# Self-organizing Logic

SHINDO Keisuke

kshindo999@gmail.com

April 8, 2025

**Abstract**

Self-organizing logic is a theoretical system for a logic circuit operation simulator that automatically generates sequential circuits based on probability, and is a systematic and generalized machine learning theory. The first feature is to perform bidirectional probability propagation using a Bayesian network. The second feature is to form an accurate logic circuit using two propagation probabilities between the nodes of the network. The third feature is to propagate activation values called assumption vectors to this network and perform comparisons and substitutions within the network. The fourth feature is to partially replicate and modify the network as necessary. By using these elements, this self-organizing logic can generate logic circuits with the most probabilistic accuracy. Furthermore, it is possible to express hierarchical states, state transitions, and even natural language. As a result, it is expected that all software elements can be generated autonomously.

**Keywords:** machine learning, bidirectional Bayesian network, sequential 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.

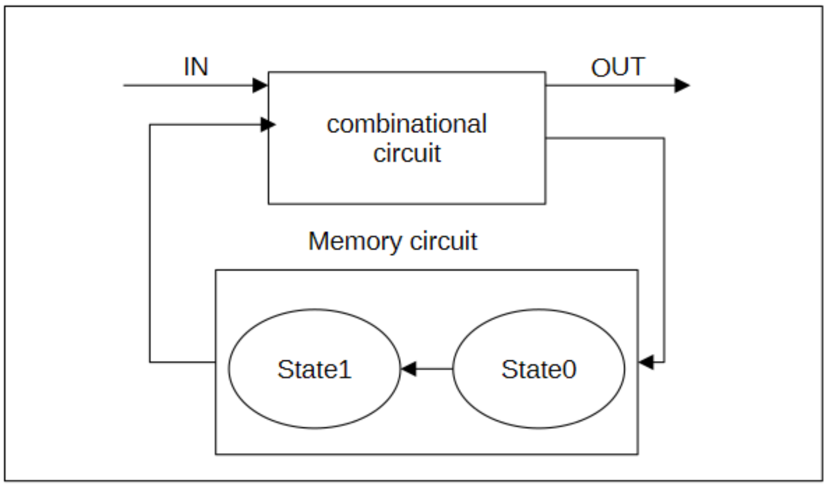

Figure 1: Sequential circuit

### 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.

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.

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.

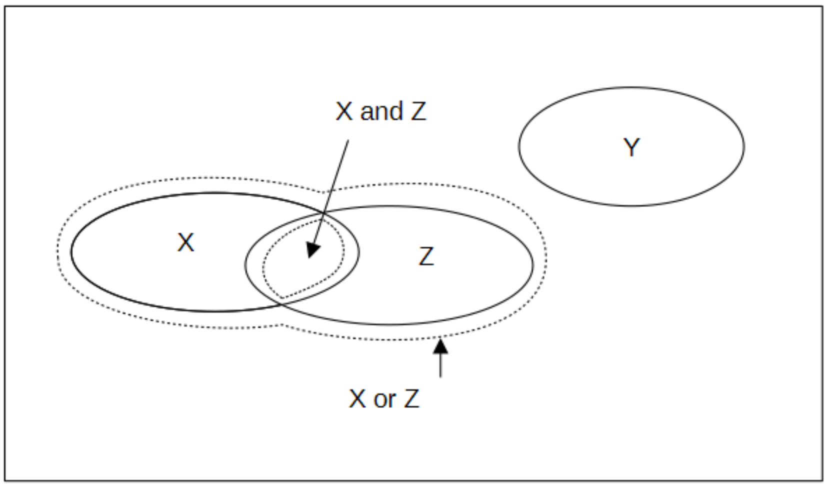

Figure 2: Basic venn diagram

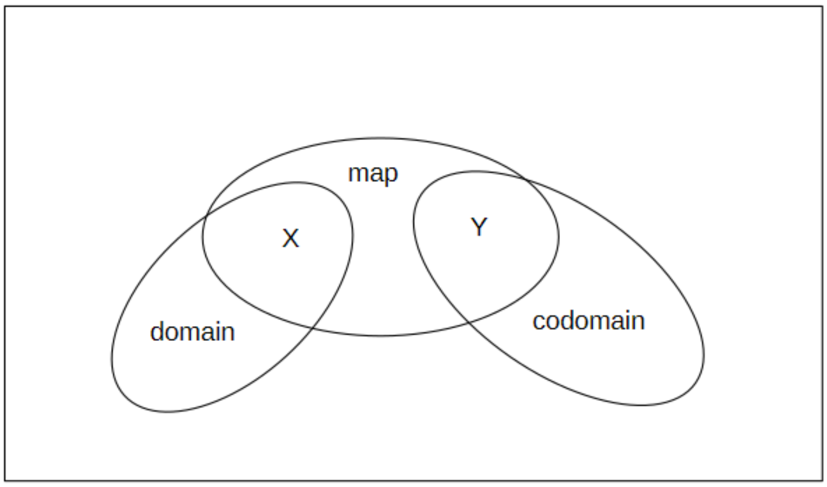

Figure 3: Set representation of mapping

## 1.2 Bidirectional binomial stochastic Bayesian network

### 1.2.1 Probability propagation

A propagation probability is assigned to each connection link between logical nodes. The logical value to be propagated is binary, 1 or 0, and the propagation probability is expressed as a value between 0 and 1.

The forward direction of the link is composed of a pair of propagation probabilities $P^{f11}$,$P^{f00}$, and the backward propagation probability is also composed of a pair of $P^{r11}$,$P^{r00}$.

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

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

The NOT operation, which inverts values, makes $P^{f11}$ and $P^{f00}$ 0. Non-NOT propagation makes $P^{f11}$ and $P^{f00}$ 1. Logical operations are also defined as probability calculations, and combine the input probabilities $P_1^{f11}$,$P_2^{f11}$,$P_3^{f11}$ of multiple logical operations.

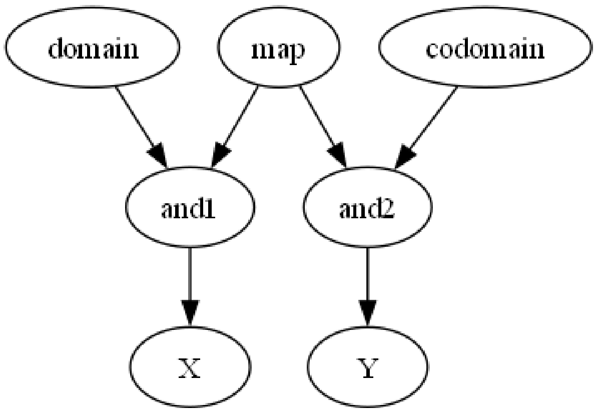

Figure 4: Representation of a mapping using backpropagation

Below is an example of AND.

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

By combining these trivial probability calculations, the propagation probability of the entire path of a logical operation between any two nodes is calculated. Unlike neural networks, no threshold functions such as sigmoid functions are used.

### 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 that propagation between any two nodes is possible. The probabilities of backward propagation, $P_r^{11}$,$P_r^{00}$, are calculated using Bayes' theorem. Let P(A) be the observation probability of the link source A and P(B) be the observation probability of the link destination B, and the following formula can be used to calculate the probability.

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

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

In other words, bidirectional belief propagation incorporates Bayes' theorem in a natural way. In the case of backward propagation of a logical operation, if all values are determined, such as the backward propagation of a value of 1 to an AND node, the propagation probability of the input links is not determined when the value of 0 is backward propagated to an AND node, but if only one input link is selected for propagation, the probability of only the selected link is determined.

## 1.3 Assumption vectors and propagation sets

SOL propagates activation values on the network, but the propagated activation value does not occupy the entire cross section of the links it propagates through as a set; rather, it considers that the set of cross sections is divided by the values of the multiple origin nodes. This propagation set is unrelated to the sets it passes through on the network, and is determined by the combination of the values of the origin nodes. For this reason, the combination of assumption values of multiple origin nodes is called an "assumption vector." The size of the propagation set is indicated by this assumption vector. Below, the assumption vector will be abbreviated to "AV" as necessary.

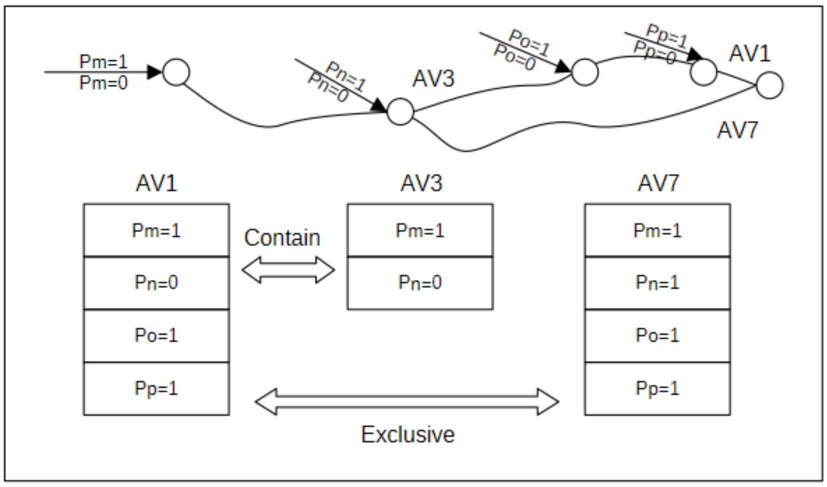

Figure 5: Assumption vectors

### 1.3.1 Synthesis of assumption vectors

When multiple assumption vectors are combined through logical operations, the resulting assumption vector is the combination of both assumption vector elements. With the logical AND operation, assumption vectors are combined when two vectors with probabilities of 1 are combined. With the logical OR operation, assumption vectors are combined when two vectors with probabilities of 0 are combined. With the logical XOR operation, or when combining vectors with intermediate probabilities between 0 and 1, assumption vectors are combined unconditionally.

- 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.

By managing the strict propagation set using the assumption vector, it is possible to form an accurate associative link between two nodes. All link formation in SOL is selected and executed according to this assumption vector.

### 1.3.3 Mapping backpropagation and propagation set expansion

The assumption vectors indicate the inclusion relationship of the sets. For various combinations of the assumptions in the assumption vector, a propagation probability is defined for each. This allows the probability of an event occurring to be calculated for each combination of assumptions.

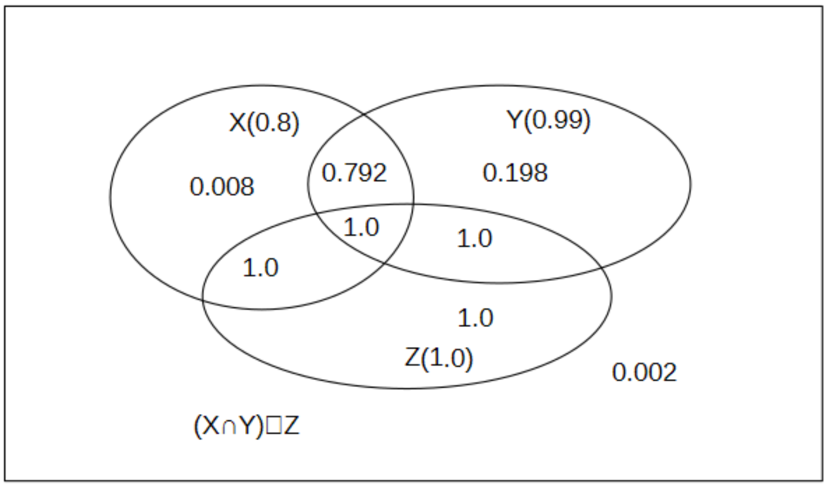

Figure 6: Assumption sets and propagation probabilities

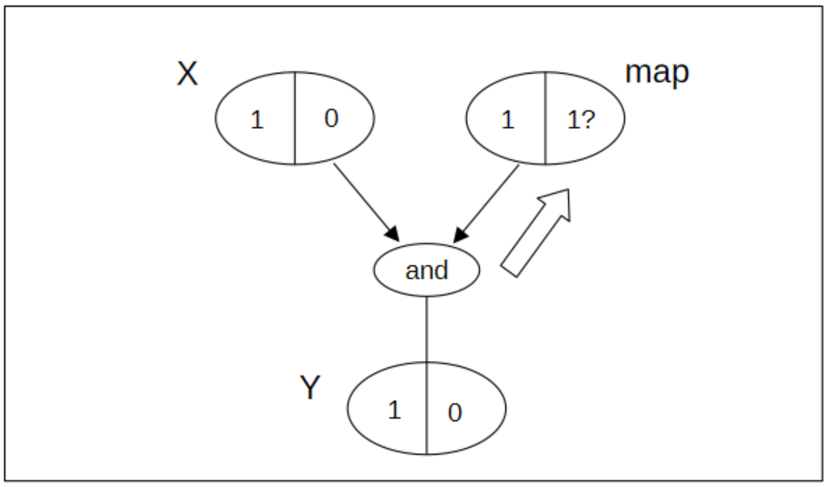

Figure 7: Back propagation

### 1.3.4 Map backpropagation and propagation set expansion

For nodes X and Y that indicate Boolean values, a method is described for propagating the activation value to node map by using the action of expanding the set when backpropagating from node Y to node map.

As shown in the diagram above, when backpropagating from output Y to input X for AND, even if the values of all nodes other than the map node to which backpropagation is directed are determined, the value of the backpropagation result is not determined. Specifically, if the input value of AND is 0 and the output value of AND is 0, the value of the map node backpropagated from AND can be either 0 or 1, with no contradiction. Therefore, backpropagation has room to expand the propagation set.

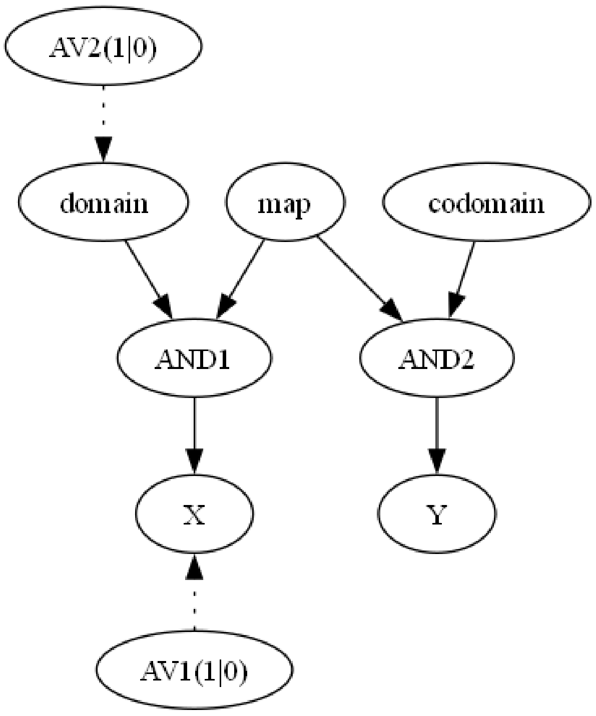

Figure 8: Map link propagation using assumption vectors

Perform back propagation from X to map. When back propagation from AND1 to map, if there is another input of the set domain, it is possible to expand the complement of the domain. Assume that AV1 propagated from X and AV2 propagated from domain are a pair of two propagation sets with complementary values. In this case, if the propagation set of X matches or is completely contained in the propagation set of the domain, the set from X is expanded and propagated to map. Expanding the propagation set means that the elements of the complementary assumption vector AV of the domain are combined to expand the propagation set, which also means that the assumption vector AV has one less element.

Since the propagation set is expanded as a result of combining the hypothesis vectors, it is highly likely that the probability is uniformly 1 or 0. This is because the propagation set combined by map is highly likely to be the same as the propagation set before adding the hypothesis element to AV1, and therefore the probability is also estimated to be a uniform value, just like the propagation set before adding the hypothesis element.

In this way, the expanded propagation set is propagated from map to AND2, and codomain and AND are taken. As a result, the expanded propagation set is propagated to Y. Here, the weak link from the map node to the AND2 node is in an undetermined state due to the small number of observations. The weak link is strengthened by positive feedback. As a result, the probability of the expansion of the set in map also approaches confirmation at the same time.

### 1.3.5 General State Representation

Using mapping, a large state can be expressed as shown below. Substates X1, X2, etc. can be hierarchically connected to further states. It is not only possible to divide and show each state by address, but also to use continuous time series or coordinates, and it is also possible to use destination nodes from other mappings or abstract nodes as conditions.

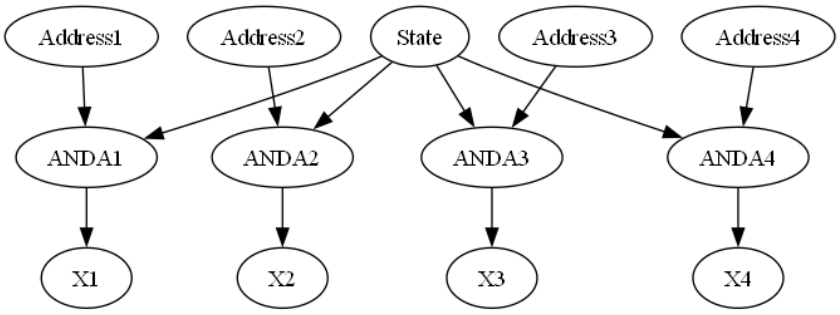

Figure 9: Multivariate state with mapping links

## 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 starting node, it is possible to reach another identical node again by passing through multiple paths in the network. For example, a collision between a node with a result observed in the past and a node with a result observed currently is a typical example.

If there is an overlap in the set of assumption vectors that are the starting points of the two routes, it is considered a collision. If the propagation probabilities of the two colliding routes are different, feedback is performed. In order to accurately calculate the amount of feedback, this probability fluctuation distribution hierarchical feedback performs a probability calculation and identifies the links to be fed back and the amount of feedback.

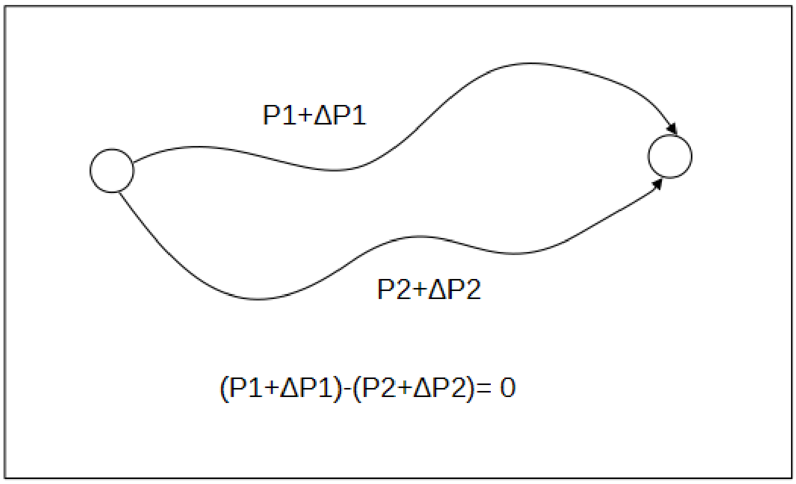

Figure 10: Difference in probability of passing two passes

### 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 from the fact that the probability of occurrence of a probability fluctuation of link n depends on the propagation probability $P_n$ of link n and the number of observations $N_n$. As a result of passing through this link multiple times, the total propagation probability $P_{total}$ is observed. If a difference occurs with the result of the total propagation probability, the difference is distributed to each link according to the weight $w_n$ of each link and fed back as follows. The weight $w_n$ of each link is calculated by the following formula (it is based on a fairly strict derivation, but it is omitted here). $P_n$ is the propagation probability of the link observed so far, and $N_n$ is the number of times feedback has been applied so far. $E_n$ is the effect coefficient for the effect of the probability fluctuation of each link on the total propagation probability, and is calculated by the total 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.

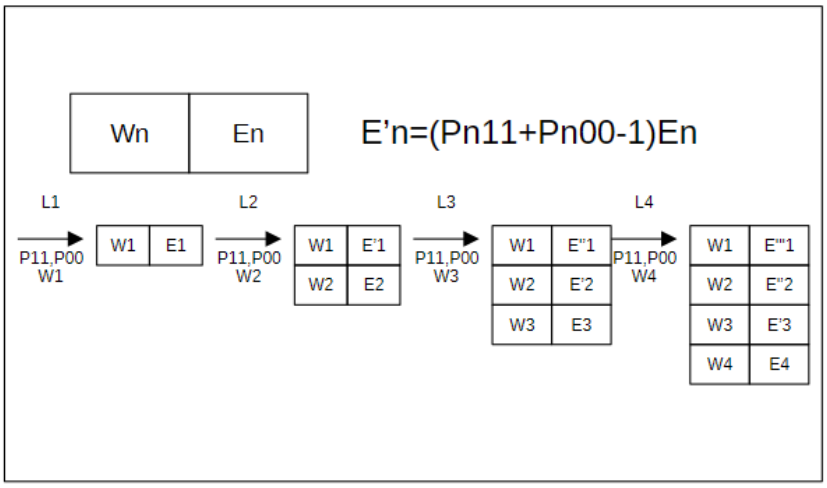

Figure 11: Propagation of weight value Wn and effect coefficient En

Once propagation is complete, the fluctuation in the overall observation probability 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 sign of the fluctuation value is consistent.

The propagation probability corrected by feedback does not become the next propagation probability of the link as it is. The larger the number of feedback observations of the link up to that point, $N_n$, the smaller the amount of feedback of the link propagation probability. $N_n$ is added using the link's effectiveness coefficient, $E_n$, to become $N'_n$.

$$N'_n = N_n + E_n$$

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

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.

### 1.4.3 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.

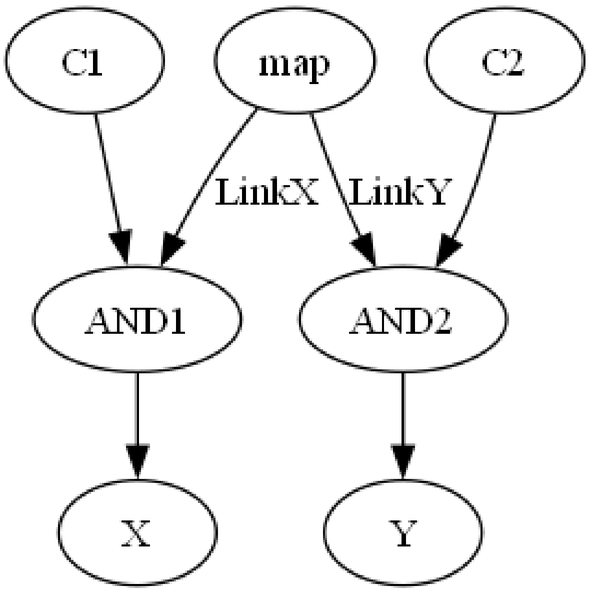

Figure 12: Forming associations from simultaneous observation

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.

After the association is formed, feedback is performed by observing the propagation probability of both $X = 1$ to $Y = 1$ and $X = 0$ to $Y = 0$. As a result, $N_y^{11}$ and $N_y^{00}$ of the associative link are further added, and the probability of both $P_y^{11}$ and $P_y^{00}$ sides is determined.

The objects X and Y that form the association are not only nodes of the observation result, but also condition nodes that are the result of feedback, link nodes that indicate causal relationships, etc., and it is possible to form associations based on probability between any subset, and it is possible to correspond to any abstract concept.

## 1.5 Stochastic autonomous logic generation algorithm

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

### 1.5.1 "Conditional" Link splitting and logical operation node insertion

If the link probability approaches 1 to 0.5 or 0 to 0.5 as a result of the feedback, the link is considered to have been negatively fed back. In other words, the uncertain probability part of the link is separated using logical operations with the current premise vector as the condition.

By managing the link propagation probability as a binary pair of $P^{11}$,$P^{00}$, the AND node or OR node to be inserted can be deterministically selected based on the direction of the propagation probability where the negative feedback occurred. If $P^{11}$ approaches 1 to 0.5, the node is an AND node, and if $P^{00}$ approaches 1 to 0.5, the node is an OR node. The XOR node is selected when both $P^{11}$ and $P^{00}$ approach 0.5, but this is limited to cases where feedback is given under the same conditions for both P11 and P00. The condition link to be added to the inserted logical operation is selected from nodes with the same premise vector.

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

### 1.5.2 "Causal" Link node activation and associative targeting from positive feedback

If the probability of the activation value passing through a random link matches the activation value of another fixed probability, the random link is positively fed back, and

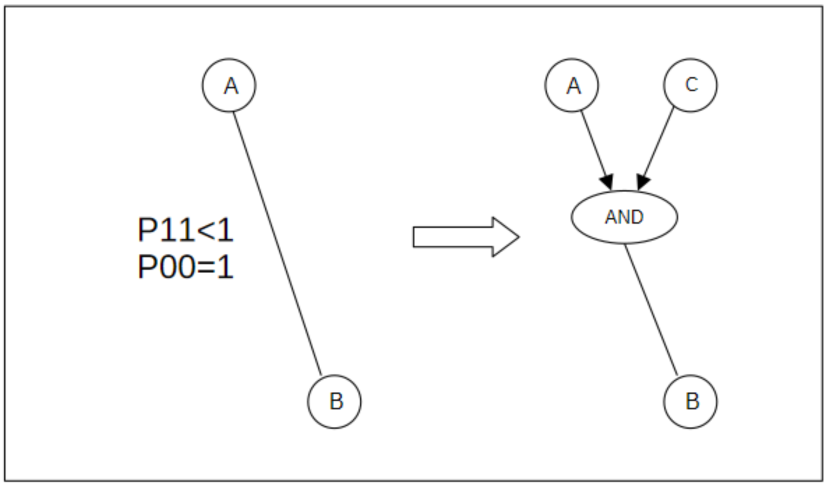

Figure 13: Insert a condition AND into the link from A to B

as a result, the "link node" corresponding to that random link is considered to be activated by the positive feedback.

The concrete entity of the link node is an unknown input node that is input to a logical operation node that is virtually added and inserted into the random link. An example is as follows. $\wedge$ is the logical operation AND.

$$B = A \wedge L$$

Back propagation to link nodes propagates the matching results of the propagation probabilities $P_A$ and $P_B$ at both ends of the link. Specifically, the propagation probability $P_L$ is expressed as follows:

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

If the probability values before and after the link are both the same, 1 and 0, the following link node is specially generated. In this case, XOR$\oplus$ is used.

$$B = A \oplus \neg L$$

The back propagation to the link node in the case of XOR is as follows.

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

This is the basic method for using the results of causal relations between two different sets in logical operations. Even if the probabilities of $P_A$ and $P_B$ are intermediate, if they match at anything other than 0.5, the probability becomes 1.

In this way, when input A and output B match as premise sets, the link node Match is activated by backpropagation. This link node is considered to be a match between A and B.

This can be thought of as a causal relationship node. Causal relationship node can be applied to conditional flow control, for example, by judging the match of numerical values. Backpropagation from XOR indicates an exact match, while backpropagation from AND or OR indicates an inclusion relationship.

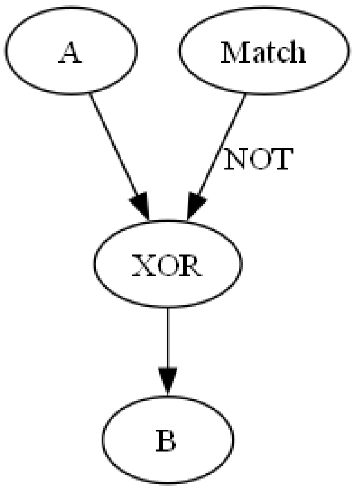

Figure 14: Representation of causal relationships by backpropagation

### 1.5.3 “Substitution” Coupling between nodes that are inclusive in terms of the propagation set.

When propagating using forward logical operations, the propagation set becomes smaller as conditions such as AND and OR are added. Here, by propagating another logical operation in the reverse direction, the conditions such as AND and OR are removed, and the propagation set before the conditions were applied can be restored and propagated. As a result, a containment relationship may be established between two distant mapping nodes. This propagation between consistent mapping nodes is called assignment. In the following example, value A is included in variable X ($A \subset X$), so Map1, which contains value A, becomes a subset of Map2, which contains variable X, and Map1 is assigned to Map2.

It is possible to perform logical operations using multiple conditions within a state, and to use hierarchical states. It is similar to pattern matching in Prolog, etc., and is a more generalized method.

### 1.5.4 “Instantiation” Deterministic partial network replication

When an activation value passes through a network, it may collide with another determined activation value, causing positive feedback and determining part of the network. As a result, it becomes possible to extract only the determined part. This is called instantiation.

In this case, if there are multiple branches in the link path, such as output links or logical operation input links, the network of the path is partially duplicated to create a kind of instance. The instance omits unnecessary link branches.

An example is shown in the figure below. The State on the right side is assigned to FunctionState as a subset in advance. From the assigned State, it passes through the function and is propagated as the result of function execution to the activation value indicating the value a. This propagation set does not overlap with the State, but it is possible to expand the propagation set of the State without contradiction. In this way, the result of the function is duplicated and added to the input state.

This “instantiation” is also used to assign function results. Instantiation can extract parts of the network structure with further determined probabilities, and at the same

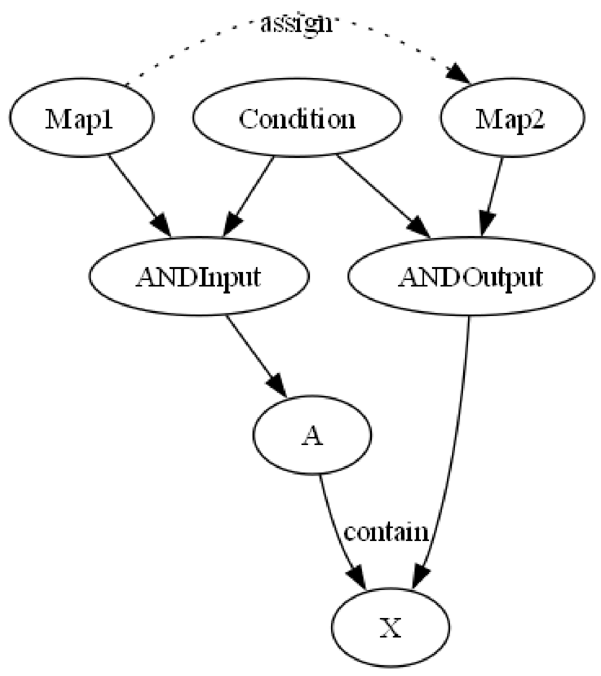

Figure 15: Assign between map nodes

time, it has the effect of reducing the number of propagation branchings, thereby reducing the number of search branchings in propagation.

### 1.5.5 “Generalization” Generalized partial duplication of a network

When an existing network is reused, a partially inconsistent part may be fed back. For example, in the case of feedback to the P00 side of a link, it is possible to partially expand and duplicate the network. This is “generalization.” The biggest difference from instantiation is that the expanded part is uncertain and needs to be confirmed in future verification.

### 1.5.6 “Selection” Selective control of multiple link propagation

There can be a large number of output links that can be connected to a node. To make link search more efficient, it is necessary to select a link from a large number of links for a condition. It is desirable to have SOL determine and make this selection process more efficient.

Controlling link selection means generating a link selection node corresponding to the selected link and making it the target of association and activation propagation. As a result, it is possible to associate the utility obtained from the activation value of the selected link with the link selection node. Conversely, the link selection node is activated from the utility, and the link corresponding to the utility is selected.

Controlling the selection of the link selection node is different from the method of controlling links by adding conditional logical operations, and can maintain the link propagation probability. Even if a link is not selected, the link propagation itself is not particularly hindered, so propagation itself is possible, but the possibility of it being selected as a propagation target is low.

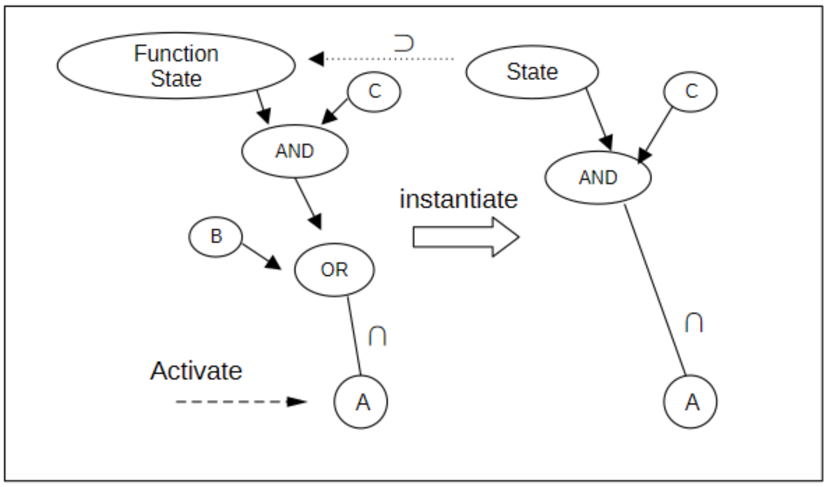

Figure 16: Duplication of a network instance

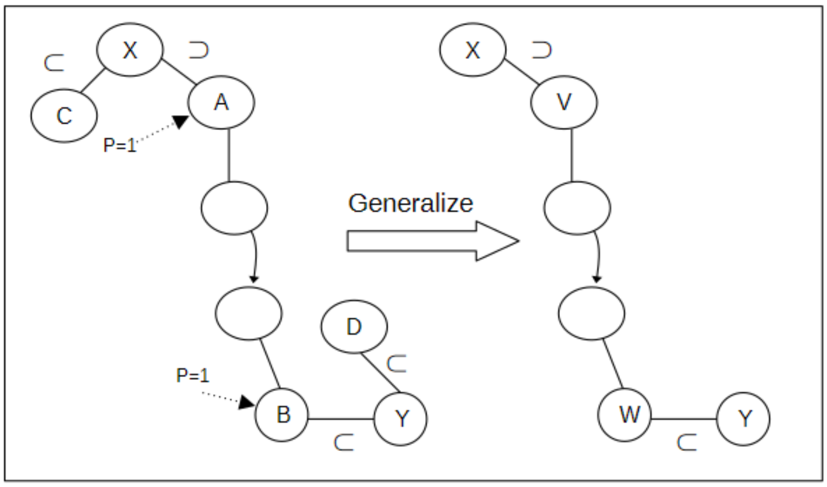

Figure 17: Generalized duplication of a network

### 1.5.7 Stochastic 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 automatically generated using the above methods. For probability fluctuations and sets, definite probability links with a propagation probability close to 100% or 0% are selected as much as possible. For links with a large number of observations and uncertain propagation probabilities, other conditions are added sequentially and replaced with definite probability links as much as possible.

In other words, unlike neural networks that use analog value weights, this probabilistic autonomous logic generation algorithm is an algorithm that attempts to digitize and reproduce the observed object as much as possible. Since definite digital logic is generated, it becomes possible to output the learning results in the form of a logical formula.

## 1.6 Autonomous Generation of Sequential Circuits

### 1.6.1 From association between fluctuations to states

A generalized observation is an act of associatively connecting a time node with multiple observation nodes observed at that time. For example, multiple bit values and a time node are combined to form a state.

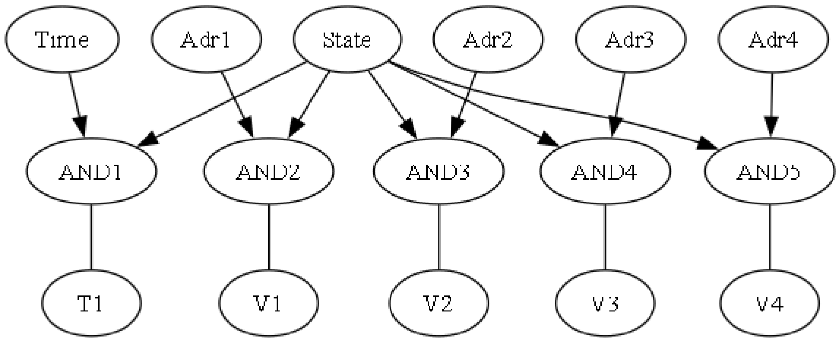

Figure 18: Form associations between time and state

### 1.6.2 Autonomous generation of logical operations between states

V1, V2, V3, and V4 are Boolean values that are observations at a certain time. Each observation is different depending on the time.

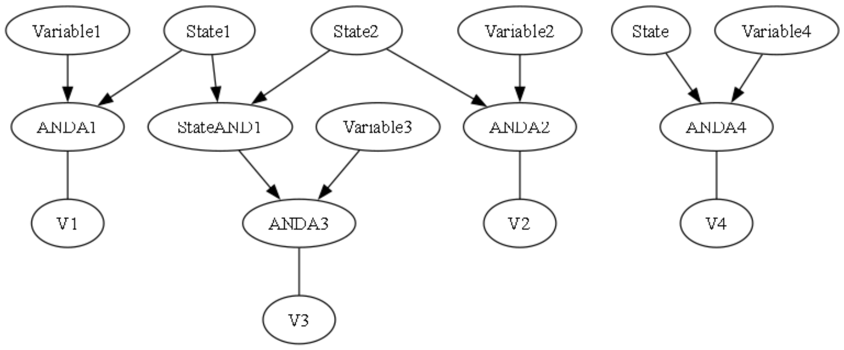

Figure 19: Associate with time passage and state

For the network of V1 to V4 connected by states, propagation occurs again from the second observation from V1 to V4. Feedback occurs due to a collision of probabilities between State and V1, generating StateAND1, and State is split into State1 and State2, which become the input condition for StateAND1. As a result, the following logical formula is formed.

$$V_3 = V_1 \wedge V_2$$

### 1.6.3 Forming associations in chronological order

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

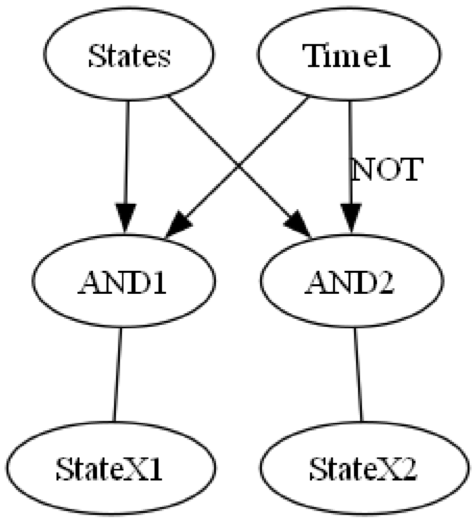

Figure 20: Associations between time variability and state

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.

#### 1.6.4 Generic Function Generation

Using the above elements, a sequential circuit is generated autonomously. Furthermore, by generalizing and duplicating the learned state, a generic function is generated in which the values before and after the mapping are generalized into variables.

The value nodes that are the subject of the mapping are Boolean values in the examples so far, but any object can be linked, such as more abstract numbers, coordinates, and character tokens. Interactions such as calculations between abstract nodes are realized using built-in functions defined outside SOL, and SOL selects the links between the built-in functions during learning.

# 2 Bidirectional logic operations

1. There are multiple real sets that divide space-time, and the containment relationship between the sets is unknown.

A real set corresponds to the material of each space-time, and has a certain extent of extension in the space-time direction. Multiple real sets can be exclusive, contain, or overlap. Relationships between all of these real sets can be derived by mapping.

2. Among multiple real sets, things that are observed simultaneously in a specific situation can be connected as a "map."

There are cases where a part of a real set is always observed "simultaneously" with a part of another real set. "Simultaneous" in this case means selecting a subset that is unrelated to the real set. In this case, a containment set that includes both subsets can be formed as a "map," regardless of the overlap relationship of the real sets themselves. This containment set is considered a "map set." It can be expected that this "map set" expresses what is generally called a causal relationship. The source of the map is completely included in the map set, but the destination of the map may only include a subset of the map set. This is due to the definition of "implication."

3. Performing logical operations between multiple mappings.

By performing a set logical operation between multiple mapping destinations, the set resulting from the logical operation becomes a subset of the final mapping destination. This is a logical operation generalized by mapping.

It is assumed that the real space-time has this kind of structure. Based on this, the aim is to reproduce the structure of real sets and mappings using SOL from observation, whatever that may be.

## 2.1 Nodes and Links

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

1. Value nodes, which are the entities of observed values
2. Logic nodes, which indicate logical operations between links
3. Joint nodes, which connect links equivalently
4. Exclusive nodes, which indicate exclusivity between links
5. Function nodes, which connect to links and input and output to the outside

Value nodes are abstract subsets that exist in space, but it is also possible to have a one-to-one correspondence between nodes and actual values such as scalars, vector values, and characters.

Logical operation nodes perform Boolean algebraic operations on input links. There are types such as 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 the equivalence of values and sets between output links.

Connection nodes are nodes that only have outputs, and indicate the equivalence of values and sets between multiple links. Equivalence of multiple links also means that feedback occurs when arriving from multiple links.

Exclusive nodes are nodes that indicate exclusivity between input links. They are roughly equivalent to NOT links between all input nodes, but are used for efficiency.

Function nodes use the actual values indicated by value nodes to perform actual operations and external input/output. They generate value nodes such as numbers that correspond to the results each time.

A link consists of a binomial set of probabilities $P^{f11}$,$P^{f00}$ to apply when traversing the link, and experience numbers $N^{f11}$,$N^{f00}$. When drawing links, they should only have arrows if the link destination is an input to a logical operation. Otherwise, they are considered equivalent links and no arrows are used.

The attributes of the link will be as follows.

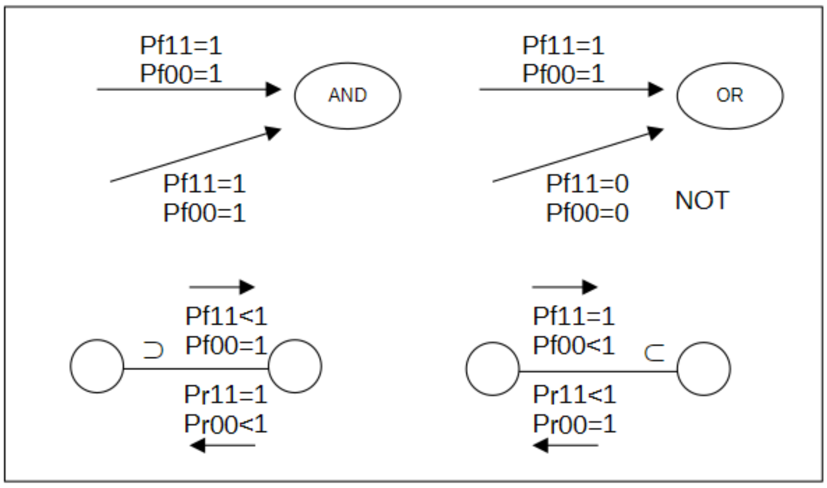

Figure 21: Link Types

Table 1: Probabilities related to link propagation

| 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 calculates the propagation probability from the origin through multiple links and logical operations.

## 2.2 Link Propagation and Logical Operations

### 2.2.1 Propagation

Link propagation is calculated using the propagation probabilities $P^{11}$, $P^{00}$.

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

### 2.2.2 NOT Propagation

If the link propagation probability is set as follows, the link itself will behave as a logical NOT.

$$P^{11} = 0$$

$$P^{00} = 0$$

### 2.2.3 AND Operation

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

$$P' = P_1 P_2 P_3 ... P_n$$

### 2.2.4 OR Operation

The results of link propagation are combined with an OR logical operation. $P_1$,$P_2$... are input and $P'$ is output.

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

### 2.2.5 XOR operation

The results of link propagation are combined with XOR logical operation. $P_1$,$P_2$... are input and $P'$ is output.

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

For XOR of three or more variables, the operation is applied recursively.

### 2.2.6 Exclusive operation

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

$$P' = P_n$$

### 2.2.7 Reverse Propagation

Reverse propagation probabilities $P_r^{11}, P_r^{00}$ are defined for each link.

For normal links, they can be calculated immediately from the forward propagation probability. The forward propagation probabilities are $P^{f11}$ and $P^{f00}$, and the reverse propagation probabilities are $P^{r11}$ and $P^{r00}$. In this case, the reverse propagation probabilities $P^{r11}$ and $P^{r00}$ are calculated using the following formula.

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

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

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

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

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

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

### 2.2.8 Entropy and link determination

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

$$S = \sum_n P_n log P_n$$

When this is binarized,

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

As a result, the probability P close to 1 or 0 has the smallest entropy S, and the probability P close to 0.5 has the largest entropy S. The objective of SOL is to make the link probability as close to 1 or 0 as possible. In other words, minimizing entropy is one of the objectives of SOL's self-organization. To achieve this, links with high entropy are split by inserting appropriate logical operations to reduce the entropy of each link.

### 2.2.9 Feedback to links

Feedback to the link separates the links vertically into multiple links for each set that passes through the link. Positive feedback, where the overall probability approaches 1 or 0, does not require separation. Negative feedback, where the overall probability approaches 0.5, is considered to be a mixture of subsets with propagation probabilities of 1 and 0 among the sets that pass through the link.

For example, if a link previously passed with probability 1, but now passes with probability 0, the propagation set with probability 0 that passed is considered to be a subset that separates the link vertically. The means of this vertical division is the insertion of logical operations.

## 2.3 Vertical splitting of links and insertion of logical operations

Links propagating from node to node can be split vertically by splitting the starting node and the ending node into subsets.

A link is split when it is observed that there is a set with a different propagation probability among the sets passing through the link. When the propagated observation probability approaches 0.5, the link is separated into an element with an observation probability of 1 and an element with an observation probability of 0. The elements of the separated link use the information of the starting node that indicates the set of elements to form a logical operation with that node.

- If the $P^{11}$ side is uncertain and the $P^{00}$ side is confirmed as 1, insert AND.
- If the $P^{00}$ side is uncertain and the $P^{11}$ side is confirmed as 1, insert OR.
- If the $P^{11}$ side is uncertain and the $P^{00}$ side is confirmed as 0, insert NOT(AND).
- If the $P^{00}$ side is uncertain and the $P^{11}$ side is confirmed as 0, insert NOT(OR).
- If both the $P^{11}$ side and the $P^{00}$ side are uncertain, there is a possibility of inserting XOR.

# 3 Mapping

## 3.1 Definition of Mapping

Logical operations can be described as the relationship between sets. However, to describe more general logic, memory elements such as flip-flops and memory that can switch between various time states are essential. Otherwise, it is impossible to create general-purpose logical operations for fluctuations in time and space.

SOL adds the concept of mapping to logical operations. Mapping connects different states in time and space and uses them as new starting points. The action of this mapping is itself a generalization of memory elements. It also makes it possible to manage the agreement of multiple states as a state. This makes it possible to use the relationship itself, such as the order between states, as a state.

This shows how to express mapping using only logical operations. In the following example, a subset X of the source set is mapped to a subset Y of the destination set. A Map node is a type of connection node.

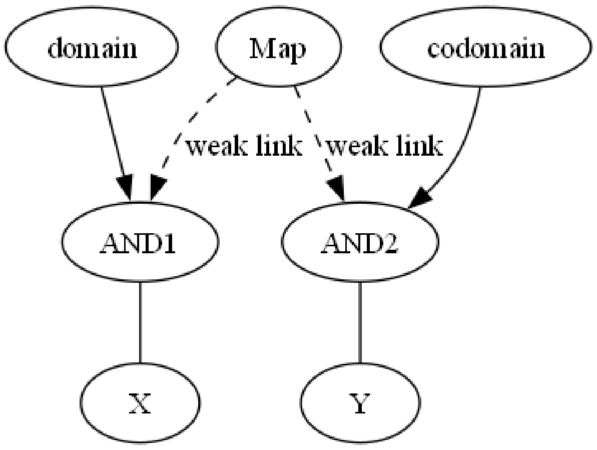

Figure 22: Map with weak links

A weak link is a link with a small number of observations, either $N^{11}$ or $N^{00}$, and can become uncertain due to negative feedback from the next observation. Further logical operations are added to make this link certain. Logical operations can be added to either of the two weak links, but only the logical operation that is consistent will be certain.

## 3.2 Back propagation and substitution through the map

This diagram shows a method for sharing the source and destination of multiple mappings. From the perspective of the Map1 node, X is the range of the mapping, and from the perspective of the Map2 node, X is the domain of the mapping. The important thing is that the Cond node that is the condition for the range of the Map1 node is the same as the Cond node that is the condition for the domain of the Map2 node.

First, the propagation from the map1 node to the map2 node is a backpropagation. The activation value at the map1 node is 1. However, by using the concept of activation value substitution, we will show why the probability of the propagation set reaching the Map2 node is uniformly 1.

The activation value with probability 1 that exists on the Map1 node is propagated from the Map1 node to X. At that time, AND is taken at the node Cond, and the

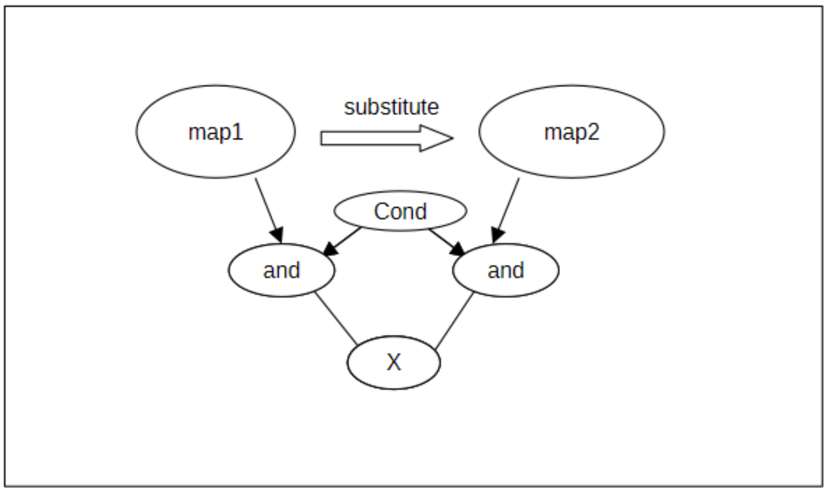

Figure 23: Substitution between maps

propagation set is divided. However, when it is propagated from X to the Map2 node, it passes through the backpropagation of AND. At that time, it is complemented by the same Cond node as the Map1 node. As a set, it becomes the same size as the activation value of the Map1 node. The activation value of the Map2 node cannot be completely determined to be either 0 or 1. However, since it is the backpropagation of AND from the value 1, it is possible to assume that all values of the activation value of the Map2 node are 1 in the propagation set.

As a result, we can assume that the activation value in the Map1 node is propagation-set-wise identical to the activation value in the Map2 node. This is considered to be an assignment in SOL. The conditions for this are as follows:

1. There is no contradiction because the value of the propagation source and the value of the determined part of the propagation destination are the same.
2. The assumption vectors of the two propagation sets are the same.

This assignment is similar to Occam's razor and is strictly speaking unfounded. However, the validity of this assignment is verified between the propagation using the assigned value and the observation. Furthermore, the conditions for validity are added by feedback.

## 3.3 Hierarchical matching by mapping and bidirectional propagation

As an example, let us take text analysis from Input. In the following example, three mutually exclusive words are activated using mutually exclusive nodes L1, L2, and L3. To determine whether they match, backpropagation from the words occurs, and InputAND is activated in reverse. It is sufficient to confirm that L1, L2, and L3 are mutually exclusive. Backpropagation from AND1 to Map1 then expands them collectively and they are no longer exclusive. The ANDs from Map1, 2, and 3 are combined to activate InputAND. Conversely, InputAND will not be activated if all elements are missing.

In this way, using bidirectional propagation makes it possible to perform set logic operations between words that collectively do not overlap.

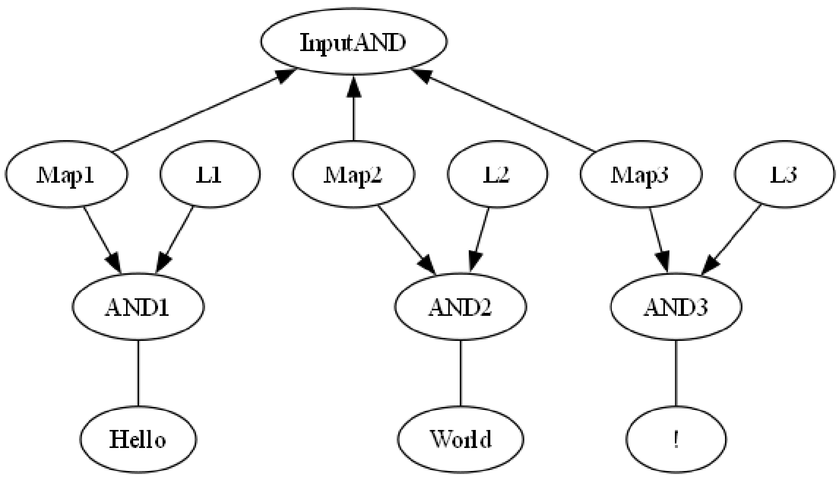

Figure 24: Natural language expression using mapping

Furthermore, the concepts of time series and distance are added to these exclusive nodes L1, L2, and L3 to represent actual text sentences.

## 3.4 Order and Natural Language

### 3.4.1 Representation method using mapping nodes and order links

This shows a method for expressing order using sets and mappings. PreviousState and CurrentState are states of successive time, and are linked by a higher-level mapping StateMap. PreviousState and CurrentState are linked by the conditions Previous1 and Next1, which indicate the order, respectively.

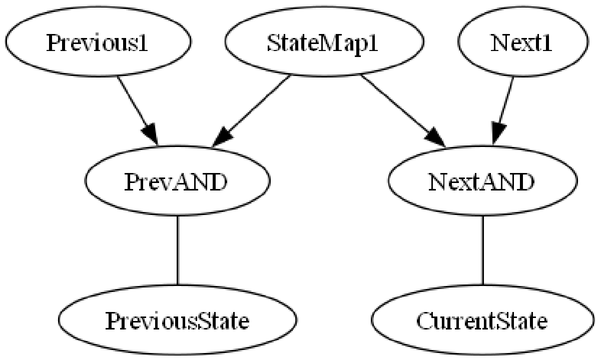

Figure 25: Representing the order between states by mapping

Previous1 and Next1 are the condition nodes used for this StateMap1, but we also consider this to be a subset of the generalized ordering Previous, Next nodes. This grouping can be used for substitution into the ordering of another network.

### 3.4.2 Sentence Matching

In the following example, three mutually exclusive words are activated using the mutually ordered links Next1 and Next2. Next1 and Next2 are also link nodes that are activated when the order between the words is established. Matching is input from the link nodes

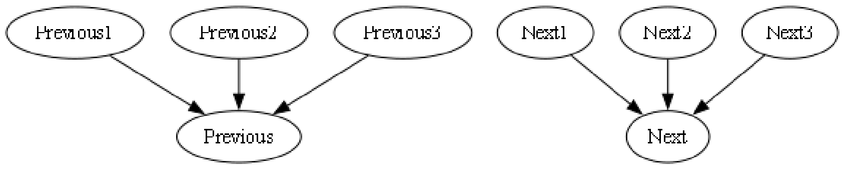

Figure 26: Grouping by rank

Next1 and Next2 to InputAND. Unlike the previous example, if the order of the words is swapped, the link nodes Next and Next2 will not be activated. In that case, no substitution is made to the map that represents the entire sentence.

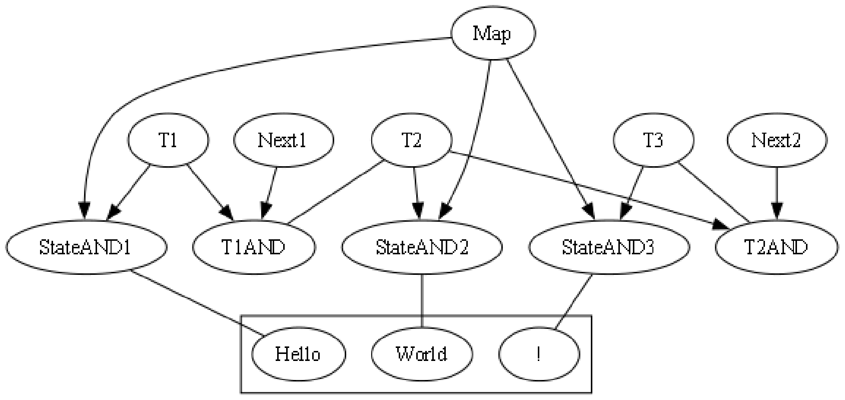

Figure 27: Representing a sentence as a mapping order

This ordered sentence network is reused. The next time a partially identical sentence is input, it is propagated to this existing network. As a result, the state of the long sentence is divided and conditions are added along the way. In this way, the sentence state is hierarchical, logical operations are added, and it is associated with other sentences and other concepts.

This method of expressing the order may seem inefficient, but it is a strict method that is not affected by the order structure in space and time. Many current machine learning methods implement the order as a simple vector, but one-dimensional vectors have limited versatility.

## 4 Bidirectional propagation of activation values

SOL propagates activation values in a similar way to neural networks, but strictly manages the propagation probability from the starting point. In addition, it manages the information of the sets propagated by the assumption vector. In addition, SOL allows the propagation of logical operations in both directions to express mappings. This makes it possible to calculate the propagation probability and propagation set between any sets in space as long as they are connected by a network.

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

1. Propagated probability

The propagated probability value is a scalar value between 0 and 1.

2. Assumption vector

An assumption vector that indicates multiple propagation sets that are the starting points of this activation.

## 4.1 Propagation of activation values

1. Probability propagation from assumptions

The activation value starts from a hypothesis. The hypothesis is that the probability of the starting node is either 1 or 0. The activation value is propagated by applying link propagation and logical operations to this starting probability.

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 parts of the sets indicated by the assumption vectors are extracted and propagated. Logical operations are not applied to parts where the assumption vectors do not overlap.

3. Backpropagation of logical operations and complementary synthesis of assumption vectors

- The set on the output side of the logical operation is propagated in reverse to the input side of the logical operation.
- When backpropagating a value of 1 from AND, the probability 1 and the assumption vector are propagated to all inputs as is.
- When backpropagating a value of 0 from AND, if there is an input where all inputs are 1 except for one, the input is determined to be 0.
- When a value of 0 is backpropagated from AND, if there is already a 0 in the input, the probability of the backpropagated input is uncertain. However, there is another way to make the uncertain probability certain. This is why backpropagation can be used for mapping.

4. Activation collision and comparison of elements of assumption vectors

When multiple activation values reach the same node via different paths, if there is a common part between the two assumption vectors, the probabilities of both are compared. If the probabilities are different, feedback is performed to make the probabilities equal.

5. Built-in Functions and Observations

When the activation value reaches the built-in function, external observations and external actions are performed. Observation means connecting the input assumption vector and the output node with a mapping.

## 4.2 Assumption Vector

### 4.2.1 Assumption Vector Elements

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

1. Source link

The elements of the assumption vector assume that the propagation probability of the source link is either 0 or 1.

2. Backpropagation selection

Backpropagating the input link from the activation value of 1 to the OR node, one of its input nodes is selected. This selection action can be considered as an assumption of the link from the OR node to the input node. The same is true for backpropagation from the activation value of 0 to the AND node.

3. Common origin in the middle of the network

When multiple propagations collide, it is wasteful to consider all the assumption vectors of the route. Therefore, an assumption is set for the common origin of multiple propagations, and the assumptions of the common paths before that are not compared.

### 4.2.2 Interaction of logical operations

When logical operations such as AND and OR are performed between propagation sets, the assumption vectors are combined. The following rules apply to this combination.

- Assumption vector elements that are different between assumption vectors are added to the destination as is.
- Identical assumption vector elements between assumption vectors are aggregated into one.
- Composition with assumption vector elements that are collectively inclusive replaces the smaller element in the propagation set.
- Composition with assumption vector elements that are collectively exclusive (different starting point probabilities) causes the propagation set itself to become an empty set.

### 4.2.3 Activation value collisions and assumption vectors

If multiple activation values reach the same connection node or logical operation node, they are considered equivalent and will collide. In terms of logical operations, the result of combining the inputs and the output will collide.

If there is an overlap in the propagation set indicated by the assumption vector, the probability values of the propagated results must match. If they do not match, feedback is executed.

## 4.3 Probability Propagation of Activation Values

The basis of SOL is to propagate activation values along links, propagate probabilities according to the link propagation probability and node probability combination, and execute the built-in function at the end of the link.

$P^{f11}$ is the propagation probability from P with probability 1 to P' with probability 1, and $P^{f00}$ is the propagation probability from P with probability 0 to P' with probability 0. This is a trivial application of probability theory.

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

.

In the AND operation, the probabilities of multiple activations are integrated and propagated to the output link. The probability calculation is performed using multiple input propagation probabilities $P_1$ , $P_2$ , $P_3$... as follows:

$$P' = P_1P_2P_3...$$

The probability calculation for the OR operation is as follows:

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

The probability calculation for the XOR operation is as follows. In the case of three or more inputs, the following formula is applied recursively:

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

When the number of output links is huge, activation is propagated only to a few links. The method for limiting the links will be provided separately.

## 4.4 Backward propagation of probability

Activation can also be performed by backpropagation, which follows the opposite direction of the links. This has a different meaning from backpropagation in neural networks.

Backpropagation is performed in the same way as the forward propagation probability calculation, using the backpropagation probabilities $P^{r11}$,$P^{r00}$ for each link. No logical operations are applied.

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

As a result, the total 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

A collision of activation values refers to the difference in the propagation probability even though the propagated hypothesis vectors overlap on multiple propagation paths that start from the same node and reach the same node.

In principle, $P = 0.5$, that is, partial overlap with a propagation set with an uncertain probability, is not a collision because it can be considered unrelated as a propagation set. Conversely, if both probabilities are certain, overlap in the propagation set can be considered a probability collision. If the side with probability 1 (or 0) completely contains probability 0.5 in terms of the propagation set, a contradiction occurs in the certainty of the side with probability 1, so it is considered a collision.

The sizes of the propagation sets are compared using the above methods, and probability collision and feedback are applied to the parts where the propagation sets completely overlap.

## 5 Hierarchical Feedback

This feedback method is a newly developed method that can accurately discover which links are causing errors in the observation probability for an unlimited number of layers in a Bayesian network.

In conventional neural networks, it is possible to affect deep link layers using back-propagation, but it is still difficult to identify the cause of errors in deeper link layers. What is called deep learning is simply a relatively deep hierarchy.

The assumptions behind this feedback are as follows:

1. Each link in SOL has a propagation probability, and logical operations are also probability calculations. The overall propagation probability is calculated after passing through multiple links and logical operations. The observed new propagation probability $P'$ is fed back to $\Delta P_n$ for each link to bring the network propagation probability closer.

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

2. The fluctuation occurrence probability for each link with probability $P$ to become probability $P'$ depends on the past observation probability $P$ and the number of observations $N$, and can be calculated using pure probability theory.

3. The fluctuation occurrence probability for the entire link is the product of the fluctuation occurrence probabilities for all links. This overall fluctuation occurrence probability is maximized.

4. The number of observations $N$ is added to each link for each feedback. The existing propagation probability is corrected according to the number of observations using the fed back propagation probability. Links with a large number of observations will have little correction due to feedback.

A calculation method was determined to satisfy the above conditions. This calculation method is called probability fluctuation distribution hierarchical feedback.

### 5.1 Binomial Propagation on Links

The propagated activation value has one propagation probability $P$. This probability $P$ is the probability of the current set of hypothesis vectors. The probability of being in the complement is $(1 - P)$.

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

$$P^{11} = 1$$

$$P^{00} = 1$$

NOT links act on the probability of the hypothesis vectors. Note that they do not invert the set of hypothesis vectors that pass through.

$$P^{11} = 0$$

$$P^{00} = 0$$

A link has parameters $N^{11}$ and $N^{00}$ that indicate the strength of the link, in addition to $P^{11}$ and $P^{00}$. This strength is added for each feedback and determines the probability of the link.

## 5.2 Bidirectional propagation and probability calculation of logical operations

Here is the method of calculating the probability in the activation value propagation in SOL. The normal calculation method for the propagation probability of a link is as follows.

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

When performing an AND operation, it is as follows. This is the propagation probability where P=1.

$$P'_1 = P_1^1 P_1^2 P_1^3 \dots$$

OR operation

$$P'_1 = 1 - P_0^1 P_0^2 P_0^3 \dots$$

The backpropagation probability of a link is calculated by applying Bayes' theorem from the forward probability of the link. $P_1$,$P_0$ are the probabilities of node A. $P^{f11}$,$P^{f00}$ are the forward propagation probabilities from node A to B. $P^{r11}$,$P_{r00}$ are the backward propagation probabilities 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 and OR operations to the input 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 from the propagation probability of the entire network currently in use to the propagation probability $P'_{total}$ of the colliding partner, which has a different probability in the same situation.

SOL feedback is applied to each of the links $P^{11}$ and $P^{00}$ through which the two colliding paths pass. The strength of the feedback varies depending on the link probability and the number of links experienced.

The feedback results $P^{11'}$ and $P^{10'}$ are corrected using correction values $\Delta P_n^{11}$ and $\Delta P_n^{00}$. The total probability of the two paths is calculated by propagating it as a polynomial with a correction value added to each. This results in a large-scale equation.

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

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

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

The total probability of each of the two paths A and B is a propagation polynomial for a large number of $\Delta P_n^{11}$ and $\Delta P_n^{00}$. Furthermore, since the equations for these two paths are equal,

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

All we need to do is find N $\Delta P_n$s that satisfy this equation and have the smallest probability of variation. However, a proper method would involve varying each $\Delta P_n$ to find a solution, which would result in a combinatorial explosion problem.

To avoid this, we provide a means to approximately find the amount of variation $\Delta P_n$ in each probability observation probability.

## 5.4 Calculation of effect coefficient

The following is an equation that shows the extent to which the link propagation probability fluctuation $\Delta P_n$ affects the final probability $P_n$. The probability that affects the probability fluctuation $\Delta P_n$ is $E_n$, and $R_n$ is a constant term.

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

This probability $P_n$ changes when further links or logical operations are applied. For this purpose, we will show an equation 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 the $P^{11}$ side of a link is used as the starting point, the equation is as follows. The propagation probability to the link is $P_{n-1}$.

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

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

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

When the $P^{00}$ side of the link is used as the starting point, 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 expressed by the following formula.

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

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

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

Next, we calculate the propagation probability for the logical operation. From now on, $P^{11}$ and $P^{00}$ are regarded as the m-th and n-th links, respectively, and are denoted as $P_m$ and $P_n$. The composite probability of inputs other than $\Delta P_n$ is $P$. $E_n'$ in AND operation is

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

$$R_n' = P R_n$$

$$E_n' = P E_n$$

$E'_n$ 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

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

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

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

Next, we will show the method of backpropagation. If backpropagation is from a logical operation, $P^{r11}$ and $P^{r00}$ are used regardless of the operation. The formula is as follows.

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

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

When backpropagating an AND operation, if the probability of the other input is determined to be $P_f$, the formula is as follows. The derivation is complicated, so we will omit it. Although the quadratic and higher terms of $\Delta P_n$ are necessary, they are often omitted.

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

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

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

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

Backpropagation of OR operation is

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

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

Backpropagation of XOR operation is

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

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

By continuously applying the above formula to the network routes, the propagation probability of the entire route can be calculated. This propagation probability is considered to be equal to the observation result $P'_{total}$. All fluctuations of multiple links can be summed up. Since the probability is less than 1, the product of probabilities will definitely converge. Although the quadratic and higher terms of $\Delta P_n$ are necessary, they are often omitted.

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

As a result, the final action probability $E_{ntotal}$ obtained by propagation is the coefficient of action for each link $\Delta P_n$.

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

## 5.5 Derivation of the formula for weight value

The probability correction value $\Delta P_n$ that is fed back to each link that has passed through the propagation can be found by rewriting and calculating all the probabilities of the propagation links in the following form.

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

For the propagation probability $P_n$ of link n, the fluctuation occurrence probability $t_n$ at which the propagation probability $P'_n$ is observed as an observation result follows a kind of binomial distribution and is obtained by the following formula.

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

Specifically, this formula uses m coin tosses with probability $P_n$ to find the probability that the sum of the results is $mP\ '_n$, and takes the limit for m. The following combination formula is used.

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

Furthermore, by multiplying by the number of past observations of the link $N_n$, the probability that the result is $mN_nP\ '_n$ is found. This is the probability of fluctuation occurring on a link where probability $P_n$ has been observed multiple times.

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

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,

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

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

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

Using the value of this combination, substitute $T_n$.

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

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

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

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 $\tau_n$. The last term converges in the limit of m, so we omit it.

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

$$-N_nP'_n \log P'_n - N_(1-P'_n)\log(1-P'_n)$$

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

Differentiate this $\tau_n$ equation with respect to $P_n$.

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

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

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

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

This allows us to calculate the weight $w_n$ of the fluctuation of $P_n$ with $\tau$ as a parameter.

$$P_n + \Delta P_n$$

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

$$w_n = \begin{cases} \dfrac{P_n(1-P_n)}{N_n} & \text{if } P_n > 0.5 \\ -\dfrac{P_n(1-P_n)}{N_n} & \text{if } P_n < 0.5 \end{cases}$$

In this way, the weight $w_n$ of each link is determined by the known probability $P_n$ of the link and the number of observations $N_n$. The sign of the weight is unified in the direction that reduces the entropy of the link probability. Note that the sign of the weight is further reversed depending on the feedback target.

## 5.6 Calculation of the probability correction value

From here, we will explain the feedback distribution method to multiple links. The observed total probability is $P'_{total}$. The fluctuation $\Delta P_n$ of each link is calculated so that it matches this probability. $E_n$ is the effect coefficient determined by the propagation probability for link n, as mentioned above.

$$P'_{total} = P_{total} + \Delta P_{total} = P_{total} + \sum_n E_n \Delta P_n + \sum_m \sum_n E_{mtotal} E_{ntotal} \Delta P_m P_n \cdots$$

Another constraint is to maximize the total fluctuation occurrence probability $T$.

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

Using $\tau_n$, this formula becomes as follows.

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

The purpose of feedback is to determine the fluctuation value of $P_n$ for each link so that the total fluctuation occurrence probability $T$ (large tau) is maximized and the entropy of each link is minimized while distributing the value of $\tau_n$ as evenly as possible.

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

## 5.7 Feedback distribution and network replication

### 5.7.1 Equal distribution feedback

When the amount of fluctuation to be fed back is small, equal distribution is performed according to the weight of the link in order to minimize the fluctuation of the entropy of the propagation probability of each link. First, replace the fluctuation occurrence probability $\tau_n$ of each link n with a common parameter $\tau$.

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

For each $\Delta P_n$, the total fluctuation $\Delta P_{total}$ is distributed using the ratio of $E_n w_n$.

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

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

Using the $\Delta\tau$ calculated in this way, the total feedback $\Delta P_{total}$ can be distributed to the $\Delta P_n$ of each link.

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

It is possible that $P'_n$ can be greater than 1 or less than 0, so in such cases $P'_n$ is saturated to 1 or 0 and $P_{total}$ is recalculated each time. After that, the saturated link is removed and $\Delta\tau$ is calculated again.

However, there are cases where equal distribution is not possible, such as when the sign of the probability is completely reversed from 1 to 0. This is because the product of the fluctuations of each link cannot be ignored. In actual feedback of binary logic, this case is more common. Here is the procedure for doing so.

### 5.7.2 Weight reversal feedback

If the effect coefficient $E_n$ of a link is close to 1, the overall probability can be reversed by simply feeding back to that link. In this case, in order to keep the propagation probability entropy of other links at a minimum, one link with the smallest propagation probability entropy and experience number N is selected and the weight is reversed. Then, feedback is applied to completely reverse the overall probability. Since there can be multiple links with an effect coefficient $E_n$ close to 1, there are multiple possibilities for link selection. For each selection, the network is replicated and feedback is applied. The priority is determined by the weight of the link.

If the overall probability to be fed back is either 1 or 0, feedback is completed by simply selecting and reversing one link. However, if the overall probability to be fed back is an intermediate value between 1 and 0, after applying the reversal of one link, an equal distribution is again performed to minimize the entropy of the other links. The purpose of the feedback is to minimize the overall link entropy and to maximize the fluctuation probabilistically, and if this purpose is met, the weight is reversed.

## 5.8 Link Propagation Probability Correction

This feedback-corrected probability does not become the probability that the link itself will update. The larger the number of observations of the link up to that point, $N_n$, the smaller the link probability correction amount will be. The value added to $N_n$ is not always 1, but is determined by the effectiveness coefficient $E_n$ used in the actual propagation and the number of observations of the other party that is fed back, $N_f$.

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

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

$$P''^{N'_n}_n = P_n{}^{N_n} P'_n{}^{N_f E_n}$$

Since there is not much difference between the multiplicative average and the additive average, it can also be approximated by the following formula.

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

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

Based on the feedback results for the calculated probability $P^{11}_n$, links where $P^{11}_n$ is in the direction of decreasing entropy, that is, where the probability approaches either 1 or 0, are considered to have received positive feedback. Conversely, links where $P^{11}_n$ is in the direction of increasing entropy, that is, where the probability approaches 0.5, are considered to have received negative feedback. It is assumed that negative feedback occurs because the link has some invisible condition, and the condition is searched for. Feedback is calculated similarly for $P^{00}_n$.

When the number of link usages $N_n$ is 0, the probability is considered to be $P_n = 0.5$. Positive feedback alone can bring the probability closer to 1 or 0, but it cannot become 1 or 0 itself.

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

# 6 Associations

## 6.1 Associative Target Selection

An association is the process of predicting a causal relationship by linking two collectively independent nodes with a map. To form an association that may have a causal relationship, the success rate is high if two nodes that fluctuated simultaneously are selected. Note that simultaneous fluctuations include all fluctuations, such as time axes and coordinates. The probability of simultaneous fluctuations, or in other words, the strength of the association, can be quantitatively determined based on the probability of fluctuations occurring at each node.

The conditions for this are as follows.

1. The probability entropy observed simultaneously is almost equal (in most cases, a definite probability of either 1 or 0).

2. The assumption vectors from the same origin are the same, or there is an inclusion relationship. Simultaneous fluctuations are considered to be cases in which the assumption vectors themselves are completely identical or have an inclusion relationship.

Here, unrelated events can be accidentally activated at the same time and become the subject of an association, but since the possibility of further simultaneous activation of an association due to chance is low, negative feedback results in an uncertain link.

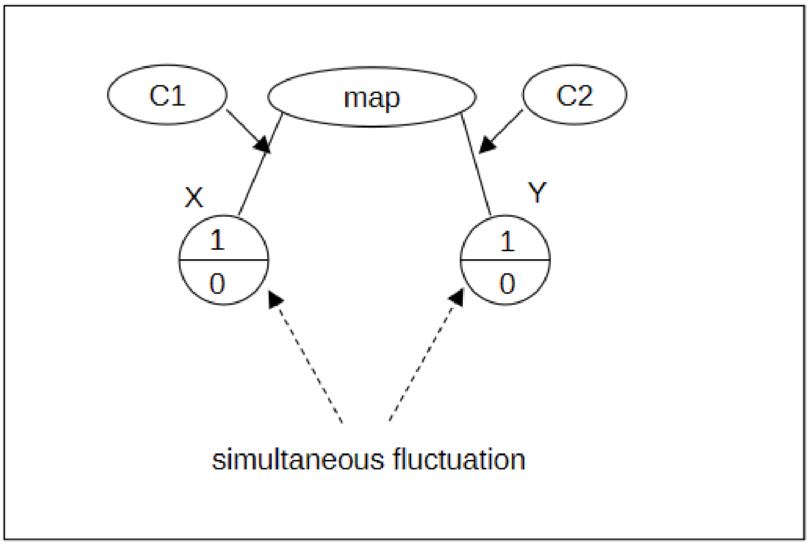

Figure 28: Create association

A node that is a candidate for association is the end point of the link to which negative feedback has been applied. By applying negative feedback, it can be considered that some condition has been added to the link.

A node that changes at the same time as this negative feedback change is considered to be possible for association. Furthermore, a change node with a low probability of value change is considered to have a high association accuracy, so it is selected with the highest priority.

An observation is an action of associating a node indicating the current space-time with a node indicating an observed value, and is basically generated each time by an external function.

## 6.2 Association Formation and Addition of Conditions

Association requires the selection or generation of two nodes that are conditions for performing the and from the map. These condition nodes are generated from the difference of the assumption vectors. In the above diagram, nodes F and C, which are the difference between AV1 and AV2, are the conditions.

## 6.3 Calculation of Propagation Probability and Number of Experiences

Associative links are not definite links, and the number of association experiences $N$ can be determined from the probability of observed simultaneous activation values.

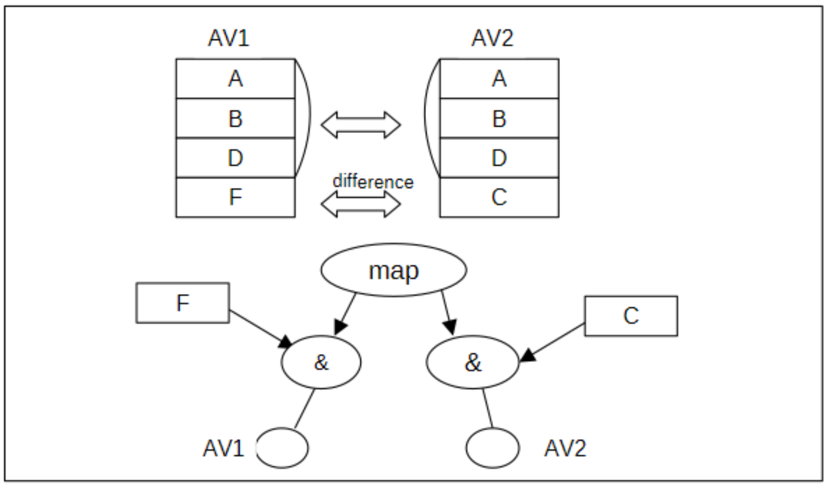

Figure 29: Difference between assumption vectors

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

The calculation method is as follows. In an unmeasured and uncertain link, that is, an associative link with an assumed propagation probability $P = 0.5$, $P = 1$ is actually observed $N^{11}$ times, and the past propagation probability $P_Y$ of the link to the associated object Y is measured as a result.

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

In other words, $P = 0.5$ can be considered as a link observed $N$ times.

When X=1, Y=1 are observed, the association probability is $P' = 1$. As a result, the initial values $P^{11}, N^{11}$ are as follows. When X=1, Y=0, $P = 0$.

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

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

When X=0, Y=0 are observed, it only affects the $P^{00}$ side. In this case, the association probability is also $P' = 1$. As a result, the initial state $P^{00}, N^{00}$ is expressed by the following formula. When X=0, Y=1, $P' = 0$.

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

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

This calculation of the number of experiences means that the fluctuating association between events that rarely occur is also highly certain of the association.

## 7 Autonomous Logic Generation Algorithm

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

## 7.1 "Conditional" Link splitting and logical operation node insertion

### 7.1.1 Feedback on associative links

When an associative link is formed, the propagation probability is uncertain, and the propagation probability is corrected by feedback. As a result, while there are some links for which a definite propagation probability of 1 or 0 is observed, there are many more indeterminate links for which the propagation probability is an intermediate value. In such cases, the indeterminate links for which both a propagation probability value of 1 and 0 are observed are brought closer to a definite probability of 1 or 0 by adding logical operation conditions. This section explains the method of selecting the conditions for this purpose.

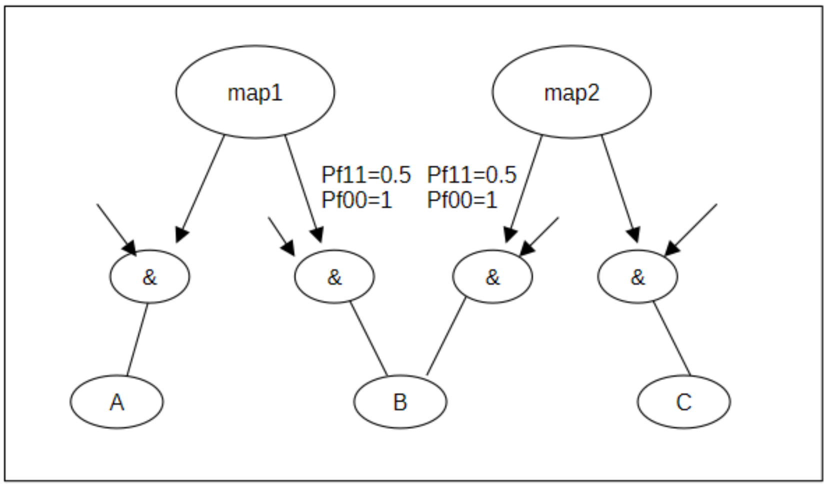

Figure 30: Feedback to an associative link

Associations are formed from A and C to B. The associations are connected by the Map1 and Map2 nodes, respectively. The propagation probability $P_{f11}$ of the two association links to B is corrected to an uncertain probability by feedback for the same assumption vector. To correct this propagation probability $P_{f11}$ to a definite value of 1 or 0, a condition is formed for the link of B. A logical operation is inserted into the feedback side of the two uncertain links, and the other becomes the input of the logical operation that becomes the condition. A logical operation is not formed unless the link is fluctuating for the same assumption vector.

### 7.1.2 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 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.

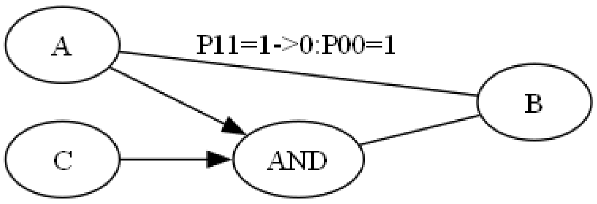

Figure 31: AND node formation

### 7.1.3 OR node

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

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

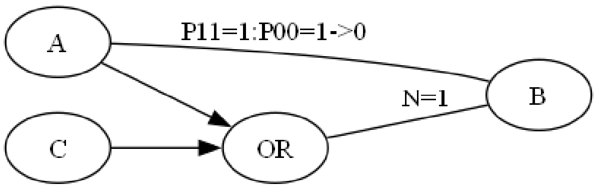

Figure 32: OR node formation

### 7.1.4 XOR node

If feedback is given to both $P^{11}$ and $P^{00}$, the link becomes random and is essentially invalid. However, if feedback is given to both $P^{11}$ and $P^{00}$ at the same time, the link becomes a candidate for joining at the XOR node.

$$A \to B \quad \begin{cases} P^{11} = 1 \to 0 \\ P^{00} = 1 \to 0 \end{cases}$$

$$C \to B \quad \begin{cases} P^{11} = 1 \to 0 \\ P^{00} = 1 \to 0 \end{cases}$$

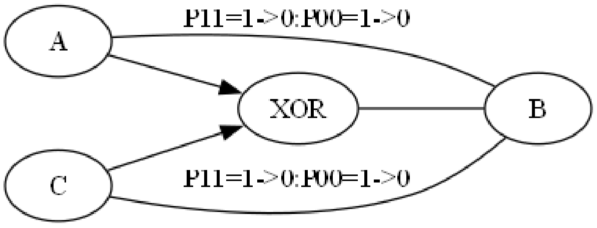

Figure 33: XOR node formation

### 7.1.5 NOT link

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

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

## 7.2 Link node activation from "causal" feedback

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

$$A \rightarrow B$$

### 7.2.1 Logical operation equivalent to a link node

As a result of feedback to the link from A to B, link node L is activated. The link from A to B is rewritten as a logical operation using L. In other words, the true identity of link node L is a new input node for the AND operation inserted into the link. If $P_A = 1$ and $P_B = 1$ change to $P_B = 0$, logical operation AND$\wedge$ is inserted.

$$B = A \wedge L$$

It becomes. If $P_B = 0$ and $P_B = 0$ change to $P_B = 1$, OR logical operation $LOR$ is inserted.

$$B = A \vee L$$

Furthermore, if feedback is observed for both $P = 1$ and $P = 0$, a link node can be implemented using the XOR logic operation $\oplus$. For convenience, the link node and the XOR input are connected by a NOT link.

$$B = A \oplus \neg L$$

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

### 7.2.2 From determining the link propagation probability to generating a link node

The probability of backpropagation to a link node using AND is given by the following formula. $P_A$ and $P_B$ are the probabilities before and after the link, and $P_L$ is the probability of the link node.

$$P_B = P_L P_A$$

The probability of propagation to a link node is given by the following formula. Note that it is defined as anything other than $P_A = 0$.

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

Note that unlike forward propagation, probability propagation to a link node cannot propagate fluctuations as is.

The propagation set by the assumption vector is propagated by substitution, with a concept similar to the backpropagation of a map. The two propagation sets $P = 1$ and $P = 0$ propagated to A are combined and propagated to the link node. As with the reverse propagation of a map, the propagation probability to a link node when $P = 0$ is indefinite, but the assumption vector propagated to the link node is substituted for links with equal propagation probability.

Furthermore, the propagation probability to a link node is calculated from the reverse propagation of XOR. First, the formula for the logical operation of XOR is used.

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

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

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

Another feature of XOR link nodes is that it is strictly possible to cancel out the assumption vector elements. If the complementary propagation sets $P = 1$ and $P = 0$ originate from the same assumption vector and the values of the assumption vector elements are complementary, the assumption vector elements can be integrated.

## 7.3 "Substitution" Propagation Connection between nodes in a collectively contained relationship

The term "assignment" used here means that when it is confirmed that the hypothetical vectors are in an inclusive relationship with respect to the activation values that have reached the multiple mapping nodes, the entire activation values are propagated between the mapping nodes.

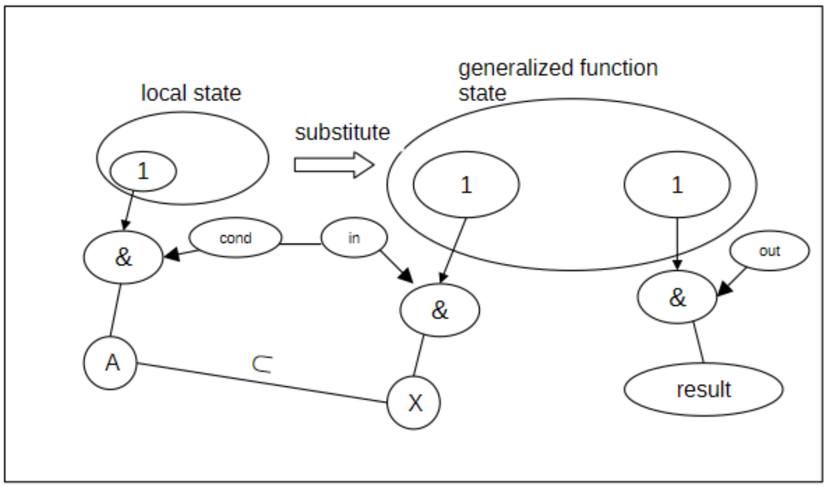

Figure 34: Substitution between states using assumption vectors

In the figure, substitution is performed between multiple states. The conditions for this substitution are explained below.

The localstate node contains a set of value A, but the rest of it is indeterminate. For value a of the localstate node, the functionstate node has set X which contains value A, so there is no inconsistency in considering the functionstate node to contain the localstate node. Therefore, the functionstate node contains the result node as a partial state.

The result node is not present in the localstate node, but there is no inconsistency in assuming that it exists in the undefined part of the localstate node. This result can be added to the localstate node in the next “instantiation”. This is the assignment of a value to the state by a function.

## 7.4 “Instantiation” Instance duplication of part of a network using propagation subsets

A part of a network is split vertically to take a subset, 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 the output of many links can be omitted.

This instantiation is performed when the propagation probability for the same propagation set is determined between the starting subset and the ending subset, and the propagation set of the network for that route is large by OR or the like. The entire route between the starting point and the ending point is separated by the propagation set.

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

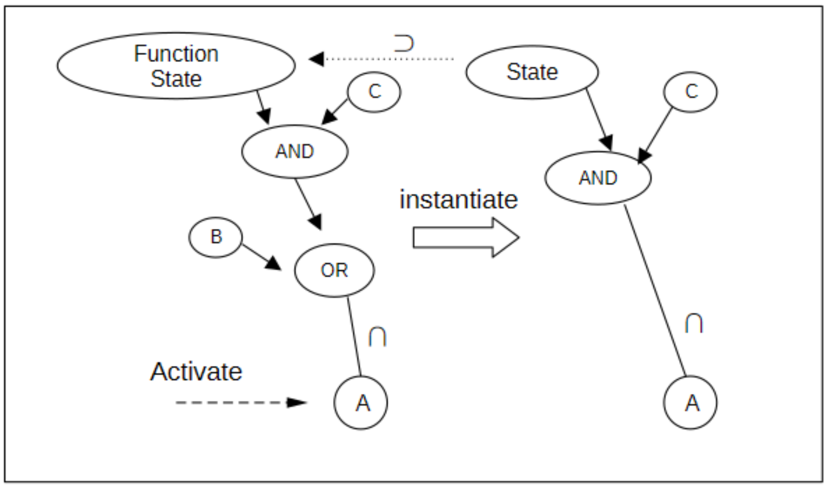

Figure 35: Partial instantiation of a network

## 7.5 “Generalization” Generalized duplication of part of a network by propagation subset

A part of the established network is replicated as a more collectively generalized network between the inclusion set of the origin and the inclusion set of the destination. This is generalization. It is similar to instantiation, but it is performed even when the propagation set of the network is not completely observed. In other words, this is a provisional network that is later corrected by feedback.

X is a generalized node that includes A. Y is a generalized node that includes B. At this point, the relationship between X and Y is undetermined. The path from X to Y is generalized when the propagation from X to Y is determined using other elements C and D of X and Y, even if A and B are activated with 0. At this time, the entire path between the starting point X and the end point Y is separated. At that time, A, which is a subset of X, and B, which is a subset of Y, are replaced with variable V and W nodes, respectively.

It should be noted here that propagation is not determined for any combination of the subsets A, C, and B, D inside X and Y, respectively. For example, propagation from A to D is not established because it has not been observed.

Generalized node X and generalized node Y may be OR nodes, but often use Exclusive nodes, which mean mutually exclusive nodes.

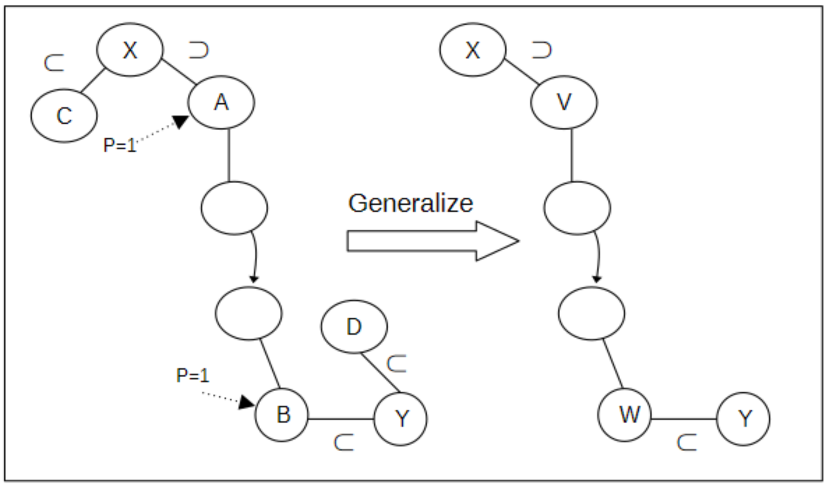

Figure 36: Partial generalization of the network

## 7.6 “Selection” Selection control of multiple link propagation

A large number of links may be connected as inputs and outputs of a node. In particular, the number of links can be more than a thousand when connecting functions. Therefore, in order to minimize 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 these large number of links is originally deterministic, and when link control is performed by logical operations, the propagation probability fluctuates and becomes uncertain. To solve this problem, a link selection node is generated separately from the control by logical operations.

Link selection activates the link selection node. This activated link selection node is linked to the fluctuating state of the rest of the network by fluctuating association. As a result, it becomes possible to select an appropriate link corresponding to the state of the network.

The link selection node can also be thought of as an association target for a subset of the SOL network itself.

## 7.7 “Convergence” network optimization

When positive feedback occurs, where two paths and probabilities match, nodes and links in the paths of the network that are small in the propagation set and do not have a large number of observations are deleted. Used in SOL network optimization. It is the reverse action of network duplication, and is performed on unused networks.

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

## 7.8 Overall operation of SOL

The entire SOL operates in the following steps using the algorithms described above.

**Algorithm 1** SOL main loop

```
1: Generate initial activation state
2: while Overall loop do
3:     Select one activated node
4:     if Input shortage of logic node then
5:         Add assumption vector element to input
6:     Probability collision determination using assumption vector
7:     if Probability collision detection then
8:         Execute feedback
9:         if Increase link entropy then
10:            Form condition
11:        else
12:            Form association
13:            Activate link node
14:        Generate instance network
15:        Generate generalized network
16:    Execute built-in function
17:    for Output links do
18:        Propagate to output link
19:        Execute logical operation
```

# 8 Sequential circuit generation

By using the states by mapping hierarchically, circuits between generalized states are generated autonomously. Furthermore, by generalizing the states with variables, etc., a function that can be used universally is generated.

## 8.1 Observation function

The state observation function forms an association between the observed current time $T_n$ and the results A, B, and C observed at the same time. The results observed at the same time at the current time are further linked by mapping as causal relationships, and the logical relationship between them is observed.

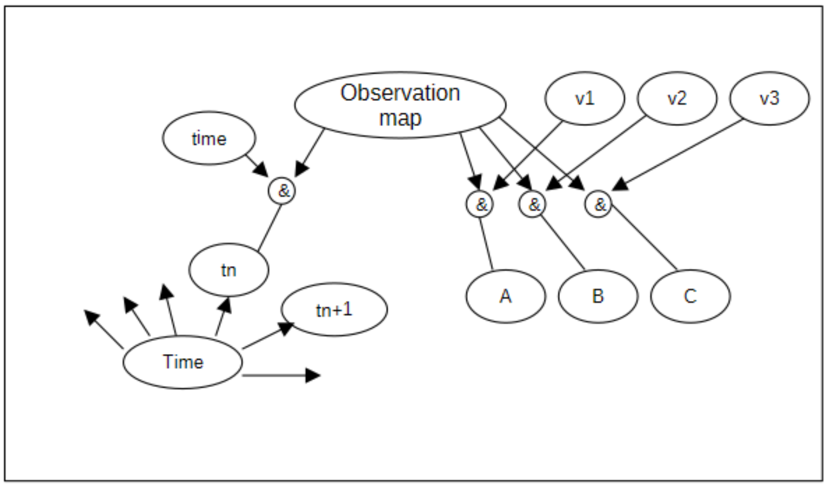

Figure 37: Observation

## 8.2 State transition generation

The state transition of a time series is formed from a mapping when time is the difference. The mapping between A, B, C for time $T_n$ and D for time $T_{n+1}$ is fed back to form a logic circuit between A, B, C, and D.

## 8.3 Function generation from generalized nodes

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

## 8.4 Function execution

The function formed by generalization assigns a state with the input values. Since the function is considered to be propagation-aggregate larger than the assigned state, the

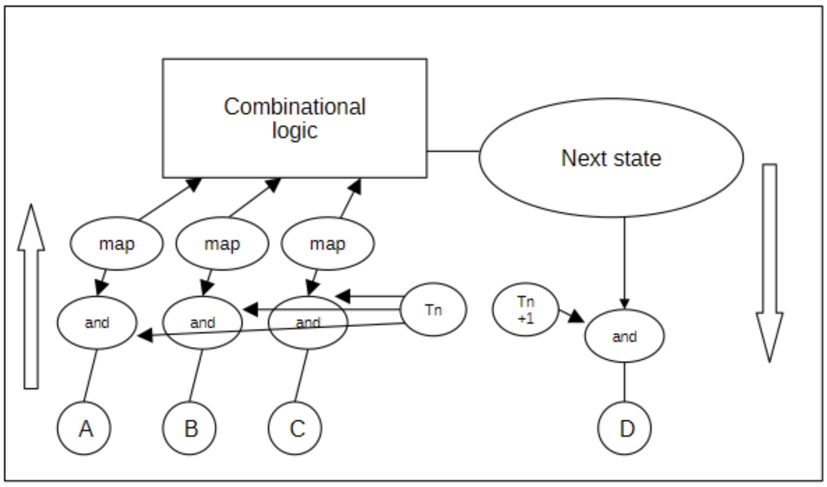

Figure 38: Sequencial circuit by map

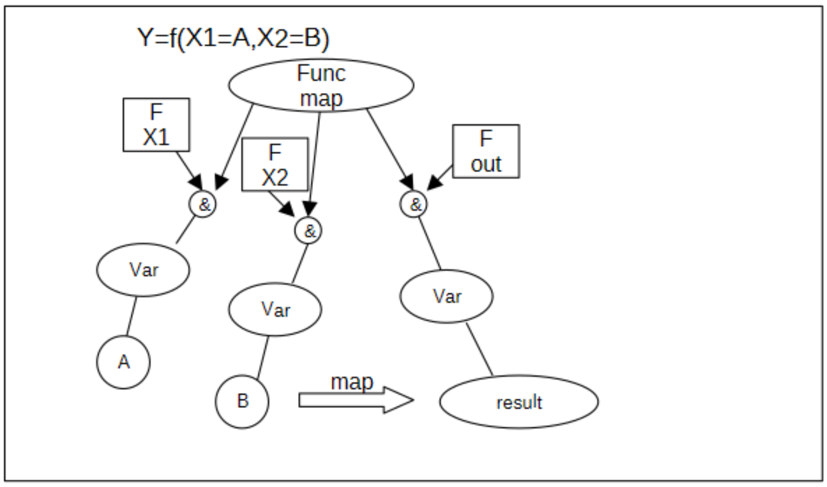

Figure 39: Function map

result value of the function is also considered to be included in the input state. This is the application of the function to the state. The applied result is then partially replicated and added to the original state during instantiation.

When feedback occurs to a result node, the feedback is also recursively applied to the function that produced the result.

The result of the function can also be a link node indicating a match, making it possible to implement a conditional test that converts a match to a 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.

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.

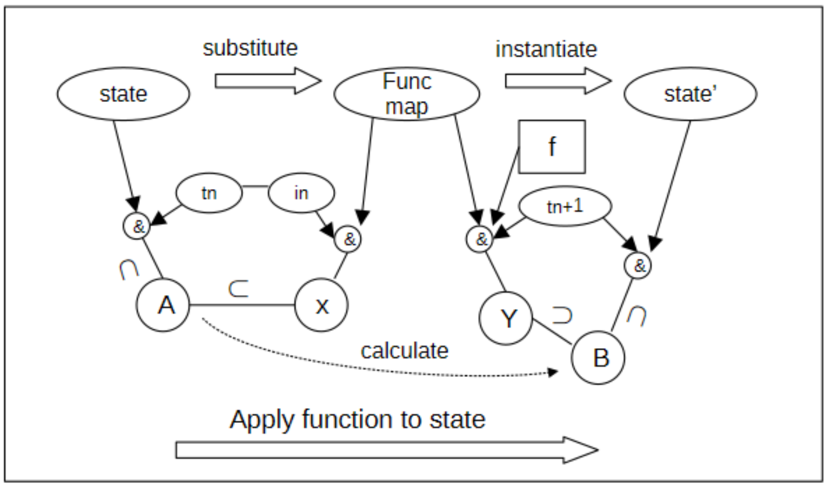

Figure 40: Apply function

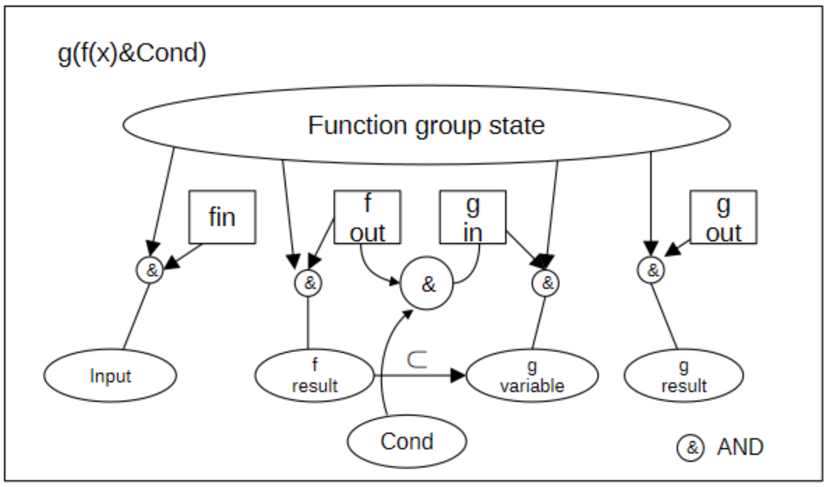

Figure 41: Function group

## 8.6 Built-in functions and their usage

A built-in function is defined as an external function that uses input nodes and external observations to select and activate output nodes. The action on the input is defined inside the built-in function. It can also be used for input and output from outside SOL. The nodes used in the built-in functions are Boolean values, but the built-in functions can obtain the actual objects corresponding to the nodes. Specifically, scalar values such as integers, vector values, objects, etc. are associated and used by the built-in functions for calculations. The results are generated as nodes corresponding to scalar values, etc., and conditionally combined with the input state.

If you want to provide feedback to scalar values with intermediate values, you need to use a feedback method that is unrelated to probability. Vector values, etc. cannot be managed by probability alone. Therefore, values with intermediate values are realized by preparing a mechanism other than SOL's feedback. For example, generating functions such as Fourier series and Gaussian noise, as well as addition and inner product operations that integrate these, are also provided as built-in functions. The selection and connection of these built-in functions can be realized by SOL's mappings and functions.

In the example shown in the figure below, the real vector value of the character Token is used to calculate the inner product using a built-in function to generate a Result value and generate the corresponding node.

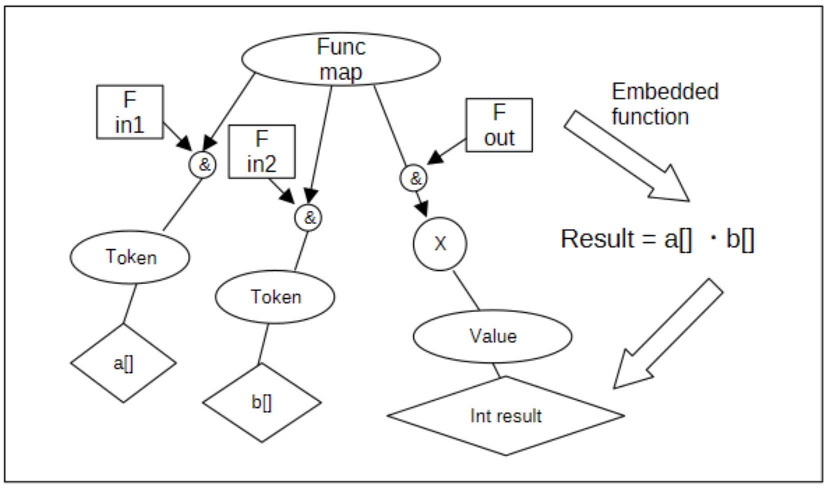

Figure 42: Embedded function

## 8.7 Comparison with Attention in Machine Learning

Attention[2], the basis of the Transformer, which has been widely used in recent years, is a mechanism that has produced remarkable results in machine learning. This mechanism has application capabilities that have not been achieved with previous logic circuits. However, this mechanism is heuristic and does not have theoretical consistency.

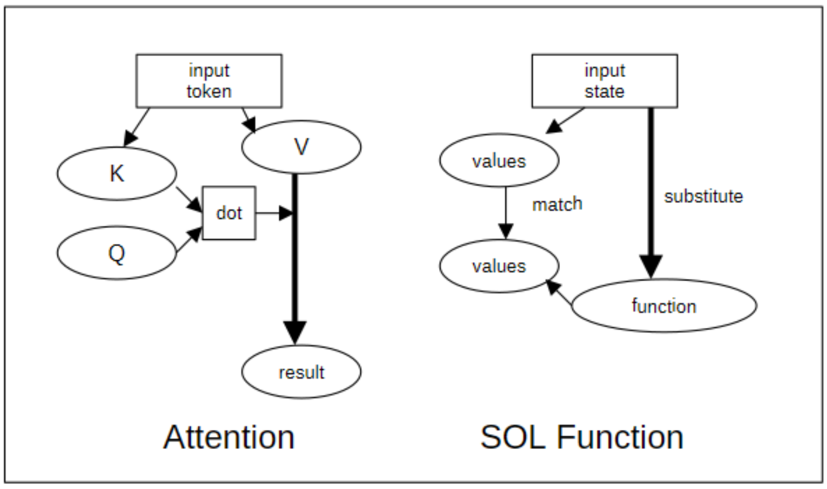

Figure 43: Similarities between the Attention mechanism and SOL functions

Attention's KQVs are both N-dimensional vectors that correspond to tokens such as characters. In contrast, SOL's states are unrestricted hierarchical data structures.

The comparison of Attention's K and Q by inner product operation corresponds to the comparison of the input conditions and input states in SOL's functions. The partial state of SOL's input State is the Query in Attention, and the comparison part of the SOL function becomes the Key in Attention. As a result, the original State is substituted into the function like the Value of Attention, and partial modification is performed. The logical operations that are the output conditions of SOL functions can be compared to the modification action on the Value of Attention.

As described above, SOL's mapping and state functions can be compared to the mechanism of Attention. If anything, SOL's functions can be considered a more generalized and qualified version of the mechanism of Attention.

## 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 automatically generates deterministic digital logic with a probability of 100% or 0% by adding logical operations and mappings according to the feedback results to the link probability connected by association. By using the propagation probability of the pair $P^{11}$ and $P^{00}$, accurate insertion of logical operations is possible. In this case, the network is

partially replicated to add logical operations. In the case of fluctuations in $P^{00}$, it is also possible to partially replicate the network and generalize it.

7. By forming associations for fluctuations in values over time and adding further logical operations for feedback to the associations, it is possible to generate generalized logic circuits between the input and output of memory, etc. Furthermore, generalized replication of the network autonomously forms general sequential circuits. The generalized sequential circuits are used many times to become functions.

8. By using mappings hierarchically, the hierarchical data structures and functions of general software can be expressed. By managing the containment relationship of the propagation sets between these mappings, we can check the match of parameters for a state with a data structure and add the result of the function to the state. To do this, we use "assignment" and "instantiation" between mappings defined in SOL.

SOL is built only on basic mathematical principles such as probability, sets, and mapping. It does not imitate biological neurons. In addition, it hardly uses any non-rigorous heuristics such as the non-linear activation function of 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 SOL. Existing machine learning such as neural networks can even be considered to be an approximate implementation of part of SOL. Therefore, if we aim for accurate and safe machine learning, we have no choice but to introduce the principles of SOL in some form.

Future challenges include demonstrating operation through implementation, making the implementation more efficient, simplifying, and parallelizing. In addition, whether SOL can comprehensively autonomously generate logic circuits and software will be a future research topic.

SOL has clear connections to mathematical logic, category theory, etc., so there is room for generalization of the theory.

# 10 Glossary

## 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.

## Activation value

The activation state passing through the nodes and links of a network, and is composed of a propagation set and a propagation probability.

## 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

This shows how link fluctuations affect the results of link propagation. Specifically, when the propagation probability of a link changes from 0 to 1, if the probability of the propagation destination in the network definitely changes to 1, the effect coefficient is 1. On the other hand, even if a link changes from 0 to 1, if it has passed through an OR operation along the way, the probability at the end will not change. In this case, the effect coefficient is 0.

### 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] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2010.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, *Attention Is All You Need*, In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.