Logical investigation of reasoning with tables

Ryo Takemura[*]   Atsushi Shimojima[†]   Yasuhiro Katagiri[‡]

Abstract

In graphical or diagrammatic representations, not only the basic component of a diagram, but also a collection of multiple components can form a unit with semantic significance. We call such a collection a "global object", and we consider how this can assist in reasoning using diagrammatic representation. In this paper, we investigate reasoning with correspondence tables as a case study. Correspondence tables are a basic, yet widely applied graphical/diagrammatical representation system. Although there may be various types of global objects in a table, here we concentrate on global objects consisting of rows or columns taken as a whole. We investigate reasoning with tables by exploiting not only local conditions, specifying the values in individual table entries, but also global conditions, which specify constraints on rows and columns in the table. This type of reasoning with tables would typically be employed in a task solving simple scheduling problems, such as assigning workers to work on different days of the week, given global conditions such as the number of people to be assigned to each day, as well as local conditions such as the days of the week on which certain people cannot work. We investigate logical properties of reasoning with tables, and conclude, from the perspective of free ride, that the application of global objects makes such reasoning more efficient.

1 Introduction

By "global objects," we mean those patterns or structures in diagrams that allow the extraction of higher-level information about the represented domain. Typical examples are a "cloud" consisting of multiple dots in a scatter plot that allows an estimation of the correlation strength of two variables [12], an "ascending staircase" made of multiple columns in a vertical bar graph that allows the observation of an increasing trend [13], and a group of adjacent contours lines in a topographical map that allows the identification of a characteristic landform of the terrain [11, 9]. The extraction of such higher-level information has been variously called "macro reading" [20], "pattern perception" [7], "direct translation" [13], and "cognitive integration" [14], and contrasted to the extraction of more concrete information from local objects, such as individual dots (scatter plot), bars (bar charts), and passing points of a contour line (geographical maps).

Given that the utilization of global objects significantly contribute to comprehension of graphical representations [5, 6, 20, 13, 21, 10, 7, 14], it is natural to suspect its *inferential* advantage. That is, applying inferential operations on global components of a diagram may lead to a simplification or other positive change of inferential processes. Despite this prospect,

[*]Nihon Univeristy, Japan. takemura.ryo@nihon-u.ac.jp
[†]Doshisha University, Japan. ashimoji@mail.doshisha.ac.jp
[‡]Future University Hakodate, Japan. katagiri@fun.ac.jp

few logical investigations have been conducted on what exact inferential advantages might be obtained by the utilization of global objects.

In our view, the paucity of logicians' interests in global objects is attributed to the difficulty of formally characterizing global objects—patterns and structures—that have been investigated in the graphics comprehension research cited above. The present paper tries to break the impasse and takes a few initial steps toward the logical explication of inferential advantages of utilizing global objects. For this purpose, we scale down the problem to the case of simple tabular representations. Simple as they may appear, rows and columns, as opposed to individual cells, can play the role of global objects and bring about a definite inferential advantage in certain natural inferential tasks. Also, rows and columns are simple enough to formally characterize, and we can define a heterogeneous logical system with tables, where the distinctive roles played by rows and columns are pinned down in the form of specific inference rules in the system. This allows us to compare logical systems with and without these inference rules, and to quantize the advantage of utilizing rows and columns on the basis of the complexity of proofs in the relevant logical system.

As it turns out, the inference rules in question involve "free rides" in the sense of Shimojima [15], and that is directly reflected in our results on computational complexity. Thus, the present work can be thought of as a quantitative analysis of the inferential advantage of free rides that nicely complements Shimojima's qualitative analysis.

In Section 2, we specify our reasoning with tables through an example. In Section 3, we define the syntax, semantics, and inference rules of our logic with tables (LT). LT is a table fragment of the heterogeneous logic with tables (HLT) of [19], in which tables and the usual first-order formulas are combined. We then investigate some logical properties, i.e., translation into a usual sentential system, soundness, and completeness of LT. In Section 4, we discuss the effectiveness of particular inference rules in LT by connecting up the notion of free ride with a computational complexity analysis.

2 A reasoning with tables

Correspondence tables are one of the most basic graphical/diagrammatical representations, and have been applied in a variety of scenarios. Shimojima [17] studied the semantic mechanism of extracting information from a given table, and discussed the mechanism of derivative meaning. In addition to the extraction of information from given tables, we can use tables more dynamically to solve a given problem. This involves constructing a table and adding pieces of information, before manipulating, and finally reading the table as illustrated in the following example.

Example 2.1 Consider four people a, b, c, d who are scheduled to work separately on one of Monday, Tuesday, Wednesday, and Friday. The following constraints are known: (1) a works on Wednesday; (2) Neither b nor c can work on Monday; (3) On Friday, either c or d should work. Under these conditions, how we can arrange who works on which day?

Let us first consider this problem without using tables. Note that, in addition to conditions (1), (2), and (3), we know that:

(4) There is a one-to-one correspondence between the persons and the days.

First, condition (1) states that "a works on Wednesday." Thus, by (4), we find that "a does not work on Monday." Then, by combining this with (2) and (4), we find that "d works on Monday."

In the given situation, (3) is equivalent to the following (5) under (4):

(5) "a does not work on Friday, and b does not work on Friday."

Because we have already determined that "d works on Monday," (4) implies that "d does not work on Friday." As the above facts can be combined to give that "Neither a nor b nor d works on Friday," we find by (4) that "c works on Friday."

As for b, because we already know that "a works on Wednesday," "c works on Friday," and "d works on Monday," we have from (4) that "b works on Tuesday."

In this way, we are able to determine the working day of a, b, c, d.

Note that in the above reasoning, the condition (4) is necessary to derive any piece of information. Further note that there are various ways to solve the above problem. For example, in the above solution, we converted condition (3) with disjunction into (5) without disjunction. Alternatively, we could have divided (3) into two cases, and examined each case individually.

Next, let us solve the same problem using a correspondence table. We construct a table in which the rows are labeled according to the workers a, b, c, d, and the columns are labeled by days, M (for Monday), T (for Tuesday), W (for Wednesday), and F (for Friday). Based on the given conditions (1), (2), and (3), we insert $\bigcirc$ into each entry (x, Y) for which "x works on Y" holds, and insert $\times$ when "x does not work on Y" holds. Thus, we obtain the table T_0 in Fig. 1. Note that we applied condition (5) instead of the given condition (3).

In terms of tables, condition (4) is divided into the following two conditions:

(6) In each row, exactly one entry should be marked as $\bigcirc$, and the other entries should be $\times$;

(7) In each column, exactly one entry should be marked as $\bigcirc$, and the other entries should be $\times$.

Thus, from the fact that the (a, W)-entry is $\bigcirc$ and (6), we find that the $(a, M), (a, T)$ entries are $\times$, as illustrated in T_1. Similarly, because the $(a, M), (b, M), (c, M)$ entries are all $\times$, we find that (d, M) must be $\bigcirc$ by (7), as illustrated in T_2.

Hence, by successively applying (6) and (7), we finally get the determined table T_7. From this, we can read off any information about the working day of a, b, c, d.

Although all entries are either $\bigcirc$ or $\times$ in the above example, in general, some entries may not be determined. For example, if we remove condition (3), we obtain a partial table in which $(b, T), (b, F), (c, T), (c, F)$ remain indeterminate.

Let us consider another example, in which the number of days worked by each person and the number of people working on a given day are changed from Example 2.1.

Example 2.2 Each person should work exactly two days, and exactly two people should work on each day. Conditions (1) and (2) are the same as in Example 2.1. Condition (3) is replaced by: (8) On Friday, c and d should work together. In this case, how can we arrange the allocation of working days to a, b, c, d?

Using tables, we are able to apply essentially the same strategy as for Example 2.1. Note that conditions (6) and (7) in Example 2.1 become the following:

	M	T	W	F
a			○	×
b	×			×
c	×			
d				

T_0

	M	T	W	F
a	×	×	○	×
b	×			×
c	×			
d				

T_1

	M	T	W	F
a	×	×	○	×
b	×			×
c	×			
d	○			

T_2

	M	T	W	F
a	×	×	○	×
b	×		×	×
c	×		×	
d	○		×	

T_3

	M	T	W	F
a	×	×	○	×
b	×	○	×	×
c	×		×	
d	○		×	

T_4

	M	T	W	F
a	×	×	○	×
b	×	○	×	×
c	×		×	
d	○	×	×	×

T_5

	M	T	W	F
a	×	×	○	×
b	×	○	×	×
c	×	×	×	
d	○	×	×	×

T_6

	M	T	W	F
a	×	×	○	×
b	×	○	×	×
c	×	×	×	○
d	○	×	×	×

T_7

Fig. 1

(9) In each row, exactly two entries should be marked as ○, and the other entries should be ×;

(10) In each column, exactly two entries should be marked as ○, and the other entries should be ×.

We begin with the following table T_8. Because the $(b, M), (c, M)$ entries are ×, we find, by (10), that $(a, M), (d, M)$ are ○, as illustrated in T_9. In a similar way, we obtain T_{10}. Then, because the entries of $(a, M), (a, W)$ are ○, we find, by (9), that (a, T) is ×, as in T_{11}, and in a similar way to Example 2.1, we finally obtain table T_{15} as follows.

	M	T	W	F
a			○	
b	×			
c	×			○
d				○

T_8

	M	T	W	F
a	○		○	
b	×			
c	×			○
d	○			○

T_9

	M	T	W	F
a	○		○	×
b	×			×
c	×			○
d	○			○

T_{10}

	M	T	W	F
a	○	×	○	×
b	×			×
c	×			○
d	○			○

T_{11}

	M	T	W	F
a	○	×	○	×
b	×	○	○	×
c	×			○
d	○			○

T_{12}

	M	T	W	F
a	○	×	○	×
b	×	○	○	×
c	×			○
d	○	×	×	○

T_{13}

	M	T	W	F
a	○	×	○	×
b	×	○	○	×
c	×	○		○
d	○	×	×	○

T_{14}

	M	T	W	F
a	○	×	○	×
b	×	○	○	×
c	×	○	×	○
d	○	×	×	○

T_{15}

In the above examples, although the number of ○ is fixed to be the same (i.e., one or two) in every row and column, this is not necessarily the case. We do not assume such a restriction in our formalization of reasoning with tables.

By investigating Examples 2.1, and 2.2, we find that there are two types of condition in our problems. One is a "constraint over the framework of a given problem" (e.g., condition (4) above), and we call these *global conditions*. In view of tables, our global conditions are constraints over the number of ○ and × in every row and column. (Although there may be various types of global conditions, we here discuss a particular kind of them, which is formally defined in Definition 3.4 below.) The other type is a "specific condition for each object" (e.g., (1) above), and we call these *local conditions.* In view of tables, local conditions specify only particular entries. Our reasoning with tables is essentially conducted by combining global conditions and local conditions.

One of the remarkable facts of our reasoning with tables is that, even if the given local and global conditions change, we are able to apply essentially the same strategy:

(I) We decompose, if necessary, the given conditions into local conditions (i.e., atomic sentences or their negation, such as "a does not work on Friday" and "b does not work on Friday") by applying logical laws (e.g., (3) $\wedge$ (4) $\to$ (5)).

(II) We construct a correspondence table using these local conditions (e.g., T_0).

(III) By applying global conditions, that is, by exploiting constraints over the number of $\bigcirc$ or $\times$ in a row or column, we further insert $\bigcirc$ and $\times$ into the table.

(IV) Finally, we extract information from the table.

Although most of the given conditions in the above example are already local conditions, more complex conditions may generally be given. In such cases, we frequently apply item (I) in the above procedure. Furthermore, a given condition, such as an implicational sentence, may be decomposed using basic information obtained after item (III). In these complex cases, we must repeat the whole procedure several times. Thus, a natural system of formalizing our reasoning with tables is a heterogeneous logical system combining tables and first-order formulas. Our formalization is based on Gentzen's natural deduction.

Reasoning tasks, such as that specified above, occur in simple scheduling problems, civil servant examinations in Japan, and in so-called logic puzzles, among others. Barker-Plummer and Swoboda [2] discussed similar problems. They formalized correspondence tables and their manipulations. Their system, consisting mainly of rules for case dividing and reduction to absurdity, is defined to be simple and have as few rules as possible. Conversely, we formalize our system based on each row and column as a global object so that they work effectively to solve a given problem.

3 Logic with tables LT

In Section 3.1, we roughly review our HLT [19], while in Section 3.2, we define the syntax and semantics of LT, which is the table fragment of HLT. We introduce the inference rules of LT in Section 3.3 and then investigate logical properties of LT, that is, the translation of tables into formulas in Section 3.4, and the completeness theorem in Section 3.5.

3.1 A heterogeneous logic with tables HLT

To formalize our heterogeneous reasoning with tables, we adopt many-sorted first-order logic, in which constants and variables of the usual first-order language are divided into two sorts: sorts of row and column. A **row-label** or **row-constant** (resp. variable) is denoted by a small letter a (resp. x), while a **col-label** (resp. variable) is denoted by a capital letter A (resp. X). (See, for example, [8] for many-sorted logic.) Then, by an **atomic formula** $B(a)$, or more formally $\bigcirc(a, B)$, we denote that "a and B are in a certain positive relation." Thus, sentences such as "a is B," "a matches B," and "a corresponds to B" are all expressed as $B(a)$. Atomic formulas and their negation without free variables are collectively called **(closed) literals**. Based on the atomic formulas, complex formulas are defined inductively as usual by using connectives $\wedge, \vee, \rightarrow, \neg, \forall, \exists$. Among such complex formulas, we distinguish "global formulas", as defined in Definition 3.4 below. Our correspondence tables are defined in Definition 3.1.

Semantics is defined as a particular case of the semantics of many-sorted logic, which is a natural generalization of the usual first-order set-theoretical semantics. (See [8] for the semantics of many-sorted logic.)

Since we are mainly concerned with the table fragment LT (Logic with Tables) of our Heterogeneous Logic with Tables HLT in this paper, we present definitions of the tables and global formulas without going into the detail for the heterogeneous system. See [19] for a formal description of the full HLT.

3.2 Syntax and semantics of LT

First we define our tables.

Definition 3.1 A **table** T is an $m \times n$-matrix over symbols $\{\bigcirc, \times, b\}$; that is, a rectangular arrangement of the symbols, in which rows are labeled by distinct row-labels $a_1, \ldots, a_m$ and columns are labeled by distinct col-labels $A_1, \ldots, A_n$.
In a specific representation of a table, we usually omit the symbol "b" and leave the entry blank. A table is said to be **determined** if there are no blank entries.
Tables T_1 and T_2 are of the **same type** if their labels are the same.

Note that according to the definition, no entry can be marked as $\bigcirc, \times, b$ at the same time.

Remark 3.2 A table T is abstractly defined as the function $T : \mathcal{R} \times \mathcal{C} \longrightarrow \{\bigcirc, \times, b\}$, where $\mathcal{R}$ (resp. $\mathcal{C}$) is some finite set of row-labels (resp. col-labels) of T.

As usual, any pair of tables, say T_1 and T_2, are identical if they have the same type, and if the $\bigcirc$, $\times$ marks of all entries in T_1 and T_2 are also identical. This is formally defined as follows.

Definition 3.3 (Equivalence of tables) Table T_1 is a subtable of T_2, written as $T_1 \subseteq T_2$, if:

- all row- and col-labels of T_1 are also those of T_2;
- for any (a_i, A_j)-entry in T_1: if it is $\bigcirc$ in T_1, it is also $\bigcirc$ in T_2, and if it is $\times$ in T_1, it is also $\times$ in T_2.

T_1 and T_2 are **(syntactically) equivalent** if $T_1 \subseteq T_2$ and $T_2 \subseteq T_1$ hold.

Note that, by definition, two specific tables that differ only in the order of their labels of rows and columns are equivalent.

Next, we define global formulas. To express sentences of the form "among n objects, there are exactly i objects that are A," we introduce a kind of counting quantifier, and write the sentence as $\exists^{i/n} x.A(x)$.

Definition 3.4 (Global formula) For fixed sets of row-labels $\mathcal{R} = \{a_1, \dots, a_m\}$ and of col-labels $\mathcal{C} = \{A_1, \dots, A_n\}$, the following forms of formulas are called **global formulas**: For any A and a,

$$\exists^{i/m} x \in \mathcal{R}.A(x), \qquad \exists^{i/m} x \in \mathcal{R}.\neg A(x), \qquad \exists^{i/n} X \in \mathcal{C}.X(a), \qquad \exists^{i/n} X \in \mathcal{C}.\neg X(a).$$

If a set of labels is clear from the context, it is abbreviated as $\exists^{i/m} x.A(x)$.

Global formulas are simply abbreviations of the appropriate first-order formulas. For example, for some row-label a and col-labels $\mathcal{C} = \{A_1, A_2, A_3\}$, the global formula $\exists^{2/3} X \in \mathcal{C}.X(a)$ is an abbreviation of the following formula:

$$\Big(A_1(a) \wedge A_2(a) \wedge \neg A_3(a)\Big) \vee \Big(A_1(a) \wedge A_3(a) \wedge \neg A_2(a)\Big) \vee \Big(A_2(a) \wedge A_3(a) \wedge \neg A_1(a)\Big).$$

By $\mathcal{G}$, we denote a set of global formulas, and by **label sets of** $\mathcal{G}$, we mean the set consisting of label sets of every global formula of $\mathcal{G}$.

As for the semantics, informally speaking, $(a, B) = \bigcirc$ in a table T means that "a certain relation exists between a and B," or more specifically, $B(a)$ holds. In the same way, $(a, B) = \times$ in T means that $B(a)$ does not hold, i.e., the negation $\neg B(a)$ holds. $(a, B) = b$ in T means that it is not determined whether or not $B(a)$ hold. Although such an informal reading of a table can be formalized in an appropriate set-theoretical domain by applying the semantics of many-sorted logic, here we informally introduce the notion of models by avoiding technical details. See [19] for the details.

Let M be a set-theoretical domain. First-order formulas are interpreted as usual, and we write $M \models \varphi$ if formula φ has a **model** M, i.e., φ holds in M. Table T has a **model** M (written as $M \models T$) if the following holds:

- if $(a, B) = \bigcirc$ in T, then $B(a)$ holds in M;
- if $(a, B) = \times$ in T, then $B(a)$ does not hold, i.e., $\neg B(a)$ holds, in M.

Let $\mathcal{G}$ be a set of global formulas. We write $M \models \mathcal{G}, T$ if $M \models \mathcal{G}$ and $M \models T$, i.e., T has a model M in which $\mathcal{G}$ holds.

Since our tables and global formulas do not contain any variables, LT is essentially the propositional logic. Thus, our model M can be considered to be a set of literals, which hold in M.

Alternatively, a determined table T can also be regarded as a model in its own right, because we can define a model M from T as follows: $B(a)$ holds in M if $(a, B) = \bigcirc$ in T; and $B(a)$ does not hold in M if $(a, B) = \times$ in T (and either one is fine if (a, B) is blank or if there is no such entry in T).

Conversely, if we have a model M of T, we are able to construct the determined instance T_M by applying the above definition in the opposite direction.

Thus, the following are equivalent:

- T has a model;
- T can be extended to a determined table by consistently inserting $\bigcirc, \times$ marks into the blank entries of T.

The semantic consequence relation in our LT is defined as follows.

Definition 3.5 (Semantic consequence) Let T_1 and T_2 be tables of the same type, and $\mathcal{G}$ be a set of global formulas whose label sets are those of T_1. T_2 is said to be a **semantic consequence** of $\mathcal{G}, T_1$, written as $\mathcal{G}, T_1 \models T_2$, if any model of $\mathcal{G}, T_1$ is also a model of T_2.

3.3 Inference rules of LT

According to Plummer-Etchemendy [1], inference rules characteristic of heterogeneous systems are generally called *transfer rules*, and they allow the transfer of information from one form of representation to another. Typical rules in Hyperproof [3, 4] are the Apply rule (from sentences to a diagram) and the Observe rule (from a diagram to sentences). Our inference rules, corresponding to the Apply rule, generally have the following form:

$$\frac{T \qquad \varphi}{T'}$$

where T and T' are our tables, and φ is a formula (more specifically, a literal or global formula). Following [3, 4], we read this rule as: "we apply φ to amplify T to T'," "we extend T to T' by adding the new information of φ to T," or "φ justifies the specific modification of T to T'." More concretely, our heterogeneous HLT has the following rules, for example:

in **rule:** By applying $A(b)$, we extend table T in which the (b, A)-entry is blank, to table T' in which the (b, A)-entry is $\bigcirc$. Similarly for $\neg A(b)$.

row **rule:** By applying global formula $\exists^{i/n} X.X(a)$, we extend T in which exactly i entries of row a are $\bigcirc$, to T' in which the other entries of row a are $\times$.

Similarly for a global formula of the form $\exists^{i/n} X.\neg X(a)$.

col **rule:** By applying global formula $\exists^{i/n} x.A(x)$, we extend T in which exactly i entries of column A are $\bigcirc$, to T' in which the other entries of A are $\times$.

Similarly for a global formula of the form $\exists^{i/n} x.\neg A(x)$.

ext **rule:** This rule corresponds to the Observe rule in Hyperproof, and we extract information in a sentential form from a given table.

In addition to these transfer rules, our heterogeneous HLT has the usual natural deduction rules for first-order formulas. See [19]. Thus, the table fragment LT consists of the above *row* and *col* rules, which are formally defined below. Our rules are defined by specifying premises (a global formula and a table) and a conclusion (a table) for each rule. See [19] for the other rules, as well as the usual natural deduction rules for formulas. (See Appendix A for the usual natural deduction style representation of our rules.)

Definition 3.6 Inference rules *row* and *col* of LT are defined as follows.

row$\times$ **rules** Premises: A global formula of the form $\exists^{i/n} X.X(a)$ (resp. $\exists^{i/n} X.\neg X(a)$), and a table T in which exactly i entries of row a are $\bigcirc$ (resp. b or $\times$) and the other entries of row a are b or $\times$ (resp. $\bigcirc$).

Conclusion: A table T' that is exactly the same as T except for the blank entries of row a, which are now $\times$.

row$\bigcirc$ **rules** Premises: A global formula of the form $\exists^{i/n}X.X(a)$ (resp. $\exists^{i/n}X.\neg X(a)$), and a table T in which exactly i entries of row a are b or $\bigcirc$ (resp. $\times$) and the other entries of row a are $\times$ (resp. b or $\bigcirc$).

Conclusion: A table T' that is exactly the same as T except for the blank entries of row a, which are now $\bigcirc$.

col$\times$ **rules** Premises: A global formula of the form $\exists^{j/m}x.A(x)$ (resp. $\exists^{j/m}x.\neg A(x)$), and a table T in which exactly j entries of column A are $\bigcirc$ (resp. b or $\times$) and the other entries of column A are b or $\times$ (resp. $\bigcirc$).

Conclusion: A table T' that is exactly the same as T except for the blank entries of column A, which are now $\times$.

col$\bigcirc$ **rules** Premises: A global formula of the form $\exists^{j/m}x.A(x)$ (resp. $\exists^{j/m}X.\neg A(x)$), and a table T in which exactly j entries of column A are b or $\bigcirc$ (resp. $\times$) and the other entries of column A are $\times$ (resp. b or $\bigcirc$).

Conclusion: A table T' that is exactly the same as T except for the blank entries of column A, which are now $\bigcirc$.

The notion of proof in LT is defined inductively as usual in natural deduction.

Definition 3.7 (Provability in LT) Let T_1 and T_2 be tables, and $\mathcal{G}$ be a set of global formulas whose label sets are those of T_1. T_2 is **provable** from $\mathcal{G}, T_1$, written as $\mathcal{G}, T_1 \vdash T_2$, if there exists a proof of T_2 from the premises of $\mathcal{G}, T_1$.

Example 3.8 A proof in LT of Example 2.1 is given in Fig. 2.

3.4 Translation of LT

We investigate our tables in terms of the usual first-order language through logic translation of tables. Our tables are translated as follows.

Definition 3.9 (Translation) A table T is **translated** into a conjunction of literals T° as follows:

$$T^\circ = \bigwedge\{A_i(a_j) \mid (a_j, A_i) = \bigcirc \text{ in } T\} \wedge \bigwedge\{\neg A_i(a_j) \mid (a_j, A_i) = \times \text{ in } T\}$$

Conversely, it is easily seen that, for any consistent conjunction of literals (without free variables), there exists a corresponding table. Thus, a table T and a consistent conjunction $\mathcal{L}$ of literals can be regarded as being interchangeable. Thus, by slightly abusing our notation, we sometimes write as $\mathcal{L} \subseteq T$.

Based on the translation of tables, manipulations of tables, i.e., *row* and *col* rules of LT are translated into combinations of natural deduction rules; see [19] for the details. The translation of an application of the *row*$\times$ rule is illustrated as follows.

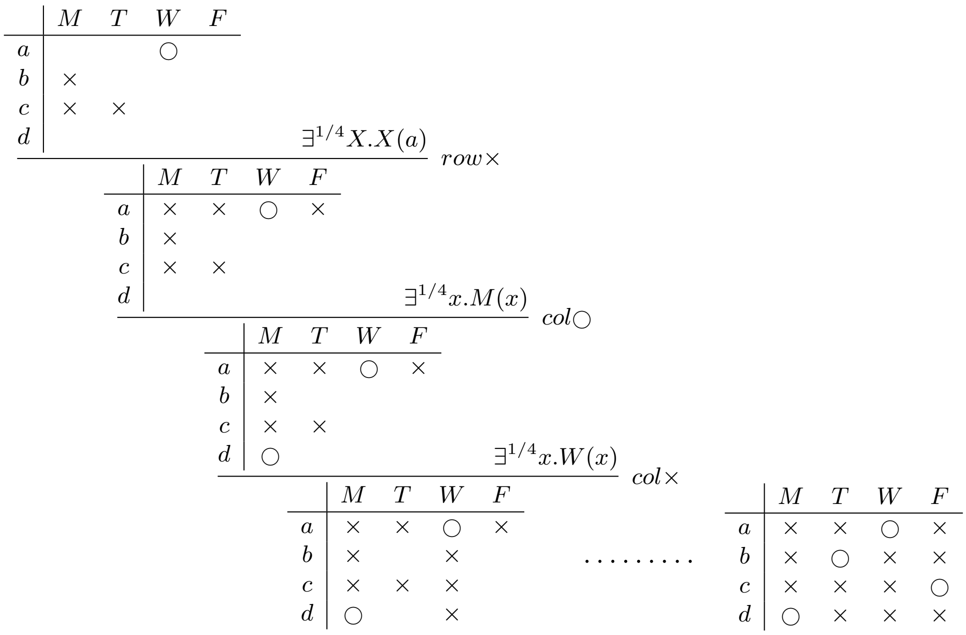

Fig. 2 A proof in LT of Example 2.1

Example 3.10 Let us consider the following application of the *row*×-rule:

$$\dfrac{\begin{array}{c|ccc} & A_1 & A_2 & A_3 \\ \hline a & \bigcirc & \bigcirc & \end{array} \qquad \exists^{2/3}X.X(a)}{\begin{array}{c|ccc} & A_1 & A_2 & A_3 \\ \hline a & \bigcirc & \bigcirc & \times \end{array}}\ row\times$$

The premise table is translated into the formula $A_1 \wedge A_2$, while the conclusion table is translated into $A_1 \wedge A_2 \wedge \neg A_3$, where we abbreviate $A_i(a)$ as A_i. Note that $\exists^{2/3}X.X(a) := (A_1 \wedge A_2 \wedge \neg A_3) \vee (A_1 \wedge A_3 \wedge \neg A_2) \vee (A_2 \wedge A_3 \wedge \neg A_1)$, where we omit trivial permutations. This application of the *row*×-rule is translated into the following proof:

$$\dfrac{\dfrac{\exists^{2/3}X.X(a) \quad \dfrac{[A_1 \wedge A_2 \wedge \neg A_3]^1}{\neg A_3} \quad \dfrac{\dfrac{\dfrac{[A_1 \wedge A_3 \wedge \neg A_2]^1}{\neg A_2} \quad \dfrac{A_1 \wedge A_2}{A_2}}{\bot}}{\neg A_3} \quad \dfrac{\dfrac{\dfrac{[A_2 \wedge A_3 \wedge \neg A_1]^1}{\neg A_1} \quad \dfrac{A_1 \wedge A_2}{A_1}}{\bot}}{\neg A_3}}{\neg A_3}\,1 \qquad A_1 \wedge A_2}{A_1 \wedge A_2 \wedge \neg A_3}$$

Proposition 3.11 (Translation) *If $\mathcal{G}, T_1 \vdash T_2$ in* LT*, then $\mathcal{G}, T_1^\circ \vdash T_2^\circ$ in the natural deduction (without tables).*

By the above theorem of translation, soundness of LT is obtained through soundness of the usual natural deduction without tables.

Proposition 3.12 (Soundness of LT) *If $\mathcal{G}, T_1 \vdash T_2$ in* LT*, then $\mathcal{G}, T_1 \models T_2$.*

3.5 Completeness of LT

Let us now investigate the completeness theorem of LT. The theorem implies, in the framework of the heterogeneous HLT, that any conjunction of consistent ground literals provable with natural deduction rules is also provable only with manipulations of tables. Unfortunately, LT is not complete with respect to our semantics; that is, for a given table T_1 and global formulas $\mathcal{G}$, there exists a table T_2 such that $\mathcal{G}, T_1 \models T_2$ but it cannot be obtained from T_1 with only *row* and *col* rules, as illustrated in the following example.

Example 3.13 Let T_1 and T_2 be the following tables. Let the global formulas $\mathcal{G}$ be $\exists^{1/4}x.X(x)$ for all $X \in \{A, B, C, D\}$ and $\exists^{1/4}X.X(x)$ for all $x \in \{a, b, c, d\}$, which implies that in every row and column, exactly one entry should be ○. Then, we have $\mathcal{G}, T_1 \models T_2$ but $\mathcal{G}, T_1 \not\vdash T_2$ since we cannot apply row, col rules to T_1.

	A	B	C	D
a	×	×		
b	×	×		
c				
d				

T_1

	A	B	C	D
a	×	×		
b	×	×		
c			×	×
d			×	×

T_2

Thus, in order to obtain completeness, we need to: (1) extend the inference rules in LT so that the above $\mathcal{G}, T_1 \vdash T_2$ holds; or (2) restrict the logical consequence relation so that the T_1 and T_2 above are not considered. For (1), we can extend LT by introducing rules for case dividing and reduction to absurdity according to [2]. However, these rules check each entry one by one, and hence they are not that effective. Thus, instead of (1), we adopt (2), thereby retaining our intuitive manipulation of tables, and restricting our tables and global formulas as follows. Thus, by retaining our rules, intuitive manipulation of tables, we restrict our tables and global formulas as follows.

Informally speaking, a table T is **uniquely determinable** under $\mathcal{G}$, if every entry of T is uniquely determined semantically to either ○ or ×, without leaving any entry blank, regardless of any model M in which $\mathcal{G}$ holds.

Definition 3.14 (Uniquely determinable) Given table T and its model M, we are able to construct the determined table T_M. Let $\mathcal{G}$ be a set of global formulas whose label sets are those of T. T is said to be **uniquely determinable** under $\mathcal{G}$, if the following hold:

- $\mathcal{G}$ and T have a model M, i.e., $M \models \mathcal{G}, T$, and
- for any model N, if $N \models \mathcal{G}, T$ then $T_M = T_N$.

We further restrict our global formulas. We call a set $\mathcal{G}\mathbf{1}$ of global formulas **all-one-global formulas**, consisting of global formulas of the following forms: $\exists^{1/n}x.Y(x)$ for every Y and $\exists^{1/n}X.X(y)$ for every y, which implies, in terms of tables, that "for every row and column, exactly one entry should be ○."

Lemma 3.15 *If $M \models \mathcal{G}, T$ and $\mathcal{G}, T \vdash T'$, then $T_M = T'_M$.*

Proof. By ⊗, we denote either ○ or ×. Then, by $\overline{\otimes}$, we denote ○ if ⊗ is ×; and × if ⊗ is ○. Assume to the contrary that $T_M \neq T'_M$. Then, since T_M and T'_M are of the same type and determined, $(a, B) = \otimes$ in T_M, but $(a, B) = \overline{\otimes}$ in T'_M for some entry (a, B). However, since by definition $M \models T_M$ and $M \models T'_M$, we have $M \models B(a)$ and $M \models \neg B(a)$, which is a contradiction. ■

The following is the main lemma used to prove our completeness.

Lemma 3.16 (Main lemma) *Let $\mathcal{G}\mathbf{1}$ be a set of all-one-global formulas. Let T be a table uniquely determinable under $\mathcal{G}\mathbf{1}$. Then, we have $\mathcal{G}\mathbf{1}, T \vdash T'$ in* LT *for some determined table T'.*

Proof. Let T' be a table obtained by applying as many *row* and *col* rules to $\mathcal{G}\mathbf{1}, T$ as possible; that is, $\mathcal{G}\mathbf{1}, T \vdash T'$, and no more *row* and *col* rules can be applied to T'. To show that this T' is the required determined table, assume to the contrary that T' is not determined, i.e., T' contains some blank entries.

Let M be a model such that $M \models \mathcal{G}\mathbf{1}, T$. Then, by the soundness (Theorem 3.12) of LT, we have $M \models T'$. The determined table defined by T' and M is the same as that of T by Lemma 3.15, which we denote by T_M. By using this T_M, we construct another determined table T_N that is different from T_M. Then, we obtain another model N of T that is different from M. This contradicts the assumption that T is uniquely determinable under $\mathcal{G}\mathbf{1}$.

In what follows, we refer to each blank entry of T' using □ notation, and denote the entry in the determined T_M as $\boxed{\bigcirc}$ or $\boxed{\times}$.

We assume that there are no entries marked by ○ in T', since, if there is a ○ entry, then all entries of the corresponding row and column are already determined to be ×, and such rows and columns do not play any role in the following construction of T_N. Thus, in the determined T_M, all entries marked by ○ are boxed.

Let all rows (resp. columns) of T_M be labeled by $a, b, c, d, \dots$ (resp. $A, B, C, D, \dots$). We usually refer to an entry of T_M by its label, say (a, B), but sometimes, we refer to an entry according to the underlying table without using labels, like (1st, 2nd), which denotes the intersection of the first row and the second column.

We illustrate our strategy to rewrite the given T_M in Fig. 3. For simplicity, we assume that the (a, A)-entry in T_M is $\boxed{\bigcirc}$. In every row and column of T_M, there are more than two boxed entries, because, if this were not the case, we would be able to apply a *row* or *col* rule to T'. In particular, there is another boxed entry, other than $(a, A) = \boxed{\bigcirc}$ in column A of T_M. Let $(b, A) = \boxed{\times}$. Then, we replace the entire row a including ○, × marks in the row by row b. After the replacement, if the ○-entry of row b, say (b, B), which appeared in the first row, becomes boxed (i.e., if $(b, B) = \boxed{\bigcirc}$), then we terminate our rewriting process. Otherwise, we continue replacing the row b by another row, say c, such that $c \neq a$ and $(c, B) = \boxed{\times}$. After the replacement, we again check whether the ○-entry in row c, which appeared in the first row, is boxed, and if not, we continue the replacement process.

	A	B	C
a	$\boxed{\bigcirc}$		
b	$\boxed{\times}$		
c		$\boxed{}$	
d			

	A	B	C
b	$\boxed{\times}$	○	
a	$\boxed{\bigcirc}$		
c		$\boxed{\times}$	
d			

	A	B	C
c	$\boxed{}$	×	○
a	$\boxed{\bigcirc}$		
b	×	$\boxed{\bigcirc}$	
d			$\boxed{\times}$

	A	B	C
d	$\boxed{}$		×
a	$\boxed{\bigcirc}$		
b	×	$\boxed{\bigcirc}$	
c		×	$\boxed{\bigcirc}$

Fig. 3

Our rewriting process is described more formally as follows:

Step 1: Let $(a, X) = \boxed{\bigcirc}$.

Step 2: Search for a row y such that $(y, X) = \boxed{\times}$ and y has not yet been replaced.

Step 3: Replace the 1st row by y.

Step 4: Search for a column Y such that $(1\text{st}, Y) = \bigcirc$.

Step 5: If $(1\text{st}, Y) = \boxed{\bigcirc}$, terminate the process.
Otherwise, search for a row z such that $(z, Y) = \boxed{\times}$ and z has not yet been replaced.

Step 6: Replace the 1st row by z.

Step 7: Return to Step 4.

When the above rewriting process has finished, we obtain T_N by renaming labels of the resulting table so that 1st, 2nd, 3rd, ... rows are labeled as $a, b, c, \dots$, respectively.

In the above rewriting process, we are able to find a row z that has not yet been replaced. This is because there are more than two boxed entries in every column.

The above process terminates after some repetition. This is because, in the first row, there are more than two boxed entries, and hence, by repeating the replacement, $\bigcirc$ appears as a boxed entry in the first row at most the number of columns minus two steps.

The resulting table T_N differs from the original T_M only with respect to the boxed entries. Note that on the one hand, $(a, A) = \boxed{\bigcirc}$ in the original T_M, and on the other, $(a, A) = \boxed{\times}$ in the resulting T_N. Furthermore, by our definition of the rewriting process, all $\bigcirc$ appear in a boxed entry, and hence, T_N and T_M differ only with respect to their boxed entries.

Since the above T_N is determined, we are able to define a model N. Then, we have $N \models \mathcal{G}\mathbf{1}, T$, since $T \subseteq T' \subseteq T_N$ by our construction. ∎

By the above main lemma, we obtain our completeness of LT.

Theorem 3.17 (Completeness of LT) *Let T_1, T_2 be tables of the same type, and $\mathcal{G}\mathbf{1}$ be a set of all-one-global formulas. Let T_1 be uniquely determinable under $\mathcal{G}\mathbf{1}$. If $\mathcal{G}\mathbf{1}, T_1 \models T_2$, then $\mathcal{G}\mathbf{1}, T_1 \vdash T_3$ in LT for some table T_3 such that $T_2 \subseteq T_3$.*

Proof. Let T_3 be a table obtained by applying as many *row* and *col* rules to $\mathcal{G}\mathbf{1}, T_1$ as possible. We show that $T_2 \subseteq T_3$. Assume to the contrary that $(a, B) = \bigcirc$ in T_2 and $(a, B) \neq \bigcirc$ in T_3 for some entry (a, B). (The same applies to the case $(a, B) = \times$ in T_2 and $(a, B) \neq \times$ in T_3.) Since T_2 and T_3 are of the same type, and since T_3 is determined by Lemma 3.16, we have $(a, B) = \times$ in T_3. Then, for any model M such that $M \models \mathcal{G}\mathbf{1}, T_1$, we have $M \models B(a)$ and $M \models \neg B(a)$, which is a contradiction. ∎

4 Effectiveness of tables from a free ride perspective

In this section, we investigate the effectiveness of our tables from the viewpoint of the notion of free ride. In Section 4.1, we introduce the notion of "free rides" of inference rules, and explain that our rules have multiple free rides. In Section 4.2, we further investigate the effectiveness of our rules in terms of the complexity of inference.

4.1 Free rides of inference rules

The **free ride** property is one of the most basic properties of diagrammatic systems that provides an account of the inferential efficacy of diagrams. By adding a certain piece of information to a diagram, the resulting diagram somehow comes to present pieces of information not contained in the given premise diagrams. Shimojima [15] called this phenomenon *free ride*, and analyzed its semantic conditions within the framework of channel theory.

By slightly extending the notion of free ride, let us call diagrammatic objects, or translated formulas thereof, **free rides** if they do not appear in the given premise diagrams or sentences, but (automatically) appear in the conclusion after adding pieces of information to the given premise diagrams. The notion of free rides enables us to analyse the effectiveness of each inference rule. (Cf. [18].) Let us illustrate free rides of the *row*× rule by the following example.

Example 4.1 (Free rides of *row*×) Let us consider the following application of the *row*× rule and its translation: By applying a global formula $\exists^{1/4}X.X(a)$, we extend table T_1 in which exactly one entry of row a is ○, to T_2 in which the other entries of row a are ×. In the translation, the double line $==$ represents the application of various rules.

$$\dfrac{\overset{T_1}{\begin{array}{c|cccc} & A_1 & A_2 & A_3 & A_4 \\ \hline a & \bigcirc & & & \end{array}} \qquad \exists^{1/4}X.X(a)}{\underset{T_2}{\begin{array}{c|cccc} & A_1 & A_2 & A_3 & A_4 \\ \hline a & \bigcirc & \times & \times & \times \end{array}}}\ row\times \quad \overset{\text{transl.}}{\Longrightarrow} \quad \dfrac{A_1(a) \qquad \exists^{1/4}X.X(a)}{A_1(a), \underbrace{\neg A_2(a), \neg A_3(a), \neg A_4(a)}_{\text{free rides}}}$$

Note that pieces of information, $\neg A_2(a), \neg A_3(a), \neg A_4(a)$ in the conclusion do not appear in premise table T_1, which is translated into $A_1(a)$. Furthermore, they do not appear explicitly, or they are indeterminate, in the given global formula $\exists^{1/4}X.X(a) := (A_1 \wedge \neg A_2 \wedge \neg A_3 \wedge \neg A_4) \vee (\neg A_1 \wedge A_2 \wedge \neg A_3 \wedge \neg A_4) \vee (\neg A_1 \wedge \neg A_2 \wedge A_3 \wedge \neg A_4) \vee (\neg A_1 \wedge \neg A_2 \wedge \neg A_3 \wedge A_4)$. Note that the global formula does not imply any definite information about which entries are ○ and which are ×. If we extend T_1 to T_2 so that $\exists^{1/4}X.X(a)$ holds, all blank entries in row a *have to be* × in T_2 since there already exists one ○ in row a of T_1. Note that, in the extension of T_1 to T_2, we do not care about the location of ○ in T_1, because the same applies even if $(a, A_2) = \bigcirc$ and the other entries are blank in T_1. The only thing that matters is the number of ○ in row a in T_1, which corresponds to the number 1/4 of the given global formula $\exists^{1/4}X.X(a)$. Then, we *freely* find that all blank entries of T_1 are × without checking each entry one by one. In other words, the pieces of information, $\neg A_2(a), \neg A_3(a), \neg A_4(a)$, are free rides of this *row*× as they do not appear explicitly in the premises.

As seen in Example 4.1, the *row*× rule has multiple free rides. By checking each rule of LT, we obtain the following proposition.

Proposition 4.2 (Free rides) *row and col rules of* LT *have multiple free rides.*

4.2 Complexity of inference

We further investigate the multiple free rides of our inference rules in terms of complexity of inference. To this end, we consider the framework of the full heterogeneous HLT, which contains our LT as a subsystem. In the heterogeneous HLT, in addition to the rules for tables, we have usual natural deduction rules for first-order formulas. Furthermore, as shown by the translation (Proposition 3.11), everything provable by using tables in LT is also provable in HLT by using formulas instead of tables. Thus, we compare, in the framework of the heterogeneous HLT, complexities of inference both with and without the use of tables.

As usual, we formally define the complexity as the length, i.e., the number of formulas and tables, of a given heterogeneous proof in HLT. We consider the restricted logical consequence relation $\mathcal{G}\mathbf{1}, T_1 \vdash T_2$ (and its translation $\mathcal{G}\mathbf{1}, T_1^\circ \vdash T_2^\circ$), where T_1, T_2 are $n \times m$-tables, and T_1 is uniquely determinable under $\mathcal{G}\mathbf{1}$.

Let us first examine inference without using tables. In order to state each formula (literal) of T_2° corresponding to an entry in T_2, we need $n \times m$ formulas. Although we need zero steps to derive such a formula when it is given in the premise T_1°, it is difficult to estimate how many steps (definitely more than one) we need to derive the formula generally. (Cf. the natural deduction proof in Example 3.10.) Thus, we assume we need at least one step to derive all the formulas of T_2°. Then, we need $n \times m$ formulas to derive T_2°. Hence, we estimate the following number of steps, i.e., formulas, to prove $\mathcal{G}\mathbf{1}, T_1^\circ \vdash T_2^\circ$:

$$\underbrace{(n \times m)}_{\text{to state}} + \underbrace{(n \times m)}_{\text{to derive}}\ .$$

Next, we examine inference using tables. Our completeness (Theorem 3.17) implies the following theorem, which states that we are able to set an upper bound on the length of inference with tables.

Theorem 4.3 (Upper bound) *Let $\mathcal{G}\mathbf{1}$ be a set of all-one-global formulas, and T be an $n \times m$-table that is uniquely determinable under $\mathcal{G}\mathbf{1}$. Let $\mathcal{L}$ be a conjunction of literals whose labels (constants) are those of T. If $\mathcal{G}\mathbf{1}, T \vdash \mathcal{L}$ in the heterogeneous* HLT*, then, with a proof of at most $n + m$ length, we have $\mathcal{G}\mathbf{1}, T \vdash T'$ for some T' such that $\mathcal{L} \subseteq T'$.*

Proof. Let $\mathcal{G}\mathbf{1}, T \vdash \mathcal{L}$ in HLT. Then, by the soundness of HLT, we have $\mathcal{G}\mathbf{1}, T \models \mathcal{L}$. Also, by the completeness of LT (Theorem 3.17), we have, with only rules for tables, $\mathcal{G}\mathbf{1}, T \vdash T'$ for some T' such that $\mathcal{L} \subseteq T'$. Note that the rules applicable to T are only *row* and *col* rules, and, by an application of one of these rules, one of the rows and columns is filled with $\bigcirc$ and $\times$ symbols. Since there are only $n + m$ rows and columns in T, we obtain T' within at most $n + m$ steps (tables). ■

In order to state each formula of the entries in T_2, i.e., to read table T_2 using the *ext* rule, we need $n \times m$ formulas. Furthermore, by the above theorem, in order to derive such formulas using tables, we need at most $n + m$ tables. Hence, we estimate the following number of formulas and tables to prove $\mathcal{G}\mathbf{1}, T_1 \vdash T_2$:

$$\underbrace{(n \times m)}_{\text{to state}} + \underbrace{(n + m)}_{\text{to derive}}\ .$$

Since the number of steps to state each formula is the same, the above comparison between inference with and without tables (or between systems with multiple free rides and with at most one free rides, more generally) is summarized as the difference between $n+m$ and $n \times m$, or between $2n$ and n^2 more concisely.

The effectiveness of our system stems from the multiple free rides of our inference rules. In contrast to the rules for deriving each entry in a table one by one, our rules, having multiple free rides, derive multiple entries at once. The bigger a given table is, the more significant the difference between $2n$ and n^2 becomes. However, note that the above result is obtained in a restricted fragment, where global formulas are restricted to all-one-global formulas $\mathcal{G}\mathbf{1}$, and where T_1 is uniquely determinable under $\mathcal{G}\mathbf{1}$. In contrast, there is no such restriction on inference using natural deduction rules without tables, although we need n^2 steps to infer. In other words, the above theorem characterizes a fragment in which our tables work effectively. Thus, in practical applications, we can divide the solution to a given problem into two phases. The first phase consists of applying our *row* and *col* rules to a given table; while the other consists of applications of the usual natural deduction rules.

5 Conclusion and future work

We studied reasoning with tables in which local and global conditions are exploited. By regarding each row and column as a global object, we formalized our logic with tables LT, which is a subsystem of the heterogeneous logic with tables HLT. LT is shown to be complete with respect to the usual set-theoretical semantics (Theorem 3.17). Our inference rules, *row* and *col* rules, are designed to take full advantage of the global objects. Thus an inferential advantage of our tables is captured as multiple free rides of our rules (Proposition 4.2 and Theorem 4.3).

Our basic research can be extended in a variety of ways. Our correspondence tables and global objects (rows and columns) can be generalized, respectively. We here discuss the following among others.

- Our completeness of LT is restricted to the fragment in which only all-one-global formulas and uniquely determinable tables are considered. The theorem can be generalized by weakening this restriction or by introducing other inference rules. We intend investigating a more general completeness in the future.
- In addition to the free ride discussed in Section 4.1, there may be another kind of free ride in our system. From a local point of view, any addition of a symbol in an entry is just that—the addition of a symbol in that entry. From a global point of view, however, it means something additional, namely, the addition of a symbol in the *row* and *column* comprising the entry. As our inference rules act upon the number of symbols in a row or in a column, such an additional global effect can add up to trigger them and advance our reasoning. In fact, such effects can be seen abundantly in the example discussed in Section 2. We leave the formalization and analysis of this type of free ride to future work.
- We concentrated on global conditions that constraints, from the perspective of tables, on the numbers of $\bigcirc$ and $\times$ symbols in each row and column. However, there may be several different global conditions such as $\forall x(A(x) \leftrightarrow \neg C(x))$, which implies that columns A and C are opposite. Characterization and analysis of these general global conditions remain to be investigated.
- Although our table consists of a matrix over symbols $\bigcirc, \times, b$, besides these symbols, we typically use other symbols as well, such as characters and numbers. In particular, numbers make numerical reasoning possible, and we may need other types of inference rules to formalize this reasoning.

References

[1] Dave Barker-Plummer and John Etchemendy, A computational architecture for heterogeneous reasoning, *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 19, No. 3, 195-225, 2007.

[2] Dave Barker-Plummer and Nik Swoboda, Reasoning with coincidence grids–A sequent-based logic and an analysis of complexity, *Journal of Visual Languages and Computing*, 22 (1), 56-65, 2011.

[3] Jon Barwise and John Etchemendy, Heterogeneous Logic, in *Logical reasoning with diagrams*, G. Allwein and J. Barwise eds., Oxford Studies In Logic And Computation Series, 1996.

[4] Jon Barwise and John Etchemendy, Hyperproof: For Macintosh, *The Center for the Study of Language and Information Publications*, 1995.

[5] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*, The University of Wisconsin Press, Madison, WI, 1973.

[6] Bertin, Jacques, *Graphics and Graphic Information*, Originally published in France in 1977, Walter de Gruyter, Berlin, 1981.

[7] W. S. Cleveland, *The Elements of Graphing Data*, Hobart Press, Summit, NJ, 1994,

[8] Herbert B. Enderton, *A Mathematical Introduction to Logic*, Second Edition, Academic Press, 2000.

[9] Kenneth J. Gilhooly and Michael Wood and Paul R. Kinnear and Caroline Green, Skill in map reading and memory for maps, *Quarterly Journal of Experimental Psychology*, 40A, 87-107, 1988.

[10] Guthrie, John T. and Weber, Shelley and Kimmerly, Nancy, Searching Documents: Cognitive Processes and Deficits in Understanding Graphs, Tables, and Illustrations, *Contemporary Educational Psychology*, 18, 186–221, 1993.

[11] Kinnear, Paul R. and Wood, Michael, Memory for topographic contour maps, *British Journal of Psychology*, 78, 395-402, 1987.

[12] Stephen M. Kosslyn, *Elements of Graph Design*, W. H. Freeman and Company, 1994.

[13] Steven Pinker, A Theory of Graph Comprehension, *Artificial Intelligence and the Future of Testing*, Roy Freedle ed., L. Erlbaum Associates, 73–126, 1990.

[14] Ratwani, Raj M. and Trafton, J. Gregory and Boehm-Davis, Deborah A., Thinking Graphically: Connecting Vision and Cognition During Graph Comprehension, *Journal of Experimental Psychology: Applied*, 14, 1, 36–49, 2008.

[15] A. Shimojima, On the Efficacy of Representation, Ph.D thesis, Indiana University, 1996.

[16] Atsushi Shimojima, Derivative Meaning in Graphical Representations, *Proceedings of 1999 IEEE Symposium on Visual Languages*, 212-219, 1999.

[17] Atsushi Shimojima, The Inferential-Expressive Trade-Off: A Case Study of Tabular Representations, *Diagrams 2002*, 116–130, 2002.

[18] Ryo Takemura, Proof theory for reasoning with Euler diagrams: a Logic Translation and Normalization, *Studia Logica*, Volume 101, Issue 1, 157-191, 2013.

[19] Ryo Takemura, A Heterogeneous Logic with Tables, *Proceedings of International Workshop on Diagrams, Logic and Cognition (DLAC 2013)*, Jim Burton, Lopamudra Choudhury, eds., CEUR Series Vol. 1132, 9-16, 2014.

[20] E. R. Tufte, *Envisioning Information*, Graphics Press, Cheshire, CT, 1990.

[21] Howard Wainer, Understanding Graphs and Tables, *Educational Researcher*, 21, 14-23, 1992.

A *row* and *col* rules of LT

$row\times$ rules: each $\square$ is blank or $\times$.

$$\dfrac{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \bigcirc & \dots & \bigcirc & \square & \dots & \square \end{array}}{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \bigcirc & \dots & \bigcirc & \times & \dots & \times \end{array}}\ \exists^{i/n}X.X(a) \qquad \dfrac{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \square & \dots & \square & \bigcirc & \dots & \bigcirc \end{array}}{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \times & \dots & \times & \bigcirc & \dots & \bigcirc \end{array}}\ \exists^{i/n}X.\neg X(a)$$

$row\bigcirc$ rules: each $\square$ is blank or $\bigcirc$.

$$\dfrac{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \square & \dots & \square & \times & \dots & \times \end{array}}{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \bigcirc & \dots & \bigcirc & \times & \dots & \times \end{array}}\ \exists^{i/n}X.X(a) \qquad \dfrac{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \times & \dots & \times & \square & \dots & \square \end{array}}{\begin{array}{c|cccccc} & A_1 & \dots & A_i & A_{i+1} & \dots & A_n \\ \hline a & \times & \dots & \times & \bigcirc & \dots & \bigcirc \end{array}}\ \exists^{i/n}X.\neg X(a)$$

$col\times$ rules: each $\square$ is blank or $\times$. **$col\bigcirc$ rules**: each $\square$ is blank or $\bigcirc$.

$$\dfrac{\begin{array}{c|c} & A \\ \hline a_1 & \bigcirc \\ \vdots & \vdots \\ a_j & \bigcirc \\ a_{j+1} & \square \\ \vdots & \vdots \\ a_m & \square \end{array}}{\begin{array}{c|c} & A \\ \hline a_1 & \bigcirc \\ \vdots & \vdots \\ a_j & \bigcirc \\ a_{j+1} & \times \\ \vdots & \vdots \\ a_m & \times \end{array}}\ \exists^{j/m}x.A(x) \qquad \dfrac{\begin{array}{c|c} & A \\ \hline a_1 & \square \\ \vdots & \vdots \\ a_j & \square \\ a_{j+1} & \bigcirc \\ \vdots & \vdots \\ a_m & \bigcirc \end{array}}{\begin{array}{c|c} & A \\ \hline a_1 & \times \\ \vdots & \vdots \\ a_j & \times \\ a_{j+1} & \bigcirc \\ \vdots & \vdots \\ a_m & \bigcirc \end{array}}\ \exists^{j/m}x.\neg A(x) \qquad \dfrac{\begin{array}{c|c} & A \\ \hline a_1 & \square \\ \vdots & \vdots \\ a_j & \square \\ a_{j+1} & \times \\ \vdots & \vdots \\ a_m & \times \end{array}}{\begin{array}{c|c} & A \\ \hline a_1 & \bigcirc \\ \vdots & \vdots \\ a_j & \bigcirc \\ a_{j+1} & \times \\ \vdots & \vdots \\ a_m & \times \end{array}}\ \exists^{j/m}x.A(x) \qquad \dfrac{\begin{array}{c|c} & A \\ \hline a_1 & \times \\ \vdots & \vdots \\ a_j & \times \\ a_{j+1} & \square \\ \vdots & \vdots \\ a_m & \square \end{array}}{\begin{array}{c|c} & A \\ \hline a_1 & \times \\ \vdots & \vdots \\ a_j & \times \\ a_{j+1} & \bigcirc \\ \vdots & \vdots \\ a_m & \bigcirc \end{array}}\ \exists^{j/m}x.\neg A(x)$$