The Inferential-Expressive Trade-Off: A Case Study of Tabular Representations

Atsushi Shimojima

School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1 Asahi-dai, Tatsunokuchi, Nomi-gun, Ishikawa, 923-1292, Japan

Abstract. Many graphical systems (e.g., Euler diagrams, maps, pictorial images, and even tables) support efficient inferences or rich presentation of information apparently at the expense of expressive flexibility. This association of inferential efficiency, expressive richness, and expressive inflexibility in a graphical system has been pointed out by various researchers (e.g., Sloman [1], Stenning and Oberlander [2]). This paper investigates the semantic mechanism of the association by closely examining a particular system of tabular representations, which, despite its simplicity, clearly exhibits all those opposing functional traits. Using a semantic framework of channel theory (Barwise and Seligman [3]), we will show that the common mechanism is a parallelism between abstraction relations in represented properties and in representing properties.

Intuitions tell us that both positive and negative traits coexist in graphics. Graphics are rich in content; they facilitate our inferences on the depicted objects or situations; but they often have severe limitations in what can be expressed. Recent studies have started revealing the semantic mechanisms behind these functional traits of graphics individually: their expressive richness (e.g., Kosslyn [4]), their potentials for efficient inferences (e.g., Barwise and Etchemendy [5]), and their expressive inflexibility (e.g., Stenning and Oberlander [2]). Now, is there any relationship among these coexisting traits? Or are they all independent from each other?

Many researchers have suggested their connection. Sloman [1] mentions a trade-off between efficient "problem-solving power" and expressive generality, suggesting that "analogical" representation systems (such as graphical systems) sacrifice the latter for the former (pp. 217). Barwise and Etchemendy [5] mention a trade-off between expressive richness and expressive flexibility. Comparing pictorial images and first-order sentences, they point out that graphics are good at expressing conjunctive information but not at expressing disjunctive information (p. 22). Barwise and Hammer [6] state that the expressive generality of a system is often incompatible with its capacity for being homomorphic to its target domain, while this capacity is a root of a system's expressive richness (pp. 47–48).

It is, however, Stenning and Oberlander [2] who have offered the clearest insight into the connection between positive and negative functional traits of graphics. According to them, "graphical representation such as diagrams limit abstraction and thereby aid processibility" (p. 98). Here, "limiting abstraction" means limited capacities of expressing weak, non-specific information, while "processibility" is capacities of supporting

M. Hegarty, B. Meyer, and N. Hari Narayanan (Eds.): Diagrams 2002, LNAI 2317, pp. 116–130, 2002.

	F_1	F_2	F_3	F_4	F_5	F_6
A	○		○			
B	○					
C	○					
D	○	○	○	○	○	
E		○		○	○	
G			○			○

Fig. 1. A well-formed representation of the system $\mathcal{R}_t$ of feature tables.

efficient inferences on the user's part. Stenning and Inder [7] even propose a general ordering of representation systems according to their expressive flexibility, and compare the inferential potentials of systems in different places in this ordering.

Thus, previous studies strongly suggest an association among (a) the potential of a graphical system to allow efficient inferences, (b) its capacity for rich presentation of information, and (c) its tendency to prohibit flexible presentations of information. What is then the semantic mechanism behind this association? Is there any common property of graphical representation systems from which these positive and negative traits all derive? Let us call this question *the trade-off problem*.

In this paper, we try to answer this question by investigating a particular system of graphical representations in detail. It is a system of tabular representations of a common type, which, despite its simple syntax and semantics, exhibit the apparent association in question. Relying on previous studies of inferential and expressive potentials of graphics, we will start with showing the exact senses in which this tabular system supports efficient inferences, enables rich expression of information, and prohibits flexible expression of information (section 1). After introducing basic semantic concepts, we will show that the tabular system has a property that can be called "trackings of capturing relations over homomorphic exhaustive sets" (section 2). This property is a near-sufficient condition for all the relevant functional traits, and hence can be considered the semantic mechanism behind the trade-off phenomenon. We will close the paper by discussing the extent and generalizability of our analysis (section 3). Discussions will be kept informal throughout this paper, for compact exposition of core ideas.

1 Reproducing the Trade-Off

The tabular system we investigate consists of simple "feature" tables such as the one in Figure 1. Here, the labels for rows, "A," "B," ... and "G," are the names of ink-jet printer models and the labels for columns, "F_1," "F_2," ... and "F_6," are the names of various functions that ink-jet printers may have. (For the ease of reference, we use these simple names for printer models and their potential functions, instead of real names and descriptions, such as "Epson PM750," "Canon M70," "Print on A3-size paper" and "Print in 720 $\times$ 360 dpi.")

The semantics of this system is natural one: if a circle appears in the intersection of the raw labeled by "X" and the column labeled by "Y," then it indicates that the

	F_1	F_2	F_3	F_4	F_5	F_6
A	○		○			
B			○	○		
C	○		○			○

Fig. 2. The result of expressing the set of information $\{(1), (2), (3)\}$ in an $\mathcal{R}_t$-table.

printer model X has the function Y; if that position is blank, it indicates that the printer model X lacks the function Y. Thus, according to the table in Fig 1, the model A has the function F_3 but the model B does not; the model G has the function F_6 but the model E does not, and so on. The syntax of the system requires that all the six printer names appear as labels for rows in alphabetical order, and that all the six function names appear as labels for columns in numerical order. Each cell of a table must either be blank or have a circle in it. We call this system of tabular representations $\mathcal{R}_t$. [1]

Now, the trade-off problem is a question on the mechanism underlying the apparent association of several functional traits of representation systems. So far, those traits have been intuitively described as "inferential efficiency," "expressive inflexibility," and "expressive richness." The first step in the analysis of the phenomenon is then to unpack these intuitive descriptions. For this paper's plan, this amounts to specifying inferential efficiency, expressive inflexibility, and expressive richness as they are found in a system of tabular representations. We start with inferential efficiency.

1.1 Inferential Efficiency

In what respect does $\mathcal{R}_t$ support efficient inferences? Consider the question whether the information (4) is a consequence of the set of information $\{(1), (2), (3)\}$.

(1) The model A has the functions F_1 and F_3 and no other functions.
(2) The model B has the functions F_3 and F_4 and no other functions.
(3) The model C has the functions F_1, F_3 and F_6 and no other functions.
(4) None of the models A, B, and C has the function F_2.

Well, yes, it is, of course. One could answer this question by directly thinking about an arbitrary situation in which (1), (2), and (3) all hold, and asking oneself whether (4) necessarily holds in that situation. Alternatively, one could express all the information (1), (2), and (3) in an $\mathcal{R}_t$-table. Noticing that a blank column under the label "F_2" in the resulting table (Figure 2), one can read off the information (4) and validly conclude that (4) is a consequence of $\{(1), (2), (3)\}$.

This alternative way is an instance of the procedure characterized as "free ride" (Shimojima [8]), which various researchers have emphasized as a main root of inferential efficiency provided by graphical systems (Sloman [1], Lindsay [9], Barwise and

[1] We discuss this particular system for the sake of concreteness, but our discussions would apply to any feature tables and, less straightforwardly, to tabular representations in general.

Etchemendy [5]). In a free ride, one does not have to think directly about the consequence relation governing the printer situation; instead, one can just express one's premises in a graphic representation and observe the result to draw read off a consequence of the premises. Setting aside the issue of the semantic mechanism of free rides, we just note for now that there are numerous combinations of premises and conclusions whose derivability can be checked in similar ways with $\mathcal{R}_t$ tables. From the present table, for example, one notes a sequence of circles under "F_3" and can read off the consequence (5).[2]

(5) All of the models A, B, and C has the function F_3.

1.2 Expressive Inflexibility

Thus, we have identified the potentials for free rides and graphical consistency proofs as two main roots of the inferential efficacy of the system $\mathcal{R}_t$. In what respect is the system expressively inflexible then? Consider the following information:

(6) Exactly two models have the function F_1.

Can an $\mathcal{R}_t$-table express this information alone, without expressing any other information? First, in order to express this information, an $\mathcal{R}_t$-table must have two circles in the column labeled "F_1, but of course, these circles must be placed in some *particular* cells in that column. Thus, the $\mathcal{R}_t$-table ends up specifying *which two* of the models A, B, C, and D have the function F_1. Secondly, the $\mathcal{R}_t$-table must have either a circle or a blank in each cell of the other columns. Thus, the $\mathcal{R}_t$-table must also specify, for each of the printer models A, B, C, and D and each function other than F_1, whether that model has that function. For these reasons, no $\mathcal{R}_t$-table can express (6) alone.

This inflexibility of expression in the system $\mathcal{R}_t$ is an instance of the property characterized as "content-specificity" or "over-specificity" by Stenning and Oberlander [2] and Shimojima [8]. Over-specificity is a system's incapability of expressing certain sets of information without expressing extra information, and hence it amounts to certain limitations on the expression of weak or abstract information. Philosophers have long considered this property as a distinguishing character of pictorial images (Berkeley [11], Hume [12], Dennett [13], Pylyshyn [14]), most notably of the system of geometry diagrams ("we can't draw a right triangle *per se*"). The property is now considered a feature of a wider range of graphical systems (such as Euler diagrams) and as the above example shows, it is also a feature of the tabular system $\mathcal{R}_t$. Note that there are many other instances of information that $\mathcal{R}_t$ is over-specific about, including:

(7) The model B has exactly two functions.

[2] Free rides are in turn one of a more general group of inferential procedures called "physical on-site inferences" (Shimojima [10]). The group contains three other inferential procedures, and precisely speaking, the inferential potential of $\mathcal{R}_t$ also consists in its capacities for these procedures. Due to space limitation, we have to omit discussions of these procedures here, but our analysis of $\mathcal{R}_t$ in terms of its potential for free rides largely applies to those procedures.

1.3 Expressive Richness

Intuitively, the $\mathcal{R}_t$-table in Figure 1 expresses the information that none of the models A, B, and C has the function F_2. Clearly, it expresses this information because, in that table, the entire area where the rows labeled "A," "B," and "C" intersect with the column labeled "F_2" is blank. Thus, in one way or another, this condition of the table, (4*), indicates the information (4).

(4*) The area where the rows labeled "A," "B," and "C" intersect with the column labeled "F_2" is all blank.
(4) None of the models A, B, and C has the function F_2.

This semantic relation, however, is distinct from the semantic rules specified in the beginning of this section. Those basic rules simply say that the appearance of a circle in a particular cell indicates the possession of a particular feature by a particular printer type, and that the blankness of a particular cell indicates the non-possession. They are, so to speak, concerned with the meanings of cell-wise states in a table. In contrast, the semantic relation from (4*) to (4) is concerned with the meaning of the blankness of a *sequence* of cells. It is area-wise, so to speak.

This second semantic relation is an instance of the phenomenon called "derivative meaning" (Shimojima [15]). Although it is different from the basic semantic rules, it clearly depends on them. Just imagine there were not the basic rule concerning the meaning of the blankness of individual cells in am $\mathcal{R}_t$-table. Then the blankness of a sequence of cells would mean nothing, and the semantic relation from (4*) to (4) would not hold. Note that the system $\mathcal{R}_t$ has many other semantic relations derived from its basic semantic relations. They include the one from (8*) to (8) and from (9*) to (9) below:

(8*) The entire column labeled "F_1" is filled with circles.
(8) All of the models has the function F_1.
(9*) More circles appear in the row labeled "D" than in the row labeled "E."
(9) The model D has more functions than the model E has.

Now, a system's capacity for such meaning derivation greatly contributes to the overall richness of expressive capacity. For example, Kosslyn [4] alludes to this phenomenon in describing the rich semantic capacities of scatter plots and other visualization forms of statistical data. The additional semantic relations such as the ones just specified is certainly a root of the semantic richness of the particular system $\mathcal{R}_t$.

Thus, we have succeeded in reproducing, in a simple tabular system, the opposition of functional traits that is apparently alluded to in the statement of the trade-off problem: (a) inferential efficiency arising from its potentials for free rides, (b) expressive inflexibility arising from its over-specific character, and (c) expressive richness arising from its potential for derivative meaning. Our underlying assumption is, of course, that this particular opposition found in $\mathcal{R}_t$ is an instance of the opposition responsible for the trade-off problem in general. However, let us defer discussions of the validity of this assumption until later, and let us concentrate on the issue of how we can identify the common semantic mechanism for these particular traits in $\mathcal{R}_t$.

2 Analysis

In our view, that common semantic mechanism consists in what we call "homomorphism of exhaustive sets," and all the three functional traits derive from it through various types of "tracking of exhaustive relations." In this section, we first define these two concepts, and then show how those semantic properties give rise to free rides, over-specificity, and derivative meaning in the system $\mathcal{R}_t$.

2.1 Homomorphism of Exhaustive Sets

Let us begin with introducing several basic terms. *Source tokens* of the system $\mathcal{R}_t$ are *particular* well-formed $\mathcal{R}_t$-tables drawn in different places, such as a piece of paper, a computer display, and a white board. Thus, a particular body of black ink printed as Figure 1 earlier in this paper is an example of a source token of $\mathcal{R}_t$. An $\mathcal{R}_t$-table printed on a different place would be counted as a different source token even if it has exactly the same structure as Figure 1. In contrast, *source types* are conditions or structures of source tokens, and therefore the same source type can hold of different source tokens. For example, the condition of an $\mathcal{R}_t$-table described as (9*) is a source type of the system $\mathcal{R}_t$. It holds of the particular $\mathcal{R}_t$-table in Figure 1, but it can hold of indefinite numbers of other $\mathcal{R}_t$-tables as far as they have more circles in the row labeled "D" than in the row labeled "E."

We distinguish *target tokens* and *target types* similarly. Target tokens are particular situations in different places and times concerning the six printer models and the six potential functions. In contrast, target types are conditions or structures of target tokens, and the same target type can hold of different target tokens. An example is the condition described as (9). It may hold of the current printer situation in Japan, but also can hold of the current printer situation in the US, as long as the printer model D has more functions than E has in both situations.

Given a source type σ and a target type θ of the system $\mathcal{R}_t$, we say σ *indicates* θ *in* $\mathcal{R}_t$, written as "$\sigma \Rightarrow_{\mathcal{R}_t} \theta$," if the semantic relation holds from σ to θ on the basis of the semantic rules for $\mathcal{R}_t$ or as "derivative" of them in the sense described in the last section. Γ of source types and a *set* Δ of target types, we say Γ is projected to Δ in $\mathcal{R}_t$, written as "$\Gamma \Rightarrow_{\mathcal{R}_t} \Delta$," if every member of Γ has some member of Δ that it indicates and if every member of Δ has some member of Γ that indicates it. We say a source token s *satisfies* a set Γ of source types if every condition in Γ holds of s. Similarly for target tokens and sets of target types. See Barwise and Seligman [3], especially chapter 20, for a more systematic presentation of these semantic concepts.

With this preparation, let us illustrate the notion of "exhaustive set." Recall that $\mathcal{R}_t$ is a special-purpose representation system, designed to display the functions of printer models A through G. The variety of functions to be displayed are F_1 through F_6. Thus, the semantically relevant structure of a table in this system can be specified by a set of 6×6 source types, specifying whether each cell has a circle or a blank space. When a set contains 6×6 of such source types corresponding to the 6×6 cells of a well-formed $\mathcal{R}_t$-table, we call the set the *cell-wise structure* of the $\mathcal{R}_t$-table. For example, the cell-wise structure of the table in Figure 1 is the set of source types listed in Figure 3, where "$\bigcirc(X, Y)$" denotes the condition that the intersection of the row labeled "X" and the

$$\{ \bigcirc(A, F_1),\ \Box(A, F_2),\ \bigcirc(A, F_3),\ \Box(A, F_4),\ \Box(A, F_5),\ \Box(A, F_6),$$
$$\bigcirc(B, F_1),\ \Box(B, F_2),\ \Box(B, F_3),\ \Box(B, F_4),\ \Box(B, F_5),\ \Box(B, F_6),$$
$$\bigcirc(C, F_1),\ \Box(C, F_2),\ \Box(C, F_3),\ \Box(C, F_4),\ \Box(C, F_5),\ \Box(C, F_6),$$
$$\bigcirc(D, F_1),\ \bigcirc(D, F_2),\ \bigcirc(D, F_3),\ \bigcirc(D, F_4),\ \bigcirc(D, F_5),\ \Box(D, F_6),$$
$$\Box(E, F_1),\ \bigcirc(E, F_2),\ \Box(E, F_3),\ \bigcirc(E, F_4),\ \bigcirc(E, F_5),\ \Box(E, F_6),$$
$$\Box(G, F_1),\ \Box(G, F_2),\ \bigcirc(G, F_3),\ \Box(G, F_4),\ \Box(G, F_5),\ \bigcirc(G, F_6) \}$$

Fig. 3. A cell-wise structure, Γ_{38}, in the system $\mathcal{R}_t$.

column labeled "Y" has a circle in it, and "$\Box(X, Y)$" denotes the condition that that intersection has a blank space. We call this particular cell-wise structure "Γ_{38}." Since each well-formed table in $\mathcal{R}_t$ has $6 \times 6 = 36$ cells, there are a total of 2^{36} different cell-wise structures in $\mathcal{R}_t$.

By definition, every cell-wise structure is satisfied by some possible $\mathcal{R}_t$-table. And clearly, every $\mathcal{R}_t$-table has exactly one cell-wise structure. That is, if $\mathcal{G}$ is the set of all cell-wise structures, the following facts hold:

(10) Every source token of $\mathcal{R}_t$ satisfies at least one member of $\mathcal{G}$.

(11) Every source token of $\mathcal{R}_t$ satisfies at most one member of $\mathcal{G}$.

(12) Every member of $\mathcal{G}$ is satisfied by some source token of $\mathcal{R}_t$.

We will indicate these facts by calling $\mathcal{G}$ an *exhaustive set* in the source domain of the system $\mathcal{R}_t$.

Note that the condition (10) implies that the set of tokens satisfying at least one member of $\mathcal{G}$ exhaust the entire set of source tokens; the condition (11) implies that the members of $\mathcal{G}$ are mutually incompatible, while (12) implies that each member is singularly consistent. Thus, if we write "$\mathrm{tok}(\Gamma_i)$" to denote the set of tokens satisfying a member Γ_i of $\mathcal{G}$, the collection $\{\mathrm{tok}(\Gamma_i) \mid \Gamma_i \in \mathcal{G}\}$ partitions the entire set of source tokens of the system $\mathcal{R}_t$.

Interestingly, the system $\mathcal{R}_t$ has a "corresponding" exhaustive set in its target. Consider the set of target types listed in Figure 4, where "$F(X, Y)$" denotes the condition that model X features the function Y, and "$L(X, Y)$" denotes the type that model X lacks the function Y. Note that this set is a complete pair-wise specification of a binary "featuring" relation from the six printer models A through G to the six functions F_1 through F_6. Generally, when 6×6 of such target types specify a possible featuring relation among these models and functions, we call the set of those types a *pair-wise structure*. Assuming all combinations of the printer models and the functions are possible, there are a total of 2^{36} pair-wise structures in $\mathcal{R}_t$. The particular pair-wise structure listed in Figure 4 will be called "Δ_{38}" in the following discussions.

By definition, every possible printer situation within the coverage of $\mathcal{R}_t$ satisfies exactly one pair-wise structure. Conversely, every pair-wise structure is satisfied by some possible printer situation. Thus, calling the set of all pair-wise structures "$\mathcal{D}$," we can say:

(13) Every target token of $\mathcal{R}_t$ satisfies at least one member of $\mathcal{D}$.

$$
\begin{array}{llllll}
\{ F(A,F_1), & L(A,F_2), & F(A,F_3), & L(A,F_4), & L(A,F_5), & L(A,F_6), \\
F(B,F_1), & L(B,F_2), & L(B,F_3), & L(B,F_4), & L(B,F_5), & L(B,F_6), \\
F(C,F_1), & L(C,F_2), & L(C,F_3), & L(C,F_4), & L(C,F_5), & L(C,F_6), \\
F(D,F_1), & F(D,F_2), & F(D,F_3), & F(D,F_4), & F(D,F_5), & L(D,F_6), \\
L(E,F_1), & F(E,F_2), & L(E,F_3), & F(E,F_4), & F(E,F_5), & L(E,F_6), \\
L(G,F_1), & L(G,F_2), & F(G,F_3), & L(G,F_4), & L(G,F_5), & F(G,F_6) \}
\end{array}
$$

Fig. 4. A pair-wise structure, Δ_{38}, in the system $\mathcal{R}_t$.

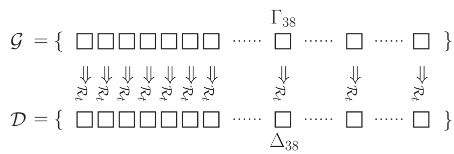

Fig. 5. The homomorphism h from the exhaustive set $\mathcal{G}$ in the source to the exhaustive set $\mathcal{D}$ in the target.

(14) Every target token of $\mathcal{R}_t$ satisfies at most one member of $\mathcal{D}$.

(15) Every member of $\mathcal{D}$ is satisfied by some target token of $\mathcal{R}_t$.

In other words, $\mathcal{D}$ is an exhaustive set in the target domain of the system $\mathcal{R}_t$.

It is this exhaustive set $\mathcal{D}$ that we earlier said to "correspond" to the exhaustive set $\mathcal{G}$. According to the semantic rules in the system $\mathcal{R}_t$, $\bigcirc(X,Y)$ indicates $F(X,Y)$, and $\square(X,Y)$ indicates $L(X,Y)$. Thus, for example, the set Γ_{38} is projected to Δ_{38} but to no other members of $\mathcal{D}$. Generally, every set in $\mathcal{G}$ has exactly one set in $\mathcal{D}$ that it is projected to. Thus, the projection in the system $\mathcal{R}_t$ provides a one-one mapping from $\mathcal{G}$ to $\mathcal{D}$. To indicate this fact compactly, we say that the exhaustive set $\mathcal{G}$ is *homomorphic* to the exhaustive set $\mathcal{D}$ in the system $\mathcal{R}_t$. We will also use "h" to denote the particular one-one mapping from $\mathcal{G}$ to $\mathcal{D}$.

Figure 5 pictures the homomorphism h from $\mathcal{G}$ to $\mathcal{D}$. Each point in the upper square represents an cell-wise structure in $\mathcal{G}$ such as Γ_{38}. Each arrow represents a semantic projection in $\mathcal{R}_t$ that relates the members of a cell-wise structure in $\mathcal{G}$ to the members of a pair-wise structure in $\mathcal{D}$ such as Δ_{38}. (The relation is a one-one correspondence in this particular case.) Then the entire collection of arrows represents the homomorphism h that maps each cell-wise structure in $\mathcal{G}$ to a unique pair-wise structure in $\mathcal{D}$. (The mapping is a one-one correspondence in this particular case.)

Now, due to this homomorphism, a large collection $\mathcal{G}$ of sets of source types semantically corresponds to a large collection of sets of target types. Moreover, by conditions (11), (12), (14), and (15) above, both collections consist of mutually incompatible, but singularly consistent sets of types. As we will see shortly, the extensive semantic correspondence between these special collections is a double-edged sword for the efficacy of the system $\mathcal{R}_t$: combined with "capturing" capacities discussed below, it creates constraint matching sufficient for on-site inferences and content inducement on the one hand, while creating constraint matching for over-specificity on the other hand.

$$\{ \bigcirc(A,F_1),\ \square(A,F_2),\ \bigcirc(A,F_3),\ \square(A,F_4),\ \square(A,F_5),\ \square(A,F_6),$$
$$\square(B,F_1),\ \square(B,F_2),\ \bigcirc(B,F_3),\ \bigcirc(B,F_4),\ \square(B,F_5),\ \square(B,F_6),$$
$$\bigcirc(C,F_1),\ \square(C,F_2),\ \bigcirc(C,F_3),\ \square(C,F_4),\ \square(C,F_5),\ \bigcirc(C,F_6),$$
$$\square(D,F_1),\ \bigcirc(D,F_2),\ \bigcirc(D,F_3),\ \bigcirc(D,F_4),\ \bigcirc(D,F_5),\ \square(D,F_6),$$
$$\square(E,F_1),\ \bigcirc(E,F_2),\ \bigcirc(E,F_3),\ \bigcirc(E,F_4),\ \bigcirc(E,F_5),\ \square(E,F_6),$$
$$\square(G,F_1),\ \square(G,F_2),\ \bigcirc(G,F_3),\ \square(G,F_4),\ \square(G,F_5),\ \bigcirc(G,F_6) \}$$

Fig. 6. Another cell-wise structure, Γ_{269}, in the system $\mathcal{R}_t$.

2.2 Tracking of a Capturing Relation

We say a type *captures* a collection of sets of types if the set of tokens supporting the type is equal to the set of tokens satisfying at least one member of the collection. For example, compare the cell-wise structure Γ_{38} cited above and the following type:

(1*) The row labeled "A" has circles in the columns labeled "F_1" and "F_3" and nowhere else.

Clearly, if an $\mathcal{R}_t$-table has the cell-wise structure Γ_{38}, then in that table, the row labeled "A" has circles in the columns labeled "F_1" and "F_3" and nowhere else. That is, every source token satisfying Γ_{38} supports the source type (1*). The same holds for the cell-wise structure Γ_{269} listed in Figure 6, for if a source token satisfies Γ_{269}, it necessarily supports (1*).

Suppose we enumerate all such cell-wise structures and call the resulting collection of cell-wise structures "$\mathcal{G}_{1^*}$." If the enumeration is complete, $\mathcal{G}_{1^*}$ should exhaust all possible "ways" in which (1*) holds. So, (1*) and $\mathcal{G}_{1^*}$ are in the following relationship:

(16) Every source token supporting (1*) satisfies at least one member of $\mathcal{G}_{1^*}$.

(17) Every source token satisfying some member of $\mathcal{G}_{1^*}$ satisfies (1*).

Exactly in this sense, we say that the type (1*) "captures" the collection $\mathcal{G}_{1^*}$. Intuitively, a type that captures a collection is a "short way" of saying the large disjunction of the conjunctions of 6×6 types contained in individual members of that collection.

The capturing relation holds between many source types of the system $\mathcal{R}_t$ and many subsets of the exhaustive set $\mathcal{G}$. For instance, the following source types capture distinct, but partly overlapping, subsets of $\mathcal{G}$:

(2*) The row labeled "B" has circles in the columns labeled "F_3" and "F_4" and nowhere else.

(3*) The row labeled "C" has circles in the columns labeled "F_1," "F_3" and "F_6" and nowhere else.

(4*) The area where the rows labeled "A," "B," and "C" intersect with the column labeled "F_2" is all blank.

(6*) The column labeled "F_1" has exactly two circles.

Let us use '$\mathcal{G}_i$' to denote the subset of $\mathcal{G}$ captured by a source type σ_i. The above six types therefore capture the subsets $\mathcal{G}_{2^*}$, $\mathcal{G}_{3^*}$, $\mathcal{G}_{4^*}$, and $\mathcal{G}_{6^*}$, respectively. The collections $\mathcal{G}_{1^*}$, $\mathcal{G}_{2^*}$, and $\mathcal{G}_{3^*}$ have 2^{30} members each, while $\mathcal{G}_{4^*}$ has 2^{33} members and $\mathcal{G}_{6^*}$ has 5×2^{30} members.

Now that we understand the notion of "capturing," let us illustrate what it is to "track" a capturing relation. We have seen that there is a one-one semantic mapping h from $\mathcal{G}$ to $\mathcal{D}$. Thus each subset $\mathcal{G}_i$ of $\mathcal{G}$ and its image $h(\mathcal{G}_i)$ under h are in a one-one correspondence. For example, $h(\mathcal{G}_{1^*})$ is the collection of pair-wise structures to which the cell-wise structures in $\mathcal{G}_{1^*}$ are projected by $\mathcal{R}_t$.

Interestingly, there is another way of describing this collection $h(\mathcal{G}_{1^*})$. It is the collection of all the "ways" in which the printer model A features the functions F_1 and F_3 and no other functions. In other words, $h(\mathcal{G}_{1^*})$ is the collection *captured* by the following target type:

(1) The model A has the functions F_1 and F_3 and no other functions.

To see this point clearly, let us see what sort of things are in $h(\mathcal{G}_{1^*})$. By definition, this collection contains all and only those sets of target types to which a member of $\mathcal{G}_{1^*}$ is projected. Since $\mathcal{G}_{1^*}$ is captured by the source type (1*), $\mathcal{G}_{1^*}$ in turn consists of all and only those cell-wise structures that share the following source types:

(18) $\bigcirc(A, F_1), \Box(A, F_2), \bigcirc(A, F_3), \Box(A, F_4), \Box(A, F_5), \Box(A, F_6)$

Since each pair-wise structure in $h(\mathcal{G}_{1^*})$ is just the element-by-element translations of a cell-wise structure in $\mathcal{G}_{1^*}$, it must contain the corresponding target types below:

(19) $F(A, F_1), L(A, F_2), F(A, F_3), L(A, F_4), L(A, F_5), L(A, F_6)$

Moreover, every pair-wise structure containing (19) is clearly the element-by-element translation of a cell-wise structure containing (18). It follows that every cell-wise structure containing (19) is the element-by-element translation of a member of $\mathcal{G}_{1^*}$, and hence is a member of $h(\mathcal{G}_{1^*})$. In sum, $h(\mathcal{G}_{1^*})$ consists of all and only those pair-wise structures containing (19).

We have thus confirmed that (1) captures $h(\mathcal{G}_{1^*})$. But in the source domain of the system $\mathcal{R}_t$, (1*) captures $\mathcal{G}_{1^*}$. This is what we call a "tracking of a capturing." Figure 7 shows the situation schematically. Intuitively, when a source type captures a collection whose correspondent collection under h is captured by a target type, it is a case of tracked capturing.

In the system $\mathcal{R}_t$, there are many cases of such tracking. For example, the source type (4*) captures the collection $\mathcal{G}_{4^*}$ whose correspondent collection $h(\mathcal{G}_{4^*})$ is captured by the following target type:

(4) None of the models A, B, and C has the function F_2.

Similarly, the source types (2*), (3*), and (6*) capture the collections whose correspondents are captured by the following target types respectively.

(2) The model B has the functions F_3 and F_4 and no other functions.
(3) The model C has the functions F_1, F_3 and F_6 and no other functions.
(6) Exactly two models have the function F_1.

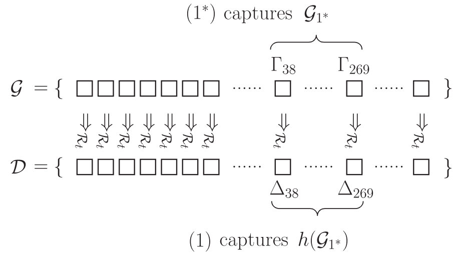

Fig. 7. The capturing of $\mathcal{G}_{1^*}$ by (1^*) tracks the capturing of $h(\mathcal{G}_{1^*})$ by (1).

2.3 Deriving the Phenomena

We have witnessed how two exhaustive sets are semantically homomorphic in the system $\mathcal{R}_t$ and how a capturing relation on one exhaustive set is tracked by a capturing relation on the other. Let us now see how these two properties of $\mathcal{R}_t$ give rise to free rides, derived meaning, as well as over-specific characters.

Derivative Meaning. Recall, from section 1.3, that the source type (4^*) indicates the target type (4) as derived meaning. This indication relation can be explained from the fact that the capturing of the collection $h(\mathcal{G}_{4^*})$ by (4) is tracked by the capturing of the collection $\mathcal{G}_{4^*}$ by (4^*). Suppose an arbitrary source token s (such as the table in Figure 1) supports the source type (4). Since (4^*) captures $\mathcal{G}_{4^*}$, s satisfies some member Γ_k of $\mathcal{G}_{4^*}$. But that member is semantically projected to some member Δ_k of $h(\mathcal{G}_{4^*})$. Since (4) captures $h(\mathcal{G}_{4^*})$, Δ_k entails (4). Thus, the token s ends up expressing (4). The indication from (4^*) to (4) is thus generated.

Similar analyses apply to other pairs of types such as (6^*) and (6). Generally, when a target type captures a collection of mutually incompatible, but singularly consistent sets of source types, and a source type tracks this capturing by capturing the semantically corresponding collection of source types, the source type has the target types as an inducible content. The homomorphism h between the exhaustive sets $\mathcal{G}$ and $\mathcal{D}$ generates quite extensive semantic correspondences of this kind, and therefore prepares much room of parallel capturings. Thus, the enrichment of the indication relation in $\mathcal{R}_t$ derived from this mechanism is quite significant.

Over-Specificity. Recall, from section 1.2, that any $\mathcal{R}_t$-table cannot express the information (6) alone, without expressing any other information. This over-specific character of $\mathcal{R}_t$ is accountable in the following way. Note (6) captures the collection $h(\mathcal{G}_{6^*})$, and this capturing is tracked by the capturing of $\mathcal{G}_{6^*}$ by (6^*). Obviously, $\mathcal{G}_{6^*}$ has a large number of elements. Correspondingly, there is a large number of elements of $h(\mathcal{G}_{6^*})$. Now, for a table s in this system to express (6), it must support the source type (6^*). Since (6^*) captures $\mathcal{G}_{6^*}$, s must satisfy one of the members of $\mathcal{G}_{6^*}$. But $\mathcal{G}_{6^*}$ contains more than one members, say, $\Sigma_1, \ldots, \Sigma_n$, which are semantically projected to distinct

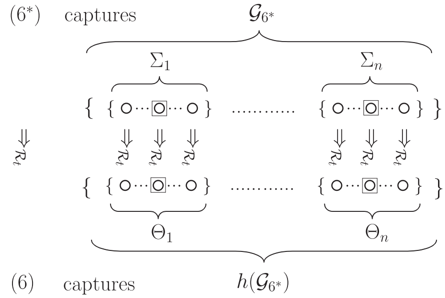

Fig. 8. Over-specificity in presenting (6), which is generated from the parallelism of the capturing of $\mathcal{G}_{6^*}$ by (6) and the capturing of $h(\mathcal{G}_{6^*})$ by (6*).

members $\Theta_1, \ldots, \Theta_n$ of $h(\mathcal{G}_{6^*})$. So, s ends up expressing one of these distinct members $\Theta_1, \ldots, \Theta_n$ of $h(\mathcal{G}_{6^*})$.

Recall $h(\mathcal{G}_{6^*})$ is a subset of the exhaustive set $\mathcal{D}$, and so its members are mutually incompatible, but singularly consistent sets of target types. It follows that for each member of $h(\mathcal{G}_{6^*})$, there is a target token that satisfies it while satisfying no other members. This in turn means that for each member of $h(\mathcal{G}_{6^*})$, there is a target token that does not satisfy it while satisfying some other member, and therefore supporting (6). Thus, each member of $h(\mathcal{G}_{6^*})$ contains some target type that does not follow from (6).

Figure 8 summarizes this situation. Each circle represents an individual source type or a target type, and each arrow represents an indication relation from a source type to an target type. Each member of $h(\mathcal{G}_{6^*})$ has a member (squared) that does not follow from (6), and each member of $\mathcal{G}_{6^*}$ has a corresponding member (squared). This mechanism forces the table s to express some unwarranted information, and generally, no $\mathcal{R}_t$-table can express (6) without thereby expressing some unwarranted information.

Thus, the inflexibility of expression in question derives from the tracking of capturing defined on homomorphic exhaustive sets. Generally, whenever a target type captures a collection with more than one members and this capturing is tracked by a capturing in the source domain, we have a case of over-specificity. This mechanism accounts for the system's inflexibility in expressing $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_7$, and numerous other target types whose capturings are thus tracked.

Free Rides. Recall, from section 1.1, that the system $\mathcal{R}_t$ allows a free ride from the set of information $\{(1), (2), (3)\}$ to its consequence (4). Recall that the target types (1),(2),(3), and (4) respectively capture the collections $h(\mathcal{G}_{1^*})$, $h(\mathcal{G}_{2^*})$, $h(\mathcal{G}_{3^*})$, and $h(\mathcal{G}_{4^*})$. These capturings in the target domain are tracked by other capturings in the source domain, namely, by the capturings of $\mathcal{G}_{1^*}$, $\mathcal{G}_{2^*}$, $\mathcal{G}_{3^*}$, and $\mathcal{G}_{4^*}$ by the source types (1*),(2*),(3*), and (4*). It is clear from the discussions on derivative meaning that this tracking makes the source types (1*),(2*),(3*), and (4*) respectively have the target types (1),(2),(3), and (4) as their derivative meaning. So, the set $\{(1^*), (2^*), (3^*)\}$ is projected to the set $\{(1), (2), (3)\}$ in $\mathcal{R}_t$, while (4*) indicates (4) in $\mathcal{R}_t$.

$$\begin{array}{ccc} \{(1^*),(2^*),(3^*)\} & \vdash & (4^*) \\ \Downarrow_{\mathcal{R}_t} & & \Downarrow_{\mathcal{R}_t} \\ \{(1),(2),(3)\} & \vdash & (4) \end{array}$$

Fig. 9. The matching of consequence relations, generated from the parallelism of capturings by (1^*),(2^*),(3^*), and (4^*) and those by (1),(2),(3), and (4).

Now, crucially, the intersection of the collections $\mathcal{G}_{1^*}$, $\mathcal{G}_{2^*}$, and $\mathcal{G}_{3^*}$ is a subset of the collection $\mathcal{G}_{4^*}$. Since the set of tokens supporting a type σ_i is always equal to the set of tokens satisfying at least one member of the collection $\mathcal{G}_i$ captured by σ_i, this immediately means that the set of tokens supporting (4^*) is a subset of the set of tokens supporting all of (1^*), (2^*), and (3^*). That is, the source type (4^*) is a consequence of the set $\{(1^*),(2^*),(3^*)\}$ of source types. Since each of the collections $\mathcal{G}_{1^*}$, $\mathcal{G}_{2^*}$, $\mathcal{G}_{3^*}$, and $\mathcal{G}_{4^*}$ is in a one-one correspondence to its counter-part in $h(\mathcal{G}_{1^*})$, $h(\mathcal{G}_{2^*})$, and $h(\mathcal{G}_{3^*})$, the fact that the intersection of $\mathcal{G}_{1^*}$, $\mathcal{G}_{2^*}$, and $\mathcal{G}_{3^*}$ is a subset of $\mathcal{G}_{4^*}$ transfers to the fact that the intersection of $h(\mathcal{G}_{1^*})$, $h(\mathcal{G}_{2^*})$, and $h(\mathcal{G}_{3^*})$ is a subset of $h(\mathcal{G}_{4^*})$. Hence, the target type (4) is a consequence of $\{(1),(2),(3)\}$, *just as* (4^*) is a consequence $\{(1^*),(2^*),(3^*)\}$. Figure 9 summarizes this situation, where $\vdash$ means a consequence relation.

Thus, we end up with an interesting matching of the consequence relations in the source domain and in the target domain. Due to the consequence on the source part, whenever one expresses the set $\{(1),(2),(3)\}$ of information by creating an $\mathcal{R}_t$-table with the structural properties $\{(1),(2),(3)\}$, the additional condition (4^*) necessarily holds of that $\mathcal{R}_t$-table. Since (4^*) indicates (4) in $\mathcal{R}_t$, this means that the table also expresses the information (4). But due to the consequence of the target domain, this additional information (4) is a consequence of the original set $\{(1),(2),(3)\}$ of information. The automatic expression of the consequence (4) thus takes place.

Generally, if capturings by source types track capturings by target types and the collections captured by some of the source types intersects within the collection captured by another source type, the system has the capacity of on-site inference of consequence for the target types whose capturings are tracked. The system $\mathcal{R}_t$ allows numerous free rides in this way, beside the one we have just seen.

3 Conclusions

In a nutshell, our analysis shows that the opposing functional traits of the system $\mathcal{R}_t$ are rooted in a semantic correspondence between an extensive structure made from source types and another extensive structure made from target types, where those source types and target type are related through basic semantic rules. Specifically, this correspondence is called a homomorphism, and the two extensive structures are called exhaustive sets. It is just like two complex structures are connected by numerous parallel strings. Look back at the picture of the homomorphism in Figure 5.

Now, a capturing is just like setting aside several components in either structure and crushing them together into a single type. Suppose one does the same to those components of the other structure that are connected to the original components with parallel strings. This process is called a tracking of capturings. With these parallel crushings, those parallel strings that used to connect the two sets of components will be also crushed into a single string. This string is a derived meaning relation—it is derived from original parallel strings established by basic semantic rules. Moreover, the system obtains potentials for free rides and over-specificity depending on what components are chosen for these parallel crushings. Thus, the original exhaustive sets connected by numerous strings is the very base on which the system $\mathcal{R}_t$ obtains various functional traits. This is the common semantic mechanism that we have been after.

If so, how much does this account clarify the trade-off phenomenon? Note that the existence of homomorphic exhaustive structures is only the base on which the various functional traits may be constructed; as such it is not a sufficient conditions for them, but only a *near*-sufficient condition. For the system $\mathcal{R}_t$ to obtain those functional traits, there must exist various trackings of capturing relations on that base. Thus questions still remain as to how various types of capturings hold and how capturings in the target domain are paralleled by capturings in the source domain. The first question is, for example, concerned with why we use abstract source types such as (20*) in addition to more basic source types such as (21*), or why we use abstract target types such as (20) in addition to more basic target types such as (21). The second question is concerned with why a matching pair of abstract types often exist over the source domain and the target domain, such as the pair of (20*) and (20).

(20*) The column labeled "F_1" is all blank.

(21*) The intersection of the row labeled "C" and the column labeled "F_2" is blank.

(20) No models has the function F_1.

(21) The model C does not have the function F_2.

Strictly speaking, these questions are matters of empirical studies on how humans individuate or construct environmental properties or conditions, but as our analysis of the system $\mathcal{R}_t$ shows, we apparently have a strong capacity for such abstractions. Combined with the homomorphism property of a graphical system, this capacity yields positive functional traits such as expressive richness and inferential efficiency. This gives a competitive edge to the system and makes it preferred and inherited over generations of users.[3] However, the same combination of our strong abstraction capacity and the homomorphism property also yields a negative trait such as expressive inflexibility. This would explain how the opposing traits coexist in many existing graphical systems.

How far can we apply our analysis to other graphical systems then? In our view, the analysis can be directly extended to account for the opposition of functional traits in the system of Euler diagrams, the system of standard geometry diagrams, and many systems of geographical maps. Each of these systems has an extensive semantic correspondence between exhaustive sets in the source and the domain, and various parallel abstractions in these structures give rise to free rides, meaning derivation, and over-specificity.

[3] I thank an anonymous reviewer to point out this historical selection process of systems.

This, however, does not mean that the existence of homomorphic exhaustive sets is the only way a system obtains the potentials for meaning derivation, free rides, and over-specificity. Our claim is solely that a homomorphism of exhaustive sets in the source and the target holds *when and where* a graphical system has inferential efficiency, expressive richness, and expressive inflexibility *together*. Thus, there may well be graphical systems with one functional trait without the others, or the degrees of their coexistence may vary. The system of Venn diagrams, for example, is a system with a fairly high capacity for free rides without severe setback from over-specificity. It would be, therefore, an interesting extension of our analysis to investigate various ways a graphical system may have the semantic properties for free rides or meaning derivation without a homomorphism of exhaustive sets, an optimal combination of functional traits.

References

1. Sloman, A.: Interactions between philosophy and ai: the role of intuition and non-logical reasoning in intelligence. Artificial Intelligence **2** (1971) 209–225
2. Stenning, K., Oberlander, J.: A cognitive theory of graphical and linguistic reasoning: Logic and implementation. Cognitive Science **19** (1995) 97–140
3. Barwise, J., Seligman, J.: Information Flow: the Logic of Distributed Systems. Cambridge Tracts in Theoretical Computer Science, 42. Cambridge University Press, Cambridge, UK (1997)
4. Kosslyn, S.M.: Elements of Graph Design. W. H. Freeman and Company, New York (1994)
5. Barwise, J., Etchemendy, J.: Visual information and valid reasoning. In Allwein, G., Barwise, J., eds.: Logical Reasoning with Diagrams. Oxford University Press, Oxford (1990) 3–25
6. Barwise, J., Hammer, E.: Diagrams and the concept of logical system. In Gabbay, D., ed.: What Is a Logical System? Oxford University Press, Oxford (1995)
7. Stenning, K., Inder, R.: Applying semantic concepts to analyzing media and modalities. In Glasgow, J., Narayanan, N.H., Chandrasekaran, B., eds.: Diagrammatic Reasoning: Cognitive and Computational Perspectives. The MIT Press and the AAAI Press, Cambridge, MA and Menlo Park, CA (1995) 303–338
8. Shimojima, A.: Operational constraints in diagrammatic reasoning. In Barwise, J., Allwein, G., eds.: Logical Reasoning with Diagrams. Oxford University Press, Oxford (1995b) 27–48
9. Lindsay, R.K.: Images and inference. In Glasgow, J.I., Narayanan, N.H., Chandrasekaran, B., eds.: Diagrammatic Reasoning: Cognitive and Computational Perspectives. The MIT Press and the AAAI Press, Cambridge, MA and Menlo Park, CA (1988) 111–135
10. Shimojima, A.: A logical analysis of graphical consistency proofs. In Magnani, L., Nersessian, N., Pizzi, C., eds.: Proceedings of MBR 2001 (tentative title). Kluwer Academic Publishers, Dordrecht, Netherlands (Forthcoming)
11. Berkeley, G.: A Treatise Concerning Principles of Human Knowledge. The Library of Liberal Arts. The Bobbs-Merrill Company, Inc., Indianapolis (1710)
12. Hume, D.: A treatise of human nature. In: The Philosophical Works of David Hume. Volume 1. Little, Brown and Company, Boston (1739)
13. Dennett, D.C.: Content and Consciousness. International library of philosophy and scientific method. Routledge and Kegan Paul, London (1969)
14. Pylyshyn, Z.W.: What the mind's eye tells the mind's brain: A critique of mental imagery. Psychological Bulletin **80** (1973) 1–24
15. Shimojima, A.: Derivative meaning in graphical representations. In: Proceedings of the 1999 IEEE Symposium on Visual Languages. IEEE Computer Society, Washington, D. C. (1999) 212–219