都留文科大学研究紀要　第87集（2018年３月）
The Tsuru University Review, No.87（March, 2018）

Diagrammatic Calculus for Order-sorted Logic

寺川　宏之　吉岡　　卓

TERAKAWA Hiroyuki, YOSHIOKA Suguru

Abstract

In the field of artificial intelligence, order-sorted logics, that have subsumption relations between sorts, are widely utilized for structural knowledge representations. Among them, dual hierarchical systems, that have subsumption relations also in events (predicates) as well as sorts (terms), can realize superb efficiency in logical reasoning. In ordinary cases, such subsumption relations organizes a lattice, the operations of ‘join’ and ‘meet’ being assumed. However, in dual systems, the description of two different lattices of predicates and sorts, makes us hard to find reasonability between atomic formulae. In this paper, we propose a representation of cellular table for the dual hierarchies, assigning a Gödel number to each node to identify its spacial position. Thus, we can describe two lattices in one table, and in addition, the reasonability between two atomic formulae is reduced to simple numerical calculation. Therefore, (i) the reasonability between two distant atomic formulae and (ii) the scope of partial negation are easily displayed, and in addition, (iii) that the whole table is adequately maintained even in case new subsumption relations are added. We implemented a deduction system on a computer, and showed its efficiency.

1　Introduction

In order-sorted logics, a set of objects are classified as a **sort** and *subsumption* relations are given in sorts. Knowledge can be expressed structurally by these subsumptions. In order to represent hierarchical knowledge, order-sorted logics are widely applied in the field of artificial intelligence [8].

In ordinary cases, such subsumption relations organizes a lattice where the operations of ‘join’ and ‘meet’ are assumed and the *top* element $\top$ and the *bottom* element $\perp$ uniquely exist, respectively. Usually these sorts are used to classify terms, that are arguments of predicates. As a special case of the logic, we consider *dual lattice* systems [12]; that is, predicates have subsumption relations as well as terms, viz., term objects organize sort hierarchies and predicates objects organize predicate hierarchy independently. For knowledge representations, this dual system is very useful. In this paper, we show an effective representation for this knowledge representation of dual lattice.

In the following section, we summarize order sorted logic and the representation with event of channel theory. In Section 3, we define a sentence for our logic. In Section 4, we suggest that the representation TDL for dual lattice, assigning a Gödel number for two hierarchies. In Section 5, we define a transformation rules of TDL. In Section 6, we prove a soundness of TDL. In Section 7, We explain the reasoning mechanism by our TDL representation, and its efficiency. In Section 8, we conclude and discuss future subjects.

2　The dual lattice structure

A sort hierarchy, for maintenance of a knowledge-base and concise representation, is very useful. A subsort declaration $a \sqsubseteq b$ has an intention for an element x of whole objects as follows:

$$\forall x[a(x) \rightarrow b(x)] \tag{1}$$

That is, it is only natural that the semantics request for a relation $[\![a]\!] \subseteq [\![b]\!]$.

Usually these sorts are used to classify terms, that are arguments of predicates. As a special case of the logic, we consider *dual lattice* systems [12]; that is, predicates have subsumption relations as well as terms, viz., term objects organize sort hierarchies and predicates objects organize predicate hierarchy independently. For example, the act ‘roasting’ imply ‘cooking.’ This relation allow us to declare the hierarchical structure between *roast* and *cook*. For knowledge representations, this dual system is very useful.

In this paper, we deal with sort hierarchy for predicate and sort, matters that require attention is, predicate hierarchy is intended for sets of state(event) [1].

Therefore, it is not suitable to represent for equation of sort as predicate like as follows:

$$\forall x[roast(x) \rightarrow cook(x)] \tag{2}$$

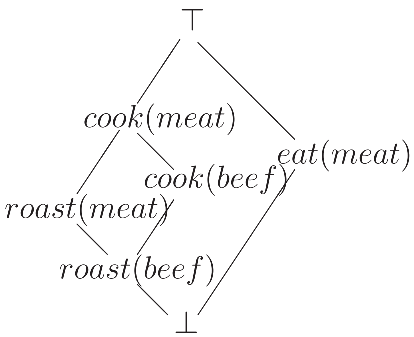

Figure 1 : Representation of dual lattice

[1] Usually, whenever a predicate p bite a term x, The $p(x)$ is interpreted that x has an attribute a property p. However, if we want to treat a sets of state, we must refer to event but property. [8, 9].

By Channel-theory[2], if there is one classify the token e by type '*roast*', it is represented as $e \models roast$. Thus, we must improve the (2) to (3).

$$\forall e[e \models roast \rightarrow e \models cook] \tag{3}$$

In this paper, '$\sqsubseteq_p$' represents a subsumption resolution of predicate, and '$\sqsubseteq_s$' represents a conventionally sort resolution. For example, if we define the $roast \sqsubseteq_p cook$, adding to $beef \sqsubseteq_s meat$, we can construct a dual lattice in Figure 1.

Whenever a hierarchical relation of dual lattice brings about a combination of (1) and (3), that is, for set $\mathcal{P}$ of predicate and set $\mathcal{S}$ of sort, $p, q \in \mathcal{P}$ and $a, b \in \mathcal{S}$,and $p(a) \sqsubseteq q(b)$, we represent as follows:

$$\forall e \forall x[e \models p \;\&\; a(x) \rightarrow e \models q \;\&\; b(x)] \tag{4}$$

where '$\sqsubseteq$' expected the suffix is a composite hierarchical structure. For correctly description above an equation, by set $\mathcal{T}$ of situated tokens and set $\mathcal{O}$ of object, we gain the equation as follows:

$$[\![p(a)]\!] \equiv \{(e, x) | e \in [\![p]\!] \subseteq \mathcal{T}, x \in [\![a]\!] \subseteq \mathcal{O}\}$$

and we define that

$$p(a) \sqsubseteq q(b) \quad \text{iff} \quad [\![p(a)]\!] \subseteq [\![q(b)]\!]$$

That is, dual sort is an element of a Cartesian-product $\mathcal{P} \otimes \mathcal{S}$ from the two group, the whole of the dual sort consist a lattice.

This composite lattice is representable by preparing the Cartesian-product from two lattice in advance. However, two lattice have a possibility of updating, it is inefficient to previously calculate a product, and it is difficult to maintain. In this paper, we show a system that the dual lattice consists dynamic. Therefore we propose a tabularization for dual lattice by using row and column. By our way, we show (i) the efficiently visual language and (ii) the easily reasoning system for dual lattice.

3 Order-sorted Logic and Sets of State

In this paper, we add a new hierarchical structure to the conventional order-sorted logic for the sets of state(event). That is, we define the sentence as follows:

Defnition 1. *The language L consists of following symbols.*

1. $S(= \{s_1, ..., s_n\})$is a finite set of **sorts**.
2. $P(= \{p_1, ..., p_n\})$is a finite set of **events**, called the *types*.
3. A set X of objects to be classified, called *elements* of a classification $\langle X, S, : \rangle$.
4. A set E of objects to be classified, called *tokens* of a classification $\langle E, P, \models_P \rangle$.
5. $\neg, \wedge, \vee, \rightarrow, \forall, \exists$ are logical connectives and quantifiers
6. (,), & is supplementary symbols.

[2]Situation theory or Information flow [5].

Defnition 2. We call the tuple $\Sigma = \langle \sqsubseteq_S, \sqsubseteq_P, :, \models_P, \otimes \rangle$ *the signature of the language L.*

1. 2-element subsets $\sqsubseteq_S$ of S called sort *subsumption* relations.
2. 2-element subsets $\sqsubseteq_P$ of P called *subsumption* relations of event.
3. A binary relation : between X and S that selecting a token out of X in a sort s.
4. A binary relation $\models_P$ between E and P that tell one which tokens are classified being of which types.
5. A binary relation $\otimes$ represent a Cartesian-product between group S and group P. And we represent $P \otimes S$ to P(S) simplify.

Defnition 3. *We define binomial operator 'union (join)' and 'intersection (meet)' to each sorts and events as follows:*

$$s_1 \sqcap s_2 = glb\{s_1, s_2\}, \qquad s_1 \sqcup s_2 = lub\{s_1, s_2\},$$
$$p_1 \sqcap p_2 = glb\{p_1, p_2\}, \qquad p_1 \sqcup p_2 = lub\{p_1, p_2\},$$

where glb is the greatest lower bound and lub is the least upper bound. We define 'top' as $\top_S = \sqcup S$, $\top_P = \sqcup$P, and 'bottom' as $\bot_S = \sqcap S$, $\bot_P = \sqcap P$.

Then, there are following mathematical properties:

$$s_i \sqsubseteq_S s_i \sqcup s, \quad s_i \sqcap s \sqsubseteq_S s_i, \quad p_i \sqsubseteq_P p_i \sqcup p, \quad p_i \sqcap p \sqsubseteq_P p_i.$$

Therefore, set of sorts S and set of events P form lattices, respectively.

Defnition 4. *The relation between subsumption relation and implication is as follows:*

1. $s_1 \sqsubseteq_S s_2$ *then* $\forall x\, p(x : s_1) \to p(x : s_2)$
2. $p_1 \sqsubseteq_P p_2$ *then* $\forall x\, p_1(x : s) \to p_2(x : s)$
3. $p_1 \sqsubseteq_P p_2$ *then* $\forall x\, \neg p_2(x : s) \to \neg p_1(x : s)$
4. $s_1 \sqsubseteq_S s_2$ *then* $\forall x\, \neg p(x : s_2) \to \forall y\, \neg p(y : s_1)$

3.1 Semantics

Defnition 5. *A structure $M = (U, I)$ satisfies the following condition:*

1. A set U is nonempty.
2. A function I satisfies as follows:
   For s $\in$ S, $I(s) \subseteq U$,
   For $s_i \sqsubseteq_S s_j$, $I(s_i) \subseteq I(s_j)$.

Defnition 6. *A structure $M' = (E, I')$ satisfies the following condition:*

1. A set E is nonempty.
2. A function I' satisfies as follows:
   For p $\in$ P, $I'(p) \subseteq E$,
   For $p_i \sqsubseteq_P p_j$, $I'(p_i) \subseteq I'(p_j)$.

Defnition 7. *For events $p, p_1, p_2 \in P$ and sorts $s_1, s_2 \in S$, we define an interpretation of the binary operator $\otimes$ as follows:*

1. $[\![p(a)]\!] \equiv \{(e, x) | e \in [\![p]\!] \subseteq \mathcal{T}$, $x \in [\![a]\!] \subseteq \mathcal{O}\}$
2. $[\![p \otimes s_1 \sqcap s_2]\!] = [\![p \otimes glb(s_1, s_2)]\!]$
3. $[\![p \otimes s_1 \sqcup s_2]\!] = [\![p \otimes lub(s_1, s_2)]\!]$
4. $[\![p_1 \sqcap p_2 \otimes s]\!] = [\![glb(p_1, p_2) \otimes s]\!]$
5. $[\![p_1 \sqcup p_2 \otimes s]\!] = [\![lub(p_1, p_2) \otimes s]\!]$

4 Tabularisation of Dual Lattice

4.1 Representation of hierarchy using the prime number label

Usually, the hierarchical structure is expressed by HASSE-diagram. Although HASSE-diagram is intelligible for our visual recognition, it is difficult to implement it on a computer. We may be able to implement the dual lattice system, using the Cartesian-product of predicates and terms; however, the representation requires tremendous memory space. In this paper, we propose a more concise **TDL(Tabularisation of Dual Lattice)**.

First we introduce the TDL, and after that, we define syntax and semantics of the representation system. The TDL is consisted of following symbols: closed-curve(cell), line, arc, ⊗, consist, ∘, negation symbol(¬). We intend to represent a lattice in one-dimensional array, that is a sequence of cells. Each cells are labeled sort or event symbol, represent a subset of the domain. Cells are articulated by **boundaries**, that is a subsumption relation. Each sort and event has a prime number that shows a **path** from ⊤ to ⊥, and when there is the **dividable** relation between two sorts, we interpret that there is a subsumption relation.

The line represents the inclusive-or, and ⊗ is used to assert the non-emptiness of a represented set.

For example, let us consider the sort hierarchy as shown in the left side of Fig 2. Then, the representation by the cell proposed in this paper is shown in Fig 3. Each sorts are assigned by power of prime number as described in Fig 3. For example, assignment of a prime number label becomes like the right side of Fig 2.

That is, a prime number 2 is assigned to the path $\{\top \rightarrow C \rightarrow A \rightarrow \bot\}$ and the factorial

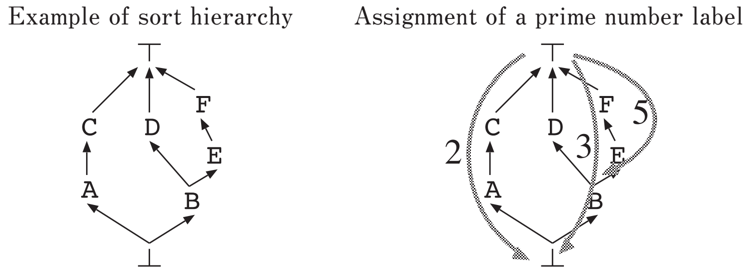

Figure 2 : Prime number labeling

$\bot^{2^2\times3\times5^2}$	$B^{3\times5^2}$	$A^{2^2}E^{5^2}$	$C^2D^3F^5$	$\top$

Figure ３: Hierarchical structure by cellular representation

of 2 is assigned according to the depth. Thus, only a related sorts has the aliquot relation by assigning the factorial of a different prime number according to a different path.

Note that the number of 3×5^2 of the sort B in Fig. 3 is dividable by 5 of F; thus we can infer that the sort F is the higher rank concept of sort B. Because we treat a dual lattice system and each of the predicate lattice and the sort lattice would be made into an array, we need to layout two arrays on a two-dimensional plane; thus, the representation of a dual lattice system would be a **table**.

Next, the formal definition to cellular representation is given.

4.2 Syntax

Alphabet The Cellular language L_C consists of following symbols.

▭	C	cell
$s_1, s_2,...$	S	sort symbols
$p_1, p_2,...$	P	state symbols(event)
2, 3,...	G	prime number label
¬	N	negation symbol
⊗	E	⊗
—	L	line
*	T	total relation

A horizontal array of cells is called a **column**, and the perpendicular array is called a **row**, and between cells is called **boundary**. Those cells which face each other by boundaries are called **adjacent-cells**.

Now, let there is a subsumption relation $\{A \sqsubseteq_S B\}$. The left side in the Fig 4 represents $\exists x(x : A)$ and $\exists x(x : B)$ by TDL. And the right side in the Fig 4 represents $\exists x(x : B \wedge x : \neg A)$.

Signature We call a tuple $\langle Dv, Dr, T\rangle$ the signature of the cellular language L_C, where:

Dv	dividable relation
Dr	drawing rules
T	transformational rules

A drawing rule assigns a prime number label to each sort and each event. A transformational rule deduces possible predicate expressions, taking a cell of predicate and a cell of term.

Extended Gödel Number Prior to introduce the extended Gödel number, we shortly review the original Gödel number [15]. For each character X_i of a formal language, we uniquely assign a natural number. Let the natural number of character X_i is n_i. Then, if given a sequence $\mathbf{S}(=X_1X_2\cdots X_v)$ of characters, we assign the number $G(S)(=2^{n_1}\cdot 3^{n_2}\cdot\cdots\cdot p_v^{n_v})$, where p_i expresses the i-th prime number, and call the Gödel number of S.

A^{2^2}	B^2
$\otimes^{2^2}$	

A^{2^2}	B^2
	$\otimes^{2}$

Figure 4 : The example of TDL

Now, we introduce our **extended Gödel number**, that is the number labels tacked to each cell. And, a prime number is assigned for every different path $X_1, X_2, \cdots, X_v$ on a HASSE-diagram, and each sort is given a value n_i to $X_i(\in S)$. If a sort is reachable from multiple paths from the superlative sort, the product of the numbers of the multiple supersorts. Henceforth, the label (number) of sort 's' is expressed by $\lceil s \rceil$ and the label of event 'p' is expressed by $\lceil p \rceil$.

Now, we write $p_1\ Dv\ p_2$ that the relation that p_1 is dividable by p_2. The number label by the binomial operator $\sqcap$ and $\sqcup$ is defined as follows:

$$\lceil s_1 \sqcap s_2 \rceil = LCM\{\lceil s_1 \rceil, \lceil s_2 \rceil\}, \quad \lceil s_1 \sqcup s_2 \rceil = GCD\{\lceil s_1 \rceil, \lceil s_2 \rceil\},$$
$$\lceil p_1 \sqcap p_2 \rceil = LCM\{\lceil p_1 \rceil, \lceil p_2 \rceil\}, \quad \lceil p_1 \sqcup p_2 \rceil = GCD\{\lceil p_1 \rceil, \lceil p_2 \rceil\}$$

where LCM denotes the least common multiple and GCD denotes the greatest common divisor. Therefore, the labels of $\top_S$, $\bot_S$, $\top_P$ and $\bot_P$ are defined that

$$\lceil \top_S \rceil = \{\lceil \bigcap_i s_i \rceil \mid s_i \in S\}, \quad \lceil \bot_S \rceil = \{\lceil \bigcup_i s_i \rceil \mid s_i \in S\},$$
$$\lceil \top_P \rceil = \{\lceil \bigcap_i p_i \rceil \mid p_i \in P\}, \quad \lceil \bot_P \rceil = \{\lceil \bigcup_i p_i \rceil \mid p_i \in P\},$$

where S is the whole set of sorts and P is the whole set of events.

Moreover, we correspond the product to $\otimes(\in P \times S)$. That is, for prime number G_1 of sort S and prime number G_2 of event P, $\lceil P \otimes S \rceil = \lceil P \rceil \times \lceil S \rceil$. Then, we gain next theorem.

[**Theorem 4.2.1**]

A prime number in some cell specify a prime number of sort and event uniquely.

[*proof*]

Now, let a prime number is $\Pi_i P_i^{f_i}$ (P_i is a prime number) in some cell. We suppose that $\Pi_i P_i^{f_i}$ get the several product. That is, for $\Pi_i P_i^{f_i}$, by the product of a prime number of sort and event,

$\Pi_i P_i^{f_i} = \Pi_j P_j^{f_j} \times \Pi_k P_k^{f_k}$ $(1 \le j, k \le i)$

$\Pi_i P_i^{f_i} = \Pi_l P_l^{f_l} \times \Pi_m P_m^{f_m}$ $(1 \le l, m \le i)$.

Now, we can consider two resolution that $\Pi_j P_j^{f_j}$, $\Pi_l P_l^{f_l} \in \{\lceil S \rceil\}$, $\Pi_k P_k^{f_k}$, $\Pi_m P_m^{f_m} \in \{\lceil P \rceil\}$. For the uniquely of prime number resolution, $GCD\{P_j^{f_j}, P_m^{f_m}\} \neq 1$. This is an inconsistent. Therefore, every prime number in some cell specify a prime number of sort and event uniquely. [Q.E.D]

Drawing Rules of TDL A drawing rule assigns a prime number label to sorts and predicates. Because each of the predicate lattice and the sort lattice becomes a one-dimensional array, the application of drawing rules results in a two-dimensional table. Prime number labels are assigned with the following algorithm:

Algorithm 1 (EGN Labeling)

1. The counter of a prime number is set to c. c ← 2.
2. Sorting declaration ($\alpha \sqsubseteq \beta$) is input. Otherwise quit.
3. If β does not exist in a cell, then *goto* 3.2.
   - 3.1 If α already exists in a cell, then $\lceil \gamma \rceil \leftarrow \lceil \gamma \rceil \times \lceil \beta \rceil$ for each of $\{\gamma \| \lceil \gamma \rceil\ Dv \lceil \alpha \rceil\}$. Otherwise, $\lceil \alpha \rceil \leftarrow c \times \lceil \beta \rceil$, $c \leftarrow$ *new* c, and *goto* 2.
   - 3.2 If α already exists in a cell, then $\lceil \beta \rceil \leftarrow c$, and for each of $\{\gamma \| \lceil \gamma \rceil\ Dv \lceil \alpha \rceil\}$, $\lceil \gamma \rceil \leftarrow \lceil \gamma \rceil \times c$. Moreover, c ← *new c* and *goto* 2.
   - 3.3 $\lceil \alpha \rceil \leftarrow c^2$, $\lceil \beta \rceil \leftarrow c$, $c \leftarrow$ *new* c, and *goto* 2.

A cell either in a column or in a row represents the depth from $\top$. Thus, those sorts in a cell have the same depth from the top of the lattice.

4.3 Semantics

Since the dividable relation in labels is a subsumption relation, we first give the following definition.

Defnition 8. *The interpretation of a label*

For a natural number n and a vertex V on a HASSE-diagram, the relation $\vdash V$ is interpreted that $(\exists m)(m\ Dv \lceil V \rceil)$. If an atomic formula is deducible, that is, $(\exists m)(m\ Dv\ p)$, we write $Bew(p)$, and is called that "p is the Gödel number of **beweisbar** sort." If there is a subsumption relation $(s_1 \sqsubseteq s_2)$ between sort s_1 and sort s_2, then we write $Bew_{\lceil s_1 \rceil}(\lceil s_2 \rceil)$, that is, $\lceil s_1 \rceil\ Dv \lceil s_2 \rceil$.

Next, we define a subsumption relation as an implication.

Defnition 9. *The interpretation of subsumption relation*

Given a set of tokens X and a set of sorts S, we define the quadruple $\langle X, S, :, \rightarrow \rangle$, where the triple $\langle X, S, : \rangle$ is a classification and ':' is a function that chooses a token out of X in a sort s and '→' is an implication. That is, for $x \in X$, $s_j \in S$, there is a relation '$[\![x]\!] \in [\![s_j]\!] \subset$U' . In addition, '$[\![\ ]\!]$' expresses an interpretation.

Then, for $\forall\ x \in X$, $\forall\ s_j \in S$, we define the sort resolution as follows:

$$x : s_j \text{ and } s_j \rightarrow s_k \text{ then } x : s_k \tag{5}$$

Now, by the formula (5) and the theorem 4.2.1, we obtain the one-to-one mapping to $\langle X, S, :, Dv \rangle$ from $\langle X, S, :, \rightarrow \rangle$. Therefore, we give the following definition (6):

$$x : s_j \text{ and } [s_j]\ Dv\ [s_k] \text{ then } x : s_k \tag{6}$$

Defnition 10. *The interpretation of event resolution*

Given a set of tokens E and a set of sorts P, we define the quadruple $\langle E, P, \models_P, \rightarrow \rangle$.

Then, for $\forall e \in E$, $\forall p_j \in P$, we define the event resolution as follows:

$$e : p_j \text{ and } p_j \rightarrow p_k \text{ then } e : p_k \tag{7}$$

Now, by the formula (7) and the theorem 4.2.1, we obtain the one-to-one mapping to $\langle E, P, \models, D_v \rangle$ from $\langle E, P, \models, \rightarrow \rangle$. Therefore, we give the following definition (8):

$$e : p_j \text{ and } [p_j]\ Dv\ [p_k] \text{ then } e : p_k \tag{8}$$

Defnition 11. *The interpretation of dual lattice structure*

Given a set of sorts S and a set of events P, we define the quadruple $\langle P, S, \otimes, \rightarrow \rangle$. Then, for (6), (8), $\forall s \in S$ and $\forall\ p_j \in P$, we give the following definition :

$$p_j \otimes s \text{ and } [p_j]\ Dv\ [p_k] \text{ then } p_k \otimes s$$

Defnition 12. *The interpretation between an elements in TDL and sets*

There is a relation of one-to-one mapping between sort and set. Let U_s is a set of sort s. In this paper, the cell included in sort s, if there is such an area, interpreted by $U_{s \setminus s_i}$, where s_i Dv s.

Now, we can express the representing facts by means of situation-theoretic terminology infons and define a homomorphism h from facts about diagrams to facts about sets as follows:

$h \ll labeled, s; i \gg\ = \ll Empty, U_s; i \gg$

$h \ll In, \otimes^p, c; i \gg\ = \ll Exist, element, U_{s^p \setminus s_i^{p_j}}; i \gg$

where p is a prime number and c is a cell.

Defnition 13. *The interpretation of transformation*

Let C is set of cells, Δ is set of Cs.

Therefore, the content of a C, $Cont(C)$, is defined as the set of the represented facts: $Cont(C) = \{h(\alpha) | C \models \alpha\}$, where h is the homomorphism, α is the wff. And $Cont(\Delta) = \cup_{C \in \Delta} Cont(C)$. Thus, let U be a set such that it is an universe of objects, and let Sit be a subset that is it closed under $\cup$ and $\setminus$, we define what it means for a basic infon to be supported by one of these situation s, as follows:

$s \models \ll Empty, x; 1 \gg$ iff $x \in s$ *and* $x = \varnothing$

$s \models \ll Empty, x; 0 \gg$ iff $x \in s$ *and* $x \neq \varnothing$

$s \models \ll Set, x; 1 \gg$ iff $x \in s$

Let Σ_1, Σ_2 be sets of infons, we define the involvement relation as follows:

$\Sigma_1 \supseteq \Sigma_2$ iff $_{\forall s \in Sit}(_{\forall \alpha \in \Sigma_1} s \models \alpha \rightarrow {}_{\forall \beta \in \Sigma_2} s \models \beta)$

Therefore, for Δ and C,we define as follows:

$\Delta \models C$ iff $\Delta \supseteq C$.

Defnition 14. *The interpretation of cell transformation*

Given a formula K, a formula K' obtained by an application of a transformational rule is a logical consequence from K.

Namely, the cellular representation can perform reasoning transitively, as long as two formulae are in the dividable relation. Thus, in our system, a step of reasoning is a numeric division.

5 Transformational Rules of TDL

The transformational rules work as deductive rules in the ordinary meaning. The rules fill cells up with atomic formulae, which are combinations of events and terms.

We define (i) the rules between cells and formulae, and (ii) the rules of transformation of cells as follows.

(i) The rules between cells and formulae

The deductive rules are defined as follows:

$\exists$ − **Apply** For $\exists x\ P(x : S)$, the ⊗s filled in the cell which corresponded P and S_i, where $[S_i]\ Dv\ [S]$. And each ⊗s labeled $[P] \times [S_i]$, joined by line.

$\exists$ − **PreApply** For $\exists x\ P(x : S)$, the ⊗s filled in the cell which corresponded P_i and S, where $[P_i]\ Dv\ [P]$. And each ⊗ s labeled $[P_i] \times [S]$, joined by line.

$\exists$ − **Observe** If there is a ⊗ in some cell, we get a formula $\exists x\ P(x : S)$, where $[\otimes] = [P] \times [S]$. And, for hierarchical structure, for $[S]\ Dv\ [S_i]$, $[P]\ Dv\ [P_i]$, we gain formulae $\exists x\ P_i(x : S_i)$.

$\neg\exists$ − **Apply** For $\neg\exists x(x : S)$, If a ⊗ filled in a column labeled S is dividable $[S]$, we erase such ⊗ and join by line the disconnected sequence. And, If the ⊗ labeled * connected, it erased too.

$\neg\forall$ − **Apply** For $\neg S$, we apply the rule of $\neg\forall$ − *Apply*, and erase the $\{S_i | [S_i]\ Dv\ [S]\}$.

$\exists$ − **¬PreApply** For $\exists x\ \neg P(x : S)$, the ¬⊗s filled in the cell which corresponded P_i and S, where P_i has a relation $[P_i]\ Dv\ [P]$. If there is a ⊗ in the corresponded cell, it erased, and join by line the disconnected sequence. And, If the ⊗ labeled * connected, it erased too.

$\forall$ − **¬PreApply** For $\forall x\ \neg P(x : S)$, the ¬⊗s filled in the cell which corresponded P_i and S_i, where P_i has a relation $[P_i]\ Dv\ [P]$ and S_i has a relation $[S_i]\ Dv\ [S]$. If there is a ⊗ in the corresponded cell, it erased, and join by line the disconnected sequence. And, If the ⊗ labeled * connected, it erased too.

¬ − **PreObserve** If there is a ¬⊗ in some cell, we get a formula $\exists x\ \neg P(x : S)$, where $[\otimes]$

$= [P] \times [S]$.

Inconsistent − Information If the whole of sequence erased by the rule of $\neg\forall -$ *PreApply*, it is inconsistent.

(ii) The rules of transformation of cells

We can apply the rules for cell as follows:

Erase the cell we erase a cell which has no label.

Erase the sequence we erase the disconnect line.

Expand the sequence we join disconnect sequences by line.

Now, we define the interpretation of negation in TDL as follow:

Defnition 15. *The interpretation of negation*

We give two kinds of negation. The first is the negation of a formula itself (classical negation), and the second is the complementary set of a formula. For a sort, diagrammatic elements erased in cell, thus it represent the fact that the sort with a negation symbol is interpreted as a complementary set. A predicate with a negation symbol is interpreted as the classical negation.

Having introduced the concept of a complementary set, we can represent the word 'except' in our ordinary language formally, that has been hard in the conventional HASSE-diagram derivation, and can realize more versatile knowledge representation.

6 Soundness

We define whenever one C is obtainable from a set Δ, the content of diagrams in Δ involves the content of C.

[**Theorem 6**] Soundness of TDL

We want to prove that if $\Delta \vdash C$, then $\Delta \models C$.

[*proof*]

Suppose that $\Delta \vdash C$. By definition, there is a sequence of $< C_1, C_2, ..., C_n >$ such that $\Delta = C_n$. And for each $1 \leq k \leq n$ either (a)there is some C' such that $C' \in \Delta$ and $C' \equiv C_k$, or (b)there is some C' such that for some $i < k$, a rule of transformation allows us to get C′ from C_i and $C' \equiv C_k$. We show by induction on the length.

(Basis Case)

That is, $C_1 \equiv C$. Since there is no previous diagram in this sequence, it should be the case that there is some C' such that $C' \in \Delta$ and $C' \equiv C_1$. Thus,

1. $C' \equiv C$

2. $Cont(C') = Cont(C)$ (by 1)
3. $Cont(C') \subseteq Cont(\Delta)$ (since $C' \in \Delta$)
4. $Cont(C) \subseteq Cont(\Delta)$ (by 2 and 3)

Therefore, $\Delta \models C$.

(Inductive Step)

Suppose that for any C if C has a length of a sequence less than n, then $\Delta \models C$. We want to show that if C has a length of a sequence n then $\Delta \models C$. That is $C_n \equiv C$. If there is some C' such that $C' \in \Delta$ and $C' \equiv C_n$, then as we proved in the basis case, $\Delta \models C$. Otherwise, it must be the case that there is some C' such that for some $i < n$, a rule of transformation allow us to get C' from C_i (*). Now, we represent a $TR(C_i)$ that we apply a transformation rule to C_i, then

$\mathrm{TR}(C_i) = C_i \cup C_j \ (C_j \in \Delta, j < n)$ —— (**)

By our inductive hypothesis, $\Delta \models C_i$ and $\Delta \models C_j$. That is,
$Cont(\Delta) \supseteq Cont(C_i)$ and $Cont(\Delta) \supseteq Cont(C_j)$. Therefore,

$Cont(\Delta) \supseteq (Cont(C_i) \cup Cont(C_j))$.

Moreover, for (*)(**),

$(Cont(C_i) \cup Cont(C_j)) \supseteq Cont(C')$.

By the transitivity of the involvement relation,

$Cont(\Delta) \supseteq Cont(C')$.

Since $C' \equiv C_n$ and $C_n \equiv C$, $C' \equiv C$. Hence, $Cont(C') = Cont(C)$. Accordingly,

$Cont(\Delta) \supseteq Cont(C)$.

Therefore, $\Delta \models C$. [*Q.E.D*]

7　The deductive system using TDL

We have implemented this TDL system on a computer. Any user give term subsumption relations and predicate subsumption relations to the system, and the system draws the cell according to the input. we could represent subsumption relations by numerical dividability. Thus, we can easily calculate the logical provability of two formulae, which were placed in distant places in the table, and we improved the visibility of the relation of two formulae. We estimate the cost for holding an information of subsumption relation to one hierarchy. For example, we consider a following set S of subsort declarations.

$$S = \{dolphin \sqsubseteq_S mammal,\ human \sqsubseteq_S mammal,\ swallow \sqsubseteq_S feather,\ mammal \sqsubseteq_S animal,\ feather \sqsubseteq_S animal\}$$

Thus, we obtain the cellular representation proposed in this paper as Figure 5.

Let the new subsumption relation $\{male \sqsubseteq_S human\}$ is added. If we represent by the Cartesian-product, since it is necessary to calculate the transitive relation ($male \sqsubseteq_S mammal$, $male \sqsubseteq_S animal$), we need complexity of $\mathcal{O}(n^2)$ for the number of sort n. On the

dolphin	human	swallow	mammal	feather	animal
2^3	$2^2 \times 3$	2×5^2	2^2	2×5	2

Figure 5 : Representation of TDL

male	dolphin	human	swallow	mammal	feather	animal
$2^2 \times 3^2$	2^3	$2^2 \times 3$	2×5^2	2^2	2×5	2

Figure 6 : The addition of new subsort declaration

other hand, we obtain the TDL proposed in this paper as Figure 6 for the new subsort declaration.

By the TDL, when we add a new subsumption relation, we need a complexity $\mathcal{O}(n)$ for the number of sort n to obtain Figure 6. Because the system rewrites only labels of a subset of sorts, the complexity of calculation is greatly cut down.

8 Conclusion

In this paper, we introduced a representation system TDL for dual lattice systems, in which both of predicates(event) and sorts have subsumption relations. Replacing a lattice for a one-dimensional array, and drawing a planer table for the arrayed lattices, we defined a deduction system that properly fill out the cells in the table. In order to identify the position of each predicate and sort in lattices, we assigned a unique Gödel number onto it, and we could represent subsumption relations by numerical dividability. Thus, we can easily calculate the logical provability of two formulas, which were placed in distant places in the table, and we improved the visibility of the relation of two formulae. This method is also advantageous in denoting the area of the negation of a formula, considering how the negation affects upon adjacent cells (by the effect of *freeride* [2]). Furthermore, the operation of adding/deleting a subsumption relation is easily implemented as algorithms to reassign numbers, and this fact enables us easy to maintain a large knowledge base in various application fields.

As future subjects, we consider a plural terms of formulae, and also consider how the two kinds of negation (classical negation and complementary set) affects each other in the table. In addition, because the current system would unexpectedly produce huge numbers, we need to improve the numbering system so as to be able to maintain larger lattices.

References

[1] A. Ota : A negative meaning, Taishukan Book, 1980. (in Japanese)

[2] A. Shimojima : Operational Constraints in Diagrammatic Reasoning, Logical

Reasoning with Diagrams, Oxford University Press, pp.27-48, 1996.

[3] E. Izuhara, T. Yoshida, and H. Atsumi : Systems of Diagrams - Diagrammatic Thinking and its Representation. Nikkagiren, 1986. (in Japanese)

[4] J. Barwise and E. Hammer : Diagrams and the Concept of Logical System, Logical Reasoning with Diagrams, Oxford University Press, pp.49-78, 1996.

[5] J. Barwise and J. Seligman : Information Flow, Cambridge Tracts in Theoretical Computer Science 44, Cambridge University Press, 1997.

[6] J.I. Glasgow and D. Papadias : Computational Imagery, Diagrammatic reasoning : cognitive and computational perspective;ed.by Janice Glasgow. N.Nari Narayanan, B. Chandrasekaran, MIT Press, pp.435-480, 1995.

[7] J. Seligman : Perspective in Situation Theory, Situation Theory and its Applications Volume 1, pp.146-191, 1990.

[8] K. Kaneiwa and S. Tojo : Event and Property in Order-Sorted Logic, Computer Software, Vol.17, No.2, pp.10-24, 2000. (in Japanese)

[9] K. Kaneiwa : Order-sorted logic programming with predicate hierarchy, Artificial Intelligence, Volume 158, Issue 2, pp.155-188, 2004.

[10] K. Kiyozuka and H. Sawamura : Considerations on the Complexity of Visual Reasoning with Sentences and Diagrams, Journal of Japanese Society for Artificial Intelligence, 14(4), 1999. (in Japanese)

[11] K. Kiyozuka and H. Sawamura : JVenn: A Visual Reasoning System with Diagrams and Sentences, Theory and Application of Diagrams, Diagrams 2000, LNAI 1889, pp.271-285, 2000.

[12] K. Nitta and S. Tojo and et al., Knowledge Representation of New Helic II, Workshop on Legal Application of Logic Programming, ICLP'94, 1994.

[13] N.G. Swoboda : Implementing Euler/Venn Reasoning System, Reasoning with Diagrammatic Representation, Papers from the 1997 AAAI Fall Symposium Technical Report FS-97-03.

[14] O. Yamazaki and K. Miwa, A change of problem solving processes by Externalization, Journal of the Japanese Cognitive Science Society, Vol.8, No.1, pp.103-116, 2001. (in Japanese)

[15] S. Maehara : Introduction of mathematical foundations, Asakura Book, 1977. (in Japanese)

[16] S. Takagi, Elementary integer theory lecture The 2nd edition, Kyoritsu shuppan Co., Ltd. 1999. (in Japanese)

[17] T. Murata, Y. Kato, M. Shimura, M. Numao : DIPS: Diagrammatic Arithmetic Problem Solver, Journal of the institute of electronics, information and communication engineers D-II, Vol.J82-D-II, No.1, pp.75-82, 1999. (in Japanese)

Received : October, 2, 2017
Accepted : November, 8, 2017