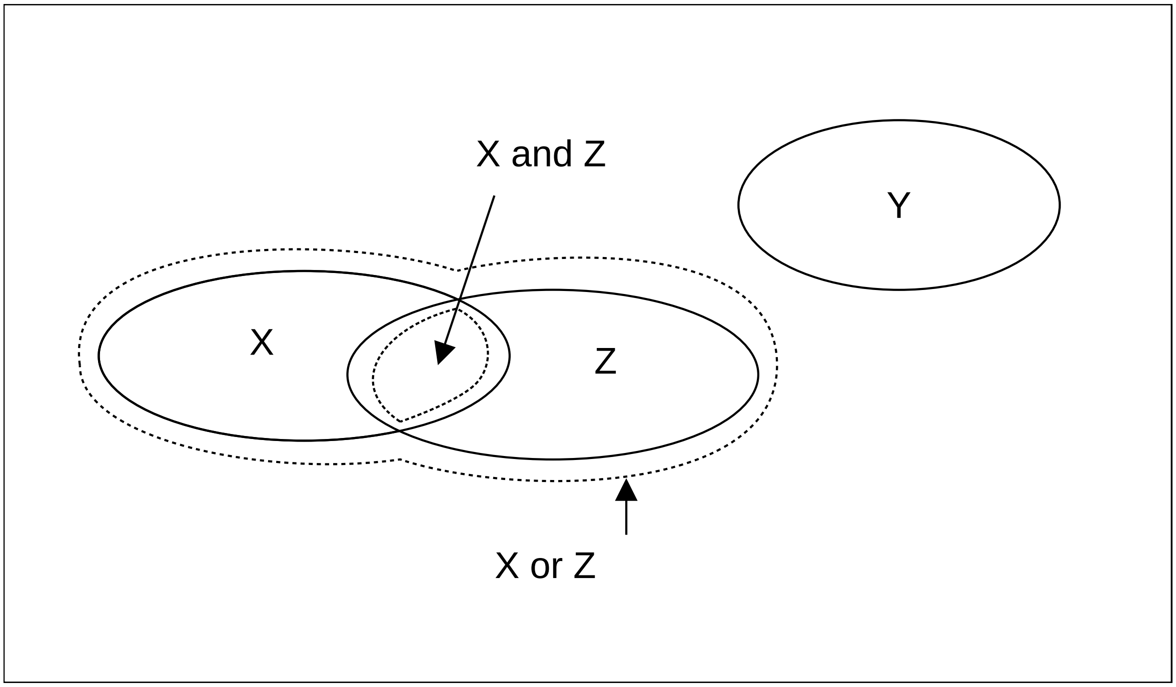
In ordinary logic algebra, the elements of AND, OR, and NOT logic operations are held as nodes, and they are joined by links to perform basic logic operations. In an ordinal circuit, a memory function such as a flip-flop FF or memory is added to this, and CPUs, GPUs, etc. all correspond to this ordinal circuit. This sequential circuit can theoretically produce any output sequence for any input sequence.



Sequencial circuit

1.1.2 Mapping by backpropagation of logic circuits

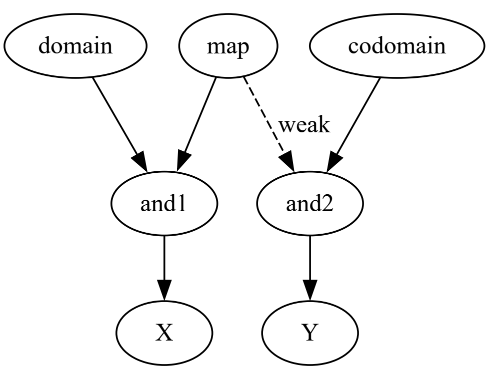
Logical operations act to cut and paste sets into each other, as expressed in Venn diagrams, etc. However, logical operations cannot be performed between exclusive subsets that do not overlap, such as X and Y in the figure below. In contrast, a mapping makes it possible to reach another set that is exclusive with respect to the input set. To do so, the sets before and after the mapping are considered as subsets of an even larger set, the Map node, which encompasses them. The input node X and the output node Y are equal to the result of ANDing with the domain and codomain nodes, respectively, for the Map node.

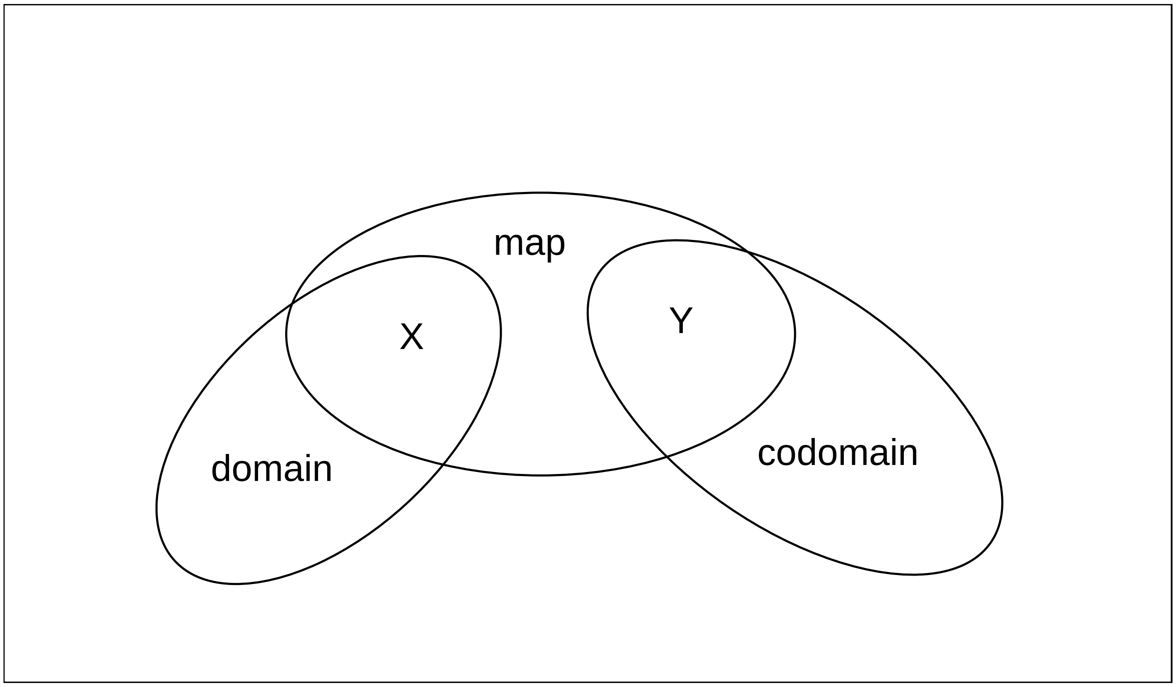


Basic venn diagram

The propagation from the mapping source X to the mapping destination Y is realized by back propagation to the Map node. In order to define probability propagation between exclusive nodes, the back propagation from AND1 to Map enlarges the set by complementing the complement part of domain. The set is then expanded by the Map node. The reason for AND instead of XOR is that the entire interior region of input node X is contained within the map.

This associative link using mapping nodes and back propagation allows the description of state transitions. Furthermore, this action enables the same behavior as memory devices such as flip-flops and memories, but in a more generalized manner. For example, a memory readout can be viewed as a mapping from a set of different spacetimes.

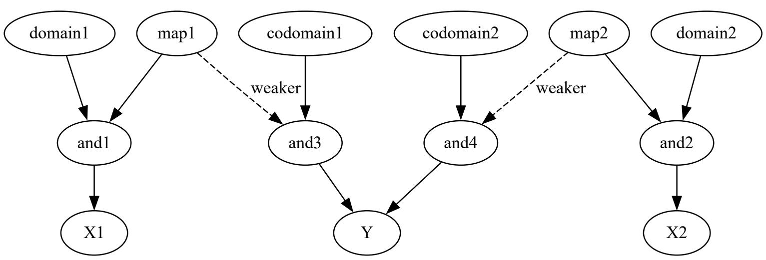


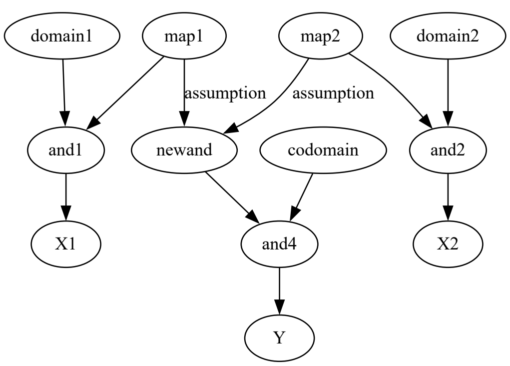


Map link by backward logic propagation

1.1.3 Inserting logical operations into indeterminate links of mappings

The link to the mapping destination Y of map1 may not have a fixed propagation probability depending on the observation conditions. In order to find the conditions under which this uncertain weak link is determined and propagated to Y, the logical operation newand with another map map2 is tentatively formed. newand is not only the logical operation AND, but also various other types of logical operations such as OR and XOR are possible, but the type of logical operation can be selected definitively according to the propagation probability of the weak link and the propagation set.

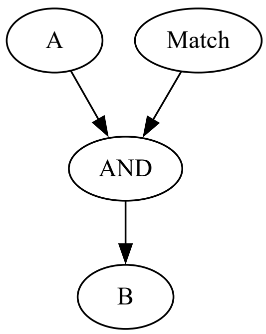




Feedback and create new logic node

1.1.4 Feedback and backpropagation to uncertain inputs

All links have variable propagation probabilities due to feedback. By using the probability of the link $A \rightarrow B$ itself, the XOR node and the link node corresponding to the condition of the link itself are generated and back propagated. This integrates the propagation sets of A and B before and after the mapping, and the link node itself becomes a set representing the link, making it possible to make the causal relation of the link itself the target of logical operations. Furthermore, this can be applied to the control of conditional statements in software by regarding the matching of state variables as a link node. This cannot be achieved by forward logic operations alone.



Back propagation indicate causal relationships

Logical operations do not require all inputs to be definite like 1 propagation to OR. However, one input to the AND in the above example will not become a definite value unless there is another input. If the Match input to AND propagates to B in an uncertain state, feedback may occur due to a collision with another value, and consistency may be achieved if the input Match is determined to be 1. In this case, back propagation of the value from AND to Match occurs.

1.2 Bidirectional binomial stochastic Bayesian network

1.2.1 Probability propagation

Each connecting link between logical nodes is given a propagation probability. The logical values to be propagated are binary values of 1 and 0, and the propagation probability is indicated by a value from 0 to 1.

The link consists of two pairs of propagation probabilities P^{f11},P^{f00}, and also specifies a pair of propagation probabilities P^{r11},P^{r00} in the opposite direction.

In the propagation link from A to B, the probability values P(A) and P(B) for A and B, respectively, are associated by two propagation probabilities P^{f11},P^{f00}.

$$P(B) = P(A)P^{f11} + (1 - P(A))(1 - P^{f00})$$

The NOT operation, which is a value inversion, sets P^{f11} , P^{f00} to 0, respectively; the non-NOT propagation sets P^{f11} , P^{f00} to 1, respectively. Logical operations are also defined as probability calculations, integrating the input probabilities P_1^{f11},P_2^{f11},P_3^{f11} of multiple logical operations. The following is an example of AND.

$$P' = P_1^{f11} P_2^{f11} P_3^{f11} ...$$

By combining these nontrivial probability calculations, the propagation probability of the entire path is obtained in the path of a logical operation between any two nodes. It differs from a neural network in that it propagates the exact probability itself and does not use any threshold function or the like.

1.2.2 Backward propagation

This section describes the probability of backward propagation of a link. For example, suppose the following AND node C is generated.

$$C = A \cap B$$

Conversely, if node C is activated with probability 1, then nodes A and B can be activated with propagation probability 1. This is backward propagation.

$$C \rightarrow B$$

Backward propagation ensures propagation between two arbitrary nodes. The probabilities of backward propagation P_r^{11} and P_r^{00} are calculated using Bayes' theorem, where P(A) is the observed probability of link source A and P(B) is the observed probability of link destination B, and the following equation is obtained

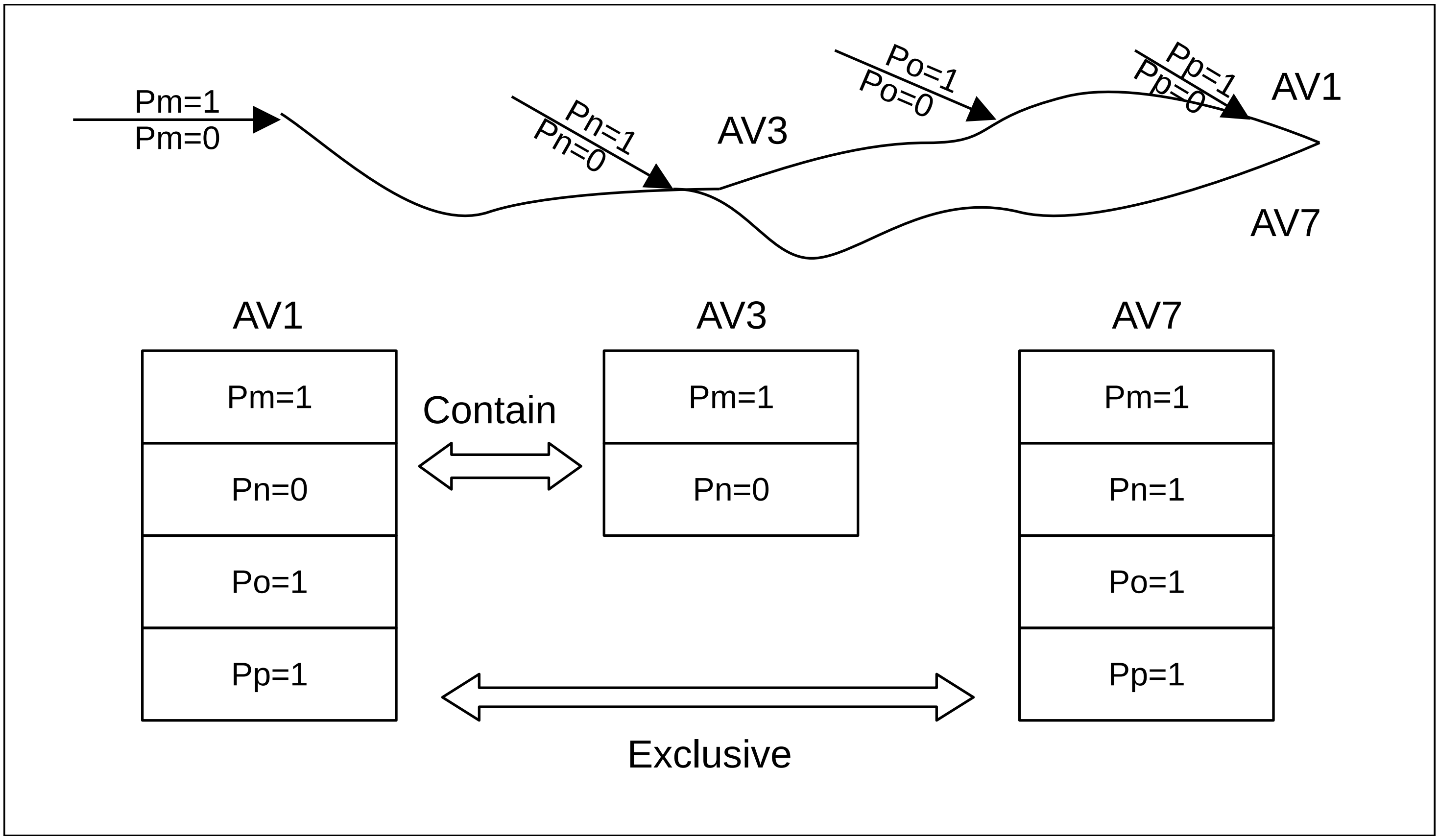
$$P_r^{11} = \frac{P(A)P^{11}}{P(B)}$$

$$P_r^{00} = \frac{(1 - P(A))P^{00}}{1 - P(B)}$$

In other words, two-way probability propagation incorporates Bayes' theorem in a natural way. Backward propagation for logical operations propagates to all input links when all values are fixed, as in the back propagation of the value 1 to the AND node; back propagation of the value 0 to the AND node does not fix the propagation probability of the input links, but if only one input link is selected for propagation, only the selected link has a fixed probability. If only one input link is selected to propagate, only the selected link will have a fixed probability.

1.3 Assumption vectors and propagation sets

SOL propagates an active state over the network, but the active state does not occupy the entire cross-section of the propagating link as a set, and we consider that the set is broken up by the values of several nodes that are the starting points. This propagation set is independent of the set that passes over the network and is determined by the combination of values of the originating nodes. For this reason, we refer to the assumed combination of multiple origin node values as an "assumption vector. Hereafter, the Assumption Vector (AV) will be abbreviated as "AV" when necessary.



Assumption vectors

1.3.1 Synthesis of assumption vectors

Multiple assumption vectors are synthesized by logical operations, resulting in an assumption vector whose assumption vector elements are synthesized from both sides.

- Assumption vector elements that are different among the assumption vectors are synthesized as they are.
- Assumption vector elements that are identical among the assumption vectors are aggregated into one.
- Composition with collectively inclusive assumption vector elements replaces the propagation collectively smaller elements.
- Composites of collectively exclusive vector elements result in the propagation set itself becoming an empty set.

1.3.2 Propagation set comparison using assumption vectors

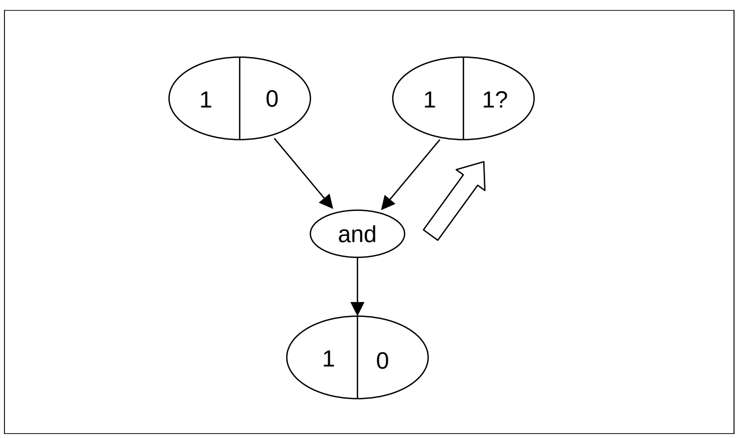
This assumption vector makes it possible to compare the magnitudes of multiple active states as a propagation set. This comparison is as a propagating set that has passed through the mapping and is independent of the actual set.

- If all the assumption set elements of the assumption vectors are identical, then both propagating sets are identical regardless of the actual set.
- If only one of the non-identical parts of the assumption set elements of the assumption vector is identical, then both propagation sets are inclusive.
- If non-identical parts of the assumption set elements of the assumption vectors are non-identical on both sides, then both propagation sets are unrelated and therefore not subject to association.
- If there is no overlap at all between the assumption set elements of the assumption vectors, the propagation itself is stopped because the propagation set vanishes.

The strict management of the propagation set using the assumption vectors allows for the formation of precise associative links between two nodes. All link formation in SOL is performed selectively according to these assumption vectors.

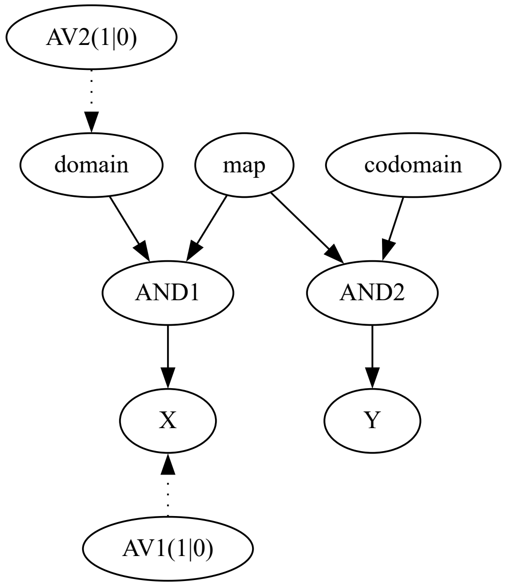
1.3.3 Mapping backpropagation and propagation set expansion

For nodes X and Y indicating Bool values, etc., we describe how to propagate back from node X to node Map to exclusive node Y by the action of expanding the set.



Propagate backward

As shown in the figure above, back propagation from output to input for AND means that the probability of the result of back propagation is not fixed, even if all other inputs are fixed. There is no contradiction in the right input being either 0 or 1.



Using assumption vectors to propagate on map link

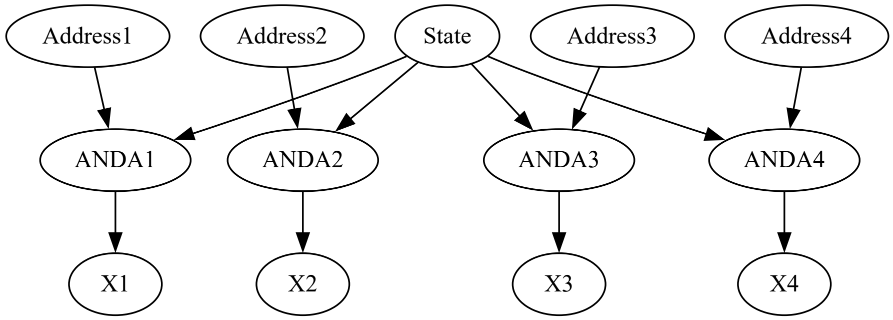
Propagate back from X to Map. If there is a separate input for the set domain when propagating back from AND1 to Map, the complement of domain can be expanded. suppose that both AV1 propagated from X and AV2 propagated from domain are pairs of two propagation sets that are complementary in value. In this case, the set from X is expanded and propagated to Map if the propagation set of X coincides with or is fully contained in the propagation set of domain. Expanding the set means that the set is expanded by combining the elements of the complementary assumption vector AV of DOMAIN, which also means that the assumption vector AV is reduced by one element.

Since the result of synthesizing the assumption vector expands the propagation set, it is highly likely that the probability is uniformly 1. The reason for this is that the propagation set synthesized by Map is likely to be identical to the propagation set before the addition of the assumption elements to AV1, so its probability is also presumed to be uniform, just like the propagation set before the addition of the assumption elements.

Thus, the propagation set is propagated from Map to AND2 in an expanded state and ANDed with codomain. As a result, the expanded propagation set is propagated to Y. Here, the link from the map node to the AND2 node is weak and undetermined, which also means that the estimation of the expansion of the set above is not yet determined. Conversely, the fact that the weak link is strengthened by positive feedback means that the probability of set expansion is also closer to being determined at the same time.

1.3.4 Expressing general states

Using mapping, a huge state can be represented as follows. Substates X1, X2, etc. can be connected to further states hierarchically. Not only can each state be shown by dividing it into addresses, etc., but also continuous time series and coordinates can be used, and even abstract nodes can be connected by different mappings.



Multi state using map links

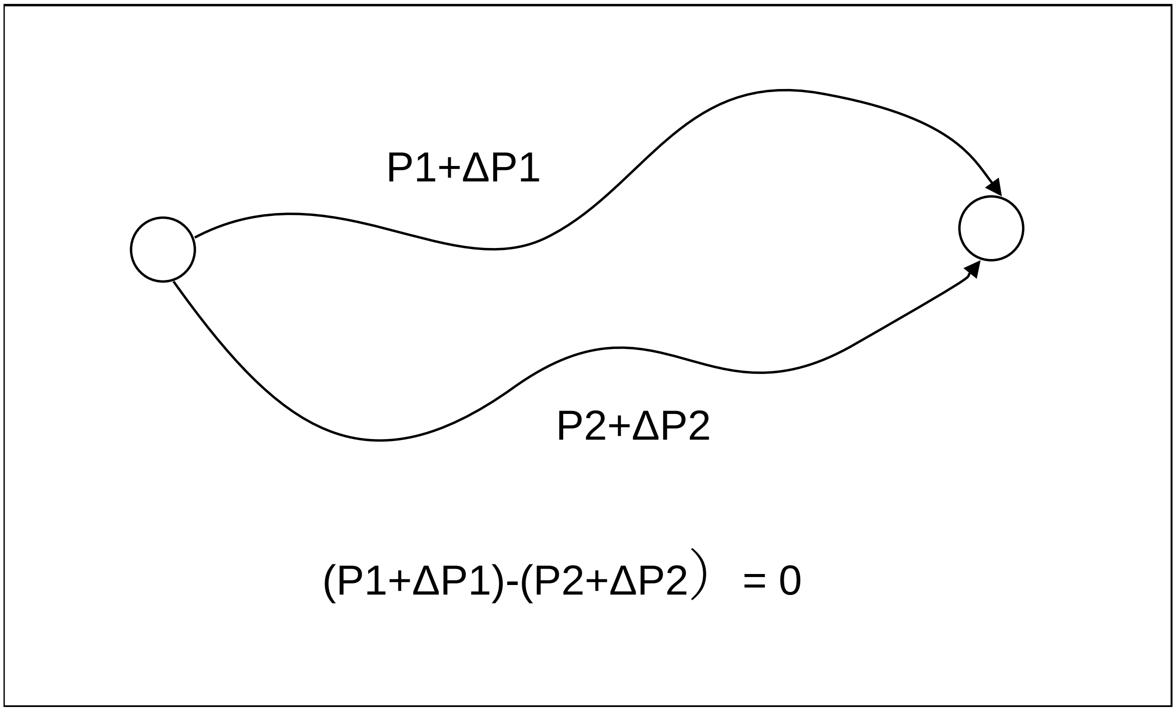
1.4 Stochastic variation distribution hierarchical feedback algorithm

To reproduce the observed probabilities by resolving the difference between the propagation probability of a path through the network in SOL and the probability of propagation of the path observed in the external observation function. We consider this to be generalized learning in SOL. The objective is almost identical to the learning of existing ordinary neural networks, but without heuristics such as activation functions, but in a more rigorous manner.

1.4.1 Propagation, collisions, and feedback

From any common origin node, it may pass through multiple paths in the network to reach another identical node once more. For example, a typical example would be a collision between a previously observed result node and the currently observed result node.

If there is an overlap in the set of hypothetical vectors that serve as the starting points of two routes, it is considered a collision. If the propagation probabilities of both colliding paths are different, feedback is provided. In order to accurately calculate the amount of feedback, probability calculations are performed in this probability variation distribution hierarchy feedback to identify the links that should be fed back and the amount of feedback. The action from the feedback from the links is described below.



Difference in probability of passing two passes

1.4.2 Calculating feedback value

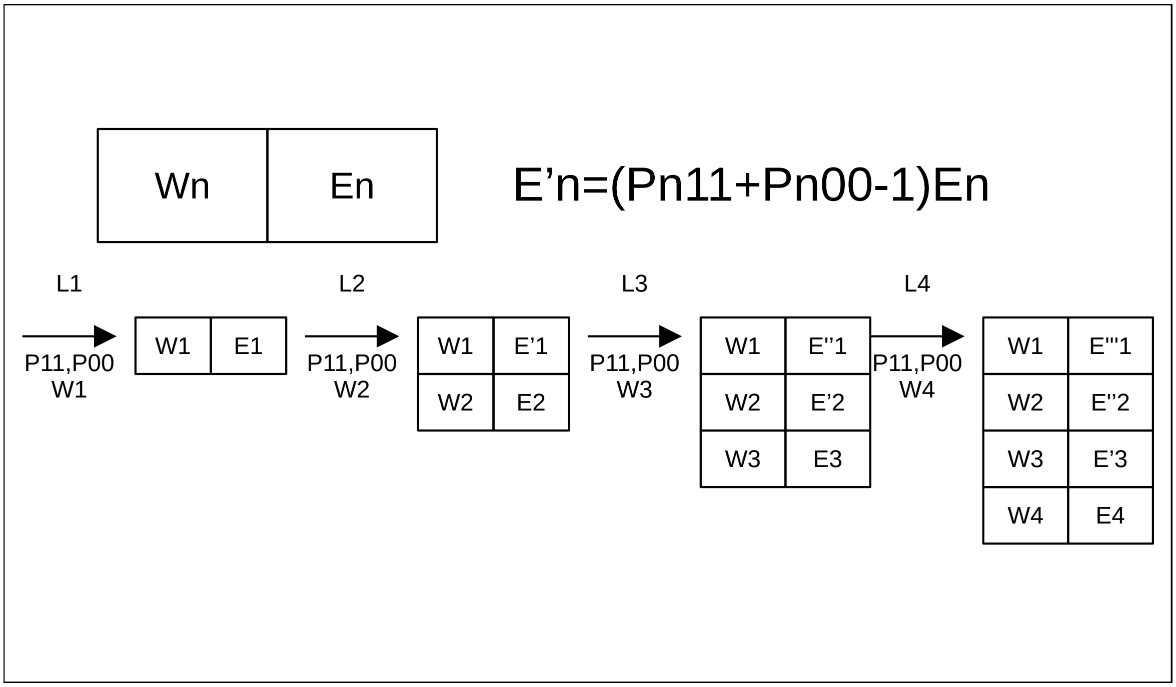
A difference in the propagation probabilities of the two paths occurs when the propagation probabilities of the two paths as a whole are observed. To resolve this difference, feedback is provided to each path. The problem here is that with hierarchical links, it is difficult to determine which and how much feedback to apply to which link. The "stochastic variable distribution hierarchical feedback" algorithm solves this problem with a more precise method. This method makes it possible to properly provide near-optimal probabilistic feedback for each link in a theoretically unlimited hierarchy with a small computational order of magnitude.

The basic concept starts with the fact that the probability of occurrence of a change in probability depends on the propagation probability P_n of the link and the number of observations N_n. Suppose that the overall propagation probability P_{total} is observed as a result of multiple passes over this link. If a difference from the result of that overall propagation probability occurs, the difference is distributed and fed back to each link according to the per-link weight w_n as follows. The per-link weight w_n is calculated by the following formula (based on a fairly rigorous derivation, but omitted here). Where P_n is the propagation probability of the link observed so far and N_n is the number of times the feedback has been applied so far. The E_n is the probability that each link has been used in the propagation and is calculated by the overall propagation.

$$w_n = \frac{P_n(1-P_n)}{N_n}$$

$$P_n' = P_n + w_n \frac{\Delta P_{total}}{\sum_n E_n w_n}$$

The above W_n and E_n are used to calculate the probability of the entire propagation. For E_n, the probabilities of individual links are applied and propagated according to the propagation of the links.



Weight propagation

Once the propagation is completed, the overall observation probability variation is distributed according to W_n, E_n so that the propagation probability of each link is as close to 0 or 1 as possible and the signs of the variation values are consistent.

The propagation probability corrected by feedback does not directly become the next propagation probability of the link. The larger the number of previous feedback observations N_n of the link, the smaller the amount of feedback of the link propagation probability. E_n is an effect coefficient for the effect of variation in the probability of each link on the overall propagation probability, and is calculated based on the overall propagation.

$$N'_n = N_n + E_n$$

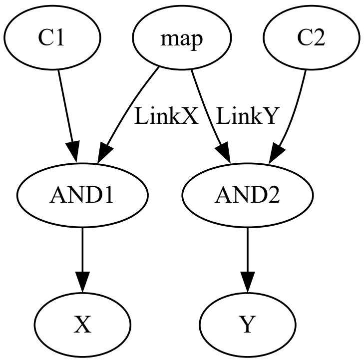
$$P'_n = \frac{N_n P_n + E_n P'_n}{N'_n}$$

This stochastic variable distribution hierarchy feedback allows the feedback to act more accurately on deep hierarchical networks, even when compared to neural network backpropagation. The reason is the application of rigorous probability theory.

1.5 Forming associations from simultaneous observations

Simultaneous observation is the basic method for defining associations between nodes that are collectively non-overlapping. When a subset of a node has a value determined simultaneously with a subset of another node, it is fundamental to associate them by mapping. Simultaneous means that the assumption vectors are congruent or entailing. However, this alone may result in a coincidental coincidence of unrelated events. For this reason, feedback is used to increase the probability of association.

Therefore, the mapping association is formed from the observed fact that the probability P of occurrence of the two values is somewhat low and that the two values are determined simultaneously. Typically, the association is formed from the simultaneous variation of node X and node Y.



Forming associations from simultaneous observation

The associative uncertainty is represented by the probabilities P^{11} and P^{00} of the link from Map to AND1 or from Map to AND2. N^{11},N^{00} of a link indicates the number of feedbacks to that link.

Let P_X,P_Y be the past observation probabilities of X and Y, respectively. If $Y = 1$ when $X = 1$, the specific initial value of the LinkY parameter is determined by the following formula. In particular, the strength increases when P_Y is sufficiently small.

$$P_y^{11} = 1$$

$$P_y^{00} = 0.5$$

$$N_y^{11} = -log_2 P_Y$$

$$N_y^{00} = 0$$

If $Y = 0$ when $X = 0$, the parameters are determined by the following formula.

$$P_y^{11} = 0.5$$

$$P_y^{00} = 1$$

$$N_y^{11} = 0$$

$$N_y^{00} = -log_2(1 - P_Y)$$

LinkX is defined similarly. Other links are considered to have definite values. Links with fixed values are not subject to feedback.

$$P^{11} = 1$$

$$P^{00} = 1$$

$$N^{11} = \infty$$

$$N^{00} = \infty$$

After the association is formed, feedback is provided by observing the propagation probabilities of both propagation from $X = 1$ to $Y = 1$ and propagation from $X = 0$ to $Y = 0$. . As a result, the associative links N_y^{11} and N_y^{00} are further added, and the probabilities of both the P_y^{11} side and the P_y^{00} side increase. It will be confirmed.

The objects X and Y that form the association can be not only observation result nodes, but also condition nodes that are the result of feedback, link nodes that indicate causal relationships, etc. In this way, associations can be formed between all subsets based on probabilities.

1.6 Stochastic autonomous logic generation algorithm

SOL uses a "probabilistic autonomous logic generation algorithm". This algorithm autonomously generates and modifies nodes and links.

To generate it autonomously for every conceivable configuration of nodes and links, it is basically considered possible with the following functions.

1.6.1 "Conditional" Link splitting and logical operation node insertion

If the feedback results in the probability of the link approaching 1 to 0.5 or 0 to 0.5, the link is considered negatively fed back. In other words, subject to the current assumption vector, the uncertain probability portion of the link is separated using logical operations.

By managing the propagation probability of the link with the binomial pair of P^{11} and P^{00}, the AND and OR nodes to be inserted can be selected deterministically from the direction of the propagation probability in which negative feedback has occurred. An AND node is selected when P^{11} approaches 0.5 from 1, and an OR node is selected when P^{00} approaches 0.5 from 1. An XOR node is selected when both P^{11} and P^{00} approach 0.5, but under identical conditions for both P11 and P00 However, it is limited to the case when the same conditions are applied to both P11 and P00. Conditional links added to inserted logical operations are selected from nodes with equal assumption tensors.

This is the basis for the autonomous generation of logic operations in SOL, which enables the formation of accurate logic operations.

1.6.2 "Causal" Link node activation and associative targeting from positive feedback

If the probabilities of two activation paths through a random link match, then the random link is positively feedback. As a result, a "link node" corresponding to that random link is considered to be activated by positive feedback.

The concrete entity of the link node is an unknown input node that is input to an exclusive logic operation (XOR) node that is inserted virtually appended to the random link.

B=A xor ¬ L

Back propagation to the link node propagates the result of matching the propagation probabilities P_A and P_B at both ends of the link. Specifically, the propagation probability P_L is the following equation.

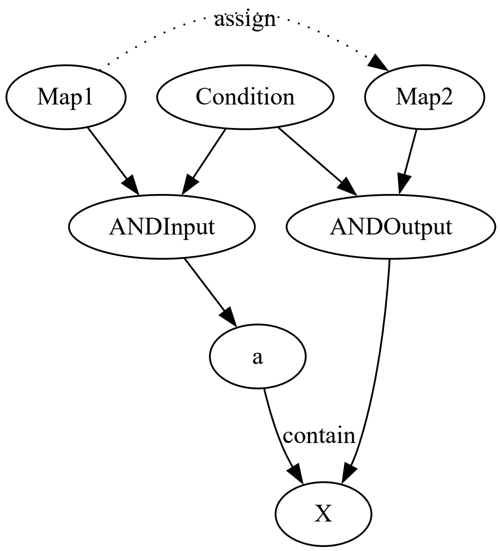
$$P_L = \frac{P_A + P_B - 1}{2P_A - 1}$$

This will be the basic way to make the very result of a causal relationship between two different sets the subject of a probability calculation. Even if the probability of P_A, P_B is an intermediate value, it will be a definite probability 1 if the match is other than 0.5.

Causality can be applied to conditional decision control from matching judgments of numerical values, etc., i.e., if-conditional statements in software.

1.6.3 "Substitution" Coupling between nodes that are inclusive in terms of the propagation set.

Propagation by logical operations in the forward direction makes the set smaller as conditions such as ANDOR are added. Propagation in the reverse direction of logical operations allows conditions to be removed and the set before the conditions are applied to be propagated as is. As a result, an inclusion relationship may be established between mapping nodes that have passed through the addition and removal of conditions. This consistent propagation between mapping nodes is called assignment. In the following example, since the value a is contained in the variable X ($a \subset X$), Map1 containing the value a is a subset of Map2 containing the variable X, and Map1 is assigned to Map2.



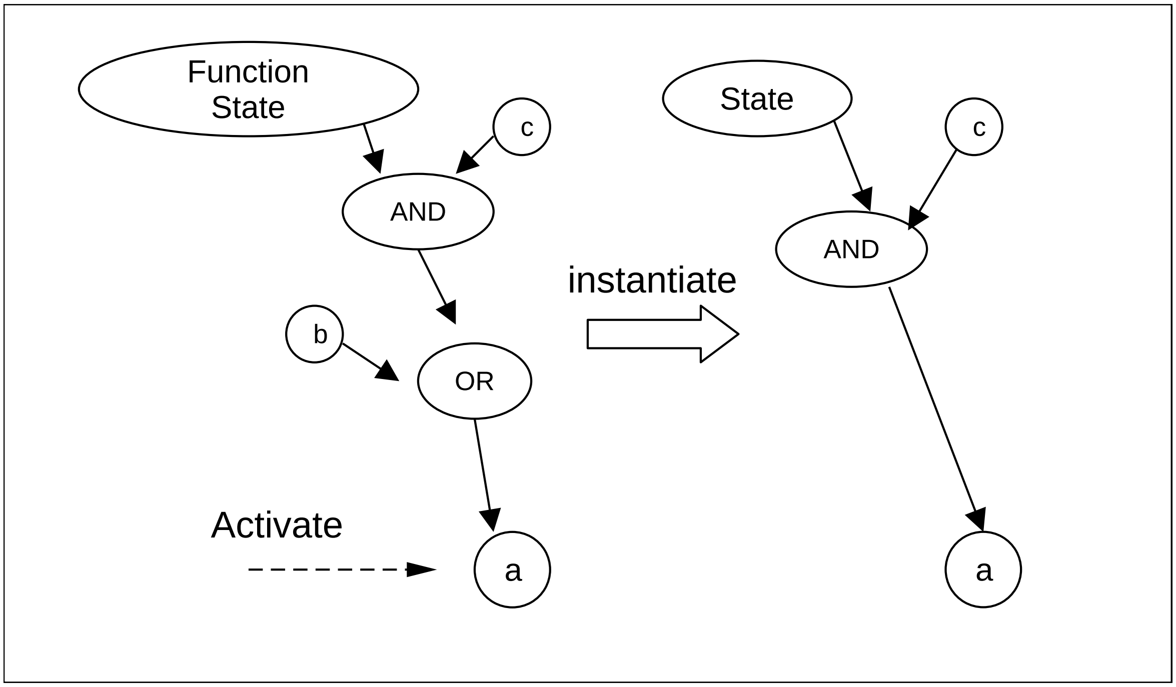
Assign between map nodes

1.6.4 "Instantiation" Instance replication of part of the network by propagation subset

Sometimes, as a result of two pathways, it is possible to expand the set of nodes. In this case, if the path of a link has multiple output links, logical operation input links, or other branches, the network of the path is partially replicated to form a kind of instance. The instance omits unnecessary link branches.

An example is shown in the figure below: FunctionState is pre-substituted with the State on the right as a subset. Propagation is performed from the assigned State through the function to the active state indicating the value a as the result of the function execution. This propagation set does not overlap with State but allows State's propagation set to be expanded without contradiction. For this reason, State is added as a duplication of the network to the result as a result of the function.

This "specialization" has the effect of reducing the search branch of the propagation by reducing the number of branches of the propagation, while at the same time allowing for the extraction of more probabilistically determined portions of the network structure. This "specialization" is also used to generate function results.



Network instantiate

1.6.5 "Selection" Selective control of multiple link propagation

For a given node, there can be a large number of output links to be combined. To make the link search efficient, link selection from a large number of links must be performed for a condition. It is essential for SOL itself to determine and streamline this selection process.

Controlling link selection means that the link selection node corresponding to the selected link is generated and made the target of association and active state propagation. As a result, it is possible to associate the utility, etc. obtained from the active state of the selected destination with the link selection node. Conversely, the link selection node is activated from the utility, and the link corresponding to the utility is selected.

Unlike the method of controlling the selection of link selection nodes by adding conditional logic operations to the links, the propagation probability of the links can be maintained. Since the propagation of the link itself is not particularly inhibited even when the link is not selected, propagation itself is possible, but the probability of the link being selected as the propagation target is low.

1.6.6 Probabilistic autonomous logic generation

These are the basic components of the probabilistic autonomous logic generation algorithm; AND, OR, XOR, NOT, mapping links, and link control can be generated automatically using the above means. For stochastic variations and sets, deterministic probability links with propagation probabilities close to 100% or 0% are selected whenever possible.

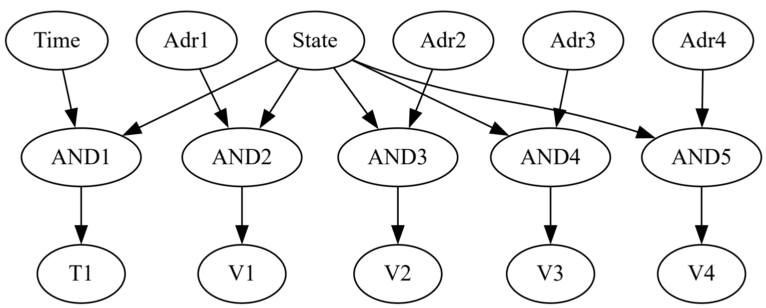
Links with a large number of observations and uncertain propagation probabilities are replaced with deterministic probability links as much as possible by sequentially adding other conditions.

In other words, unlike neural networks and other algorithms that use analog value weights, this probabilistic autonomous logic generation algorithm attempts to reproduce the observed object by digitizing it as much as possible. Since deterministic digital logic is generated, the output of the learning results in the form of logical expressions is possible.

1.7 Example of autonomous generation of sequential circuits

1.7.1 From association between fluctuations to states

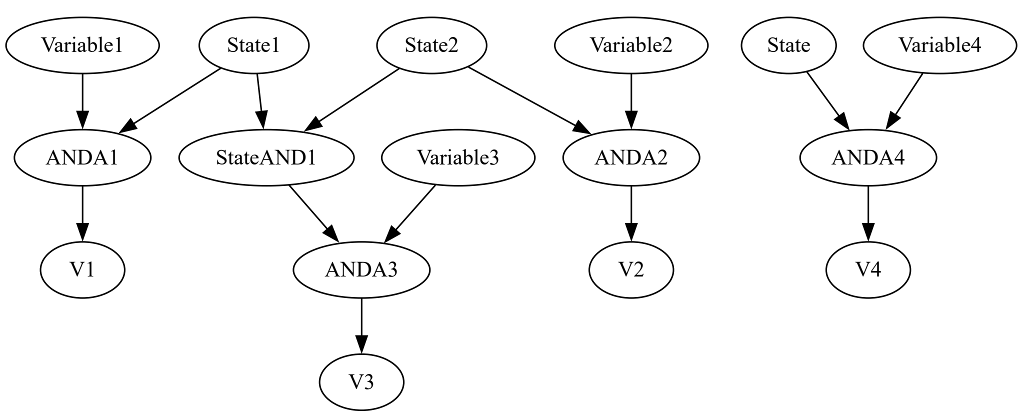
Think of a generalized observation as the action of associatively linking a time node with multiple observation nodes observed at that time. In other words, a group of multi-bit values and time nodes is generated.



Form associations between time and state

1.7.2 Autonomous generation of logical operations between states

X1, X2, X3, and X4 are the observed results at a certain time and are Bool values. Each observation result differs depending on the time.



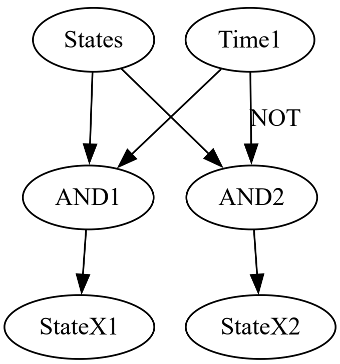
Associate with time passage and state

Propagation is performed again from the observations from V_1 to V_4 for the network of pre-formed ordered circuits.

Feedback is generated from the probability collision between State and V1 to generate StateAND1, and State is split into State1 and State2 to become the input condition for StateAND1. As a result, the following logical equation is formed autonomously.

$$V_3 = V_1 and V_2$$

1.7.3 Forming associations in chronological order



Associations between time variability and state

Before and after the Time1 node, StateX1 and StateX2 are combined in an exclusive and continuous time series by the variation association. This results in an association between StateX1 and StateX2 at adjacent times. Furthermore, the link from States to AND2 is weak and requires a condition, which forms a logical expression between States to form an sequencial circuit.

The nodes of the SOL are Bool values in this example, but further abstract numbers, coordinates, character tokens, or any other object can be linked. Interactions such as arithmetic operations between abstract nodes are realized by built-in functions, etc., and the SOL will choose the concatenation of values between built-in functions with probability.

2 Bidirectional logic operations

2.1 Assumptions about real space-time and mappings

1. there are multiple sets that divide space-time, and the inclusion relation between the sets is unknown.
   Each set corresponds to a substance in space-time, etc., and has some extent in space-time direction. The sets can be exclusive, inclusive, or partially overlapping. For all of these relationships, we derive associations by mapping.
2. The sets can be connected as a "map" if they are observed to coincide in probability.
   If one subspace of a real set belongs to another subspace, the other subspace may necessarily coincide with it probabilistically. In this case, an inclusion set including both subsets can be formed as a "mapping" regardless of the overlap relation of the real set itself. This inclusive set is regarded as a "mapping set. We can expect that this "mapping set" expresses what is generally called a causal relation. The mapping source is completely contained in the mapping set, but the mapping destination may contain only a subset with respect to the mapping set. This depends on the definition of "implication".
3. the logical operations between the mappings are the conditions for the mapping to be valid.
   By performing collective logical operations on multiple mappings, the set of results of the logical operations is a subset of the final mapping set. This is the logical operation generalized by the mapping.

We assume that the spacetime of reality is structured in this way. We then aim to reproduce in SOL from observation whatever the structure of the sets and mappings of reality may be.

2.2 Nodes and Links

SOL is a bidirectional network consisting of nodes and links. All of the following types of nodes are joined by links: 1.

Value nodes, which are the entities of the observed values
Joint nodes that equivalently connect links
Logic nodes that indicate logical operations between links 4.
Exclusive node that indicates exclusivity between links 5.
Funcition node for input/output to/from the external world by coupling with links

Value nodes are abstract subsets in space, but actual values such as scalars, vector values, and characters can be mapped one-to-one to nodes.

Boolean nodes perform Bool algebraic operations on input links; there are several types, including AND, OR, and XOR.
Logical operations such as AND, OR, and XOR apply both logical operations and set operations to multiple inputs. Output links not only propagate the results of logical operations, but also indicate that the values and sets of output links are equivalent to each other.

Joint nodes have only outputs, indicating that the values of multiple links are equivalent to the set. The fact that multiple links are equivalent also means that feedback is generated by arrival from multiple links.
Exclusive nodes are nodes that indicate exclusivity between input links. It is almost equivalent to a NOT link between all input nodes but is used for efficiency.

The Function node performs the actual operation or external input/output using the actual values indicated by the Value node. The Value node, such as a numerical value equivalent to the result, is generated each time.

A link consists of a binary pair of probabilities P^{f11},P^{f00} and empirical numbers N^{f11},N^{f00} to be applied when passing through the link.

The attributes of the link will be as follows.

content	**symbol**
forward propagation probability of the link	P^{f11},P^{f00}
forward experience probability of the link	N^{f11},N^{f00}
backward propagation probability of the link	P^{r11},P^{r00}
backward experience probability of the link	N^{r11},N^{r00}

Activation (active state) calculates the propagation probability from the origin through multiple links and logical operations.

2.3 Link Propagation and Logic Operations

Propagation

Link propagation is computed using propagation probabilities P_{11}, P_{00}.

$$P' = P_{11}P + (1 - P_{00})(1 - P)$$

NOT propagation

If the propagation probability of the link is set as follows, the link itself shows the NOT behavior of the logical value.

$P_{11}=0$P_{00}=0$$

AND operation

The results of link propagation are combined by AND logical operation. P_1,P_2... are input and P' is output.

$$P'=P_1P_2P_3...P_n$$

OR operation

The results of link propagation are combined by OR logic operation. Input P_1,P_2... are input and P' is output.

$$P'=1-(1-P_1)(1-P_2)(1-P_3)...(1-P_n)$$

XOR operation

The results of link propagation are combined by XOR logic operation. Input P_1,P_2... are input and P' is output.

$$P'=P_1(1-P_2)+(1-P_1)P_2$$

.

For XOR with 3 or more variables, the operation is applied recursively.

Exclusive operation

The result of link propagation is input to the Exclusive operation. In this case, propagation is performed without applying the operation.

$$P'=P_n$$

Backward propagation

The backward propagation probabilities P_r^{11}, P_r^{00} are defined for each link.

For normal links, this is obtained immediately from the forward propagation probability. Let P^{f11} and P^{f00} be the forward propagation probabilities and P^{r11} and P^{r00} be the reverse propagation probabilities. In this case, the propagation probabilities P^{r11} and P^{r00} in the reverse direction are calculated using the following equations.

$$P^{f11}P^{r11}+(1-P^{f11})(1-P^{r11})=1$$

$$P^{r11}=\frac{P^{f00}}{P^{f11}+P^{f00}-1}$$

$$P^{r00} = \frac{P^{f11}}{P^{f11} + P^{f00} - 1}$$

In the case of back propagation from logical operations such as OR, AND, etc., it is calculated according to Bayes' theorem. In addition to the usual information about the link, the propagation source probability of the link is P and the propagation destination probability is $P^{'}$ \$. If both the propagation source and destination probabilities are not available, the backward propagation probability cannot be calculated.

$$P^{r11} = \frac{PP^{f11}}{P'}$$

$$P^{r00} = \frac{(1-P)P^{f00}}{1-P'}$$

Entropy and Link Determination

Entropy for probability can be calculated from the formula for binary entropy.

$$S = \sum_n P_n log P_n$$

Binarizing this gives.

$$S = PlogP + (1-P)log(1-P)$$

This results in a minimum entropy S for probability P close to 1 or 0 and a maximum entropy S for probability P close to 0.5 The goal of SOL is to make the probability of the link as close to 1 or 0 as possible. In other words, minimizing entropy is one of the goals of SOL's self-organization. To this end, links with high entropy are split up with the insertion of appropriate logical operations to lower their respective entropies.

Feedback on links

Feedback to links separates longitudinally into multiple links for each set that passes through a link. Positive feedbacks with an overall probability approaching 1 or 0 do not require separation. Negative feedback with an overall probability approaching 0.5 is considered to be a mixture of subsets with propagation probabilities of 1 and 0 in the set passing through the link.

For example, if the link was previously passed with probability 1, but this time passed with probability 0, the propagation set with probability 0 that passed is considered to be the subset that separates the link vertically. The means of this vertical separation is the insertion of the following logical operation.

2.4 Vertical partitioning of links and logical operation

insertion

A link that propagates from node to node can be vertically partitioned by splitting the link into subsets of origin and destination nodes, respectively.
The link is partitioned when it is observed that among the sets that pass through the link, there are sets with different propagation probabilities. When the propagated observed probability approaches 0.5, the elements are separated into elements with an observed probability of 1 and elements with an observed probability of 0.
The separated elements form a logical operation with the node using the information of the origin node that indicates the set of elements.

- If P^{11} side is indeterminate and P^{00} side is 1-determinate, insert AND
- If P^{00} side is indeterminate and P^{11} side is 1-determinate, insert OR
- If P^{11} side is indeterminate and P^{00} side is 0-determinate, insert NOT(AND)
- If P^{00} side is indeterminate and P^{11} side is 0-determinate, insert NOT(OR)
- If both P^{11} and P^{00} sides are indeterminate, insert a possible XOR

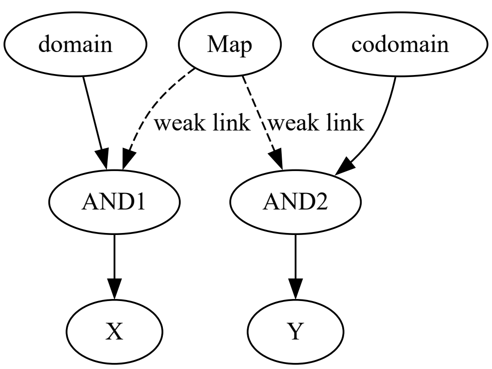
3 Mapping

3.1 Definition of a mapping

Logic operations can be described as set algebra, taking the starting point as a set. But to describe more general-purpose logic, it is essential to have a storage element, such as a flip-flop or a memory, to switch between various states of time. Otherwise, it would not be possible to create general-purpose logic operations for variations in time and space.

SOL has added the concept of mapping to logic operations. A mapping connects different states of time and space and uses them as a new starting point. This mapping action is also the generalization of the storage element. Furthermore, it allows for the management of the coincidence of multiple states as states. This also makes it possible to use the order and other relations between states themselves as states.

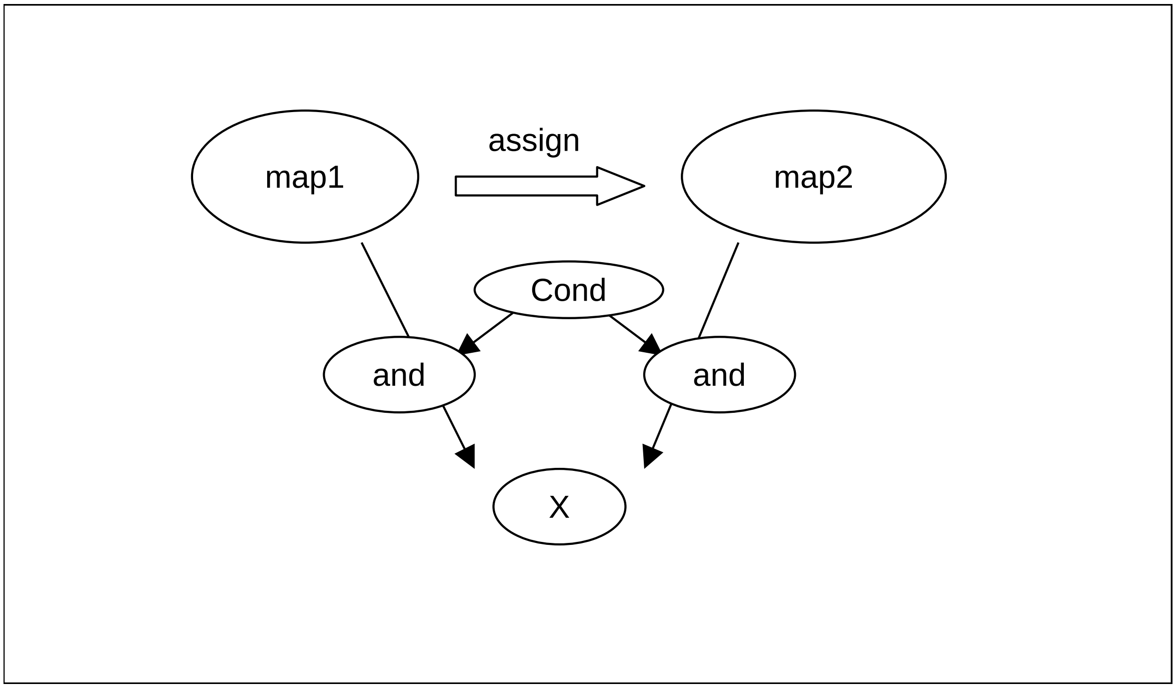
We will show how to express a mapping using only logical operations. In the following example, we map a subset X of the source set to a subset Y of the destination set.



Map with weak links

WeakLink is a link with a small number of observations of N_{11} or N_{00}, and becomes uncertain due to negative feedback from the next observation. Logical operations will be added to separate the definite probability part of this link. A logical operation can be added to either of the two weaklinks, but the logical operation with consistency will be determined.

3.2 Back propagation and substitution through the map



Substitution between maps

This diagram shows how to share the source and destination of a map among multiple maps: from the perspective of the Map1 node, X is the value domain, and from the perspective of the Map2 node, X is the definition domain. Importantly, the Cond node that conditions the value range of the Map1 node is the same as the Cond node that conditions the definition range of the Map2 node.

Here, the active state is propagated from the map1 node to the map2 node; the value of the active state at the map1 node is 1. Propagation from the map1 node to X is normal propagation, while propagation from X to the map2 node is reverse propagation. However, there is no guarantee that the probability expanded at the map2 node will uniformly have a value of 1. But using the idea of active state substitution, I will show why the probability of the propagation set reaching the Map2 node can be uniformly 1.

An active state with probability 1 that exists on the Map1 node propagates from the Map1 node to X. In doing so, it takes AND at node Cond and the propagation set is broken up. However, when it propagates from X to the Map2 node, it goes through the back propagation of AND. At that time, it is complemented by the same Cond node as on the Map1 node side. The active state of the Map2 node cannot be completely specified as either 0 or 1. However, since the active state of the Map2 node is the back propagation of the AND from the value 1, it is possible to assume that the active state of the Map2 node has the value 1 for the entire propagation set. It cannot be assumption that the value is 0.

We can assume that the active state in the Map1 node is identical to the active state in the Map2 node. We consider this to be an substitution in SOL. The conditions for this are as follows

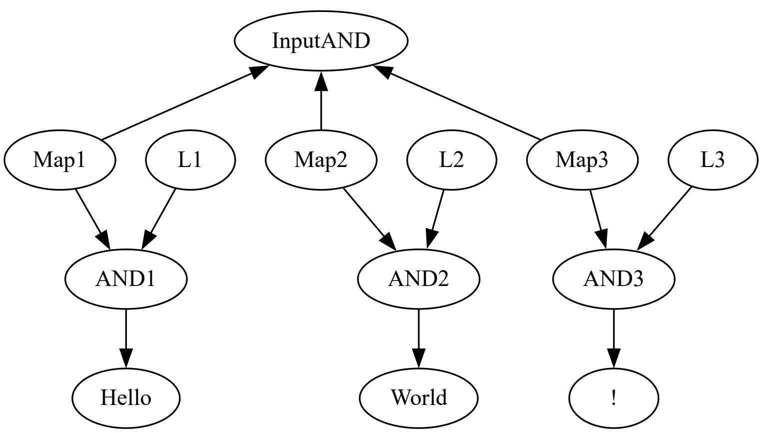
1. there is no contradiction because the value of the source of propagation is equal to the value of the definite part of the destination of propagation.
2. the assumption vectors of the two propagation sets are identical.

Strictly speaking, this idea is Occam's razor. However, the validity of this substitution will be verified through future observations, and additional conditions will be added through feedback as necessary.

3.3 Hierarchical matching by mapping and bidirectional propagation

One example is text analysis from Input input. In the following example, three mutually exclusive words are activated using mutually exclusive nodes L1, L2, and L3. The match decision propagates backwards from the words and InputAND is activated in reverse; L1, L2, and L3 need only be confirmed to be mutually exclusive. Then the back propagation from AND1 to Map1 is collectively expanded to be non-exclusive; ANDs from Map1, 2, and 3 are combined to activate InputAND. Conversely, without all elements, InputAND would not be activated.

Thus, bi-directional propagation allows set-logic operations to be performed between words that do not overlap collectively.

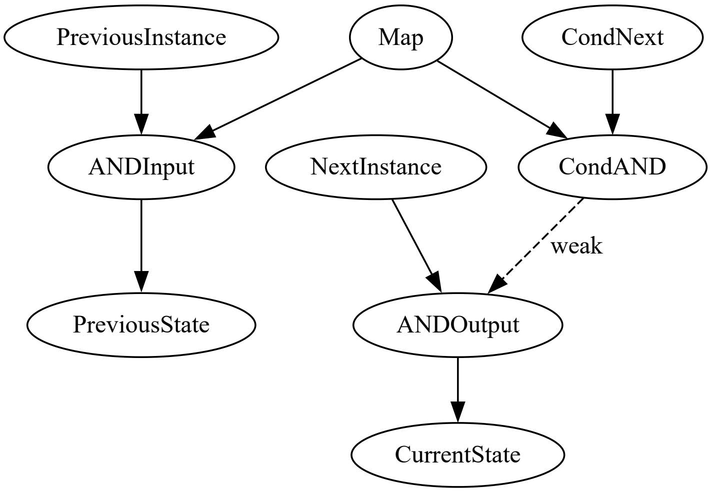


Natural language expression using mapping

3.4 Hierarchy and space-time

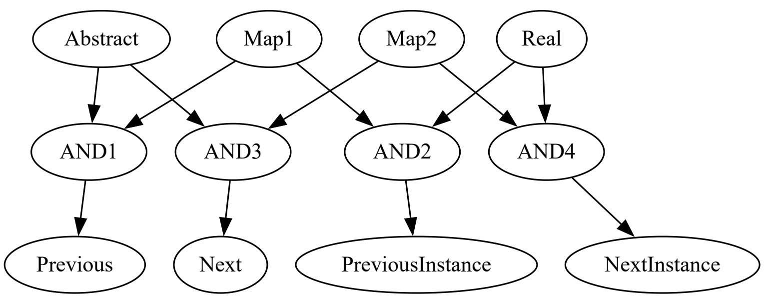
Representation of Ordered Links by Mapping Nodes

A method for expressing order using sets and mappings is shown. PreviousState and CurrentState are continuous time states, and they are connected by a mapping. If there is an order relationship between PreviousState and NextState, CondNext is activated by backpropagation. The NextInstance node and the PreviousInstance node, which are mapping conditions, are both generated for each mapping.



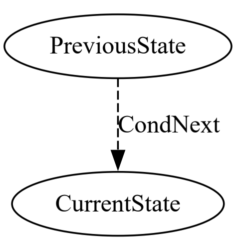
Representing the order between states by mapping

The PreviousInstance node is an abstract subset of the Previous node, but is connected to the Previous by a mapping. Since Next in one mapping can be Previous in another mapping, the PreviousInstance node as a set can also overlap with the NextInstance node in another mapping. Therefore, it is connected by mapping to the Previous node, which exists only in abstract space. The same goes for the NextInstance node.



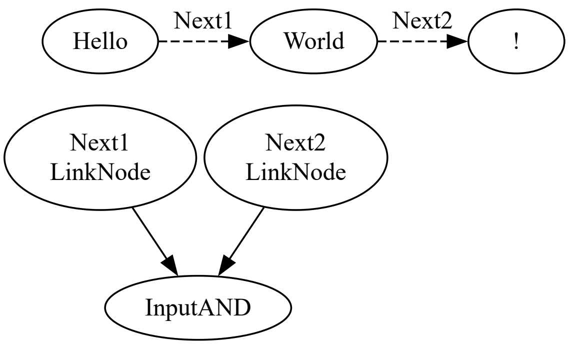
Previous,Next abstraction and instance generation for each mapping

These descriptions may be omitted as necessary and expressed as follows. CondNext is a link node that indicates that there is an order between PreviousState and CurrentState.



Match Determination by Linked Nodes for Words Containing Order

In the example below, three mutually exclusive words are activated using mutually ordered links Next1 and Next2. Next1 and Next2 are also link nodes that are activated when the order between words is established. The match determination is input from each link node Next1 and Next2 to InputAND. Unlike the previous example, if the order of the words is swapped, the link nodes Next1 and Next2 will not be activated, so InputAND will not be activated.



Natural language expression using sequences

This method of expressing ordinals may seem inefficient, but it is a rigorous method that is not dependent on the structure of space-time.

4 Bidirectional propagation of activation values

SOL propagates active states similar to neural networks, but strictly manages the probability of propagation from the starting point. In addition, it manages the information of the propagated set by means of assumption vectors. In addition to this, SOL allows bidirectional propagation of logical operations and the representation of mappings by them. This makes it possible to obtain propagation probabilities between any nodes in the space of sets, as long as they are connected by a network.

For links in the SOL network, the following Activation objects that indicate activation values are distributed and propagated for each link to generate Activations hierarchically. The information held by Activation is basically as follows.

1. Propagated Probability

This is the propagated probability value, and the value is a scalar value from 0 to 1.

2. Assumption Vectors

This is a hypothetical vector indicating the propagation set of this Activation.

3. Propagating link

Indicates the link that this Activation is passing through. Feedback will be applied to this link.

4.Parent activations

Indicates the information of the propagation source Activation. Multiple Activations are synthesized at the logical operation node. Joint nodes inherit only one activation from the propagation source.

4.1 Propagation flow of active states

1. probability propagation from the assumption
   The active state originates from an assumption. The assumption assumes that the starting probability of the node is 1 or 0. The active state is propagated by applying link propagation and logical operations to this starting point probability. 2.
2. logical operations between assumption vectors and integration of multiple assumption vectors
   When performing logical operations using multiple inputs with different assumption vectors, only the overlapping portions of the set indicated by the assumption vectors are extracted and propagated. Logical operations are not applied to the portions of the assumption vectors that do not overlap. 3.
3. back propagation of logical operations and complementary composition of assumption vectors

The set on the output side of the logical operation is propagated back to the input side of the logical operation.
When propagating the value 1 backward from AND, probability 1 and the assumption vector are propagated to all inputs without any change.
When propagating backwards from an AND with a value of 0, the input is fixed as 0 if all the inputs have a value of 1 except for one input.
When propagating backwards from an AND with the value 0, if there are already zeros in more inputs, the input that propagates backwards has an indeterminate probability. However, there is another way to determine the indeterminate probability. This is the reason why back propagation can be used for mapping.

4. Active state collision and element comparison of assumption vectors
When multiple active states pass through different paths and arrive at the same node, if there is a common part between the assumption vectors of both states, the probabilities of both states are compared. If the probabilities are different, feedback is provided to equalize the probabilities. 5.
5. when the active state reaches the built-in function, external observations and external actions are made. An observation is the activation of an output corresponding to a set of inputs.

4.2 Assumption Vector

Assumption Vector Elements

Each element of the assumption vector means an assumption that either the propagation probability P^{11} or P^{00} of a link from a particular node to a node is 1 or 0. By assuming that the value of a node is 1 or 0, the propagation set ahead is divided into two.

1. Origin node

The elements of the assumption vector are basically Boolean values indicating the probability of the node serving as the starting point. This can be regarded as an effect of assuming an implicit link from an implicitly existing node to the origin node. This element is mainly used for mapping conditions.

2. Backpropagation selection

Back-propagate the input links from the activation of value 1 to the OR node and select one of its input nodes. This effect can be viewed as an assumption of a link from the OR node to the input node. The same goes for backpropagation from the activation of value 0 to the AND node. This is also used to substitute values to variables.

3. Common starting point in the middle of the network

When multiple propagations collide, it is wasteful to consider all assumption vectors of the path. Therefore, assumptions are set for a common origin of multiple propagations, and previous common paths are not compared for assumptions.

Interaction of logical operations

When logical operations such as AND and OR are performed between propagation sets, the assumption vectors are synthesized. The synthesis in this case follows the following rules.

- Assumption vector elements that differ among assumption vectors are synthesized as they are.
- Identical assumption vector elements among assumption vectors are combined into one.
- Composition with a inclusive assumption vector element replaces a propagating collectively smaller element.
- Composites of exclusive vector elements result in the propagation set itself becoming an empty set.

Collision of Active Values and Assumption Vector

When multiple active values arrive at the same Joint node or logical operation node, they are considered equivalent and subject to collision. In the case of logical operations, the composite result of the input and the output are collision targets.

If there is overlap in the propagation set indicated by the assumption vector, the probability values of the propagated results must match. If there is no match, feedback is performed.

4.3 Probability Propagation of Activation Values

The basis of SOL is to propagate the activation value (Activation) along the link, propagating probabilities according to the link's propagation probability and the node's probability composition, and then performing the embedded function at the end of the link.

The P^{f11} is the propagation probability from probability 1 of P to probability 1 of P', and P^{f00} is the propagation probability from probability 0 of P to probability 0 of P'. This is a self-evident application of probability theory.

$$P' = PP^{f11} + (1 - P)(1 - P^{f00})$$

.

The AND operation integrates the probabilities of multiple activations and propagates them to the output link. The probability calculation uses the input propagation probabilities P_1 , P_2 , $P_3 \dots$$ are used to calculate the following.

$$P' = P_1 P_2 P_3 \dots$$

The probability calculation for the OR operation is as follows.

$$P' = 1 - \{(1 - P_1)(1 - P_2)(1 - P_3)...\}$$

If the number of output links is huge, the activation is limited to a few links and the activation is propagated. The method of link limitation is provided separately.

4.4 Backward propagation of probability

Activation can also be backward activation, which follows the reverse direction of the link. This is different from back propagation in neural network techniques.

Backward propagation is performed in the same way as the probability calculation of forward propagation, using the back propagation probabilities P^{r11} and P^{r00} for each link. Logical operations on nodes do not apply.

$$P' = PP^{r11} + (1 - P)(1 - P^{r00})$$

As a result, the overall propagation probability is generalized as a polynomial for multiple links as follows.

$$P = f(P_1^{f11}, P_1^{f00}, P_2^{r11}, P_2^{r00}, P_3^{f11}, ...)$$

4.5 Collision of active values

An active value collision is a situation in which multiple propagation paths originating from the same node and reaching the same node have different propagation probabilities despite the overlap of the propagated assumed vectors.

In principle, a partial overlap with a propagation set with $P = 0.5$, i.e., with indeterminate probability, is not a collision because it can be considered irrelevant as a propagation set. Conversely, if the probabilities are both determinate, any overlap in the propagation sets can be considered a probability collision. If the probability 0.5 is completely encompassed by the probability 1 (or 0) side in terms of the propagation set, it is considered a collision because there is a contradiction in the determinacy of the probability 1 side.

The above measures are used to compare the size of the propagation sets, and probability collision and feedback are applied to the part of the propagation set that completely overlaps.

5 Probability Variation Distribution Hierarchy Feedback

This is a newly developed method of feedback that can accurately discover which links are responsible for errors in observed probabilities for an unlimited hierarchy of Bayesian networks.

Conventional neural networks can influence the deeper link layers using backpropagation, but it is still difficult to identify error sources for deeper link hierarchies. Even what is called deep learning is only a relatively deep hierarchy.

The assumptions of this feedback are as follows

1. SOL links have propagation probabilities, and logic operations are also probability calculations. The overall propagation probability of the result of passing through multiple links and logical operations is obtained. Feedback is given to the ΔP_n per link to bring the network propagation probability closer to the new observed propagation probability P'.

$$P' = P_1 P_2 ... (P_n + \Delta P_n) ... P_{x-1} P_x = E_n (P_n + \Delta P_n)$$

2. the probability of occurrence of variation that makes the link of each probability P become probability P' depends on the past observation probability P and the number of observations N, and can be calculated by pure probability theory.
3. the probability of occurrence of variation for the entire link is the product of the probabilities of variation occurrence for all links. This overall variation occurrence probability is maximized.

The number of observations N is added to each individual link for each feedback. The feedback propagation probabilities are used to correct the existing propagation probabilities according to the number of observations. For links with a large number of observations, the correction by feedback will be small.

The calculation method was determined to satisfy the above conditions. This calculation method is called stochastic variable distribution hierarchical feedback.

5.1 Binary pair probability propagation on a link

The propagating active value has a single propagation probability P. This probability P is the probability of being in the set of assumption vectors currently being carried. The probability of being in the complementary set is (1-P).

When the activation value passes through a probabilistically determined link, the probability is carried according to the following propagation probabilities.

$$P^{11} = 1$$

$$P^{00} = 1$$

A probabilistically determined NOT link will swap the probabilities of two sets, the positive set and the complementary set. Specifically, the propagation probabilities of the link are as follows. Note that it does not invert the set of assumption vectors that pass through.

$$P^{11} = 0$$

$$P^{00} = 0$$

Apart from P^{11} and P^{00}, the link has a parameter N^{11} , N^{00}, which indicates the strength of the link. This strength is added to each feedback to establish the probability of the link.

5.2 Bidirectional propagation and probability calculation of logical operations

We show a probability calculation method for activation value propagation in SOL. In general terms, the method of calculating the propagation probability is as follows.

$$P' = PP^{11} + (1 - P)(1 - P^{00})$$

AND operation

$$P_1' = P_1^1 P_1^2 P_1^3 \dots$$

$$P_0' = 1 - P_1'$$

OR operation

$$P_0' = P_0^1 P_0^2 P_0^3 \dots$$

$$P_1' = 1 - P_0'$$

The backpropagation probability of a link is calculated from the forward probability of a link by applying Bayes' theorem. P_1,P_0 is the probability of node A. P^{f11},P^{f00} are forward propagation probabilities from node A to B. Let P^{r11},P_{r00} be the required backward propagation probability from node B to node A.

$$\{P^{f11} + (1 - P)(1 - P^{f00})\}P^{r11} = P_1 P^{f11}$$

$$P^{r11} = \frac{P^{f11}}{P^{f11} + P_0(1 - P^{f00})}$$

$$\{P^{f11}(1 - P^{f11}) + (1 - P)P^{f00}\}P^{r00} = (1 - P)P^{f00}$$

$$P^{r00} = \frac{1 - P^{f00}}{P^{f11}(1 - Pf^{11}) + (1 - P)P^{f00}}$$

Backward propagation from logical operations such as AND operations and OR operations to inputs is also calculated using Bayes' theorem. However, it is necessary to use the observation probability P(A) of the link input and the observation probability P(B) of the logical operation output.

$$P^{r11} = \frac{P(A)P^{f11}}{P(B)}$$

$$P^{r00} = \frac{(1 - P(A))P^{f00}}{1 - P(B)}$$

As described above, the propagation probability synthesis for each node in SOL is faithful to probability theory and is self-evident. Nonlinear elements such as sigmoid functions are not used.

5.3 What does feedback apply to?

The feedback is to adjust P_{total}, calculated with the propagation probability of the entire network currently in use, to the propagation probability P'_{total} of the colliding party with different probabilities in the same situation. The SOL feedback is applied to P^{11} , P^{00} each of the links traversed by the two colliding paths. Its strength depends on the probability of the link and the number of experiences of the link.

The resulting $P^{11'}$, $P^{10'}$ that are fed back are corrected using the correction values ΔP_n^{11} , ΔP_n^{00}. The overall probabilities of the two paths are calculated by propagating them as polynomials, each with an additional correction value. The large equation is.

$$P_n^{11'} = P_n^{11} + \Delta P_n^{11}$$

$$P_n^{00'} = P_n^{00} + \Delta P_n^{00}$$

$$P_{total} = (P_1^{11'})(P_2^{11'})(1 - P_3^{11'})...$$

The overall probabilities of the two paths A and B, respectively, are propagation polynomials for a large amount of ΔP_n^{11} and ΔP_n^{00}. Furthermore, from the equality of these two path expressions,

$$P_{totalA} - P_{totalB} = \Delta P_{total} = 0$$

We can find N of ΔP_n that satisfy this equation and have the minimum probability of variation. However, in a proper way, this would be a combinatorial explosion problem, in which each ΔP_n is varied to find a solution.

Therefore, we provide a means to approximate the variation of each probability observation probability ΔP_n.

5.4 Calculation of effect coefficient E_n

The following is an expression that shows how much the variation in link propagation probability ΔP_n affects the final probability P_n. The probability acting on the probability difference ΔP_n is E_n, and R_n is a constant term.

$$P_n = R_n + E_n \Delta P_n$$

This effect coefficient changes as further links and logical operations are applied. For this purpose, we will show a formula that inputs E_n and R_n and outputs E_n' and R_n'. The link propagation probabilities are P^{11} and P^{00}, respectively.

When starting from the P^{11} side of a certain link, the formula is as follows. Let P_{n-1} be the propagation probability up to the link.

$$R_n + E_n \Delta P^{11} = (1 - P^{00})(1 - P) + P_{n-1} \Delta P^{11}$$

$$R_n = (1 - P^{00})(1 - P)$$

$$E_n = P_{n-1}$$

When starting from the P^{00} side of the link, the formula is as follows.

$$R_n + E_n \Delta P^{00} = P^{11} P_{n-1} + 1 - P_{n-1} + (P_{n-1} - 1) \Delta P^{00}$$

$$R_n = P^{11} P_{n-1} + 1 - P_{n-1}$$

$$E_n = P_{n-1} - 1$$

Propagation through links that are not the origin is as follows.

$$R_n' + E_n' \Delta P_n = 1 - P^{00} + (P^{11} + P^{00} - 1) Rn + (P^{11} + P^{00} - 1)(E_n \Delta P_n)$$

$R_n' = 1 - P^{00} + (P^{11} + P^{00} - 1)Rn$$

$$E_n' = (P^{11} + P^{00} - 1) E_n$$

Below, propagation probabilities for logical operations are determined. Hereinafter, P^{11} and P^{00} are regarded as the m-th and n-th links, respectively, and are expressed as P_m and P_n. Let P be the composite probability of inputs other than ΔP_n. E'_n in AND operation is

$$R'_n + E'_n \Delta P_n = PR_n + PE_n \Delta P_n$$

$$R'_n = PR_n$$

$$E'_n = PE_n$$

E'_n in OR operation is

$$R'_n + E'_n \Delta P_n = P + R_n - PR_n + (1-P)E_n \Delta P_n$$

$$R'_n = P + R_n - PR_n$$

$$E'_n = (1-P)E_n$$

E'_n in XOR operation is

$$R'_n + E'_n \Delta P_n = P + R_n - 2PR_n + (1-2P)E_n \Delta P_n$$

$$R'_n = P + R_n - 2PR_n$$

$$E'_n = (1-2P)E_n$$

If it is back propagation from a logical operation, P^{r11} and P^{r00} are used regardless of the operation. The formula is as follows.

$$R'_n = 1 - P^{r00} + (P^{r11} + P^{r00} - 1)Rn$$

$$E'_n = (P^{r11} + P^{r00} - 1)E_n$$

In the back propagation of the AND operation, if the probability of other inputs is determined to be P_f, the formula is as follows. The derivation is complicated, so we omit it. The quadratic and subsequent terms of ΔP_n are omitted.

$$P' = P_f P_r = (R + E\Delta P_n)(R' + E'\Delta P_n)$$

$$R' + E'\Delta P_n = \frac{P'}{R} - \frac{P'E}{R^2}\Delta P_n - \frac{P'E^2}{R^3}(\Delta P_n)^2 ...$$

$$R' = \frac{P'}{R}$$

$$E' = \frac{P'E}{R^2}$$

The back propagation of the OR operation is

$$R' = \frac{1 - P'}{1 - R}$$

$$E' = \frac{(1 - P')E}{(1 - R)^2}$$

The back propagation of the XOR operation is

$$R' = \frac{P' - R}{1 - 2R}$$

$$E' = \frac{(2P' - 1)E}{(1 - 2R)^2}$$

By continuously applying the above equation to the routes of the network, the propagation probability of the entire route can be determined. We assume that this propagation probability is equal to the observation result P'_{total}. All changes in multiple links can be summed. Since the probability is less than 1, the product of probabilities definitely converges, so the term of the product of two or more ΔP_n is approximately omitted.

$$P'_{total} = R_{total} + \sum_n E_{ntotal} \Delta P_n$$

As a result, the final action probability E_{ntotal} obtained through propagation becomes the effect coefficient on each link ΔP_n.

$$\sum_n E_{ntotal} \Delta P_n \approx P'_{total} - P_{total}$$

5.5 Derivation of the formula for the weight value w_n.

The probability correction value ΔP_n that is fed back to each of the links passed in the propagation is obtained by rewriting and calculating all the probabilities of the propagation links in the following form, respectively.

$$P'_n = P_n + \Delta P_n$$

For the propagation probability P_n of link n, the probability of occurrence of variation t_n for which the resulting propagation probability P'_n is observed follows a kind of binomial distribution and is obtained by the following formula.

$$t_n = \lim_{m \to \infty} \{ \binom{m}{mP'_n} P^{mP'_n} (1 - P_n)^{m(1 - P'_n)} \}^{-m}$$

This formula, in concrete terms, uses m coin tosses with probability P_n to find the probability that the sum of the outcomes is mP'_n and takes the limit for m. The following combination equation is used.

$$\binom{m}{mP'_n} = \frac{m!}{mP'_n!m(1-P'_n)!}$$

Furthermore, multiplying by the number of previous observations of the link N_n, we obtain the probability that the result is $mN_nP\,'_n$. This is the probability of occurrence of variation in the link with probability P_n observed more than once.

$$T_n = \lim_{m\to\infty}\{\binom{mN_n}{mN_nP'_n} P^{mN_nP'_n}(1-P_n)^{mN_n(1-P'_n)}\}^{-m}$$

First, the following "Stirling's approximation formula" is used to calculate the value of the combination.

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\{1-\frac{1}{12n}+\frac{1}{288n^2}+...\}$$

The last constant other than 1 is omitted as an approximation, and the following equation is used

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

Substituting into the combination equation using this,

$$\binom{mN_n}{mN_nP'_n} = \frac{mN_n!}{(mN_nP'_n)!\{mN_n(1-P'_n)\}!}$$

$$\sim \frac{\sqrt{2\pi mN_n}}{\sqrt{2\pi mN_n 2\pi mN_nP'_n(1-P'_n)}}\left(\frac{N_n}{e}\right)^{mN_n})^{-mN_nP'_n}\left(\frac{N_n(1-P'_n)}{e}\right)^{-mN_n(1-P'_n)}$$

$$= \frac{1}{\sqrt{2\pi mN_nP'_n(1-P'_n)}}\frac{1}{P_n'^{mN_nP'_n}(1-P'_n)^{mN_n(1-P'_n)}}$$

Using the value of this combination, substitute T_n.

$$T_n = \lim_{m\to\infty}\{P_n^{mN_nP'_n}(1-P_n)^{mN_n(1-P'_n)}P_n'^{-mN_nP'_n}(1-P'_n)^{-mN_n(1-P'_n)}$$

$$\{2\pi mN_nP'_n(1-P'_n)\}^{-1/2}\}^{-m}$$

$$= P_n^{N_nP'_n}(1-P_n)^{N_n(1-P'_n)}P_n'^{-N_nP'_n}(1-P'_n)^{-N_n(1-P'_n)}\lim_{m\to\infty}\{2\pi mN_nP'_n(1-P'_n)\}^{-m/2}$$

This T_n is the probability that P'_n is observed in the link, and we can find the variation of P_n to average this probability variation over all links. Take both sides log and replace the left side by τ_n. The last term converges in the limit of m, so we omit it.

$$\tau_n = \log T_n = N_nP'_n\log P_n + N_n(1-P'_n)\log(1-P_n)$$

$$-N_n P_n' \log P_n' - N_(1 - P_n') \log(1 - P_n')$$

$$= N_n\{P_n' \log P_n + (1 - P_n') \log(1 - P_n) - P_n' \log P_n' - (1 - P_n') \log(1 - P_n')\}$$

Differentiate by P_n for this τ_n expression.

$$\frac{d\tau}{dP_n} = N_n\{\frac{P_n'}{P_n} - \frac{1 - P_n'}{1 - P_n}\}$$

$$= N_n\{\frac{P_n'(1 - P_n) - (1 - P_n')P_n}{P_n(1 - P_n)}\}$$

$$= N_n \frac{P_n' - P_n}{P_n(1 - P_n)}$$

$$P_n' = P_n + \frac{P_n(1 - P_n)}{N_n} \frac{d\tau_n}{dP_n}$$

As a result, the weight w_n of the fluctuation of P_n with τ as a parameter can be calculated.

$$\Delta P_n = w_n \Delta\tau_n$$

$$w_n = \frac{P_n(1 - P_n)}{N_n}$$

Thus, the weight w_n of each link was determined by the known probability P_n of the link and the number of observations N_n.

5.6 Calculating the probability correction value ΔP_n.

We now describe the feedback distribution method for multiple links. Let $P^{'}_{total}$ be the overall observed probability. The variation ΔP_n of each link is calculated to match this probability. The

$$E_n$$

are the effect coefficients determined by the propagation probability for link n, as described above.

$$P^{'}_{total} = P_{total} + \Delta P_{total} = P_{total} + \sum_n E_n \Delta P_n$$

Yet another constraint is to maximize the overall variation occurrence probability T.

$$t_{all} = \prod_n t_n$$

This equation, using τ_n, is as follows.

$$T = \log t_{all} = \sum_n \log t_n = \sum_n \tau_n$$

The purpose of the feedback is to determine the value of variation of P_n per link so that T, which is the log of this overall variation occurrence probability, is maximized and the value of τ_n per link is distributed evenly.

$$\Delta P_{total} = \sum_n E_n w_n \frac{d\tau_n}{dP_n}$$

To do so, we replace the probability of occurrence of variation τ_n of each link n by the common parameter τ. In other words, an equal distribution is made according to the weights.

$$\Delta\tau = \Delta\tau_n$$

For each ΔP_n, the entire variation ΔP_{total} is distributed using the ratio $E_n w_n$.

$$\Delta P_{total} = \sum_n E_n w_n \Delta\tau$$

$$\Delta\tau = \frac{\Delta P_{total}}{\sum_n E_n w_n}$$

Using $\Delta\tau$ thus obtained, the overall feedback ΔP_{total} can be distributed for each link's ΔP_n.

$$P'_n = P_n + \Delta P_n = P_n + w_n \Delta\tau$$

It is possible for P'_n to be more than 1 or less than 0. In such cases, P'_n is saturated to 1 or 0 and P_{total} is recalculated each time. After that, remove the saturated links and calculate $\Delta\tau$ again.

5.7 Propagation probability correction of the link using feedback observation number N.

This feedback-corrected probability is not the probability that the link itself will update as it is. The larger the number of previous observations N_n of the link, the smaller the link probability correction. The additive value of the feedback is not 1, but is determined by the effect coefficient E_n used for the actual propagation and the number of observations N_f of the partner being fed back.

$$N'_n = N_n + N_f E_n$$

As a result, the final probability of update P''_n is as follows.

$$P_n''^{N_n'} = P_n^{N_n} P_n'^{N_f E_n}$$

Since the multiplicative and additive averages are not so different, the following equation can also be used to approximate them.

$$P_n'' = \frac{N_n P_n + N_f E_n P_n'}{N_n'}$$

P_n is a mixture of links P^{11}, P^{00} and feedback is applied to each of the links P^{11}, P^{00}.

Based on the feedback results to the calculated probability P_n^{11}, links for which P_n^{11} approaches the direction of entropy decrease, i.e., probability is either 1 or 0, are considered to have received positive feedback. Conversely, a link for which P_n^{11} approaches the direction of increasing entropy, i.e., probability 0.5, is considered to have been negatively fed back. We consider negative feedback to be caused by some unseen condition on that link, and search for that condition. Feedback is calculated for P_n^{00} in the same way.

If the link has a frequency of use of 0, it is considered to have probability $P_n = 0.5$. Positive feedback alone can bring us closer to probability 1 or 0, but not to probability 1 or 0 itself.

This conditional link formation is implemented as part of the autonomous generative mapping logic circuit algorithm described below.

6 Associations

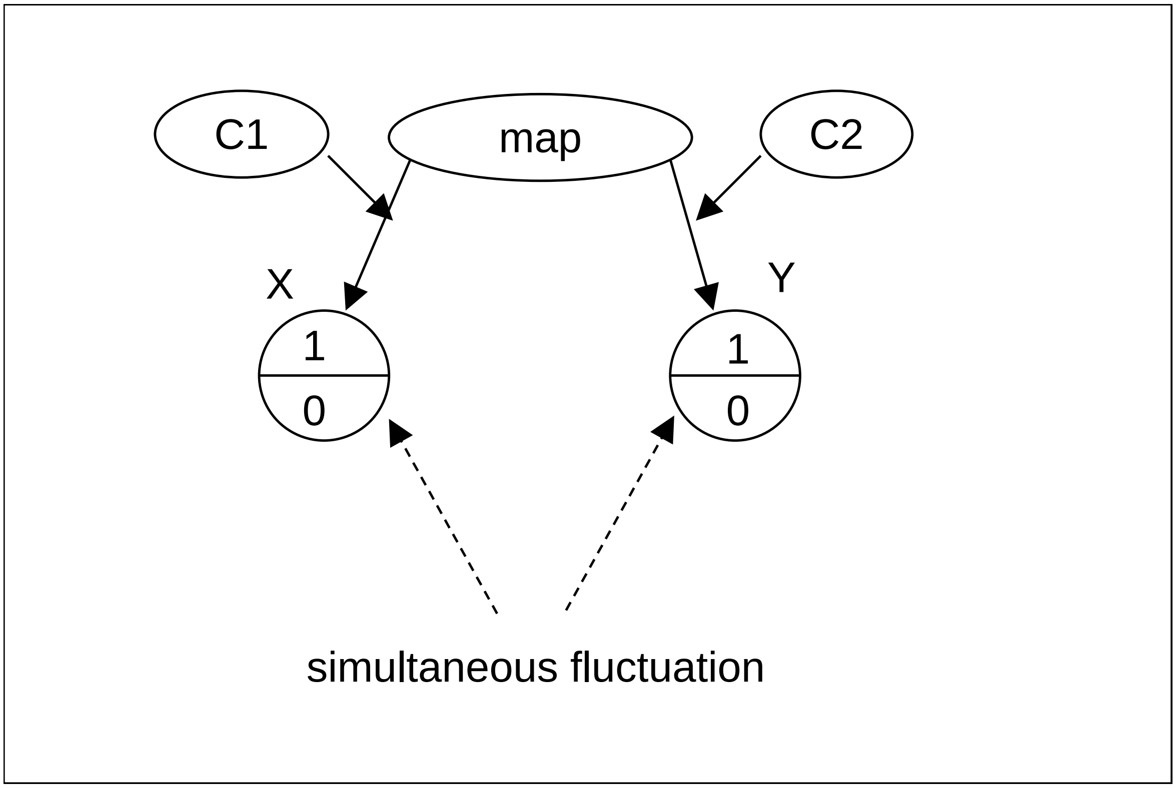
6.1 Associative Target Selection

Association is the action of predicting a causal relationship between two collectively independent nodes by connecting them with a map. This is because the most efficient way to derive possible causal associations between all events is to select only those that vary at the same time. In this case, however, variation can be any variation in time axis, coordinates, etc.
The simultaneous observation of value fluctuations at two nodes makes them candidates for associations. If all variations are observed at the same time, association is assured, but even if all variations are not observed, partial observation alone is a possibility for association. Specifically, the probability of simultaneous variation, or the value of the association, can be quantitatively determined by the probability of occurrence of variation at each node.

The conditions for this are as follows

1. simultaneously observed probability entropies are approximately equal (in most cases, a definite probability of either probability 1 or 0)
2. the assumption vectors from the same starting point are identical or have an inclusion relationship. Simultaneous variation is itself considered to be a case where the assumption vectors are exactly the same or there is an inclusion relationship.

Here, it is possible for unrelated events to be simultaneously activated by chance and become targets of association, but chance associations are fed back by multiple observations to become indeterminate links.



Create association

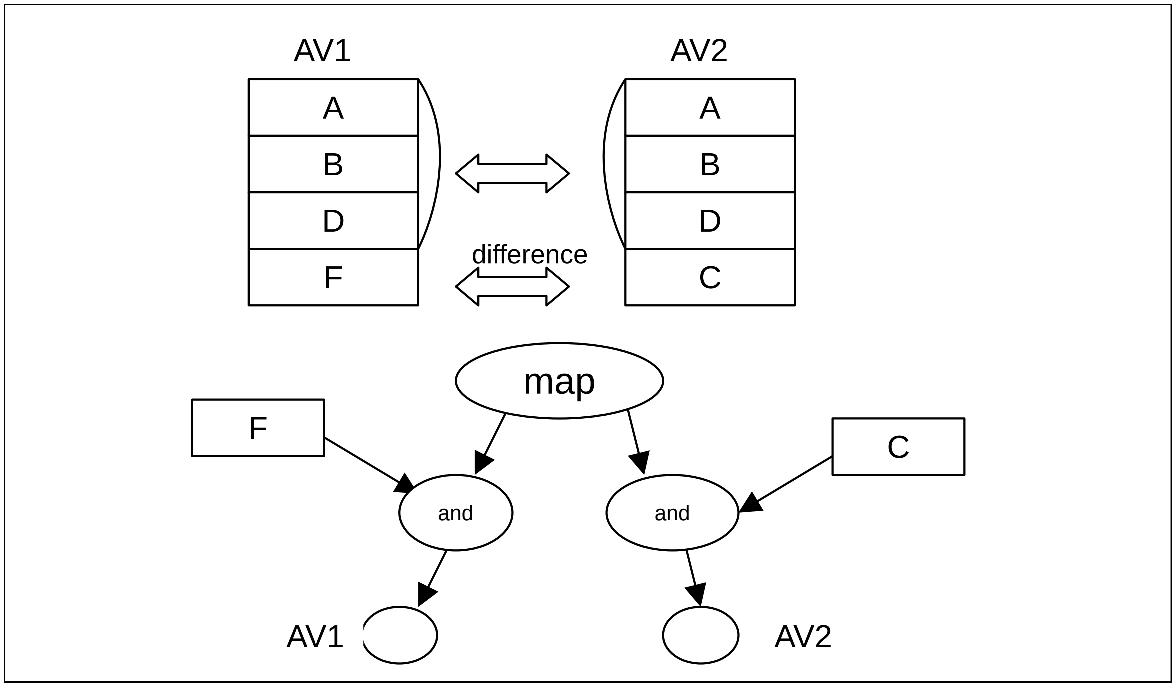
Candidate associations from feedback

A candidate node for association is the endpoint of a link to which negative feedback has been applied. The fact that negative feedback has been applied can be regarded as adding some conditions to the link.
Links to which this negative feedback is applied at the same time are considered to be potentially associative. However, there can be a large number of nodes that are simply negative feedback candidates. It would be very inefficient to make associations with all of them. Then, how to accurately select only meaningful associations? As a method to achieve this, the highest priority is given to selecting associations that have a low probability of node change and for which variation is observed at the same time.

Candidate associations based on observations as a typical example

An observation is the activation of a subset of nodes that represent the observed state. This activation is typically the action of mapping the current time-space to a node indicating the observed value, and is basically generated for each observation.

6.2 Association Formation and Addition of Conditions



Difference between assumption vectors

The associations need to select or generate two nodes that are the conditions for doing and from the map. These condition nodes are generated from the difference of the assumption vectors. In the figure above, the condition is the difference between AV1 and AV2, node F and node C, respectively.

6.3 Calculation of propagation probability and number of experiences

An associative link is not a definite link, and the number of experiences \$$N$ of its associations can be determined from the observed probability of simultaneous activation.

$$N^{11} = log_2(P_Y)$$

The method for this calculation is as follows. Consider the past propagation probability P_Y of the link to the associative target Y as a result of $N^{11} observations of$ P=1
$in reality on an unmeasured and uncertain link, i.e., an associative link with an assumed pro$
P=0.5\$.

$$0.5^{N^{11}} = P_Y$$

That is,

$$P = 0.5$$

can be regarded as a link that has been observed N times.

When $X = 1, Y = 1$ is observed, the association probability is $P' = 1$. As a result, the initial value of P^{11}, N^{11} is the following equation: $P = 0$ when X=1,Y=0.

$$N^{11} = log_2(P_Y)$$

$$P^{11} = \frac{0.5 + P'N^{11}}{1 + N^{11}}$$

When X=0,Y=0 is observed, it acts only on the P^{00} side. In this case, the association probability is also $P' = 1$. As a result, the initial state of P^{00}, N^{001} is represented by the following equation: $P' = 0$ when X=0,Y=1.

$$N^{00} = log_2(1 - P_Y)$$

$$P^{00} = \frac{0.5 + P'N^{00}}{1 + N^{00}}$$

The calculation of this empirical number implies that the association of variation between events that rarely occur also has a high degree of certainty of association.

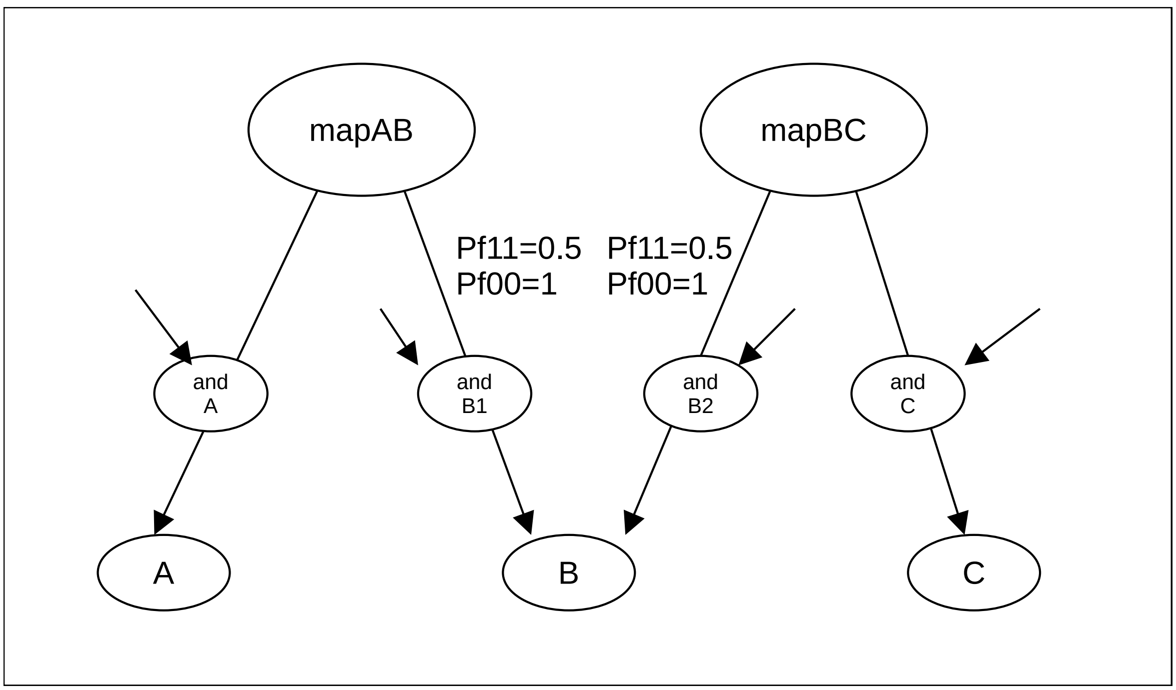
7 Probabilistic Autonomous Logic Generation Algorithm

This algorithm is a general method for autonomously generating and modifying nodes and links.

7.1 "Conditional" Link splitting and logical operation node insertion

Feedback on associative links

The propagation probability of an associative link is uncertain at the time it is formed, and the propagation probability is corrected by feedback. Links with uncertain propagation probabilities are made to have a definite probability of 0 or 1 by adding conditions. We will explain how to select conditions for this purpose.



Feedback to map links

An association is formed from A and C to B respectively. Each association is connected by a Map1 node and a Map2 node.
Then, the propagation probabilities P_{f11} of the two associative links to B are corrected by feedback for the same assumption vector and become uncertain probabilities. In order to correct this propagation probability to a definite value of 1 or 0, a condition is created for the link of B. A logical operation is inserted into one of the two

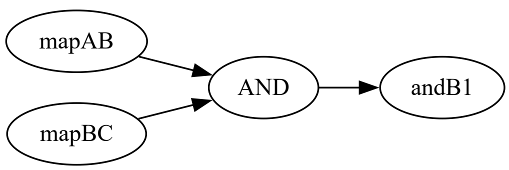
uncertain links, and the other becomes the input for the logical operation that is the condition. No logical operation is formed unless the feedback is for the same assumption vector.

AND node

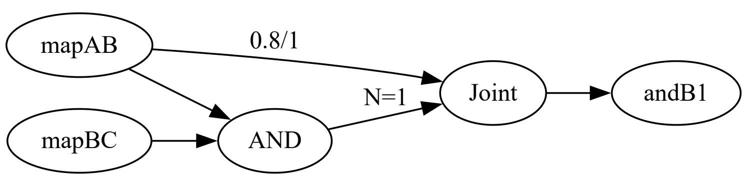
Conditioning is performed for link feedback to correct the link. Assume that the link for which feedback is provided is the link from node A to node B. In the following example, we assume that the propagation probability of P^{11} has been reduced to 0.8 by feedback.

$$A \to B \quad \begin{cases} P^{11} = 1 \to 0.8 \\ P^{00} = 1 \end{cases}$$

Use condition C corresponding to this feedback to add conditions. The condition is AND.



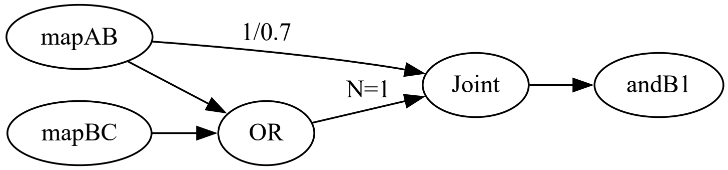
To be precise, condition C is based on one observation and cannot replace all existing links. To do this, we will maintain the existing indeterminate link and connect it with Joint.



OR node

If feedback is made on P_{00}, then the condition node C is ORed together.

$$A \to B \quad \begin{cases} P^{11} = 1 \\ P^{00} = 1 \to 0.7 \end{cases}$$

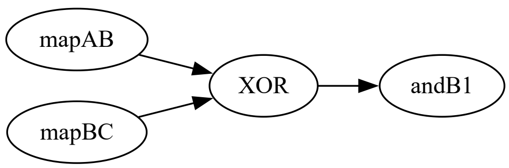


XOR node

If feedback is given to both P_{11} and P_{00}, the link will be random and basically invalid. However, if feedback of P_{11} and P_{00} is performed simultaneously on two links, they become candidates for joining at the XOR node.

$$A \rightarrow B \quad \begin{cases} P^{11} = 1 \rightarrow 0.5 \\ P^{00} = 1 \rightarrow 0.5 \end{cases}$$

$$C \rightarrow B \quad \begin{cases} P^{11} = 1 \rightarrow 0.5 \\ P^{00} = 1 \rightarrow 0.5 \end{cases}$$



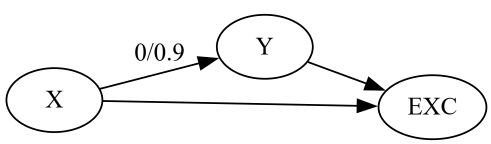
NOT link

If the propagation probabilities of both P_{11} and P_{00} are close to 0, it is a NOT link.

$$X \neg \rightarrow Y \quad \begin{cases} P^{11} = 0.01 \\ P^{00} = 0 \end{cases}$$

Formation of exclusive node

The Exclusive node exhibits the same behavior as XOR in two variables, but differs in that all elements are completely exclusive in multi-variables. The purpose of this exclusive node is to efficiently represent exclusive events that occur frequently in nature, provided that P11=0, but P00 must be greater than or equal to 0 (typically, close to 1).



7.2 "Causal" Link node activation and associative targeting from feedback

When feedback to the link below occurs, a link node is generated from nodes A and B before and after the link.

$$A \rightarrow B$$

Logical operations equivalent to link nodes

As a result of the feedback from A to B, link node L is activated. Rewrite the link from A to B as a logical operation using L. That is, the true identity of link node L is a new input node for the AND operation inserted into the link. If $P_B = 1$ changes to $P_B = 0$ for $P_A = 1$, insert the following logical operation.

B=A AND L

If $P_B = 0$ changes to $P_B = 1$ for $P_A = 0$, insert the following logical operation.

B=A OR L

Furthermore, if feedback of fluctuations for both $P = 1$ and $P = 0$ is observed, a link node using an XOR node can be implemented. The link node and the input of the XOR are connected by a NOT link for convenience.

B= A xor ¬ L

When feedback occurs due to a collision of activation values, it is propagated back to the link node.

From determining link propagation probability to generating link nodes

The probability of back propagation to a link node using AND is as follows. Let P_A and P_B be the probabilities before and after the link, and P_L be the probability of the link node.

$$P_B = P_L P_A$$

The propagation probability to link nodes is as follows. However, please note that it is defined other than $P_A = 0$.

$$P_L = \frac{P_B}{P_A}$$

It must be noted that probability propagation to link nodes cannot propagate fluctuations as they are, unlike forward propagation.
Propagation sets based on assumption vectors are propagated by assignment in a similar way to backpropagation of mappings. The two propagation sets $P = 1$ and $P = 0$ propagated to A are combined and propagated to the link node. Similar to backpropagation of mappings, the propagation probability to the link node when $P = 0$ is indefinite, but the assumption vector propagated to the link node is substituted for links with equal propagation probability. .

Additionally, find the propagation probability from the backward propagation of XOR to the link node. First, use the formula for the logical operation of XOR.

$$P_B = P_L P_A + (1 - P_L)(1 - P_A)$$

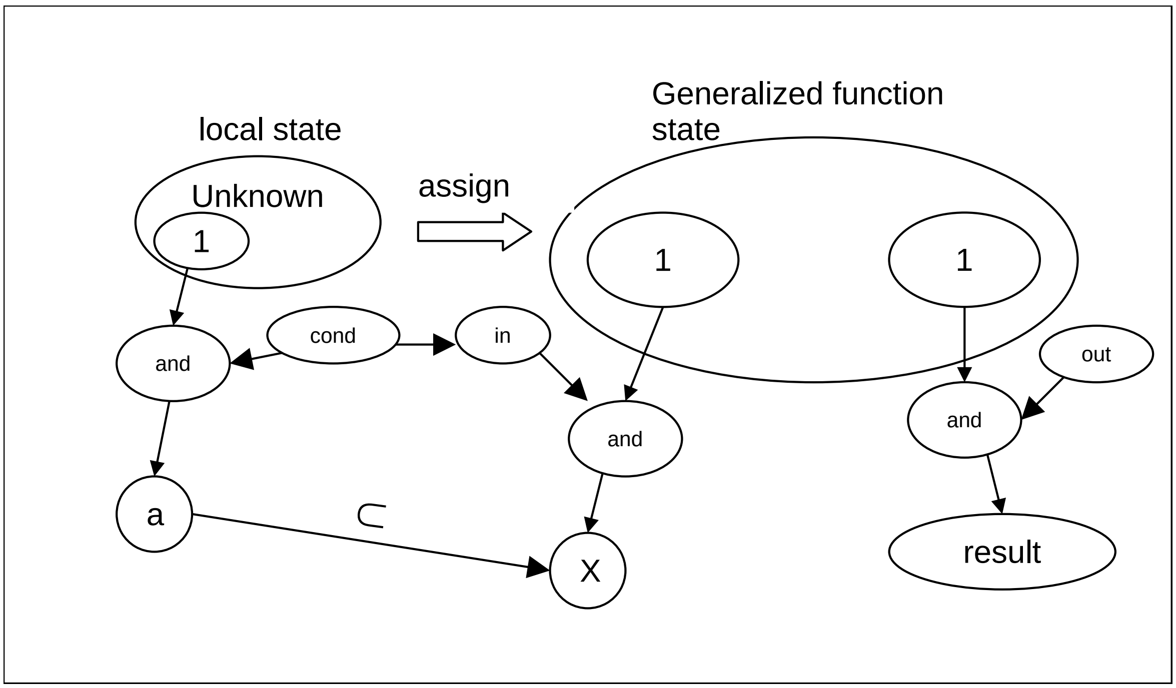
$$P_L = \frac{P_A + P_B - 1}{2P_A - 1}$$

This formula is 1 if the probabilities of P_A and P_B are the same, and 0 if they are complementary (provided that $P_A \neq 0.5$).

Another feature of the XOR link node is that cancellation of hypothetical vector elements is strictly possible. If the complementary propagation sets of $P=1$ and $P=0$ originate from the same assumption vector and the values of the assumption vector elements are complementary, then integration of assumption vector elements becomes possible. .

7.3 "Substitution" Coupling between nodes that are inclusive in terms of the propagation set.

Substitution is the propagation of the entire active value between nodes once it is confirmed that the respective assumption vectors are inclusive for each active value that reaches multiple nodes.



Substitute state into function

In the figure, assignments are made between multiple states. The conditions for this substitution will be explained.

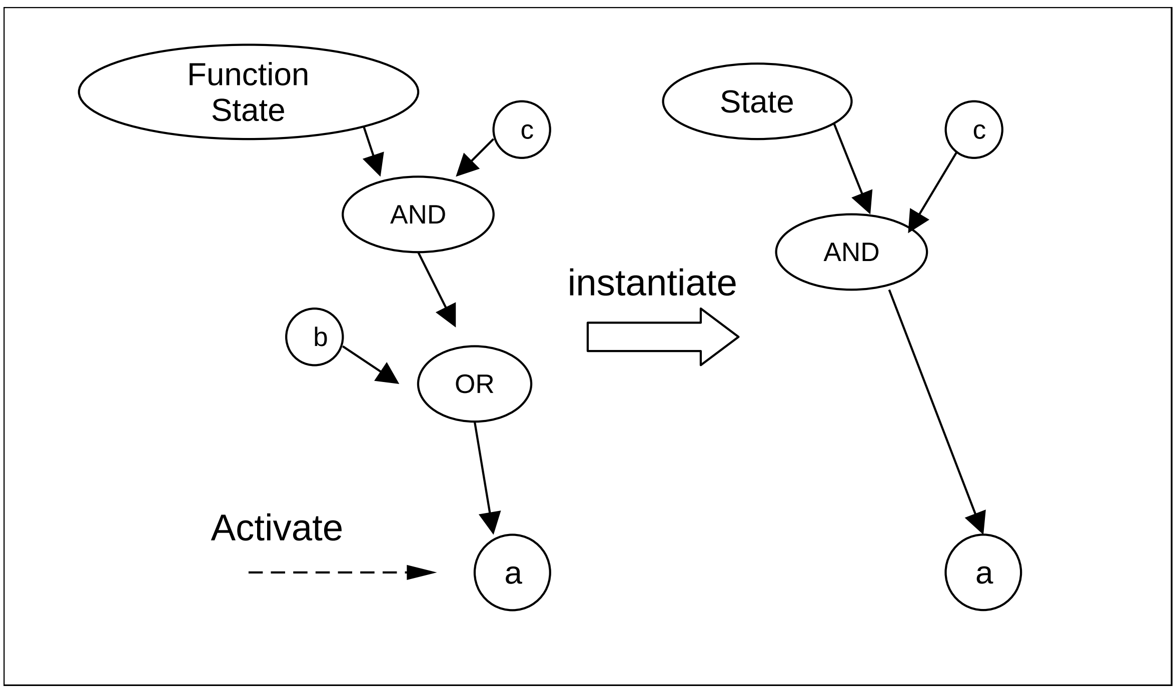
The localstate node contains a partial state for a value, but other parts are indeterminate. Here, since the functionstate node has a set X that includes the value a for the value a of the localstate node, there is no contradiction in assuming that the functionstate node includes the localstate node. As a result, the functionstate node includes the Result node as a partial state.
Although the Result node included as another partial state in the function state node is not present in the localstate node, it is consistent to assume that it also exists in the undefined portion of the localstate node. This Result can be added to the localstate node in the next "instantiation".

7.4 "Instantiation" Instance replication of part of the network by propagation subset

Part of the network is divided vertically, subsets are taken, and a simplified network is generated as an instance. The links generated by instantiation are structurally the same as existing links, but the selection of many link outputs can be omitted.

This instantiation is performed when the propagation probability for the same propagation set is determined between the subset of the origin and the subset of the destination, and the propagation set of the network for that route is increased by OR etc. It is. The entire path between the origin and destination is separated by a propagation set.

It can be used for purposes such as adding the value of the result of a function to the input State.

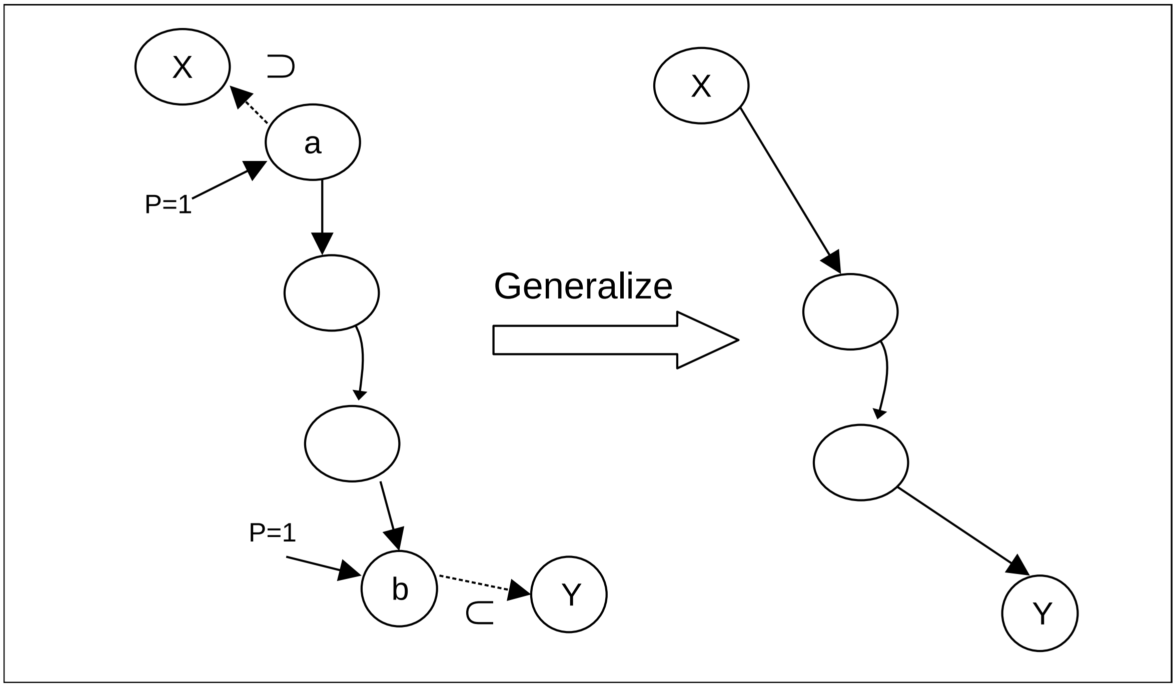


Network instantiation

7.5 "Generalization" Generalized replication of part of the network by propagation subset

A part of the established network is replicated as a more collectively generalized network using the inclusive set at the starting point and the inclusive set at the end point. This is generalization. Similar to instantiation, but performed even when the propagation set of the network is not fully observed. In other words, this is a temporary network that will be modified later with feedback.

This generalization is performed when propagation is established between the starting point subset and the destination subset. At this time, the entire route between the starting point and the ending point is expanded to some extent as a set and separated. In particular, a, which is a subset of X, is replaced with a variable X whose set is larger than a. Similarly, replace b with variable Y.



Network generalization

7.6 "Selection" Selective control of multiple link propagation

A large number of links can be connected to a given node. In particular, for connections between functions, the number can be in the order of 10,000 or more. In order to minimize the search time, it is necessary to selectively control link propagation and propagate only to necessary links. The problem here is that the propagation probability of a large number of links is originally deterministic, and when link control is performed using logical operations, the propagation probability fluctuates. To solve this problem, a link selection node is generated separately from control by logical operations.

Selection of a link activates a link selection node. This activated link selection node and other network states are connected by association between fluctuations. Typically, the operation of a function and the input of the function are connected by a association between fluctuations, and the input link of the function is selected by a link selection node.

The link selection node can also be considered to be associated with a subset of the SOL network itself.

7.7 "Convergence" Link optimization

When positive feedback occurs where two paths and probabilities match, links of paths that are small in terms of propagation set and do not have a large number of observations are deleted. Used for SOL link optimization. This is the opposite effect of "Instantiation" but it is performed on networks that are not used.

```
(A&(B|C)&D)|(A&(B|C)) = (A&(B|C))
```

7.8 "Integration" Integrating logical expression nodes with the same logic

If the nodes before and after a link connected in series are nodes of the same type, such as ANDs, and if the probability of a link between nodes is 1, the nodes are combined. If there are different AND elements, extract and separate only the different elements. Used for optimizing logical networks.

```
A&(B&C)=A&B&C
A|(B|C)=A|B|C
```

Multiple nodes of the same type directly connected by a link from a single node will attempt to integrate if other inputs are nearly the same.

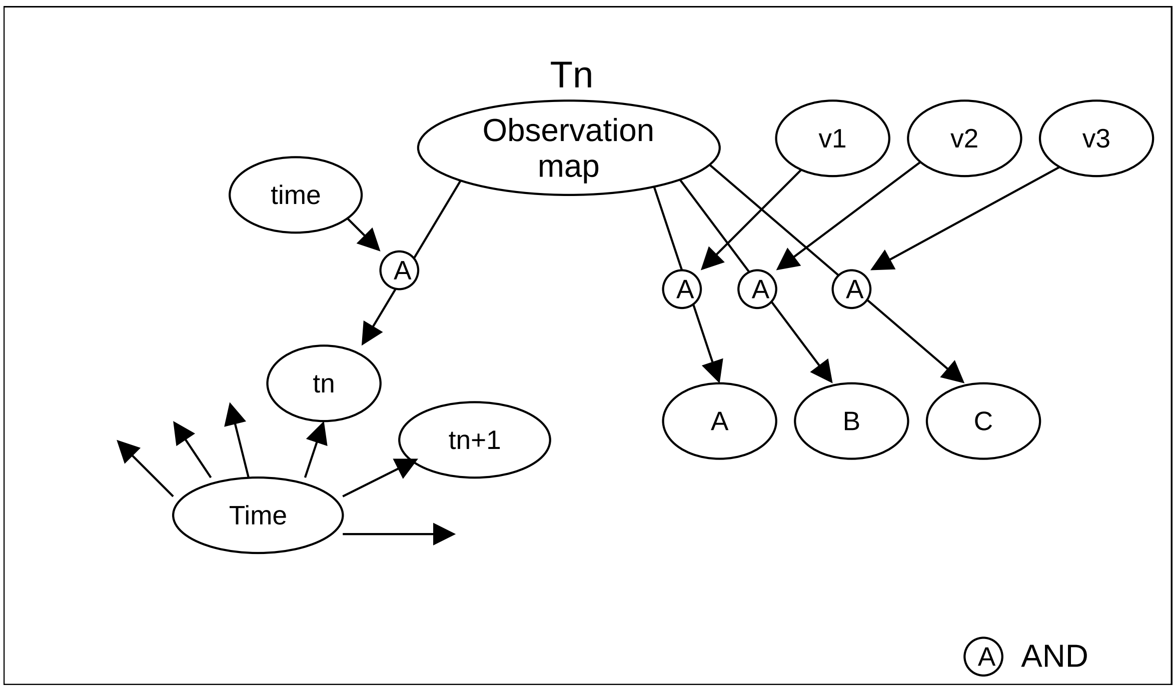
```
(A&B&C),(A&B&C&D)=(A&B&C)&D
```

8 Sequential circuit generation

By hierarchically using the concept of states based on mapping, we autonomously generate circuits between generalized states. Furthermore, by generalizing the state with variables etc., functions that can be used for general purposes are generated.

8.1 Observation function

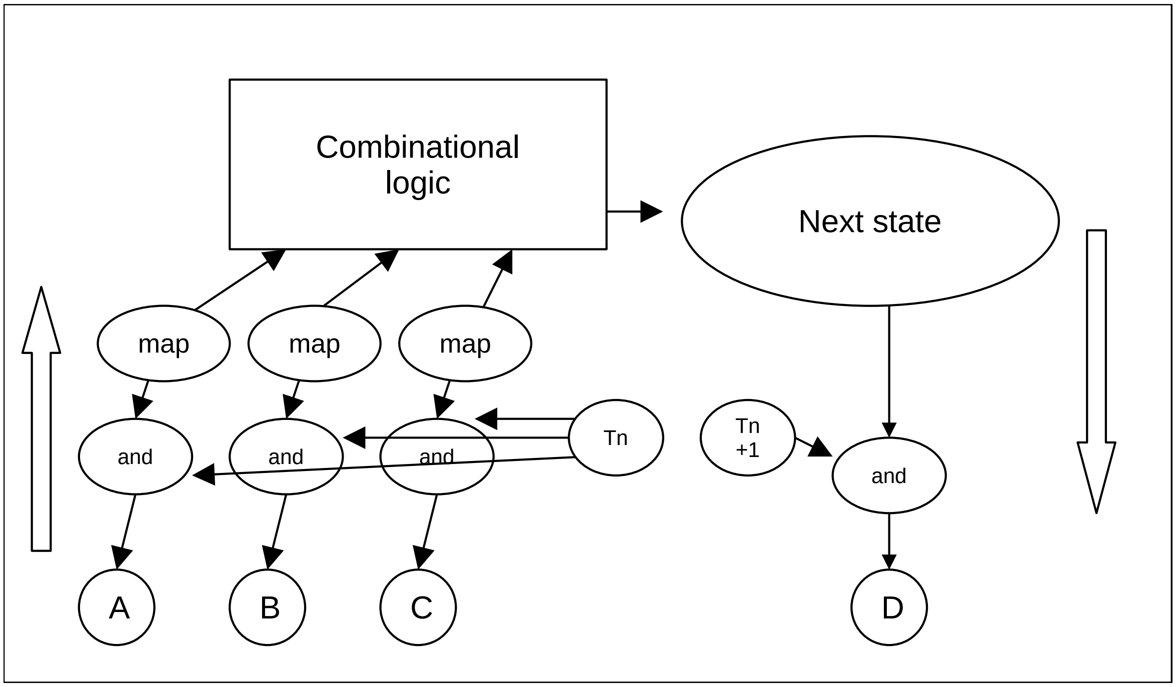
The state observation function forms an association between the observed current time T_n and the simultaneously observed results A, B, C, and D. Results observed simultaneously at the current time are further connected by mapping as a causal relationship, and a logical relationship between them is observed.



Observation

8.2 State transition generation

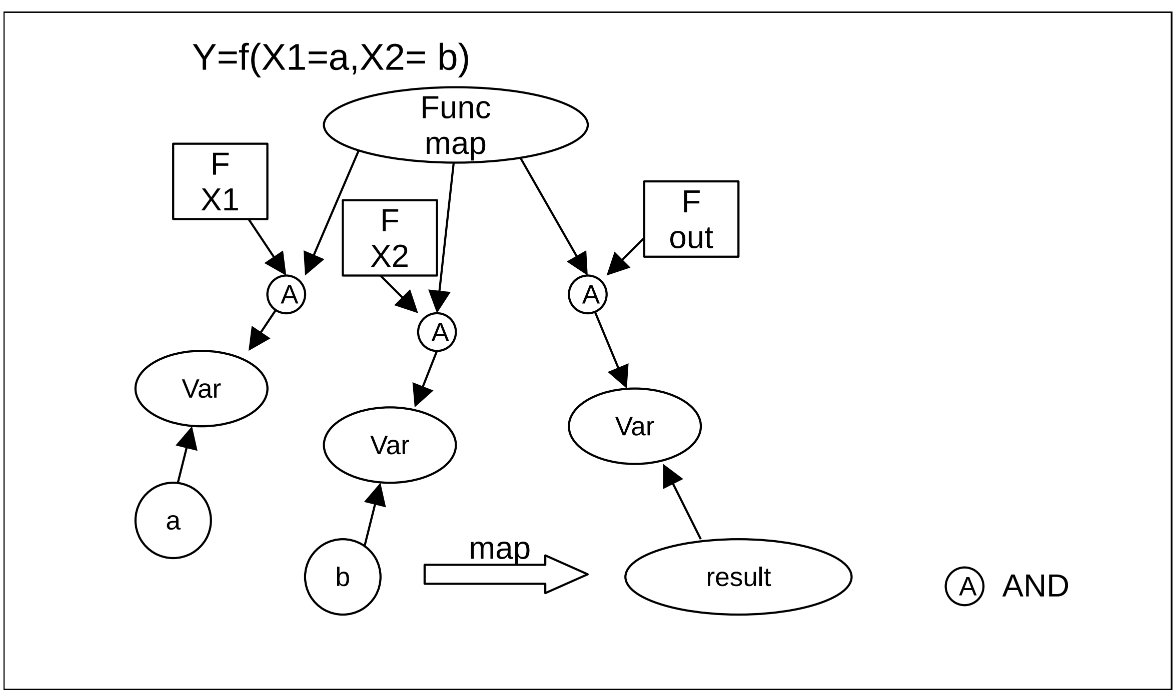
Time-series state transitions are formed from mapping when time is a difference. The mapping of A, B, C to time T_n and time T_{n+1} is fed back to form a logic circuit between A, B, and C.



Sequencial circuit by map

8.3 Generate function from generalization node

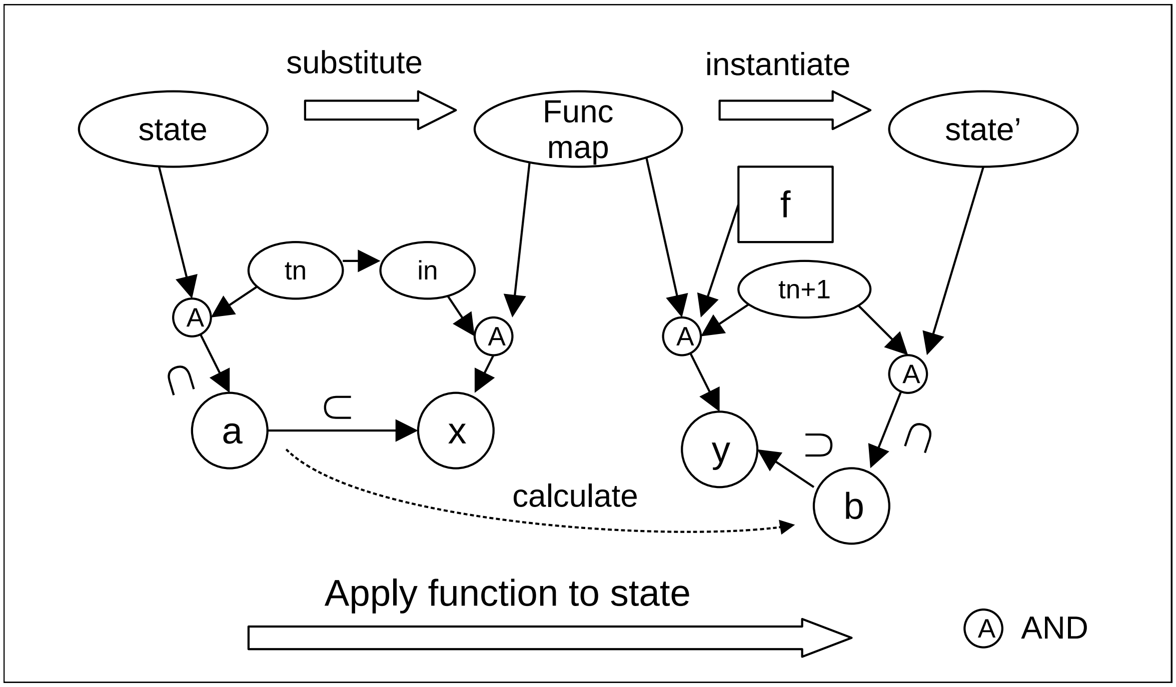
The mapping formed between observed nodes such as a, b, and result is expanded and transformed into an abstract node by the feedback to P^{00} to the link. For example, the value a is "generalized" to become the variable X1. Similarly, the value b is generalized and becomes the variable X2. The association between X1, X2, and out thus generalized is expected to be a function. Note that the relationship between individual a, b, results is expressed by another mapping, and this different mapping is selected by a function.



Function map

8.4 Function application

A function formed by generalization assigns a state with an input value. Since a function is considered to be larger in propagation set than the state to which it is assigned, the value of the result of the function is also considered to be included in the input state. This is the application of a function to a state. The applied result is then partially duplicated and added to the original state by instantiation.

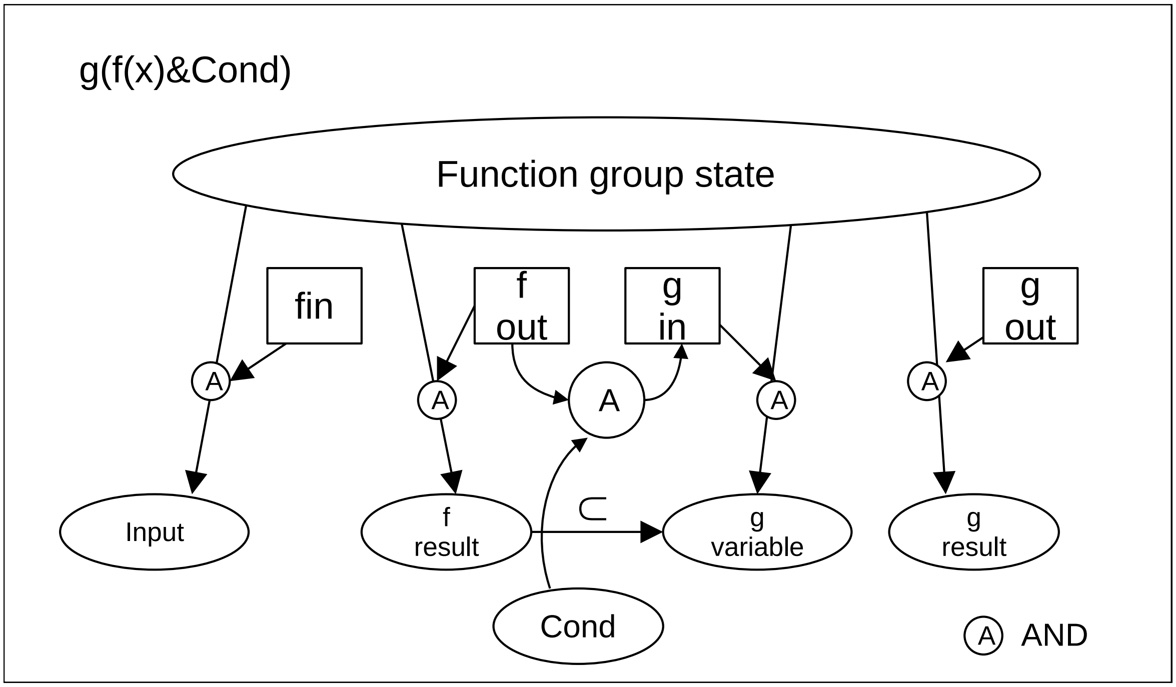


Apply function

The result of the function can also be used as a link node to indicate a match. In other words, it becomes possible to implement conditional judgments that convert matching judgments to Boolean.

8.5 Grouping functions

It shows how to apply the generated functions continuously. In order to continuously apply functions f and g, add a network that connects the output of f and the input of g.

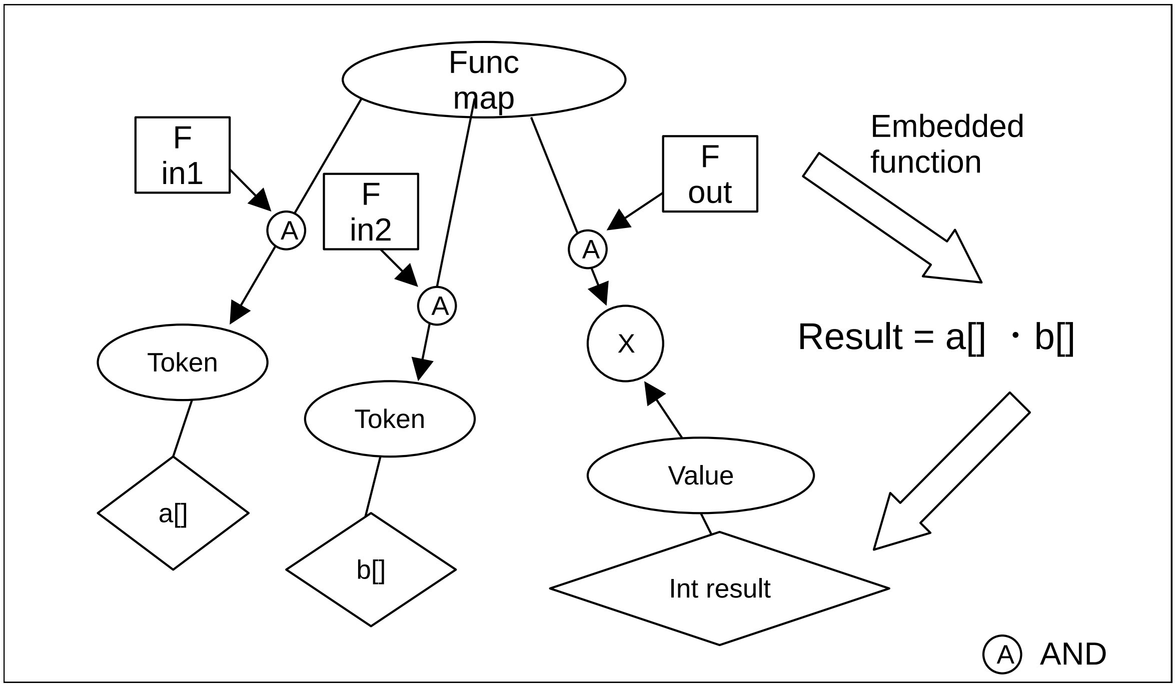


Function group

Insert a condition and an AND node into the output of function f, and connect it to the input of function g, so that the functions are connected depending on the condition. By making this condition the result of another function, it is possible to realize any combination of functions, including recursion.

8.6 Built-in functions and their usage

A built-in function is defined as an external function that activates only an output node for an input node. Actions on input are defined in built-in functions. It can also be used for input/output with the outside of SOL. The nodes used in built-in functions are Boolean values, but built-in functions can retrieve the actual object corresponding to the node. Specifically, built-in functions use scalar values such as Integer, vector values, objects, etc. for calculations. For each result, a node corresponding to the value of the result is created, and a built-in function connects the corresponding entities. In the example below, the real vector value of the character Token is used to calculate the inner product using the built-in function to generate the Result value and generate the corresponding node. [2]



Embedded function

9 conclusion

To integrate and summarize the above contents, this self-organizing logic has the following characteristics.

1. Bidirectional logic realizes the concept of mapping by applying backpropagation to logical operations between sets. Mappings can be used to express relationships between arbitrary sets that do not overlap in space and time. This allows sequential circuits to be described using only logic circuits.
2. By setting the propagation probability of the binary pair P^{11}, P^{00} in the connecting link between logical nodes and enabling bidirectional propagation of the link, bidirectional binary propagation is possible. It becomes a stochastic Bayesian network. Backpropagation is Bayes' theorem itself. Bidirectional propagation through the mapping allows exact propagation probability calculations between any two sets.
3. Collective probability propagation using assumption vectors strictly manages propagation subsets that pass through links and mappings using assumption vectors, and uses partial matches and differences of assumption vectors to generate information between any two related nodes. Set inclusion relationships can be calculated. When multiple probability propagations collide at the same node and both assumption vectors match, feedback is provided to both paths depending on whether the probabilities match or do not match. In particular, when feeding back an uncertain link to a definite value, a link node indicating a match is activated.
4. The bidirectional binomial stochastic Bayesian network automatically accurately feedback-corrects all link propagation probabilities by using a probability fluctuation distribution hierarchical feedback algorithm based on the past propagation probabilities of links and the number of observations.
5. Association probabilistically connects collectively exclusive nodes whose fluctuations are observed at the same time to create a mapping. The object of association is not only the observed value, but also the condition nodes and link nodes of logical operations that occur in feedback. This effect allows associations to be formed between any subset.
6. The probabilistic autonomous logic generation algorithm generates deterministic digital logic with a probability of 100% or 0% by adding logical operations, mappings, and link nodes according to the feedback results to link probabilities connected by associations. will be automatically generated. By using the propagation probability of the pair P^{11}, P^{00}, it is possible to insert logical operations accurately. Inserting an OR node for fluctuations in P^{00} also serves to generalize a constant into a variable.
7. Generate generalized logic circuits between inputs and outputs such as memory by forming associations based on changes in values over time and adding further logical operations to the feedback to the associations. It became possible to do this, and general sequential circuits were formed autonomously. A generalized sequential circuit is used many times and becomes a function.
8. By using mappings hierarchically, it is possible to express hierarchical data structures and functions in general software. By managing the inclusion relationship of the propagation set between these mappings, it

is possible to determine whether parameters match the data structure and reflect the results of the function in the data structure. For this purpose, we use "substitution" and "instantiation" between mappings defined in SOL.

This SOL is built solely on basic mathematical principles such as probability, sets, and mapping. We do not imitate biological neurons. Additionally, they rarely use less rigorous heuristics, such as nonlinear activation functions in neural networks. Therefore, it is closer to a rigorous mathematical theory than general machine learning, and it is almost certain that autonomous generation software is an extension of this SOL. Existing machine learning such as neural networks can even be considered to be an approximate implementation of part of this SOL. Therefore, if we are aiming for accurate and safe machine learning, we have no choice but to introduce this SOL principle in some way.

Future challenges include verifying operation through implementation, increasing implementation efficiency, simplification, and parallelization. Furthermore, whether SOL can comprehensively autonomously generate logic circuits and software will be a future research topic.

The relationship with mathematical logic, category theory, etc. is obvious, so there is room for generalization of the theory. The idea of probability propagation is also strongly related to things such as path integrals in quantum mechanics, and I believe that some form of integration with physics will prove their mutual validity. [3]

Glossary

SOL (Self-organizing Logic)

A system that automatically generates a bidirectional network consisting of nodes corresponding to logic circuits and links with binomial probabilities.

Value node

A value node corresponds to a value that is the starting point of a logical operation. It is activated by an assumption and serves as the starting point of an assumption vector. Back propagation from logical operations connected by links is also a type of assumption.

Logic node

A logic operation is performed on the probabilities of multiple input links, and the resulting probabilities are output to the output links. While performing the logic operation, it also synthesizes the assumption vector.

Joint node

Outputs the inputs of multiple links as they are. No probability operation is performed between inputs.

Function node

Performs an external function on the inputs of a link to activate the result node.

Link

Combines nodes with a binomial set of propagation probabilities. Input probabilities and return probabilities. The reverse probability is also defined separately.

Link node

When the propagation probability of a link changes due to feedback, a node corresponding to the link is generated and the comparison result of the values before and after the link is propagated. Specifically, it corresponds to the input backpropagation of a two-input AND node or XOR node. If the probabilities before and after link propagation are equal, 1 is propagated to the link node. If the probabilities before and after link propagation do not match, an uncertain value (a value between 0 and 1) is propagated to the link node.

Link selection node

A node for selecting the propagation of a specific link. There is one selection node for each link. By activating this node, the corresponding link is preferentially selected.

SOL network

It is a type of Bayesian network that consists of nodes and links and propagates probabilities. Since links have propagation probability weights, they are by definition networks rather than graphs. It is characterized by the existence of logic operation nodes and the fact that the propagation direction of logic operations is bidirectional.

Map

From a node that represents a map, multiple subset nodes are connected via links and conditional AND nodes. This relationship between subset nodes is considered a general map.

Assumption vector

Assume the probability value of a specific link is either 0 or 1. It becomes a propagation starting point by assuming an implicit input link to the actual value node. A plurality of these assumptions come together to form a binary vector. Logical operations integrate multiple asusmption vectors. A combination node compares multiple asusmption vectors and performs feedback if there is overlap in the assumption vectors.

Propagation

Propagation of probability values through links. The propagation is always the probability value from the origin, which is a value between 0 and 1. The assumption vector is propagated at the same time.

Reverse propagation

The action of retracing the input links of a logical operation. Some logical operations perform the integration of assumption vectors.

Propagating set

A subset of the cross section through the nodes and links of a network, determined by the assumption vectors alone. It is independent of the set created by the logical operations of the nodes.

Feedback

In bidirectional probability propagation in a network, when two paths reach the same end point from the same starting point, the propagation probabilities of the two paths and the assumed vector are compared. If the hypothetical vectors overlap and the propagation probabilities are not the same, feedback is used to correct the propagation probabilities of the two paths so that they are the same. At that time, feedback correction is given preferentially to uncertain links with a small number of observations.

Effect coefficient

Shows how link variations affect the results of link propagation. Specifically, it is 1 if the propagation probability from the link to the feedback collision point is 1, and it is 0 if the propagation probability is 0 due to a NOT link or the like.

Weight

This is a value that indicates how much the variation of an individual link affects the result of link propagation. If the link propagation probability (value of P^{11} or P^{00}) is close to 0 or 1, the weight is close to 0, and if the propagation probability is close to 0.5, the weight is maximum. The weight decreases as the number of link feedback increases.

Association between fluctuations

Association between nodes whose observed values fluctuate at the same time is made by a mapping. Simultaneous fluctuations means that the probability values of the same element of the assumption vector were observed before and after the fluctuation, respectively.

Substitution

An action that virtually treats multiple nodes reached by the same assumption vector as having an inclusive relationship. Even if only a partial match, such as parameters, is confirmed, the entire set of nodes is considered to be related if there are no contradictions.

Instantiation

Replicate a part of the network as an instance. Since only the part of the propagation set that has been determined is extracted, some of the logical operations in the middle are omitted.

Generalization

Generalize and replicate a part of the network. A network with a larger set of nodes than the observed nodes is generated.The nodes with larger propagation sets are still only partially observed and thus probabilistically uncertain at this point.

Embedded function

Provided to perform inputs and outputs to and from the outside of SOL. It starts by propagating probabilities to inputs, and then uses real-valued nodes corresponding to the inputs to propagate probabilities to real-valued nodes corresponding to the results. Numerical calculations and other operations are also performed by this embedded function.

References

[1] Artificial Intelligence A Modern Approach
Stuart Russel, Peter Norvig

[2] Attention Is All You Need
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

[3] Pictureing Quantum Processes A First Couse in Quantum Theory and Diagrammatic Reasoning
Bob Coecke, Aleks Kissinger