A Heterogeneous Logic with Tables
(Extended Abstract)

Ryo Takemura
Nihon University, Japan
Email: takemura.ryo@nihon-u.ac.jp

***Abstract*—Correspondence tables are a basic yet widely applied graphical/diagrammatical representation method. We investigate a certain type of reasoning with tables by exploiting local conditions, which specify the data in some table entries, and global conditions, which are constraints over every row and column. To formalize such a reasoning, we introduce a heterogeneous logic with tables by combining the usual first-order formulas in the framework of natural deduction. In order to formalize tables and formulas in the same framework, we apply the syntax and semantics of many-sorted logic.**

I. Introduction

Since the 1990s, logicians have studied diagrammatic and graphical representations from a formal logical viewpoint. As a result, it has been shown that they can be studied as formal objects equivalent to logical formulas. Formal syntax and semantics have been defined, logical properties of diagrammatic systems, such as soundness and completeness, have been investigated, and proof-theory has also been developed. Furthermore, studies on the characteristics of diagrammatic systems, such as their expressive power, drawing techniques, effective usage, and cognitive properties, have also been conducted. For more information on these studies, see proceedings, e.g., [10], [7], [5], of Diagrams conference series on the theory and application of diagrams.

On the basis of such research, a natural next step is to study heterogeneous reasoning, combining various diagrammatic, graphical, and sentential representations. Jon Barwise, one of the first logicians to investigate the logical status of diagrams, pointed out the importance of studying heterogeneous systems in an early work [1], and he introduced a heterogeneous system combining graphical representations and first-order formulas on a blocks world, e.g., [1], [3]. His Hyperproof project has recently been extended to the Openproof project. Heterogeneous systems based on Venn (and Euler) diagrams and first-order formulas have also been studied, e.g., [8], [11]. Furthermore, a heterogeneous system based on spider diagrams and its implementation has been developed [12].

In this paper, we study a heterogeneous system based on first-order formulas and tables. Tables and their usage have been studied in the diagrammatic reasoning community, for example, in [4], [9], [2]. Our tables are correspondence tables represented by a rectangular arrangement of given data such as symbols, characters, or numbers. Tables are one of the most basic graphical representations, and have been applied for various usages. From a cognitive science viewpoint, Shimojima [9] studied the semantic mechanism of extracting information from a given table.

In addition to the static reading of given tables, we can use tables dynamically to solve a given problem. This involves constructing a table and adding pieces of information, before manipulating, and finally reading the table. For example, let us consider a situation where we assign a working day to each of six people. We should establish a one-to-one correspondence between the six people and the working days in a week. In such a situation, we are usually given further pieces of information about who can or cannot work on specific days. We are able to solve such a problem effectively by constructing a correspondence table. (See Examples 1 and 2.) In such a problem, we are, in general, given certain constraints over the framework of the given problem such as a one-to-one correspondence between people and days, which we call *global conditions*, and we are also given some particular pieces of information such as information about who can or cannot work on specific days, which we call *local conditions*. This type of reasoning task is found, for example, in civil servant examinations in Japan and in so-called logic puzzles, and is one of the most natural reasoning that could be formalized in combination with the first-order formulas.

Barker-Plummer and Swoboda [2] discussed similar problems. They consider n-ary relationships among objects, and their system is defined to be simple and have as few rules as possible. On the other hand, we concentrate on basic tables representing the binary relationship between objects, and we design our inference rules so that we take full advantage of the effectiveness of our tables. Furthermore, our system is heterogeneous by combining tables and first-order formulas.

We propose a heterogeneous logic with tables based on Gentzen's natural deduction. We first illustrate the type of reasoning discussed in this paper. We then introduce the syntax and semantics of our heterogeneous logic with tables, HLT. In order to formalize tables and formulas in the same framework, we apply the syntax and semantics of many-sorted logic.

II. A heterogeneous logic with tables

In Section II-A, we first specify, through some examples, the type of reasoning considered in this paper. We then define the syntax of HLT in Section II-B, and its semantics in Section II-C. We introduce the inference rules of HLT in Section II-D, and define, in Section II-E, the translation of tables into formulas. This demonstrates the soundness and completeness theorems of HLT via the theorems of many-sorted logic.

A. Examples of reasoning with tables

Let us investigate the following examples of reasoning with tables, and then discuss the characteristics of such reasoning.

	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

Example 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, conditions (3) and (4) imply the following (in fact, under (4), conditions (3) and (5) are equivalent):

(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.” Then, 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 ○ into each entry (x, Y) for which “x works on Y” holds, and insert × when “x does not work on Y” holds. Thus, we obtain the table T_0 shown in Fig. 1. Note that we applied condition (5) (stated previously) 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 ○, and the other entries should be ×;

(7) In each column, exactly one entry should be marked as ○, and the other entries should be ×.

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

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

Although all entries are either ○ or × 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 1.

Example 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 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 1. Note that conditions (6) and (7) in Example 1 become the following:

(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 ×.

	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}

Fig. 2

We begin with the table T_8 in Fig. 2. Because the $(b, M), (c, M)$ entries are $\times$, we find, by (10), that $(a, M), (d, M)$ are $\bigcirc$, as illustrated in T_9. In a similar way, we obtain T_{10}. Then, because the entries of $(a, M), (a, W)$ are $\bigcirc$, we find, by (9), that (a, T) is $\times$, as in T_{11}, and in a similar way to Example 1, we finally obtain table T_{15}.

In the above examples, although the number of $\bigcirc$ 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 1 and 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 every row and column. 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: that is, we check each row or column, and apply appropriate global conditions controlling rows and columns. Thus, based on the local and global conditions, our reasoning using tables is conducted as follows.

(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) $\rightarrow$ (5)).
(II) We construct a correspondence table using these local conditions (e.g., T_0).
(III) By applying the 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 examples 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 from 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 here is based on Gentzen's natural deduction.

B. Syntax of HLT

In order to formalize our heterogeneous reasoning with tables, we adopt two-sorted logic, in which constants and variables of the first-order language are divided into two sorts: sorts of row and column. Although usually distinguished explicitly by sort symbols in many-sorted logic, we distinguish the two by lower- and uppercase letters: a constant (resp. variable) of the row sort is denoted by a (resp. x), and that of the column sort is denoted by A (resp. X). (See, for example, [6] for many-sorted logic.) Then, by $\bigcirc(a, B)$ we mean "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 $\bigcirc(a, B)$.

We begin by specifying some vocabulary.

Definition 3 (Vocabulary)**.** We use the following symbols.

Row-constants: $a_1, a_2, \ldots$; Col-constants: $A_1, A_2, \ldots$
Row-variables: $x_1, x_2, \ldots$; Col-variables: $X_1, X_2, \ldots$
Predicate: $\bigcirc(\ ,\ )$

Constants and variables are collectively called **terms**, and are denoted by $s, t, u, \ldots$.

Using the above symbols, we construct our formulas as follows.

Definition 4. An **atomic formula** is of the form $\bigcirc(s, t)$ for a row-constant/variable s and a col-constant/variable t. Based on atomic formulas, complex **formulas** are defined inductively as usual:

$$\varphi := \quad \bigcirc(s, t) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \neg\varphi$$

$$\forall x.\varphi \mid \exists x.\varphi \mid \forall X.\varphi \mid \exists X.\varphi$$

In particular, we denote the negation of an atom $\neg\bigcirc(s,t)$ by $\times(s,t)$. Formulas of the forms $\bigcirc(s,t)$ and $\times(s,t)$ are collectively called **literals**. Literals containing no variables are called **ground literals** (or closed literals), and they are denoted by $\alpha, \alpha_1, \alpha_2, \ldots$. We denote by $\overline{\alpha}$ the ground literal $\times(a,B)$ when α is $\bigcirc(a,B)$, and $\bigcirc(a,B)$ when α is $\times(a,B)$. Formulas containing no free variables are said to be **closed**.

We sometimes write $\bigcirc(s,t)$ as $t(s)$, and $\times(s,t)$ as $\neg t(s)$.

In order 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 $\exists^{i/n}x.A(x)$, i.e., $\exists^{i/n}x.\bigcirc(x,A)$.

Definition 5 (Global formula)**.** For fixed sets of row-constants $\mathcal{R} = \{a_1,\ldots,a_n\}$ or of col-constants $\mathcal{C} = \{A_1,\ldots,A_n\}$, the following forms of formulas are called **global formulas**: For any A and a,

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

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

Global formulas are simply abbreviations of the appropriate first-order formulas. Let σ be a permutation of the given row-constants $a_1,\ldots,a_n$, and S_n be the set of all their permutations. We then define $\exists^{i/n}x \in \mathcal{R}.A(x)$ as:

$$\bigvee_{\sigma\in S_n}\Big(A(a_{\sigma 1})\wedge\cdots\wedge A(a_{\sigma i})\wedge\neg A(a_{\sigma i+1})\wedge\cdots\wedge\neg A(a_{\sigma n})\Big)$$

We treat $\exists^{i/n}x \in \mathcal{R}.\neg A(x)$ in a similar manner.

Let σ be a permutation of the given col-constants $A_1,\ldots,A_n$, and S_n be the set of all their permutations. We then define $\exists^{i/n}X \in \mathcal{C}.X(a)$ as:

$$\bigvee_{\sigma\in S_n}\Big(A_{\sigma 1}(a)\wedge\cdots\wedge A_{\sigma i}(a)\wedge\neg A_{\sigma i+1}(a)\wedge\cdots\wedge\neg A_{\sigma n}(a)\Big)$$

We again treat $\exists^{i/n}X \in \mathcal{C}.\neg X(a)$ similarly.

Example 6. For example, for some row-constant a and col-constants $\mathcal{C} = \{A_1, A_2, A_3\}$, the global formula $\exists^{2/3}X \in \mathcal{C}.X(a)$ is the abbreviation of the following formula:

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

where we omit trivial permutations such as $A_2(a)\wedge A_1(a)\wedge\neg A_3(a)$, which is equivalent to the first disjunct in the above.

Next, we define our tables.

Definition 7. 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-constants $a_1,\ldots,a_m$ and columns are labeled by distinct col-constants $A_1,\ldots,A_n$.

In a specific representation of a table, we usually omit the symbol "b" and leave the entry blank.

Remark 8. 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-constants (resp. col-constants) of T.

Note that no entry can be marked with more than one of $\bigcirc, \times, b$ at the same time.

As usual, any pair of tables, say T_1 and T_2, are identical if they consist of the same constants, 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 9 (Equivalence of tables)**.** A table T_1 is a subtable of T_2, written as $T_1 \subseteq T_2$, when:

- all row- and col-constants of T_1 are also those of T_2;
- for any (a_i, A_j)-entry of 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** when $T_1 \subseteq T_2$ and $T_2 \subseteq T_1$ hold.

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

C. *Semantics of* HLT

We now define the semantics of our HLT as a particular case of the semantics of many-sorted logic. (See [6] for the semantics of many-sorted logic.)

Definition 10. A **structure** M is (M_{row}, M_{col}, I), where: M_{row} and M_{col} are disjoint non-empty domains for the row and column sorts, respectively.
I is an interpretation function such that:

- $I(a) \in M_{row}$ for each row-constant a;
- $I(A) \in M_{col}$ for each col-constant A;
- $I(\bigcirc) \subseteq M_{row}\times M_{col}$.

Definition 11. A **valuation** v in M is a function that assigns every row-variable x an entity of M_{row}, i.e., $v(x) \in M_{row}$, and every col-variable X an entity of M_{col}, i.e., $v(X) \in M_{col}$.

Definition 12 (Truth conditions)**.** The notion of satisfaction of a formula φ in a structure M with a valuation v, written $M \models \varphi[v]$, is defined inductively as follows:

- $M \models \bigcirc(s,t)[v]$ if and only if $(I(s)[v], I(t)[v]) \in I(\bigcirc)$, where $I(s)[v] = I(a)$ when s is a row-constant a, and $I(s)[v] = v(x)$ when s is a row-variable x; Similarly for t;
- $M \models \neg\varphi[v]$ if and only if $M \not\models \varphi[v]$;

  Truth conditions for other connectives $\wedge, \vee, \rightarrow$ are defined as usual;
- $M \models \forall x.\varphi[v]$ if and only if, for all $m \in M_{row}$, $M \models \varphi[v(x \mapsto m)]$,

$$\frac{\begin{matrix}\vdots & \vdots\\ \varphi & \psi\end{matrix}}{\varphi\wedge\psi}\wedge I \qquad \frac{\begin{matrix}\vdots\\ \varphi_1\wedge\varphi_2\end{matrix}}{\varphi_i}\wedge E_{i\,(i=1,2)} \qquad \frac{\begin{matrix}\vdots\\ \varphi_i\end{matrix}}{\varphi_1\vee\varphi_2}\vee I_{i\,(i=1,2)} \qquad \frac{\begin{matrix} & [\varphi]^n & [\psi]^n\\ \vdots & \vdots & \vdots\\ \varphi\vee\psi & \theta & \theta\end{matrix}}{\theta}\vee E, n$$

$$\frac{\begin{matrix}[\varphi]^n\\ \vdots\\ \psi\end{matrix}}{\varphi\to\psi}\to I, n \qquad \frac{\begin{matrix}\vdots & \vdots\\ \varphi & \varphi\to\psi\end{matrix}}{\psi}\to E \qquad \frac{\begin{matrix}[\varphi]^n\\ \vdots\\ \bot\end{matrix}}{\neg\varphi}\neg I, n \qquad \frac{\begin{matrix}\vdots & \vdots\\ \varphi & \neg\varphi\end{matrix}}{\bot}\neg E \qquad \frac{\begin{matrix}\vdots\\ \bot\end{matrix}}{\varphi}\bot E \qquad \frac{\begin{matrix}[\neg\varphi]^n\\ \vdots\\ \bot\end{matrix}}{\varphi} RAA, n$$

$$\frac{\begin{matrix}\vdots\\ \varphi\end{matrix}}{\forall x.\varphi}\forall Ir \qquad \frac{\begin{matrix}\vdots\\ \varphi\end{matrix}}{\forall X.\varphi}\forall Ic \qquad \frac{\begin{matrix}\vdots\\ \forall x.\varphi\end{matrix}}{\varphi[x:=t]}\forall Er \qquad \frac{\begin{matrix}\vdots\\ \forall X.\varphi\end{matrix}}{\varphi[X:=t]}\forall Ec$$

where, in $\forall Ir$ (and $\forall Ic$), the variable x (resp. X) may not occur free in any open hypothesis on which φ depends; in $\forall Er$ (and $\forall Ec$), t is a row-variable/constant (resp. col-variable/constant).

$$\frac{\begin{matrix}\vdots\\ \varphi[x:=t]\end{matrix}}{\exists x.\varphi}\exists Ir \qquad \frac{\begin{matrix}\vdots\\ \varphi[X:=t]\end{matrix}}{\exists X.\varphi}\exists Ic \qquad \frac{\begin{matrix} & [\varphi]^n\\ \vdots & \vdots\\ \exists x.\varphi & \psi\end{matrix}}{\psi}\exists Er, n \qquad \frac{\begin{matrix} & [\varphi]^n\\ \vdots & \vdots\\ \exists X.\varphi & \psi\end{matrix}}{\psi}\exists Ec, n$$

where, in $\exists Ir$ (and $\exists Ic$), t is a row-variable/constant (resp. col-variable/constant); in $\exists Er$ (and $\exists Ec$), x (resp. X) may not occur free in ψ nor in any open hypothesis on which ψ depends, except in φ.

Fig. 3 Inference rules for formulas in HLT

where $v(x \mapsto m)$ is the valuation that is exactly the same as v except for x, and x is assigned to m;

- $M \models \forall X.\varphi[v]$ if and only if, for all $m \in M_{col}$, $M \models \varphi[v(X \mapsto m)]$;
- $M \models \exists x.\varphi[v]$ if and only if there exists $m \in M_{row}$ such that $M \models \varphi[v(x \mapsto m)]$;
- $M \models \exists X.\varphi[v]$ if and only if there exists $m \in M_{col}$ such that $M \models \varphi[v(X \mapsto m)]$;
- $M \models T$ for a table T if and only if $M \models \bigcirc(s,t)$ for any entry (s,t) of T that is $\bigcirc$, and $M \models \times(s,t)$ for any entry (s,t) of T that is $\times$.

M is said to be a **model** of φ, written as $M \models \varphi$, when $M \models \varphi[v]$ holds for any valuation v in M.

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

Definition 13 (Semantic consequence)**.** Let Γ be a set of closed formulas, let $\mathcal{G}$ be a set of global formulas, and let T be a table. A closed formula φ is said to be a **semantic consequence** of $\Gamma, \mathcal{G}, T$, written as $\Gamma, \mathcal{G}, T \models \varphi$, when any model of Γ, $\mathcal{G}$, and T is also a model of φ.

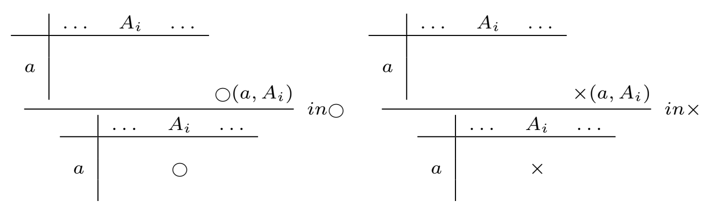
D. Inference rules of HLT

In this section, we introduce the inference rules of HLT, which consist of the usual natural deduction rules for formulas and rules of table manipulation.

Definition 14 (Inference rules)**.** The inference rules of HLT consist of the following rules for formulas and rules for tables.

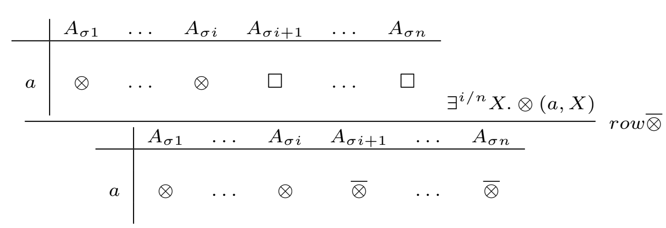
Rules for formulas are the rules for $\wedge, \vee, \to, \neg, \forall, \exists$, and $\bot E$ and RAA listed in Fig. 3.

Rules for tables are the following in, row, col, ext rules:



In order to describe the same types of rules, we use the symbol $\otimes$, which denotes either $\bigcirc$ or $\times$. Furthermore, $\overline{\otimes}$ denotes $\bigcirc$ if $\otimes$ is $\times$, and denotes $\times$ if $\otimes$ is $\bigcirc$. The following four rules of row and col are duplicated depending on whether $\otimes$ is $\bigcirc$ or $\times$.

Let σ be a permutation of columns $A_1, \ldots, A_n$. Under the permutation, we assume that entries marked as $\bigcirc$ and those marked as $\times$ are grouped.



where each $\square$ is blank or $\overline{\otimes}$.

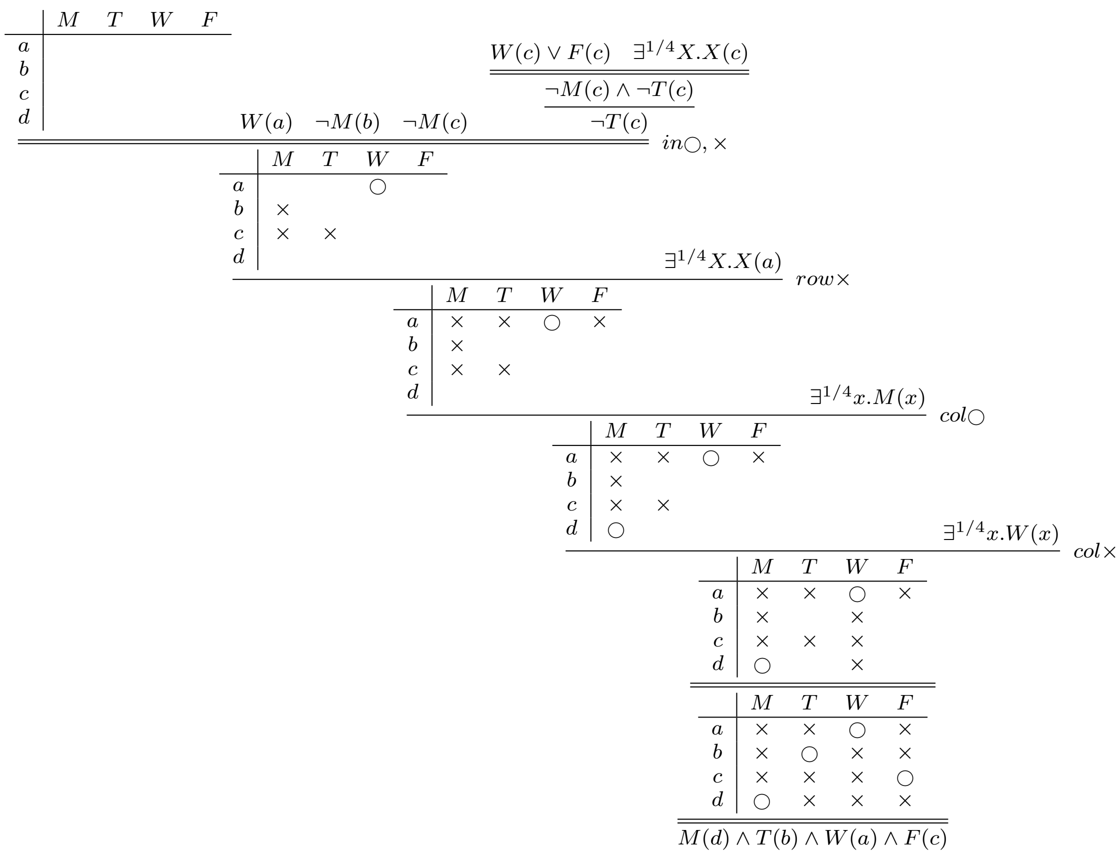
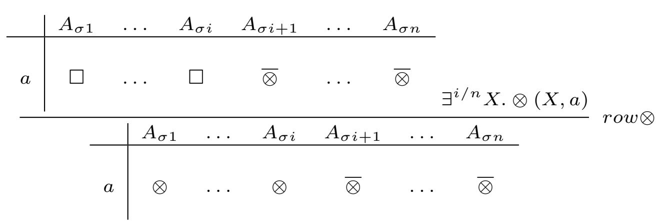
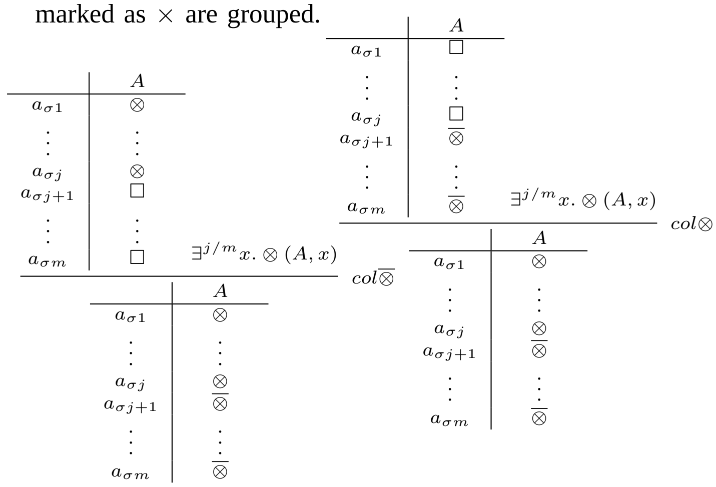


Fig. 4 A proof in HLT of Example 1. In the figure, each double line is an abbreviation of some applications of inference rules.



where each $\square$ is blank or $\otimes$.

Let σ be a permutation of rows $a_1, \ldots, a_m$. Under the permutation σ, we assume that entries marked as $\bigcirc$ and those marked as $\times$ are grouped.



where each $\square$ is blank or $\overline{\otimes}$. where each $\square$ is blank or $\otimes$.

$$\frac{T}{\bigwedge\{\bigcirc(a_j, A_i) \mid (a_j, A_i) \text{ is } \bigcirc \text{ in } T\} \wedge \bigwedge\{\times(a_j, A_i) \mid (a_j, A_i) \text{ is } \times \text{ in } T\}}\ ext$$

The in-rule enables us to convert ground literals into a table, and, conversely, the ext-rule enables us to convert a table into a conjunction of literals. The row and col rules pertain to the manipulation of tables.

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

Definition 15. Let Γ be a set of closed formulas, let $\mathcal{G}$ be a set of global formulas, and let T be a table. A closed formula φ is provable from $\Gamma, \mathcal{G}, T$, written as $\Gamma, \mathcal{G}, T \vdash \varphi$, when there exists a proof of φ from the open assumption $\Gamma, \mathcal{G}, T$.

Example 16. A proof in HLT of Example 1 is given in Fig. 4.

E. Translation of tables

We give a translation of the rules for tables into those for usual natural deduction without tables, from which the soundness and completeness theorems of HLT are obtained.

We first define the following terminology.

Definition 17. A conjunction of literals is said to be **consistent** when it has a model.

We define the translation of tables into a formula. This is completely parallel to the ext-rule of HLT.

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

Fig. 5 A translation given in Example 20

Definition 18. Each table T is **translated** into a conjunction of (consistent ground) literals T° as follows:

$$
\begin{aligned}
T^\circ = & \bigwedge\{\bigcirc(a_j, A_i) \mid (a_j, A_i) \text{ is } \bigcirc \text{ in } T\} \\
& \wedge \bigwedge\{\times(a_j, A_i) \mid (a_j, A_i) \text{ is } \times \text{ in } T\}
\end{aligned}
$$

Conversely, it is easily seen that, for any consistent conjunction of ground literals, there exists a corresponding table. Thus, we may regard a table T and a consistent conjunction $\mathcal{L}$ of ground literals to be interchangeable, and (slightly abusing our notation) we write, for example, $\mathcal{L} \subseteq T$.

Based on the translation of tables, we give a translation of table manipulations into rules of natural deduction. (See also Example 20 below.)

Theorem 19 (Translation). *If $\mathcal{G}, T_1 \vdash T_2$ with only row and col rules in* HLT*, then $\mathcal{G}, T_1^\circ \vdash T_2^\circ$ in natural deduction without tables.*

Proof: It is sufficient to give a translation of row and col rules into combinations of rules of natural deduction without tables.

Because all cases are treated in a similar way, we show only the following $row\times$-rule (we assume that all $\square$ of the rule are blank for simplicity).

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

From our translation of tables, the table in the premises of the $row\times$-rule is translated into $A_{\sigma 1}(a) \wedge \cdots \wedge A_{\sigma i}(a)$. Note also that another premise of the $row\times$-rule $\exists^{i/n}X.X(a)$ is $\bigvee_{\tau \in S_n}\Big(A_{\tau 1}(a) \wedge \cdots \wedge A_{\tau i}(a) \wedge \neg A_{\tau i+1}(a) \wedge \cdots \wedge \neg A_{\tau n}(a)\Big)$. Given these two formulas, we construct a natural deduction proof.

We denote by $\mathbf{A}_{\tau i}$ the conjunction $A_{\tau 1}(a) \wedge \cdots \wedge A_{\tau i}(a)$, and by $\neg\mathbf{A}_{\tau n}$ the conjunction $\neg A_{\tau i+1}(a) \wedge \cdots \wedge \neg A_{\tau n}(a)$. Then, we can write $\exists^{i/n}X.X(a)$ as $\bigvee_{\tau \in S_n}(\mathbf{A}_{\tau i} \wedge \neg\mathbf{A}_{\tau n})$.

In order to apply the $\vee E$-rule of natural deduction, we divide the following cases according to the form of each disjunct $\mathbf{A}_{\tau i} \wedge \neg\mathbf{A}_{\tau n}$ of the above $\exists^{i/n}X.X(a)$, and we derive $\neg\mathbf{A}_{\sigma n}$ in each case.

1) When $\tau = \sigma$, or when $\mathbf{A}_{\sigma i} \leftrightarrow \mathbf{A}_{\tau i}$ holds, we apply the $\wedge E$-rule of natural deduction to $\mathbf{A}_{\tau i} \wedge \neg\mathbf{A}_{\tau n}$ to obtain $\neg\mathbf{A}_{\tau n}$, which is equivalent to $\neg\mathbf{A}_{\sigma n}$.
2) Otherwise, we apply the $\wedge E$-rule to $\mathbf{A}_{\tau i} \wedge \neg\mathbf{A}_{\tau n}$, and obtain $\neg\mathbf{A}_{\tau n}$. Note that at least one of the conjuncts of $\neg\mathbf{A}_{\tau n}$ contradicts a conjunct of $\mathbf{A}_{\sigma i}$ by the definition of the permutation for $\exists^{i/n}X.X(a)$. Hence, by applying the $\bot E$-rule, we obtain $\neg\mathbf{A}_{\sigma n}$.

Therefore, in every case, we obtain $\neg\mathbf{A}_{\sigma n}$ from $\mathbf{A}_{\tau i} \wedge \neg\mathbf{A}_{\tau n}$. Thus, by applying the $\vee E$-rule, we obtain a proof of $\neg\mathbf{A}_{\sigma n}$.

As we have $\mathbf{A}_{\sigma i}$ by the initial assumption, by applying the $\wedge I$-rule, we obtain $\mathbf{A}_{\sigma i} \wedge \neg\mathbf{A}_{\sigma n}$, which is the translation of the conclusion of the given $row\times$-rule. ■

Example 20. Let us consider the following application of the $row\times$-rule:

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

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 abbreviate $A_i(a)$ as A_i and omit trivial permutations. This application of the $row\times$-rule is translated as in Fig. 5.

The soundness of HLT is obtained straightforwardly by the above theorem of translation, because each rule for tables is translated into a combination of rules for formulas by Theorem 19, and each rule for formulas, i.e., of many-sorted logic, is known to be sound.

Theorem 21 (Soundness of HLT). *Let Γ be a set of closed formulas, $\mathcal{G}$ be a set of global formulas, T be a table, and φ be a closed formula. If $\Gamma, \mathcal{G}, T \vdash \varphi$ in* HLT*, then $\Gamma, \mathcal{G}, T \models \varphi$.*

Because our inference rules and semantics of HLT "without tables" are just particular cases of those of many-sorted logic, the following completeness theorem of HLT holds via the theorem of many-sorted logic. (See [6] for the completeness of many-sorted logic.)

Theorem 22 (Completeness of HLT). *Let Γ be a set of closed formulas, $\mathcal{G}$ be a set of global formulas, T be a table, and φ be a closed formula. If $\Gamma, \mathcal{G}, T \models \varphi$, then $\Gamma, \mathcal{G}, T \vdash \varphi$ in* HLT.

Proof: Based on the translation of tables, a table T and the consistent set of ground literals T° are equivalent. Thus, $\Gamma, \mathcal{G}, T \models \varphi$ is equivalent to $\Gamma, \mathcal{G}, T^\circ \models \varphi$. Then, by the completeness theorem of many-sorted logic, we have $\Gamma, \mathcal{G}, T^\circ \vdash \varphi$. Again, by the translation (more precisely, by the ext-rule of HLT for T), we have $\Gamma, \mathcal{G}, T \vdash \varphi$ in HLT. ■

III. Conclusion and Future Work

We investigated reasoning with tables by exploiting local and global conditions. This type of reasoning is not just a mere puzzle, but can be applied to simple scheduling problems. In order to formalize our reasoning with tables, we introduced a heterogeneous logic with tables, HLT, with a combination of first-order formulas. For the formalization of HLT, we applied the syntax and semantics of many-sorted logic. Although we actually applied two-sorted logic, as we studied the most basic binary relationship between objects, our system can be extended to cover the n-ary relationship between objects, as in [2], by applying the general many-sorted logic.

We defined inference rules for HLT that consist of the usual natural deduction rules for the first-order formulas and table manipulation rules. These were designed to reflect our intuitive and effective manipulation of the tables. We investigated the translation of tables and their manipulations into formulas and natural deduction inference rules. Then, based on these translations, we obtained soundness and completeness theorems of HLT by reducing them to the usual many-sorted logic theorems.

In this paper, for the sake of simplicity, we concentrated on the most basic global formulas, such as "among n objects, there are 'exactly i objects' ...." Our system can be easily extended by introducing the following forms for the global formulas:

- $\exists^{\leq i/n} x.A(x)$ meaning "among n objects, there are 'at most' i objects that are A," and
- $\exists^{\geq i/n} x.A(x)$ meaning "among n objects, there are 'at least' i objects that are A."

Similarly, we could consider $\exists^{\leq i/n} X.X(a)$ and $\exists^{\geq i/n} X.X(a)$, as well as negative literals. These formulas are also just abbreviations of appropriate first-order formulas, similar to the definition of $\exists^{i/n}$ in Section II-B.

Based on the informal analysis of Examples 1 and 2 and the translation between tables and formulas (Theorem 19 and Example 20), we may point out the following differences, for example, between the usual sentential systems and our table system.

(1) In a sentential system, on the one hand, we need to derive each statement corresponding to an entry of a table one by one. On the other hand, in our table system, by an application of our rule, a number of entries are filled with $\bigcirc$ or $\times$ at once, that is, a number of statements are derived at once. This suggests that the number of steps of inference is reduced by our row and col rules as illustrated in Example 20.

(2) However, the above difference appears in a fragment where our tables and row and col rules work most effectively. In other words, our table fragment of HLT, consisting only of rules for tables (row, col, in, ext rules) without rules for formulas, is not complete with respect to our semantics. That is, there exists a formula which is semantically valid, but is not provable with only rules for tables. In contrast, the natural deduction system without tables is complete, and there are no restriction on a given problem.

(3) Once a table is constructed, and sentences are translated into the table (for example, "a works on Wednesday ($W(a)$)" is translated as "(a, W)-entry is $\bigcirc$," and "exactly one person should work on Wednesday ($\exists^{1/4} x.W(x)$)" is translated as "there is exactly one $\bigcirc$ in column W"), reasoning with the table is conducted by completely graphical or geometrical manipulations on the table, i.e., by dealing only with the number of $\bigcirc, \times$ symbols in each row and column. Furthermore, there are two ways, in a sentential system, to derive a statement, say "d works on Monday ($M(d)$)": deriving $M(d)$ from $\neg M(a), \neg M(b), \neg M(c)$ by checking every person on Monday, and from $\neg T(d), \neg W(d), \neg F(d)$ by checking every possible working day of d. These derivations correspond, in our table system, to an application of col rule and row rule, respectively.

These comparison between sentential systems and our table system should be investigated by combining logic and cognitive science methods.

REFERENCES

[1] Gerard Allwein and Jon Barwise, eds., *Logical reasoning with diagrams*, Oxford Studies In Logic And Computation Series, 1996.

[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, A Computational Architecture for Heterogeneous Reasoning, *Proceedings of the Seventh Conference on Theoretical Aspects of Rationality and Knowledge*, 1-14, 1998.

[4] Jon Barwise and Eric Hammer, Diagrams and the concept of logical system, in *Logical reasoning with diagrams*, G. Allwein and J. Barwise, eds., Oxford Studies In Logic And Computation Series, 1996.

[5] Philip T. Cox, Beryl Plimmer, Peter Rodgers eds., *Diagrammatic Representation and Inference: 7th International Conference, Diagrams 2012*, Lecture Notes in Computer Science, Vol. 7352, 2012.

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

[7] Ashok K. Goel, Mateja Jamnik, N. Hari Narayanan eds., *Diagrammatic Representation and Inference: 6th International Conference, Diagrams 2010*, Lecture Notes in Computer Science, Vol. 6170, 2010.

[8] Eric Hammer, Reasoning with Sentences and Diagrams, *Notre Dame Journal of Formal Logic*, Volume 35, Number 1, 73-87, 1994.

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

[10] Gem Stapleton, John Howse, John Lee Eds., *Diagrammatic Representation and Inference: 5th International Conference, Diagrams 2008*, Lecture Notes in Computer Science, Vol. 5223, 2008.

[11] Nik Swoboda and Gerard Allwein, Heterogeneous Reasoning with Euler/Venn Diagrams Containing Named Constants and FOL, *Electronic Notes in Theoretical Computer Science*, 134, 153-187, 2005.

[12] Matej Urbas and Mateja Jamnik, Heterogeneous Proofs: Spider Diagrams Meet Higher-Order Provers, *Interactive Theorem Proving*, Lecture Notes in Computer Science, Volume 6898, 376-382, 2011.